Enabling Long-range Exploration in Minimization of Multimodal Functions

Jiaxin Zhang[1] **Hoang Tran**[1] **Dan Lu**[2] **Guannan Zhang**[1]

[1]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
[2]Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

Abstract

We consider the problem of minimizing multimodal loss functions with a large number of local optima. Since the local gradient points to the direction of the steepest slope in an infinitesimal neighborhood, an optimizer guided by the local gradient is often trapped in a local minimum. To address this issue, we develop a novel nonlocal gradient to skip small local minima by capturing major structures of the loss's landscape in black-box optimization. The nonlocal gradient is defined by a directional Gaussian smoothing (DGS) approach. The key idea of DGS is to conducts 1D long-range exploration with a large smoothing radius along d orthogonal directions in $\mathbb{R}^d$, each of which defines a nonlocal directional derivative as a 1D integral. Such long-range exploration enables the nonlocal gradient to skip small local minima. The d directional derivatives are then assembled to form the nonlocal gradient. We use the Gauss-Hermite quadrature rule to approximate the d 1D integrals to obtain an accurate estimator. The superior performance of our method is demonstrated in three sets of examples, including benchmark functions for global optimization, and two real-world scientific problems.

1 INTRODUCTION

We consider the problem of minimizing multimodal loss functions in the black-box setting, i.e., searching for the global minimum of a d-dimensional loss function $F(\boldsymbol{x})$, given access to only function queries. This is motivated by several applications, e.g., hyperparameter tuning of neural networks [Real et al., 2017], reinforcement learning [Salimans et al., 2017], and generating adversarial examples[Chen et al., 2017], in which the loss function's gradient is inaccessible. An extensive amount of effort are devoted to this topic. We refer to [Rios and Sahinidis, 2009, Larson et al., 2019] for broad overviews on black-box optimization.

The local gradient, i.e., $\nabla F(\boldsymbol{x})$, is the most commonly used quantities to guide optimization. When $\nabla F(\boldsymbol{x})$ is not directly accessible, we usually reformulate $\nabla F(\boldsymbol{x})$ as a functional of $F(\boldsymbol{x})$. One class of methods for reformulation is Gaussian smoothing (GS) [Salimans et al., 2017, Maheswaranathan et al., 2019]. GS smooths the landscape of the loss function with d-dimensional Gaussian convolution, then represents $\nabla F(\boldsymbol{x})$ by the gradient of the smoothed function. Monte Carlo (MC) sampling is used to estimate the Gaussian convolution. Several studies have been performed to improve GS. Most of them focus on enhancing quality of the MC estimators by e.g., variance reduction [Maggiar et al., 2018], exploiting historical data [Maheswaranathan et al., 2019, Meier et al., 2019], employing active subspaces [Choromanski et al., 2019], and searching on latent low-dimensional manifolds [Sener and Koltun, 2020]. Despite the improvements, existing work did not address the challenge of applying the local gradient to optimizing high-dimensional multimodal loss functions. Since the local gradient points to the direction of the steepest slope in an infinitesimal neighborhood, an optimizer guided by the local gradient is often trapped in a local minimum.

We develop a novel nonlocal gradient to skip small local minima by capturing major structures of the loss's landscape. The nonlocal gradient is defined by a directional Gaussian smoothing (DGS) approach, so we refer to our nonlocal gradient as *DGS gradient* hereinafter. The key idea behind the DGS gradient is to conduct 1D long-range explorations along d orthogonal directions in $\mathbb{R}^d$, each of which defines a nonlocal directional derivative as a 1D integral. Then, the d directional derivatives are assembled to form the DGS gradient. Compared with the standard GS approach, our DGS method can use large smoothing radius to achieve long-range exploration along the orthogonal directions. This enables the DGS gradient to provide better search directions than that provided by the local gradient, which makes the

Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021), PMLR 161:1639–1649.

DGS gradient particularly suitable for minimizing multimodal loss functions.

For accurate and efficient calculation of the DGS gradient, we use the Gauss-Hermite (GH) quadrature rule [Quarteroni et al., 2007, Abramowitz and Stegun, 1972] to estimate the d 1D integrals. It has been theoretically proved that the GH quadrature can achieve much higher accuracy that MC methods in estimating the 1D integrals in the DGS gradient. By leveraging such property, the GH quadrature ensures an accurate calculation of the DGS gradient even with a large smoothing radius.

Summary of contributions. Our contribution in this paper is three fold: (1) we develop the DGS gradient for long-range exploration with a large smoothing radius, which advances the global search in high-dimensional black-box optimization; (2) we develop an accurate and efficient GH-based estimator to calculate the DGS gradient; and (3) we demonstrate the superior performance of our method in minimizing several high-dimensional multimodal benchmark functions, and solving two real-world scientific and engineering problems.

Related works. The literature on black-box optimization is extensive. We review four types of methods that are closely related to this work (see [Rios and Sahinidis, 2009, Larson et al., 2019] for thorough reviews). (1) *Random search*. This type of methods first randomly generates search directions, then estimates directional derivatives (or performs direct search) to update the current states. Examples include two-point approaches [Flaxman et al., 2005, Nesterov and Spokoiny, 2017, Duchi et al., 2015, Bubeck and Cesa-Bianchi, 2012], coordinate-descent algorithms [Jamieson et al., 2012], three-point methods [Bergou et al., 2019], and binary search with adaptive radius [Golovin et al., 2020]. A theoretical analysis of two-point schemes based on GS is presented in the seminal paper [Nesterov and Spokoiny, 2017] and extended in [Ghadimi and Lan, 2013] for non-convex and in [Shamir, 2017] for non-smooth loss functions. Even though these methods have good scalability with the dimension, current studies focus on estimating local derivatives rather than nonlocal exploration. (2) *Local gradient estimation*. The most straightforward way for local gradient estimation is to use finite difference. An alternative is to use linear interpolation in a small neighborhood of the current state [Berahas et al., 2019b]. Another way is to estimate the local gradient by averaging multiple directional estimates by two-point schemes. The GS-based evolutionary strategy (ES) [Salimans et al., 2017, Mania et al., 2018, Maheswaranathan et al., 2019, Choromanski et al., 2019] can be assigned to this category. Some studies augmented ES by integrating the estimated local gradient with different gradient-based algorithms, such as alternating direction method of multipliers [Liu et al., 2017], the adaptive momentum method [Chen et al., 2019], and conditional gradient methods [Balasubramanian and Ghadimi, 2018]. A comparison of local gradient estimation methods is summarized in [Berahas et al., 2019a]. (3) *Smoothing techniques*. Sphere smoothing is a method similar to GS and is discussed in [Flaxman et al., 2005]. Analysis of GS applied to step functions is presented in [Addis et al., 2005]. Other strategies transform a non-convex and noisy optimization problem to a convex optimization problem, such as p-th power transformation [Li et al., 2001] and ℓ^2 regularization [Carlsson, 2019]. An algorithm for estimating computational noise affecting a smooth simulation was developed in [Moré and Wild, 2011]. (4) *Orthogonal exploration*. Sampling along orthogonal directions was also investigated in black-box optimization. Finite difference can be viewed as a deterministic sampling along orthogonal directions. A recent work [Choromanski et al., 2018] introduced the orthogonal sampling into GS. The main differences between our method and [Choromanski et al., 2018] are: (i) we perform nonlocal exploration to define a new nonlocal gradient, while the study in [Choromanski et al., 2018] performed orthogonal sampling to approximate the local gradient, and (ii) we use the GH quadrature for our DGS gradient estimation and [Choromanski et al., 2018] used (quasi) MC estimator for the local gradient estimation.

2 BLACK-BOX OPTIMIZATION

We consider the following black-box optimization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ consists of d inputs, and $F : \mathbb{R}^d \to \mathbb{R}$ is a d-dimensional loss function. We assume that the gradient $\nabla F(\boldsymbol{x})$ is unavailable, and $F(\boldsymbol{x})$ is only accessible via function evaluations. In this work, we are particularly interested in minimizing multimodal loss functions, i.e., $F(\boldsymbol{x})$ has a large number of local minima.

We briefly review the standard GS methods [Flaxman et al., 2005, Nesterov and Spokoiny, 2017] for estimating the local gradient. The smoothed loss is defined by $F_\sigma(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(0,\mathbf{I}_d)}\left[F(\boldsymbol{x}+\sigma\boldsymbol{u})\right]$, where $\mathcal{N}(0,\mathbf{I}_d)$ is the d-dimensional standard Gaussian distribution, and $\sigma > 0$ is the smoothing radius. The standard GS [Salimans et al., 2017] represents the $\nabla F_\sigma(\boldsymbol{x})$ as an d-dimensional integral and estimates it by drawing M random samples $\{\boldsymbol{u}_m\}_{m=1}^M$ from $\mathcal{N}(0,\mathbf{I}_d)$, i.e.,

$$\begin{aligned}\nabla F_\sigma(\boldsymbol{x}) &= \frac{1}{\sigma}\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(0,\mathbf{I}_d)}\left[F(\boldsymbol{x}+\sigma\boldsymbol{u})\,\boldsymbol{u}\right] \\ &\approx \frac{1}{M\sigma}\sum_{m=1}^{M} F(\boldsymbol{x}+\sigma\boldsymbol{u}_m)\boldsymbol{u}_m.\end{aligned} \tag{2}$$

The MC estimator in Eq. (2) is usually used as an unbiased estimator of the local gradient $\nabla F(\boldsymbol{x})$ by exploiting the fact that $\lim_{\sigma\to 0}\nabla F_\sigma(\boldsymbol{x}) = \nabla F(\boldsymbol{x})$.

Conceptually, the standard GS-based gradient ∇F_σ could help skip local minima using a large smoothing radius σ. However, σ is often set to a small value in practice, especially for high-dimensional problems, in order to guarantee the accuracy of the MC estimator for the d-dimensional integral $\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(0,\mathbf{I}_d)}\left[F(\boldsymbol{x}+\sigma\boldsymbol{u})\,\boldsymbol{u}\right]$ in Eq. (2). Hence, the standard GS cannot provide the desired long-range exploration capability for capturing major structures of multimodal losses.

3 OUR METHOD: A NONLOCAL GRADIENT VIA DIRECTIONAL GAUSSIAN SMOOTHING

In this section, we address the lack of long-range exploration of the standard GS-based local gradient by answering the following question:

Question*: Can we define a new Gaussian smoothing method that only involves very low-dimensional integrals, in order to achieve long-range exploration in minimizing multimodal loss functions?*

The answer is yes. Our intuition comes from the fact that each component of a gradient (no matter it's a local or a nonlocal gradient) is a directional derivative that only explores how a function varies along one direction. Nevertheless, such feature is missing in the standard GS formula in Eq. (2), where each partial derivative of ∇F_σ involves a full d-dimensional exploration. This motivates us to define the DGS gradient. The key idea of our method is as follows:

Key idea*: We conduct 1D long-range explorations along d orthogonal directions in $\mathbb{R}^d$, each of which defines a nonlocal directional derivative as a 1D integral. The Gauss-Hermite quadrature is used to estimate the d 1D integrals to provide accurate estimation of the DGS gradient.*

3.1 THE DGS GRADIENT

To proceed, we first define a 1D cross section of $F(\boldsymbol{x})$ as

$$G(y\,|\,\boldsymbol{x},\boldsymbol{\xi}) = F(\boldsymbol{x}+y\,\boldsymbol{\xi}),\ \ y\in\mathbb{R},$$

where $\boldsymbol{x}$ is the current state of $F(\boldsymbol{x})$ and $\boldsymbol{\xi}$ is a unit vector in $\mathbb{R}^d$. Note that $\boldsymbol{x}$ and $\boldsymbol{\xi}$ can be viewed as parameters of the function G. We define the Gaussian smoothing of $G(y)$, denoted by $G_\sigma(y)$, by

$$\begin{aligned} G_\sigma(y\,|\,\boldsymbol{x},\boldsymbol{\xi}) &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} G(y+\sigma v\,|\,\boldsymbol{x},\boldsymbol{\xi})\,\mathrm{e}^{-\frac{v^2}{2}}\,dv \\ &= \mathbb{E}_{v\sim\mathcal{N}(0,1)}\left[G(y+\sigma v\,|\,\boldsymbol{x},\boldsymbol{\xi})\right], \end{aligned} \tag{3}$$

which is the Gaussian smoothing of $F(\boldsymbol{x})$ along the direction $\boldsymbol{\xi}$ in the neighbourhood of $\boldsymbol{x}$. The derivative of $G_\sigma(y|\boldsymbol{x},\boldsymbol{\xi})$ at $y=0$ can be represented by a 1D expectation

$$\mathscr{D}[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi})] = \frac{1}{\sigma}\,\mathbb{E}_{v\sim\mathcal{N}(0,1)}\left[G(\sigma v\,|\,\boldsymbol{x},\boldsymbol{\xi})\,v\right], \tag{4}$$

where $\mathscr{D}[\cdot]$ denotes the differential operator. Since Eq. (4) is a 1D integral, it is easier to conduct long-range exploration with a large smoothing radius σ.

For a matrix $\boldsymbol{\Xi} := (\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_d)$ consisting of d orthonormal vectors, we can define d directional derivatives like Eq. (4) and assemble *the DGS gradient* as

$$\nabla_{\sigma,\boldsymbol{\Xi}}[F](\boldsymbol{x}) = \boldsymbol{\Xi}^\top \begin{bmatrix} \mathscr{D}\left[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi}_1)\right] \\ \vdots \\ \mathscr{D}\left[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi}_d)\right] \end{bmatrix}. \tag{5}$$

We emphasize that the differences between $\nabla_{\sigma,\boldsymbol{\Xi}}[F]$ and ∇F_σ in Eq. (2) are two-fold: (i) $\nabla_{\sigma,\boldsymbol{\Xi}}[F]$ is used in the nonlocal setting with a large σ while ∇F_σ is used in the local setting with a small σ; (ii) $\nabla_{\sigma,\boldsymbol{\Xi}}[F]$ consists of d 1D integrals while ∇F_σ consists of one d-dimensional integral.

3.2 THE GAUSS-HERMITE QUADRATURE ESTIMATOR

The DGS gradient in Eq. (5) is not practical until an accurate estimator is provided. As each component of $\nabla_{\sigma,\boldsymbol{\Xi}}[F](\boldsymbol{x})$ is a 1D integral, the GH quadrature rule [Quarteroni et al., 2007, Abramowitz and Stegun, 1972] is perfectly suitable for approximating the integrals with high accuracy. By doing a simple change of variable in Eq. (4), the GH rule can be directly used to obtain the following estimator for $\mathscr{D}[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi})]$, i.e.,

$$\begin{aligned} &\widetilde{\mathscr{D}}^M[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi})] \\ =&\frac{1}{\sqrt{\pi}\sigma}\sum_{m=1}^{M} w_m\,F(\boldsymbol{x}+\sqrt{2}\sigma v_m\boldsymbol{\xi})\sqrt{2}v_m, \end{aligned} \tag{6}$$

where $\{v_m\}_{m=1}^M$ are the roots of the M-th order Hermite polynomial and $\{w_m\}_{m=1}^M$ are quadrature weights. Both v_m and w_m can be found online or in [Abramowitz and Stegun, 1972]. It was proved in [Abramowitz and Stegun, 1972] that the error of the estimator in Eq. (6) is

$$\left|(\widetilde{\mathscr{D}}^M-\mathscr{D})[G_\sigma]\right| \le C\frac{M\,!\sqrt{\pi}}{2^M(2M)\,!}\sigma^{2M-1}, \tag{7}$$

where $M!$ is the factorial of M, and C is independent of M and σ. In comparison, the error of an MC estimator is on the order of $1/\sqrt{M}$. Applying the GH quadrature to each component of $\nabla_{\sigma,\boldsymbol{\Xi}}[F](\boldsymbol{x})$ in Eq. (5), we obtain the final estimator for the DGS gradient:

$$\widetilde{\nabla}^M_{\sigma,\boldsymbol{\Xi}}[F](\boldsymbol{x}) := \boldsymbol{\Xi}^\top \begin{bmatrix} \widetilde{\mathscr{D}}^M\left[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi}_1)\right] \\ \vdots \\ \widetilde{\mathscr{D}}^M\left[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi}_d)\right] \end{bmatrix}, \tag{8}$$

which requires a total of $M \times d$ function evaluations. An illustration of the difference between the DGS gradient and the local gradient is given in Figure 1.

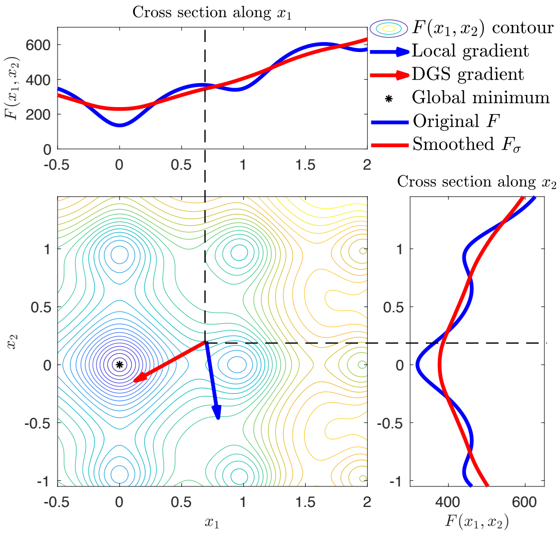

Figure 1: Illustration of the nonlocal exploration of the DGS gradient. In the central plot, the blue arrow points to the local gradient direction and the red arrow points to the DGS gradient direction. The top and right plots show the directionally smoothed functions along the two axes. Because the DGS gradient captures the major structure of F along both directions, it provides a direction pointing much closer to the global minimizer than the local gradient.

In the following, we discuss some important features of the DGS gradient and the GH quadrature estimator.

- ***The DGS gradient vs. the local gradient.*** The DGS gradient is designed to be used with a relatively large smoothing radius σ. Thus, the DGS gradient in Eq. (5) and the GH quadrature estimator in Eq. (8) should not be viewed as approximations to the local gradient. In fact, Figure 1 illustrates that the DGS gradient can provide better search directions than the local gradient attributed to its large smoothing radius. Following experiments further demonstrate the superior performance of the DGS gradient to the local gradient in optimizing high-dimensional benchmark functions (Figure 2 and Table 1).
- ***Scalability with the dimension***: The GH quadrature estimator in Eq. (8) needs $M \times d$ function evaluations per iteration. This may not be ideal compared to the $4 + 3\log(d)$ evaluations per iteration required by most MC-based local gradient estimators. However, when comparing the performance using the loss decay with the total number of function evaluations, i.e., #(func eval per iteration) × #(iteration), our experimental results show that the DGS gradient has an overall better performance than several state-of-the-art baseline methods in minimizing multimodal functions. This is because the DGS gradient provides a good quality[1] of search directions which significantly reduces the number of iterations. The efficiency of the DGS gradient can be further improved by applying dimension reduction techniques (e.g., in [Choromanski et al., 2019]) to reduce d in the $M \times d$ complexity of the GH quadrature, which will be explored in future study.
- ***Applicability to constrained optimization.*** The DGS gradient can be easily integrated into a gradient-based algorithms, e.g., gradient descent, Adam, and even constrained optimization algorithms. This feature is particularly useful for solving optimization problems arising from scientific and engineering applications, where constraints are usually imposed to ensure certain physical laws or engineering requirements.

Algorithm 1: The DGS algorithm

```
hyperparameters: $M$: # GH quadrature points; $\lambda_t$: learning rate; $\alpha$: the scaling factor for the rotation $\Delta\Xi$; $r, \beta$: the mean and radius for sampling $\sigma$; $\gamma$: the tolerance for triggering random perturbation.
Input: The initial state $\boldsymbol{x}_0$
Output: The final state $\boldsymbol{x}_T$
Set $\Xi = \mathbf{I}_d$, and $\sigma_i = r$ for $i = 1, \dots, d$
for $t = 0, \dots T - 1$ do
    Evaluate $\{G(\sqrt{2}\sigma_i v_m \,|\, \boldsymbol{x}_t, \boldsymbol{\xi}_i)\}_{m=1,\dots,M}^{i=1,\dots,d}$
    for $i = 1, \dots, d$ do
        Compute $\widetilde{\mathscr{D}}^M[G_{\sigma_i}(0 \,|\, \boldsymbol{x}_t, \boldsymbol{\xi}_i)]$ in Eq. (6)
    end for
    Assemble $\widetilde{\nabla}^M_{\sigma,\Xi}[F](\boldsymbol{x}_t)$ in Eq. (8)
    Set $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \lambda_t \widetilde{\nabla}^M_{\boldsymbol{\sigma},\boldsymbol{\Xi}}[F](\boldsymbol{x}_t)$
    if $\|\widetilde{\nabla}^M_{\sigma,\Xi}[F](\boldsymbol{x}_t)\|_2 < \gamma$ then
        Generate $\Delta\Xi$ and update $\Xi = \Xi + \Delta\Xi$
        Generate $\sigma_i$ from $\mathcal{U}(r - \beta, r + \beta)$
    end if
end for
```

Random perturbation of Ξ and σ. The estimator in Eq. (8) is deterministic for a fixed Ξ and σ, making our approach short of random exploration. To alleviate this issue, we add random perturbations to Ξ and σ. First, we add a small random rotation $\Delta\Xi$ to Ξ. To make $\Xi + \Delta\Xi$ orthonormal, we generate $\Delta\Xi$ as a random skew-symmetric matrix $\Delta\Xi = -\Delta\Xi^\top$ with small-value entries (controlled by $\alpha > 0$) The Gram–Schmidt operation is then used to ensure the othornormality of $\Xi + \Delta\Xi$. The perturbation of σ is conducted by drawing d random samples (one for each direction) from a uniform distribution $\mathcal{U}(r - \beta, r + \beta)$ with $\beta \ll r$. The random perturbation can be triggered by various types of indicators, e.g., the magnitude of the DGS gradient, the number of iterations completed since last perturbation.

Asymptotic consistency. Even though the DGS gradient is designed to be used with a large smoothing radius σ, it is

[1] See Eq. (9) and Table 1 for the metric of how to compare the quality of search directions provided by different methods.

interesting to ask the following question from mathematical perspective: "*Does the DGS gradient converge to the local gradient as σ approaches to zero?*" This question can be answered easily for $F(\boldsymbol{x}) \in \mathcal{C}^{1,1}(\mathbb{R}^d)$. In this case, there exists $L > 0$ such that $\|\nabla F(\boldsymbol{x}+\boldsymbol{\xi}) - \nabla F(\boldsymbol{x})\| \leq L\|\boldsymbol{\xi}\|$, $\forall \boldsymbol{x}$, $\boldsymbol{\xi} \in \mathbb{R}^d$ ($\|\cdot\|$ denotes the L^2 norm in this work). Then, the difference between $\widetilde{\nabla}^M_{\sigma,\Xi}[F]$ and ∇F can be bounded by

$$\left\|\widetilde{\nabla}^M_{\sigma,\Xi}[F] - \nabla F\right\|^2 \leq \frac{2C^2\pi d(M\,!)^2}{4^M((2M)\,!)^2}\sigma^{4M-2} + 32dL^2\sigma^2,$$

where the first term on the right hand side comes from the GH quadrature and the second term measures the difference between ∇F and $\nabla_{\sigma,\Xi}[F]$. It is easy to see the asymptotic consistency, i.e., $\lim_{\sigma\to 0}\left|\nabla F(\boldsymbol{x}) - \widetilde{\nabla}^M_{\sigma,\Xi}[F](\boldsymbol{x})\right| = 0$ for $M > 2$ regardless of the choice of Ξ. Additional discussion about the consistency is provided in Section 4 of the Supplementary Material.

4 EXPERIMENTS

We present the experimental results using three sets of problems. All experiments were implemented in Python 3.6 and conducted on a set of cloud servers with Intel Xeon E5 CPUs. We compare the DGS method with the following (a) ES-Bpop: the standard OpenAI evolution strategy in [Salimans et al., 2017] with a big population (i.e., using the same number of samples as DGS), (b) ASEBO[2]: Adaptive ES-Active Subspaces for Blackbox Optimization [Choromanski et al., 2019] with a population of size $4 + 3\log(d)$, (c) IPop-CMA: the restart covariance matrix adaptation evolution strategy with increased population size [Auger and Hansen, 2005], (d) Nesterov: the random search method in [Nesterov and Spokoiny, 2017], (e) FD: the classical central difference scheme, (f) Cobyla: constrained optimization by linear approximation method in [Powell, 1994] (g) Powell: a conjugate direction method without calculating derivatives [Powell, 1964], (h) DE: differential evolution [Storn and Price, 1997], (i) PSO: particle swarm optimization in [Kennedy and Eberhart, 1995], and (j) TuRBO: trust region Bayesian optimization [Eriksson et al., 2019]. The information of the codes used for the baselines is provided in Section 1 of the Supplementary Material. Our code is available at `https://github.com/HoangATran/AdaDGS`.

4.1 TESTS ON HIGH-DIMENSIONAL BENCHMARK FUNCTIONS

We test the performance of the DGS method on six high-dimensional benchmark functions [El-Abd, 2010], including $F_1(\boldsymbol{x})$: Ellipsoidal, $F_2(\boldsymbol{x})$: Sharp Ridge, $F_3(\boldsymbol{x})$: Ackley, $F_4(\boldsymbol{x})$: Rastrigin, $F_5(\boldsymbol{x})$: Schaffer's F7, and $F_6(\boldsymbol{x})$: Schwefel. Even though our method is designed for multimodal functions, we still include two single-modal functions F_1 and F_2 for demonstration. To make the test functions more general, we applied the following linear transformation to $\boldsymbol{x}$, i.e.,

$$\boldsymbol{z} = \mathbf{R}(\boldsymbol{x} - \boldsymbol{x}_{\text{opt}}),$$

where $\mathbf{R}$ is a rotation matrix making the functions non-separable and $\boldsymbol{x}_{\text{opt}}$ is the optimal state. Then we substitute $\boldsymbol{z}$ into the standard definitions of the benchmark functions to formulate our test problems. Details about those functions are provided in Section 1 of the Supplementary Material.

The hyperparameters of the DGS method are fixed for the six test functions. Specifically, we used $M = 5$ GH quadrature points. A quadratic decay schedule is used for both the smoothing radius and the learning rate where the maximum number of iterations is set to 200. The initial (maximum) learning rate is 5% of the diagonal length of the d-dimensional initial search domain, and the terminal (minimum) learning rate is 1% of the initial learning rate. The initial smoothing radius is 5 times of the length of the initial search domain for each variable, and the terminal smoothing radius is 1% of the initial radius. We turned off the random perturbation by setting $\gamma = 0$. The hyperparameters used for the baseline methods are given in Section 1 of the Supplementary Material. For each test function, we performed 20 trials, each of which has a random initial state, a random rotation matrix $\mathbf{R}$ and a random location of $\boldsymbol{x}_{\text{opt}}$.

The comparison between DGS and the baselines in the 2000D case are shown in Figure 2 and Table 1. The DGS has the best performance overall. In particular, DGS demonstrates significantly superior performance in optimizing the highly multimodal functions F_3, F_4 and F_5. For the ill-conditioned functions F_1 and F_2, DGS can match the performance of the best baseline method, e.g., IPop-CMA. For F_6, all the methods fail to find the global minimum because it has no globally major structure to exploit. How to optimize such kind of functions remains an open question.

We provide additional evidence to explain the superior performance of DGS. We use the averaged cosine distance in Table 1 to measure the quality of the search directions provided by each method,

$$\text{Cos\_Dist} = \frac{1}{T}\sum_{t=1}^{T}\left(1 - \frac{\langle \boldsymbol{x}_t - \boldsymbol{x}_{t-1}, \boldsymbol{x}_{\text{opt}} - \boldsymbol{x}_{t-1}\rangle}{\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|\|\boldsymbol{x}_{\text{opt}} - \boldsymbol{x}_{t-1}\|}\right), \tag{9}$$

where T is the number of iterations of an optimization path[3]. The Cos_Dist of the test cases in Figure 2 are shown in Table 1. We have the following findings: (1) DGS provides smallest Cos_Dist in most cases, which demonstrates that the DGS gradient is very close to the direction pointing to the global minimum (even for functions with many local

[2] We do not report comparison with some recent work on ES, e.g., [Akimoto and Hansen, 2016, Loshchilov et al., 2019], as they underperform ASEBO as shown in [Choromanski et al., 2019].

[3] The optimization paths are different for different methods.

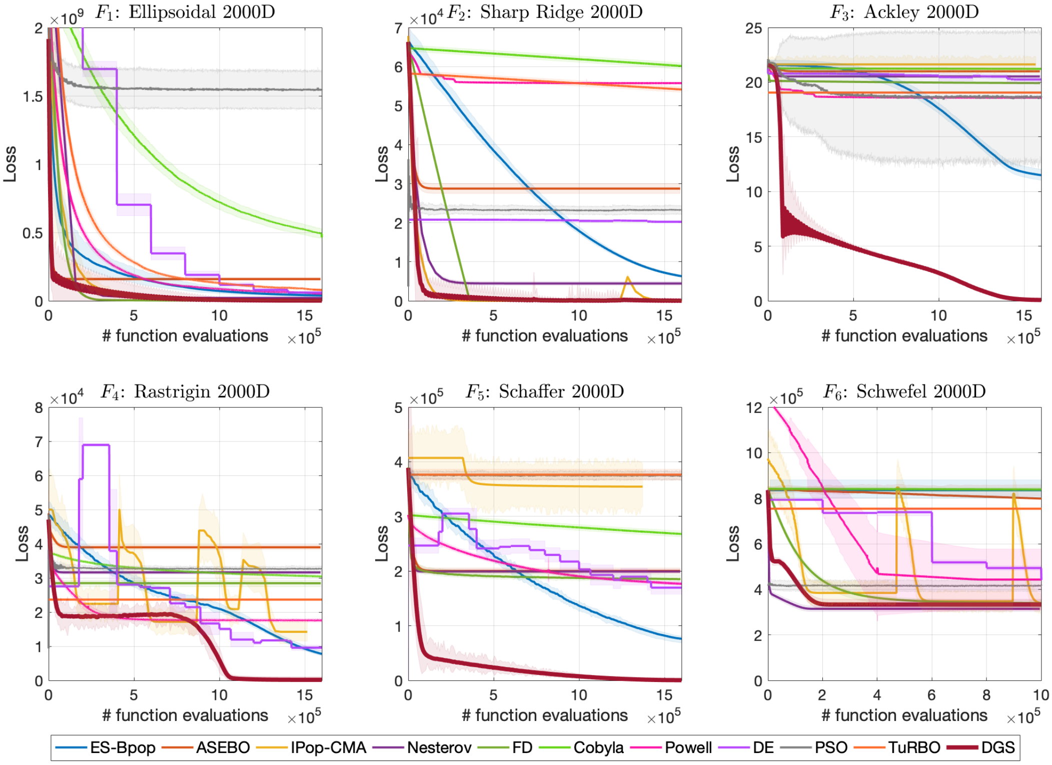

Figure 2: Comparison of the loss decay w.r.t. # function evaluations for the 6 benchmark functions in 2000D. Each curve is the mean of 20 independent trials and the shaded areas represent [mean-3std, mean+3std]. The global minimum is $F(\boldsymbol{x}_{\text{opt}}) = 0$ for all the six functions. The DGS has the best performance overall, especially for the highly multimodal functions F_3, F_4, F_5. All the methods fail to find the global minimum of F_6 which has no global structure to exploit.

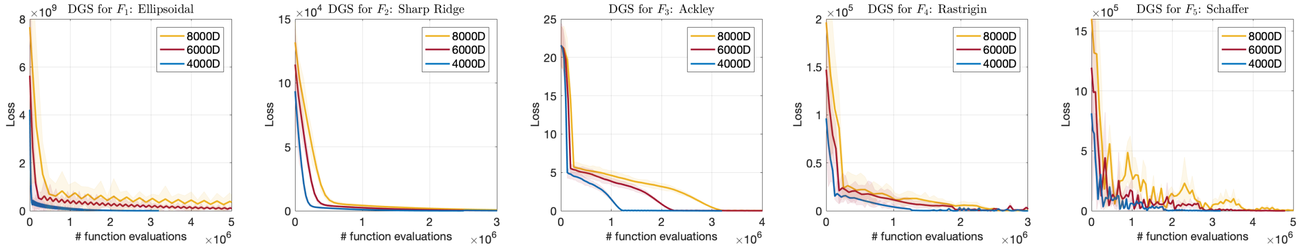

Figure 3: Tests on DGS's scalability to 4000D, 6000D and 8000D. The hyperparameters are the same as the 2000D case. The DGS still achieves promising performance, even though the number of function evaluations increases with the dimension.

minima, e.g., F_3, F_4, F_5). Table 1 implies that being able to find good search directions helps DGS greatly reduce the number of iterations to the achieve its superior performance. (2) FD achieves similar performance as DGS for F_1, F_2, where the local gradients also point to the global minimum, but FD is trapped in local minima for multi-modal F_3, F_4, F_5. (3) As Nesterov randomly selects search directions, it is reasonable that most search directions are perpendicular to $\boldsymbol{x}_{\text{opt}} - \boldsymbol{x}_{t-1}$. (4) ES-Bpop, ASEBO and IPop-CMA have too few random samples to capture the optimal direction $\boldsymbol{x}_{\text{opt}} - \boldsymbol{x}_{t-1}$.

We also test the DGS method in 4000D, 6000D and 8000D to illustrate its scalability with the dimension. We do not test F_6 because DGS failed to optimize F_6 in 2000D. The hyperparameters are set the same as the 2000D cases. The results are shown in Figure 3. The DGS method still achieves promising performance, even though the number of total function evaluations increases with the dimension.

4.2 APPLICATION OF THE DGS GRADIENT TO CONSTRAINED TOPOLOGY OPTIMIZATION FOR ARCHITECTURE DESIGN

We demonstrate the applicability of the DGS gradient to constrained optimization using a real-world topology opti-

	$F_1(\boldsymbol{x})$	$F_2(\boldsymbol{x})$	$F_3(\boldsymbol{x})$
DGS	**0.163** ± **0.02**	0.131 ± 0.03	**0.065** ± **0.01**
ES-Bpop	0.343 ± 0.08	0.476 ± 0.12	0.680 ± 0.05
ASEBO	0.946 ± 0.07	0.999 ± 0.08	0.999 ± 0.07
Nesterov	0.987 ± 0.13	0.992 ± 0.10	0.955 ± 0.02
FD	0.201 ± 0.02	**0.093** ± **0.02**	0.954 ± 0.07
IPop-CMA	1.083 ± 0.04	1.039 ± 0.02	0.999 ± 0.03
Cobyla	0.988 ± 0.03	0.993 ± 0.03	1.053 ± 0.02
Powell	0.308 ± 0.04	0.957 ± 0.03	0.994 ± 0.03
DE	0.599 ± 0.06	0.776 ± 0.04	1.021 ± 0.05
PSO	1.044 ± 0.03	0.503 ± 0.03	0.989 ± 0.03
TuRBO	0.371 ± 0.12	0.961 ± 0.13	0.993 ± 0.09
	$F_4(\boldsymbol{x})$	$F_5(\boldsymbol{x})$	$F_6(\boldsymbol{x})$
DGS	**0.198** ± **0.05**	**0.271** ± **0.03**	1.053 ± 0.02
ES-Bpop	0.710 ± 0.11	0.810 ± 0.04	1.000 ± 0.02
ASEBO	0.993 ± 0.04	0.998 ± 0.05	0.912 ± 0.04
Nesterov	0.998 ± 0.10	0.995 ± 0.10	0.939 ± 0.02
FD	1.099 ± 0.05	0.991 ± 0.01	1.032 ± 0.01
IPop-CMA	1.053 ± 0.09	0.999 ± 0.02	1.000 ± 0.03
Cobyla	1.011 ± 0.01	0.945 ± 0.08	1.022 ± 0.04
Powell	0.915 ± 0.13	0.925 ± 0.09	0.932 ± 0.10
DE	0.687 ± 0.16	0.930 ± 0.09	1.017 ± 0.06
PSO	0.999 ± 0.03	0.689 ± 0.11	**0.899** ± **0.04**
TuRBO	0.943 ± 0.08	0.735 ± 0.12	0.962 ± 0.02

Table 1: The mean and the standard deviation of the cosine distance in Eq. (9) for the test cases in Figure 2. The cosine distance is in the range $[0, 2]$. The smaller the cosine distance, the better the performance of a method. $\text{Cos\_Dist} = 0$ means the two vectors point to the same direction; $\text{Cos\_Dist} = 1$ means the two vectors are perpendicular. Our method achieves the smallest cosine distance for the multi-modal functions F_3, F_4, F_5, which explains its superior performance.

mization (TO) problem. TO has recently attracted attentions in machine learning [Hoyer et al., 2019, Li et al., 2020]. We use DGS-based TO to design a 2D vertical cross section of a bridge from random initial guesses (see Figure 6). The design domain is meshed by 120 × 40 elements, each of which is a design variable ranging from 0 (void) to 1 (solid). By assuming the bridge is symmetric, the total number of independent design variables is 2400 (60 × 40) which is the dimension of the optimization problem. The constraints include (i) 20% volume constraint, i.e., the volume of solid materials (black pixels) in Figure 6 cannot exceed 480 (2400 × 0.2), (ii) unit uniform load on the top and one fixed supports from the bottom. The goal is to optimize the material layout to achieve maximum load-carry capability of the bridge. A conceptually good design is shown in Figure 5 (Bottom-Right).

The challenges in TO include highly nonconvex and multimodal loss functions and rigid constraints. Extensive research efforts have been made on developing exclusive constrained optimization algorithms for TO. The state-of-the-art is Method of Moving Asymptotes (MMA) [Svanberg, 1987], which is a gradient-based method. However, MMA is limited to seek optima using local gradients, either via adjoint method or FD. Here, we address this issue by inserting the DGS gradient into the MMA framework, and exploit the nonlocal exploration ability of the DGS gradient to find a better design. The hyperparameters of DGS are $M = 5$, $\alpha = 0.1$, $r = 0.25$, $\beta = 0.2$ and $\gamma = 0.01$. Note that there is no learning rate λ in this case because the update step of design variable is achieved by MMA optimizer. The hyperparameters for the baselines is given in Section 2 of Supplementary Material. Figure 6 shows the iterative optimization procedure using the DGS-based MMA optimizer.

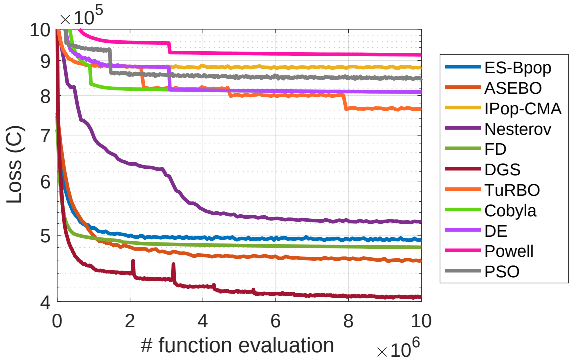

Figure 4: Loss decay for the constrained topology optimization problem for architecture design.

Figure 4 summarizes the results. We ran each algorithm for 5 times with random initial guesses and plot the mean loss decay. The DGS gradient leads to faster convergence and better final design than the baselines. FD converges fast initially but is quickly trapped into a local minimum. ASEBO and ES-Bpop perform similar to FD. Nesterov may perform well eventually but converges slowly. The IPop-CMA has the worst performance because the simple Lagrangian penalty is insufficient to enforce the constraints. The performances are shown by the final design in Figure 5. More details and discussion can be found in Section 2 of the Supplementary Material.

4.3 INFERENCE OF HYDRAULIC CONDUCTIVITY FIELD IN SUBSURFACE ENVIRONMENTS

We demonstrate the performance of DGS in solving an inference problem in groundwater modeling. Hydraulic conductivity measuring the ease of liquid flow through porous media is an important parameter in predicting contaminant transport in groundwater. However, hydraulic conductivity is very difficult to measure and typically inferred from hydraulic heads (easier to measure). In this work, we use a fully connected neural network (FNN) to approximate a 2D

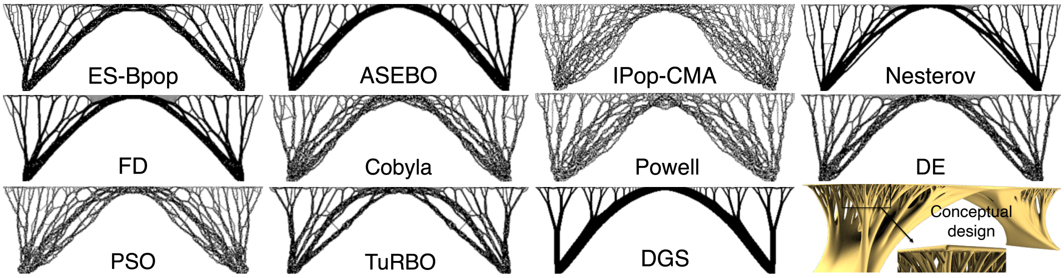

Figure 5: Comparison of final typologies. The DGS-based design shows a strong hierarchical tree feature that matches the good conceptual design (the bottom-right subfigure). IPop-CMA tends to a blurry topology. The other algorithms show many local/minor features that have negative impacts on load-carry capability and bridge construction.

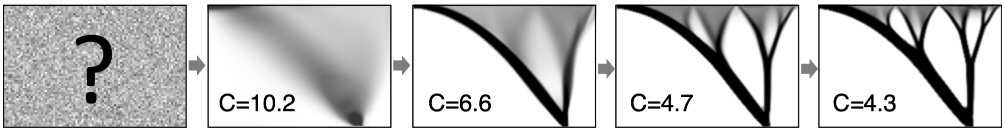

Figure 6: Illustration of DGS-based TO design process from random initial guess. The topology of the bridge architecture tends to be more and more clear as the loss function value $C(\times 10^5)$ decreases.

hydraulic conductivity field (Figure 7 (left)). The FNN has one hidden layer with 64 neurons. The input is the 2D spatial coordinates and the output is hydraulic conductivity values. $tanh(\cdot)$ is used as the activation. The training data are hydraulic head samples randomly selected at 50 locations. To map the output of the FNN to the training data space, we need to run a blackbox groundwater simulator MODFLOW [Harbaugh, 2005] which solves a second-order parabolic partial differential equation. The loss function is defined as the mean squared error between the predicted hydraulic heads and the training data[4]. The hyperparameters for the DGS method are $M = 5, \alpha = 0.1, r = 0.1, \beta = 0.1, \gamma = 0.001$ and $\lambda_t = 0.99\lambda_{t-1}$ with $\lambda_0 = 0.1$. The hyperparameters for the baseline methods and the parameter values for MODFLOW are given in Section 3 of the Supplementary Material.. The results in Figure 7 (right) clearly demonstrate the much faster convergence of our DGS method compared to the baselines.

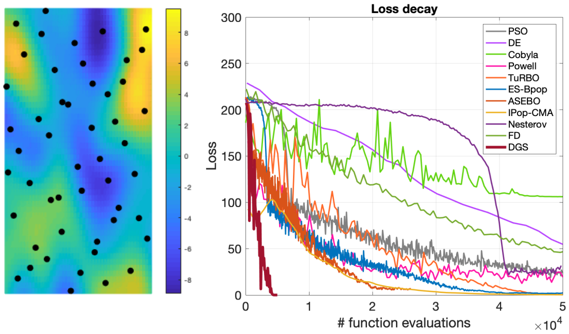

Figure 7: (Left): the target hydraulic conductivity field and the 50 locations (black dots) for collecting hydraulic head data. (Right): comparison of the loss decay w.r.t. # function evaluations for predicting the hydraulic conductivity field using hydraulic head data.

5 DISCUSSION

The current version of the DGS method has some limitations, including (1) *Non-adaptive hyperparameters.* We used a fixed set of hyperparameters for the six benchmark functions and achieved superior results, but we observed some nonoptimal performance of DGS for some hyperparameter values. For example, the fluctuation of the DGS's loss decay for the Ackley in Figure 2 is caused by the nonoptimal learning rate decay. Another important hyperparameter in DGS is the smoothing radius σ. A small σ can result in an insufficiently smoothed loss function, such that the optimizer may be trapped in a local minimum. In contrast, if σ is too large, the loss function is overly smoothed, then the convergence may slow down. How to adaptively adjust the smoothing radius is still an open question. (2) *Suboptimal solution for loss functions without globally major structures.* Figure 2 shows that the DGS cannot find the global minimum of the Schwefel function that does not have a global structure. This could happen in real-world applications. For example, although our method outperforms the baselines in solving the TO problem, we cannot verify that the design obtained by our method is globally optimal. (3) *Sampling complexity.* Since the GH estimator requires $M \times d$ samples per iteration, the DGS gradient is less practical for a large d and for a limited computing budget. For example, when $d = 10000$ but only 10000 function evaluations are affordable, then the

[4] The training data is generated by running MODFLOW with the true hydraulic conductivity field.

current version of the DGS method is inapplicable.

Although the DGS method shows impressive performance in minimizing high-dimensional problems, no method can have superior performance in all dimensions. We tested several 20D cases (see the Supplementary Material), and found that DGS is not competitive for very low-dimensional problems, specifically given a limited number of function evaluations. For example, Bayesian optimization may be a better option than the DGS method for low-dimensional problems [Wu et al., 2017, Eriksson et al., 2018].

Future work. (1) *An adaptive DGS method*. We plan to incorporate line search techniques into DGS to replace the learning rate for better exploiting the good quality of search directions provided by the DGS gradient. In addition, we plan on incorporating the adaptation strategy used in CMA-ES to help adaptively adjust the smoothing radius σ. (2) *Dimension reduction for DGS*. We plan to apply dimension reduction techniques [Choromanski et al., 2019], such as active subspace and sliced linear regression, to reduce the dimensionality and then construct the DGS gradient in the reduced subspace. This strategy will help reduce the sampling complexity in the current version of the DGS method.

Acknowledgements

This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under the contract ERJK359; and by the Artificial Intelligence Initiative at the Oak Ridge National Laboratory (ORNL). This research used resources of the Compute and Data Environment for Science at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

M. Abramowitz and I. Stegun, editors. *Handbook of Mathematical Functions*. Dover, New York, 1972.

Bernardetta Addis, Marco Locatelli, and Fabio Schoen. Local optima smoothing for global optimization. *Optimization Methods and Software*, 20(4-5):417–437, 2005.

Y. Akimoto and N. Hansen. Projection-based restricted covariance matrix adaptation for high dimension. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, page 197–204. Association for Computing Machinery, 2016. ISBN 9781450342063.

A. Auger and N. Hansen. A restart cma evolution strategy with increasing population size. In *2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1769–1776 Vol. 2, 2005.

K. Balasubramanian and S. Ghadimi. Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 3459–3468, 2018.

A. S. Berahas, L. Cao, K. Choromanskiv, and K. Scheinberg. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *arXiv:1905.01332*, 2019a.

Albert S Berahas, Liyuan Cao, Krzysztof Choromanski, and Katya Scheinberg. Linear interpolation gives better gradients than gaussian smoothing in derivative-free optimization. *arXiv preprint arXiv:1905.13043*, 2019b.

E. Bergou, E. Gorbunov, P. Richtárik, and P. Richtárik. Stochastic three points method for unconstrained smooth minimization. *arXiv: 1902.03591*, 2019.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5 (1):1–122, 2012. ISSN 1935-8237.

M. Carlsson. On convex envelopes and regularization of non-convex functionals without moving global minima. *J. of Optim. Theory and Applications*, 183(1):66–84, 2019.

P. Chen, H. Zhang, Y. Sharma, J. Yi, and C. Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, page 15–26, 2017.

X. Chen, S. Liu, K. Xu, X. Li, X.Lin, M. Hong, and D. Cox. Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization. In *NeurIPS*, 2019.

K. Choromanski, M., V. Sindhwani, R. Turner, and A. Weller. Structured evolution with compact architectures for scalable policy optimization. *International Conference on Machine Learning*, pages 969–977, 2018.

K. Choromanski, A. Pacchiano, J. Parker-Holder, Y. Tang, and V. Sindhwani. From complexity to simplicity: Adaptive es-active subspaces for blackbox optimization. In *Advances in Neural Information Processing Systems 32*, pages 10299–10309. Curran Associates, Inc., 2019.

John C. Duchi, Michael I. Jordan, Martin J. Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61: 2788–2806, 2015.

M. El-Abd. Black-box optimization benchmarking for noiseless function testbed using artificial bee colony algorithm. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*, page 1719–1724, 2010.

D. Eriksson, M. Pearce, J. Gardner, R. Turner, and M. Poloczek. Scalable global optimization via local bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 5496–5507, 2019.

David Eriksson, Kun Dong, Eric Hans Lee, David Bindel, and Andrew Gordon Wilson. Scaling gaussian process regression with derivatives. In *NeurIPS*, 2018.

Abraham D. Flaxman, Adam Tauman Kalai, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *Proceedings of the 16th Annual ACM-SIAM symposium on Discrete Algorithms*, pages 385–394, 2005.

S. Ghadimi and G. Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Daniel Golovin, John Karro, Greg Kochanski, Chan-Soo Lee, Xingyou Song, and Qiuyi Zhang. Gradientless descent: High-dimensional zeroth-order optimization. *ArXiv*, abs/1911.06317, 2020.

Arlen Harbaugh. Modflow-2005, the u.s. geological survey modular ground-water model: the ground-water flow process. *US Geol Surv Tech Methods 6-A16*, 2005.

Stephan Hoyer, Jascha Sohl-Dickstein, and Sam Greydanus. Neural reparameterization improves structural optimization. *NeurIPS 2019 Deep Inverse Workshop*, 2019.

K. Jamieson, R. Nowak, and B. Recht. Query complexity of derivative-free optimization. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, page 2672–2680, 2012.

James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.

Jeffrey Larson, Matt Menickelly, and Stefan M. Wild. Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019.

D. Li, X. L. Sun, M. P. Biswal, and F. Gao. Convexification, concavification and monotonization in global optimization. *Annals of Operations Research*, 105(1):213–226, 2001.

Y. Li, X. Li, M. Li, Y. Zhu, B. Zhu, and C. Jiang. A hybrid lagrangian-eulerian method for topology optimization. *arXiv preprint arXiv:2003.01215*, 2020.

Sijia Liu, Jie Chen, Pin-Yu Chen, and Alfred O. Hero. Zeroth-Order Online Alternating Direction Method of Multipliers: Convergence Analysis and Applications. *arXiv e-prints*, art. arXiv:1710.07804, October 2017.

I. Loshchilov, T. Glasmachers, and H. Beyer. Large scale black-box optimization by limited-memory matrix adaptation. *IEEE Transactions on Evolutionary Computation*, 23(2):353–358, 2019.

Alvaro Maggiar, Andreas Wachter, Irina S Dolinskaya, and Jeremy Staum. A derivative-free trust-region algorithm for the optimization of functions smoothed via gaussian convolution using adaptive multiple importance sampling. *SIAM Journal on Optimization*, 28(2):1478–1507, 2018.

N. Maheswaranathan, L. Metz, G. Tucker, D. Choi, and J. Sohl-Dickstein. Guided evolutionary strategies: Augmenting random search with surrogate gradients. *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search of static linear policies is competitive for reinforcement learning. In *NeurIPS*, 2018.

Florian Meier, Asier Mujika, Marcelo Matheus Gauy, and Angelika Steger. Improving gradient estimation in evolutionary strategies with past descent directions. *Optimization Foundations for Reinforcement Learning Workshop at NeurIPS 2019*, 2019.

J. Moré and S. Wild. Estimating computational noise. *SIAM J. Scientific Computing*, 33:1292–1314, 2011.

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

M. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162, 1964.

Michael JD Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer, 1994.

Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical Mathematics*, volume 332. Springer Science Business Media &, 2007.

E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, and A. Kurakin. Large-scale evolution of image classifiers. *International Conference on Machine Learning (ICML)*, pages 2902–2911, 2017.

L. Rios and N. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *J Glob Optim*, 56:1247–1293, 2009.

Tim Salimans, Jonathan Ho, Xi Chen, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Ozan Sener and Vladlen Koltun. Learning to guide random search. In *International Conference on Learning Representations*, 2020.

Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *J. Mach. Learn. Res.*, 18(1):1703–1713, 2017.

R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. of global optimization*, 11(4):341–359, 1997.

K. Svanberg. The method of moving asymptotes—a new method for structural optimization. *International J. for numerical methods in engineering*, 24(2):359–373, 1987.

Jian Wu, Matthias Poloczek, Andrew Gordon Wilson, and Peter I Frazier. Bayesian optimization with gradients. *Advances in Neural Information Processing Systems*, 2017: 5268–5279, 2017.