ORNL/SPR-2022/2717

Final Report on Field Work Proposal ERKJ358: Black-box training for scientific machine learning models

PI: Guannan Zhang
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Email: zhangg@ornl.gov

October 2022

OAK RIDGE National Laboratory

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

CONTENTS

1. Executive summary

The overarching goal of this project is to develop a scalable black-box training capability for scientific machine learning (SciML) problems that are non-trainable with existing automatic differentiation (AD)-based algorithms. AD assumes that a loss function can be decomposed into a sequence of elementary operations whose derivatives are known. This assumption is violated when the loss function includes a black-box physical model (e.g., a legacy simulator). The current strategy, converting a black-box simulator to an AD-enabled code via differential programming, is inflexible and time-, labor-consuming. Thus, black-box optimization is a main workhorse for training SciML models, e.g., in scientific reinforcement learning, hyper-parameter fine tuning, designing SciML models with adversarial robustness, etc.

New SciML methodology. An optimizer guided by the local gradient is often trapped in local optima of highly non-convex loss functions, which occurs very often in training SciML models, e.g., neural networks. Unlike existing work trying to approximate the local gradient, we developed a novel nonlocal gradient via directional Gaussian smoothing (DGS), which can be used in the black-box optimization setting (i.e., the loss function is only accessible via function evaluations). The key idea of the DGS gradient is to conduct one-dimensional long-range exploration with a large smoothing radius along d orthogonal directions in a d-dimensional parameter space, each of which defines a nonlocal directional derivative as a one-dimensional integral. The main features of our method include: (1) *Accuracy:* the Gauss-Hermite quadrature is used to replace Monte Carlo to approximate the DGS gradient, so that a significantly reduced number of function evaluations are needed to achieve the prescribed error tolerance; (2) *Nonlocality:* the long-range exploration capability enables the DGS-based gradient descent optimizer to skip local minima and capture the global structure of non-convex loss functions. (3) *Scalability*: The calculation of the DGS gradient requires an ensemble of loss functions evaluations that can be computed completely in parallel within each iteration of the gradient descent. (4) *Adaptivity:* no hyper-parameter fine tuning is needed to achieve a robust performance of our method. In addition to algorithm development, we also conducted theoretical analysis on the performance of the DGS gradient in optimizing non-convex problems. The theoretical results not only explain why our method outperforms the existing methods in solving non-convex problems, but also provide practical guidance to the users of our method.

Scientific impact. Through collaborations with universities and other laboratories, we have applied our methods to a variety of scientific and machine learning problems, including:

- Model calibration of the liquid mercury spallation target at the Spallation Neutron Source. This is a collaboration with a BES-funded project on ML for improving accelerator and target performance.
- Training heat conduction models for additive manufacturing simulation at the Manufacturing Demonstration Facility. This is a collaboration with the ExaAM project in the Exascale Computing Project.
- Black-box adversarial attack against AI models, which aims at finding vulnerabilities of AI models and designing adversarial robust AI models.
- Our method has been chosen to be used for model calibration in an NNSA project on multiphysics modeling of phenomena and processes that are of high importance for nonproliferation applications.

Deliverables. The outcome of this project have been summarized in 10 scientific publications. The source code of the adaptive DGS gradient descent algorithm has been uploaded to the github page: https://github.com/HoangATran/AdaDGS. More information about this project can be found at https://csmd.ornl.gov/project/black-box-training-scientific-machine-learning-models.

Staff and collaborators. ORNL staff who contributed to this project include Guannan Zhang, Matthew T. Bement, Hoang Tran, Yousub Lee, Benjamin Stump, John Coleman, Jiaxin Zhang (Intuit AI Research, former staff member at ORNL), Sirui Bi (Walmart Global Technology, former postdoc at ORNL). Primary external collaborators include Majdi Radaideh (University of Michigan), Zhongshun Shi (University of Tennessee), Mingjun Lai (University of Georgia), Zhaiming Shen (University of Georgia), Wenpei Gao (North Carolina State University), Jacob Smith (North Carolina State University), Xiaojian Bai (Louisiana State University), Kyle Ma (McMaster University).

2. New black-box optimization methods for SciML

2.1 A nonlocal gradient via directional Gaussian smoothing

We consider the problem of minimizing multi-modal loss functions with a large number of local optima, i.e.,

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

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. Our method significantly outperforms the existing methods in minimizing multimodal loss functions with global structures.

The gradient estimation methods with Gaussian smoothing modify the landscape of $F(\boldsymbol{x})$ with Gaussian convolution, then minimize the smoothed loss function $F_\sigma(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}(0,\mathbf{I}_d)}\left[F(\boldsymbol{x}+\sigma\boldsymbol{u})\right]$, where $\mathcal{N}(0,\mathbf{I}_d)$ is the standard Gaussian distribution, and σ determines the extent of smoothing. The local gradient $\nabla F_{\boldsymbol{\sigma}}(\boldsymbol{x})$ can be written as a d-dimensional integral and estimated by MC sampling,

$$\nabla F_\sigma(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}\sigma}\int_{\mathbb{R}^d} F(\boldsymbol{x}+\sigma\boldsymbol{u})\boldsymbol{u}\,\mathrm{e}^{-\frac{1}{2}\|\boldsymbol{u}\|_2^2}\,d\boldsymbol{u} \approx \frac{1}{M\sigma}\sum_{m=1}^{M} F(\boldsymbol{x}+\sigma\boldsymbol{u}_m)\boldsymbol{u}_m, \tag{2}$$

where the MC estimator can be combined with any gradient-based algorithm (e.g., gradient descent, Adam). *The major drawback is that the error of the MC estimator in Eq. (2) is on the order of* $\varepsilon \sim O(d\sigma/\sqrt{M})$. When the dimension d is large (e.g., on the order of thousands) and the computing budget (upper bound of M) is given, practitioners constantly face the dilemma that a required accuracy (a small ε) and a desired smoothing effect (a relatively big σ) cannot be achieved simultaneously.

2.1.1 The nonlocal DGS gradient

We first define a 1D function $G(y\,|\,\boldsymbol{x},\boldsymbol{\xi}) := F(\boldsymbol{x}+y\,\boldsymbol{\xi}), y\in\mathbb{R}$, which is a 1D cross section of $F(\boldsymbol{x})$ in Eq. (1) along the direction determined by the unit vector $\boldsymbol{\xi}\in\mathbb{R}^d$. We perform Gaussian smoothing on $G(y)$, and the smoothed function is $G_\sigma(y\,|\,\boldsymbol{x},\boldsymbol{\xi}) := \mathbb{E}_{v\sim\mathcal{N}(0,1)}\left[G(y+\sigma v\,|\,\boldsymbol{x},\boldsymbol{\xi})\right]$, which is the smoothed $F(\boldsymbol{x})$ along the direction $\boldsymbol{\xi}$ in the neighborhood of $\boldsymbol{x}$. The derivative of $G_\sigma(y|\boldsymbol{x},\boldsymbol{\xi})$ at $y=0$ is written as

$$\mathscr{D}[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi})] = \frac{1}{\sqrt{2\pi}\sigma}\int_{\mathbb{R}} F(\boldsymbol{x}+\sigma v\,\boldsymbol{\xi})\,v\,\mathrm{e}^{-\frac{v^2}{2}}\,dv, \tag{3}$$

where $\mathscr{D}$ denotes the differential operator with respect to y. For a matrix $\boldsymbol{\Xi} := (\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_d)$ consisting of d orthonormal vectors, we can define d directional derivatives like those in Eq. (3) and assemble our nonlocal gradient operator, denoted by $\nabla_{\sigma,\boldsymbol{\Xi}}[F]$, as

$$\nabla_{\sigma,\boldsymbol{\Xi}}[F](\boldsymbol{x}) := \Big[\mathscr{D}[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi}_1)], \cdots, \mathscr{D}[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi}_d)]\Big]\,\boldsymbol{\Xi}.$$

We can exploit that each component of $\nabla_{\sigma,\boldsymbol{\Xi}}[F](\boldsymbol{x})$ only involves a 1D integral so that Gauss quadrature rules can be used to approximate the integrals with high accuracy. For the Gaussian kernel in Eq. (3), we use the Gauss-Hermite (GH) quadrature rule. After changing variables in Eq. (3), a GH-based estimator of $\mathscr{D}[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi})]$ can be defined by

$$\mathscr{D}[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi})] \approx \widetilde{\mathscr{D}}^M[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi})] := \frac{1}{\sqrt{\pi}\sigma}\sum_{m=1}^{M} w_m\, F(\boldsymbol{x} + \sqrt{2}\sigma v_m\boldsymbol{\xi})\,\sqrt{2}v_m, \tag{4}$$

where w_m and v_m, $m = 1, \ldots, M$, are the GH quadrature weights and abscissae, respectively, and M is the number of function evaluations. The final estimator of the DGS gradient can be obtained by applying the GH rule to all the components of $\nabla_{\sigma,\boldsymbol{\Xi}}[F](\boldsymbol{x})$, i.e.,

$$\widetilde{\nabla}^M_{\sigma,\boldsymbol{\Xi}}[F](\boldsymbol{x}) := \Big[\widetilde{\mathscr{D}}^M[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi}_1)], \cdots, \widetilde{\mathscr{D}}^M[G_\sigma(0\,|\,\boldsymbol{x},\boldsymbol{\xi}_d)]\Big]\,\boldsymbol{\Xi}. \tag{5}$$

The DGS gradient has several key features:

- ***Accuracy***: Gaussian quadrature is much more accurate than the MC estimator, so a smaller number of samples are needed to achieve the prescribed error $\varepsilon > 0$.
- ***Nonlocality***: Large exploration radius σ can be used to capture *global* structures of the loss landscape without losing accuracy, which helps escape from local minima.
- ***Scalability***: The calculation of the DGS estimator in Eq. (5) requires $M \times d$ evaluations of $F(\boldsymbol{x})$, but those evaluations are mutually independent and completely *parallelizable*.
- ***Portability***: The DGS gradient can be integrated into most gradient-based training algorithms, including algorithms with constraints.
- ***Consistency***: Although the DGS gradient is designed for the nonlocal setting with a big σ value, it converges to the local gradient as $\sigma \to 0$, that is, $\lim_{\sigma\to 0}\left|\nabla F(\boldsymbol{x}) - \nabla_{\sigma,\boldsymbol{\Xi}}[F](\boldsymbol{x})\right| = 0$.

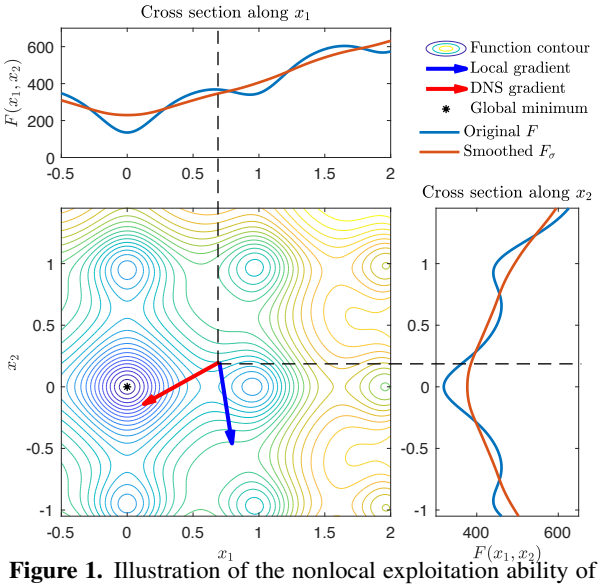


Figure 1. Illustration of the nonlocal exploitation ability of the DGS gradient. The blue arrow points to the *local* gradient direction, and the red arrow points to the DGS gradient direction where the directionally smoothed functions along the two axes are the red curves in the top and right sub-figures. Because the DGS smoothing captures the nonlocal features, the DGS gradient points to a much better direction (i.e., closer to the global minimum) than the local gradient.

2.1.2 Numerical experiments on benchmark problems

We consider the following benchmark functions (shown in 2D) whose definition can be found on https:

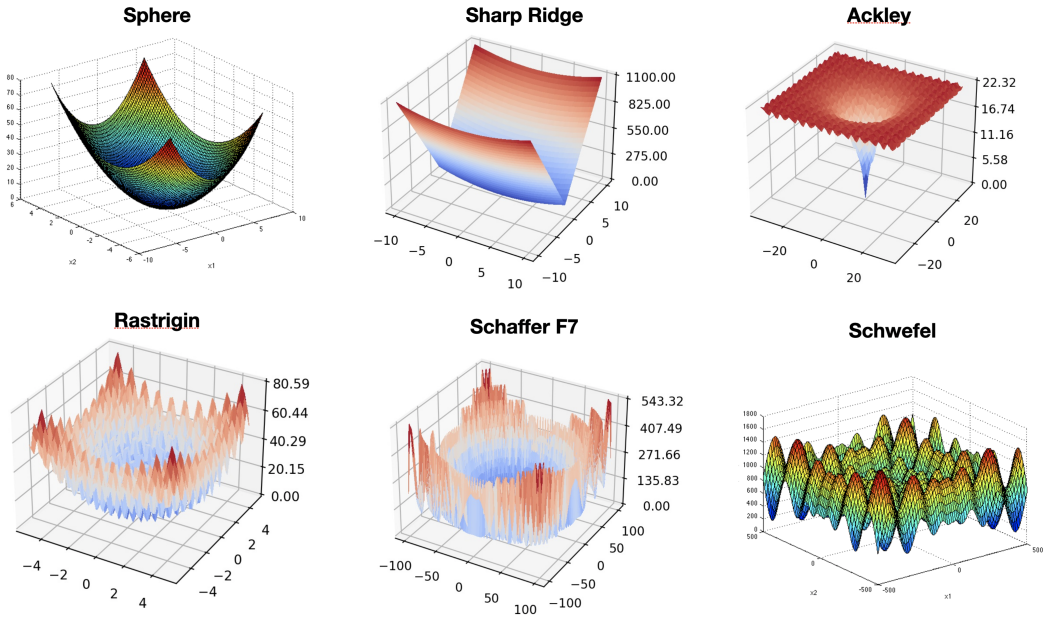


Figure 2. The 2D visualization of the benchmark functions used to test our DGS gradient descent method. The Sphere and Sharp Ridge functions are convex functions. The Ackley, Rastrigin and Schaffer F7 functions are non-convex functions with global structures; the Schwefel function is non-convex without a global structure.

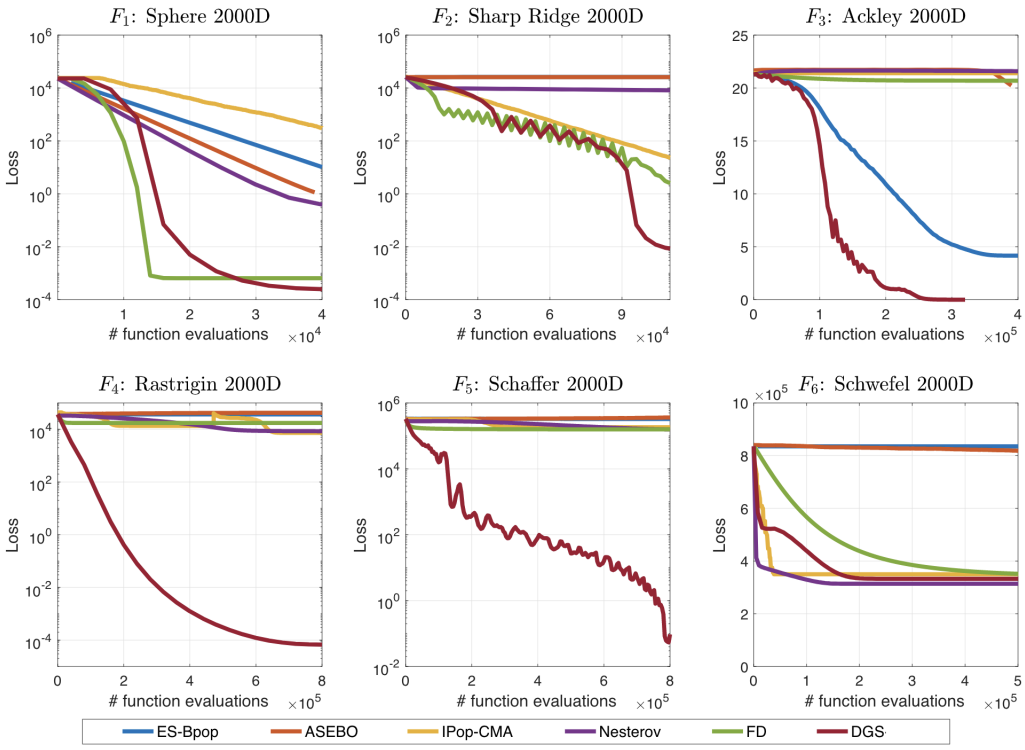


Figure 3. Figure: ES-Bpop: classic MC-based gradient estimation, ASEBO: MC-based gradient estimation with dimension reduction, IPop-CMA: CMA-ES method with random restarts, Nesterov: the Nesterov's random search, FD: finite difference, DGS: our method.

//www.sfu.ca/~ssurjano/optimization.html. We compared our method with the baselines in 2000-dimensional spaces.

The results are given in Figure 3. The DGS has the best performance overall. In particular, DGS demonstrates significantly superior performance in optimizing the highly multimodal functions, i.e., the Ackley, the Rastrigin and the Schaffer functions. For the ill- conditioned function, i.e., the Sharp Ridge function, DGS can match the performance of the best baseline method, e.g., IPop-CMA. For the Schwefel function, 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.

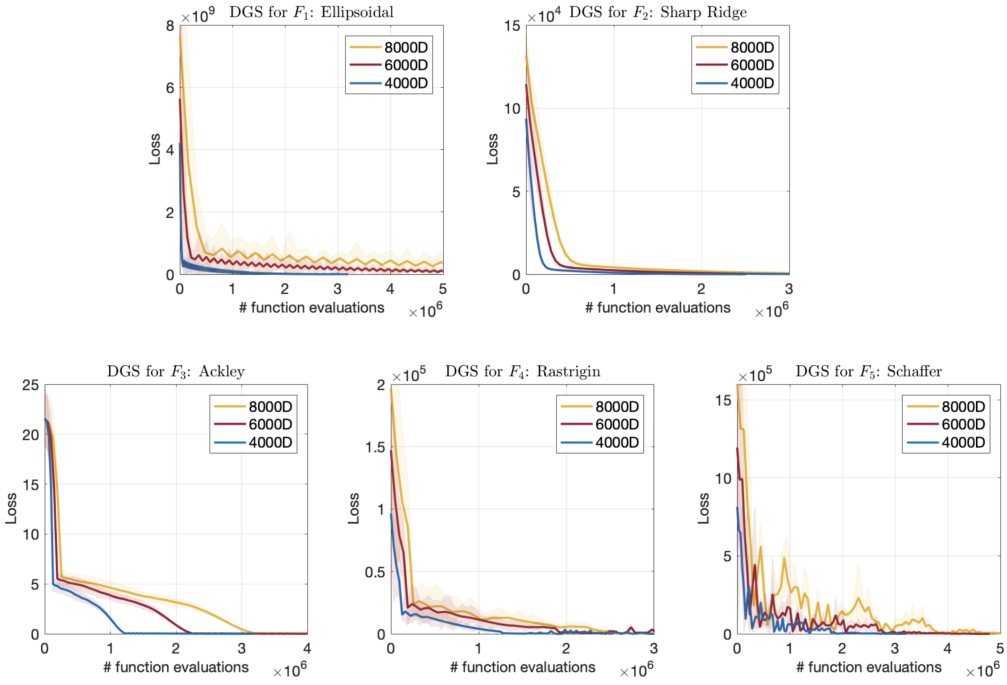


Figure 4. Tests on DGS's scalability with respect to the dimension. We test the same set of benchmark functions (except for Schwefel) in 4000D, 6000D and 8000D. The hyperparameters are the same as the 2000D case. The DGS still achieves promising performance, even though the number of function evaluations increases with the dimension.

We also test the DGS method in 4000D, 6000D and 8000D to illustrate its scalability with the dimension. We do not test Schwefel because DGS failed to optimize it in 2000D. The hyperparameters are set the same as the 2000D cases. The results are shown in Figure 4. The DGS method still achieves promising performance, even though the number of total function evaluations increases with the dimension.

2.1.3 Theoretical analysis

We first analyze the performance of the DGS gradient in the local setting (i.e., σ is very small) for minimizing convex functions, to set up a baseline. Then, we move to the nonlocal setting for minimizing non-convex functions to show the advantages of the long-range exploration of the DGS gradient.

The local setting for convex problems. In the local setting, we assume that

- $F \in C^{1,1}(\mathbb{R}^d)$ if there exists $L > 0$ such that

$$\|\nabla F(\boldsymbol{x} + \boldsymbol{\xi}) - \nabla F(\boldsymbol{x})\| \leq L\|\boldsymbol{\xi}\|, \ \forall \boldsymbol{x}, \ \boldsymbol{\xi} \in \mathbb{R}^d.$$

- $F \in C^{1,1}(\mathbb{R}^d)$ is a strongly convex function if there exists $\tau > 0$ such that for any $\boldsymbol{x}, \boldsymbol{\xi} \in \mathbb{R}^d$,

$$F(\boldsymbol{x} + \boldsymbol{\xi}) \geq F(\boldsymbol{x}) + \langle \nabla F(\boldsymbol{x}), \boldsymbol{\xi} \rangle + \frac{\tau}{2}\|\boldsymbol{\xi}\|^2.$$

Then we have the following theorem on how accurate the DGS gradient in approximating the local gradient.

Theorem 1 (The DGS gradient approximates the local gradient)**.** *Let $\boldsymbol{\Xi} = \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_d\}$ be a set of orthonormal vectors in $\mathbb{R}^d$ and F be a function in $C^{1,1}(\mathbb{R}^d)$, $\forall 1 \le i \le d$. Then*

$$\|\widetilde{\nabla}^M_{\boldsymbol{\sigma},\boldsymbol{\Xi}}[F](\boldsymbol{x}) - \nabla F(\boldsymbol{x})\|^2 \le \frac{2C_0^2\pi(M!)^2}{4^M((2M)!)^2}\sum_{i=1}^d \sigma_i^{4M-2} + 32L^2\sum_{i=1}^d \sigma_i^2,$$

where M is the number of Gauss-Hermite quadrature points, σ is the smoothing radius.

Observation: To reduce the error in approximating the local gradient, we only need to reduce the smoothing radius to as small as possible.

We now analyze how the DGS gradient performs in the gradient descent scheme, i.e.,

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \lambda\widetilde{\nabla}^M_{\boldsymbol{\sigma},\boldsymbol{\Xi}}[F](\boldsymbol{x}_t),$$

for minimizing the strongly convex loss function $F(\boldsymbol{x})$.

Theorem 2 (The accuracy of the DGS gradient descent in minimizing $F(\boldsymbol{x})$)**.** *Assume*

- *F is a strongly convex function in $C^{1,1}(\mathbb{R}^d)$,*
- *$\{\boldsymbol{x}_t\}_{t\ge 0}$ is generated by the DGS-based gradient descent with $\lambda = \dfrac{1}{8L}$.*

Then, for any $t \ge 0$, we have

$$F(\boldsymbol{x}_t) - F(\boldsymbol{x}^*) \le \frac{1}{2}L\left[\delta_{\boldsymbol{\sigma}} + \left(1 - \frac{\tau}{16L}\right)^t(\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 - \delta_{\boldsymbol{\sigma}})\right].$$

Here,

$$\delta_{\boldsymbol{\sigma}} = \left(\frac{128}{\tau^2} + \frac{16}{\tau L}\right)L^2\sum_{i=1}^d \sigma_i^2 + \left(\frac{8}{\tau^2} + \frac{1}{2\tau L}\right)\frac{C_0^2(M!)^2\pi}{4^M((2M)!)^2}\sum_{i=1}^d \sigma_i^2,$$

where τ, $\boldsymbol{\sigma}$ are the convexity and smoothing parameters and M is the number of Gauss-Hermite samples.

Observation: We need to reduce the error δ_σ to achieve the convergence, i.e., $F(\boldsymbol{x}_t) \to F(\boldsymbol{x}^*)$. Again, in the local setting, all we need to do is to reduce the smoothing radius to a very small value. Therefore, even though the DGS-based gradient descent has comparable performance to other methods, e.g., the finite difference, the DGS gradient does not show its long-range exploration advantage in the local setting.

The nonlocal setting for non-convex problems. The situation is significantly changed when optimizing non-convex functions. Here, we assume the loss function $F = \phi + \eta$ is a noisy approximation of a convex function ϕ

- $\phi \in C^{1,1}(\mathbb{R}^d)$: $\|\nabla\phi(\boldsymbol{x}+\boldsymbol{\xi}) - \nabla\phi(\boldsymbol{x})\| \le L\|\boldsymbol{\xi}\|$, $\forall \boldsymbol{x}, \boldsymbol{\xi} \in \mathbb{R}^d$.
- ϕ is strongly convex $\phi(\boldsymbol{x}+\boldsymbol{\xi}) \ge \phi(\boldsymbol{x}) + \langle\nabla\phi(\boldsymbol{x}), \boldsymbol{\xi}\rangle + \frac{\tau}{2}\|\boldsymbol{\xi}\|^2$, $\forall \boldsymbol{x}, \boldsymbol{\xi} \in \mathbb{R}^d$.

We consider two scenarios of the noise function η, i.e.,

1. *periodic:* cross-sections of η along $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_d$ are periodic,
2. *band-limited:* the power spectra of cross-sections of η along $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_d$ possess a uniform positive lower bound,

We proved the following two theorems for the two scenarios of the noise function.

Theorem 3 (Periodic noise)**.** *Let $F = \phi + \eta$, where ϕ is strongly convex and*

- *cross sections $\eta(\cdot|\boldsymbol{x}, \boldsymbol{\xi}_i)$ of η along $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_d$ are periodic functions with period $1/\alpha$*
- *$|\eta^{(n)}(y|\boldsymbol{x}, \boldsymbol{\xi}_i)| \leq C, \ \forall y \in \mathbb{R}, \ i \in \{1, \ldots, d\}$, with $n = 1$ or $n = 2$.*

Let $\{\boldsymbol{x}_t\}_{t\geq 0}$ be generated by DGS-based gradient descent with $\lambda = 1/(16L)$ and $\boldsymbol{x}^*$ be the global minimum of ϕ. Then, for any $t \geq 0$, we have

$$\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2 \leq \left(1 - \frac{\tau}{32L}\right)^{t+1} \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + \delta_{\boldsymbol{\sigma}}, \tag{6}$$

$$\text{with } \ \delta_{\boldsymbol{\sigma}} = \left(\frac{4}{\tau^2} + \frac{1}{4L\tau}\right)\left(C\frac{\pi(M!)^2 d}{4^M((2M)!)^2}\sigma^{4M-2} + 48L^2 d\sigma^2 + Cde^{-4\pi^2\alpha^2\sigma^2} r(\alpha, \sigma)\right),$$

where $r(\alpha, \sigma) = \left(1 + \frac{1}{\alpha^2\sigma^2}\right)$ if $n = 1$ and $r(\alpha, \sigma) = \left(1 + \log^2\left(1 + \frac{1}{2\pi^2\alpha^2\sigma^2}\right)\right)$ if $n = 2$.

Theorem 4 (band-limited noise)**.** *Let $F = \phi + \eta$, where ϕ is strongly convex and the cross sections $\eta(\cdot|\boldsymbol{x}, \boldsymbol{\xi}_i)$ of η along $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_d$ are high frequency signals with power spectrum being zero on $(-\alpha, \alpha)$ and uniformly bounded by K on $\mathbb{R}$. Let $\{\boldsymbol{x}_t\}_{t\geq 0}$ be generated by DGS-based gradient descent with $\lambda = 1/(16L)$ and $\boldsymbol{x}^*$ be the global minimum of ϕ. Then, for any $t \geq 0$, we have*

$$\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2 \leq \left(1 - \frac{\tau}{32L}\right)^{t+1} \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + \delta_{\boldsymbol{\sigma}}, \tag{7}$$

$$\text{with } \ \delta_{\boldsymbol{\sigma}} = \left(\frac{4}{\tau^2} + \frac{1}{4L\tau}\right)\left(C\frac{\pi(M!)^2 d}{4^M((2M)!)^2}\sigma^{4M-2} + 48L^2 d\sigma^2 + \frac{CK^2 d}{\sigma^4} e^{-4\pi^2\alpha^2\sigma^2}\right).$$

Observation: The quantity δ_σ measures the error between the current state $\boldsymbol{x}_t$ and the optimal state $\boldsymbol{x}^*$. Unlike the local setting in Theorem 2, the error δ_σ has an extra term (in red) that comes from the noise function η. We can see that we need to use a relatively large smoothing radius σ to balance the three error terms. Moreover, the lower the noise's frequency α, the larger the smoothing radius σ should be to balance the total error. This justifies the importance of the long-range exploration in minimizing functions with low-frequency noises.

2.2 AdaDGS: the adaptive nonlocal gradient descent algorithm

2.2.1 The AdaDGS algorithm

In the DGS gradient descent scheme, i.e.,

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \lambda_t \widetilde{\nabla}^{\boldsymbol{M}_t}_{\sigma_t, \boldsymbol{\Xi}_t}[F](\boldsymbol{x}_t), \tag{8}$$

we have four hyperparameters, i.e., the orthogonal coordinate system $\boldsymbol{\Xi}_t$, the number of GH quadrature points $\boldsymbol{M}_t$, the smoothing radius σ_t, and the learning rate λ_t, in the DGS gradient descent method in Eq.(8). We develop an approach to adaptively adjust these hyperparameters.

Adaptive $\boldsymbol{\Xi}_t$ for nonlocal exploration using second-order information. The local gradient is independent of the choice of the coordinate system, but the DGS gradient depends on $\boldsymbol{\Xi}_t$ when having a big smoothing

radius σ_t. We observed that the performance of the DGS gradient descent is not satisfactory for ill-conditioned problems, e.g., the Rosenbrock function. We can use second-order information to precondition the DGS gradient. A natural choice is to incorporate the DGS gradient into the BFGS framework. However, because we need to take the difference $\boldsymbol{y}_t = \widetilde{\nabla}^{\boldsymbol{M}_t}_{\sigma,\boldsymbol{\Xi}}[F](\boldsymbol{x}_{t+1}) - \widetilde{\nabla}^{\boldsymbol{M}_t}_{\sigma,\boldsymbol{\Xi}}[F](\boldsymbol{x}_t)$ in each iteration, we cannot change the orthogonal system $\boldsymbol{\Xi}$. Inspired by the fact that the inverse Hessian is the covariance matrix when the loss function is Gaussian, we incorporate the covariance matrix adaptation (CMA) into our method to adjust the coordinate system $\boldsymbol{\Xi}_t$. We start with an identity covariance matrix for $t = 0$, i.e., $\mathbf{C}_0 = \mathbf{I}$, and set the orthogonal system $\boldsymbol{\Xi}_t$ to be the orthogonal system determined by the eigenvectors of $\mathbf{C}_t$. When $t > 0$, we update the covariance matrix $\mathbf{C}_t$ by

- Generate β candidate samples, denoted by $\boldsymbol{z}_i \in \mathbb{R}^d$, from $\mathcal{N}(\boldsymbol{x}_t, \sigma_t^2 \mathbf{C}_t)$;
- Sort the samples based on their loss values,

$$\{\boldsymbol{z}_{i:\beta} \,|\, i = 1, \ldots, \beta\} = \{\boldsymbol{z}_i \,|\, i = 1, \ldots, \beta\} \;\text{ with }\; F(\boldsymbol{z}_{1:\beta}) \le \cdots \le F(\boldsymbol{z}_{\beta:\beta});$$

- Update $\mathbf{C}_t$ using the best μ candidates $\boldsymbol{z}_{1:\beta}, \ldots, \boldsymbol{z}_{\mu:\beta}$ with $\mu < \beta$, i.e.,

$$\mathbf{C}_{t+1} = (1 - c_\mu)\mathbf{C}_t + c_\mu \sum_{i=1}^{\mu} w_i \frac{\boldsymbol{z}_{i:\beta} - \boldsymbol{x}_t}{\sigma_t} \left(\frac{\boldsymbol{z}_{i:\beta} - \boldsymbol{x}_t}{\sigma_t} \right)^T.$$

Once $\mathbf{C}_{t+1}$ is obtained, we perform eigendecomposition of the covariance matrix $\mathbf{C}_{t+1}$ and define the orthogonal system $\boldsymbol{\Xi}_{t+1}$ to be the matrix consisting of the eigenvectors, i.e.,

$$\mathbf{C}_{t+1} = \mathbf{U}_{t+1}\mathbf{S}_{t+1}\mathbf{U}_{t+1}^T \;\text{ and }\; \boldsymbol{\Xi}_{t+1} = \mathbf{U}_{t+1}.$$

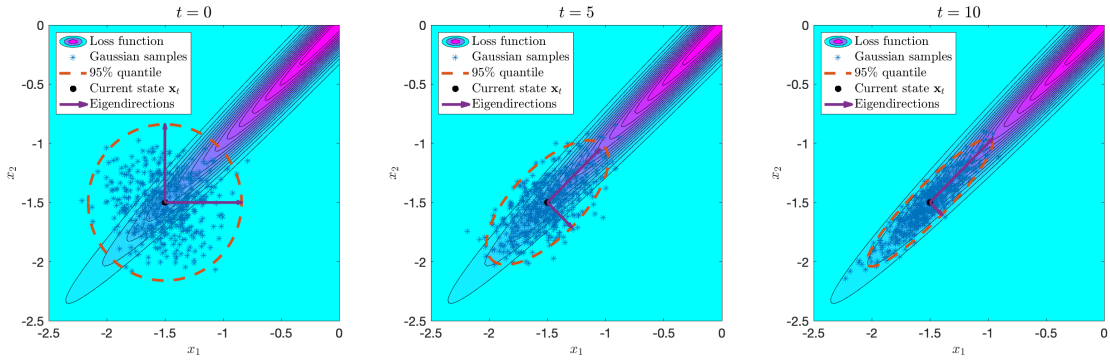


Figure 5. Illustration of covariance matrix adaptation for capturing the second-order information of the loss landscape. For simplicity, we fix the state $\boldsymbol{x}_t$ and only update $\mathbf{C}_t$. The contour represents an ellipsoidal loss landscape; the stars represent the samples drawn from the Gaussian distribution $\mathcal{N}(\boldsymbol{x}_t, \sigma_t^2 \mathbf{C}_t)$. The red dashed lines represent the 95% quantile of $\mathcal{N}(\boldsymbol{x}_t, \sigma_t^2 \mathbf{C}_t)$. It can be seen that the covariance matrix adaptation successfully captures geometry of the loss function. Exploring along the eigendirections to compute the DGS gradient will accelerate the optimization process.

Adaptive M_t for dimension reduction We can exploit the eigenvalues in $\mathbf{S}_{t+1}$ to perform dimension reduction, i.e., truncating the directions with very small eigenvalues, and only computing the directional derivatives of the DGS gradient along the important directions. Specifically, we denote by $s_{t+1,1}, \ldots, s_{t+1,d}$ the diagonal entries of $\mathbf{S}_{t+1}$, i.e., the eigenvalues of the covariance matrix $\mathbf{C}_{t+1}$. For each dimension j, we set the number of GH quadrature points $M_{t+1,j} = 0$ if the eigenvalue $s_{t+1,j}$ satisfies

$$\frac{s_{t+1,j}}{\sum_{j=1}^{d} s_{t+1,j}} < \gamma,$$

where $\gamma \in (0, 1)$ is a prescribed threshold. Using zero quadrature points in the j-th direction, i.e., $M_{t+1,j} = 0$, is equivalent of setting the corresponding partial derivative $\widetilde{\mathscr{D}}^{M_{t+1,j}}[G_{\sigma_t}(0 \,|\, \boldsymbol{x}, \boldsymbol{\xi}_{t,j})]$ to zero. This strategy can help reduce the overall computational cost for calculating the DGS gradient.

Adaptive λ_t: nonlocal and local backtracking line search We introduce a two-stage, i.e., nonlocal and local, backtracking line search approach to choose the learning rate λ_t in each iteration. For multimodal landscapes, choosing one candidate solution along the DGS gradient direction according to a *single* learning rate may make insufficient progress. The backtracking line search is easy to implement and help overcoming the sensitivity to the learning rate selection that affects the performance of the original DGS method.

Adaptive smoothing radius σ_t The adaptation of σ_t needs to satisfy several properties to be able to handle different types of loss landscapes. First, σ_t needs to be relatively large in the early phase of the optimization for nonlocal exploration and skipping local minimum. Second, σ_t needs to converge to zero as the state $\boldsymbol{x}_t$ approaches the global minimum. Third, a reset strategy is needed to reset σ_t to the initial value, just in case that $\boldsymbol{x}_t$ is trapped in a local minimum due to overly fast shrinking of σ_t.

We observe that the learning rate λ_t, obtained from the line search, is a good indicator of the distance between the current state and the global minimum. In this work, the smoothing radius σ_t is adjusted based on the learning rate learned from the line search. The initial radius σ_0 is set to be on the same scale as the width of the search domain. At iteration t, we set σ_t to be the mean of the smoothing radius and the learning rate from iteration t, i.e.,

$$\sigma_{t+1} = \frac{1}{2}\left(\sigma_t + \frac{1}{s}\sum_{i=t-s}^{t} \|\boldsymbol{x}_i - \boldsymbol{x}_{i-1}\|\right), \tag{9}$$

because both quantities indicate the landscape of the loss function. In this case, σ_t will gradually converge to zero as $\boldsymbol{x}_t - \boldsymbol{x}_{t-1}$ goes to zero. On the other hand, to verify if $\boldsymbol{x}_t$ is trapped in a local minimum, we will reset σ_{t+1} to σ_0 when $\frac{1}{s}\sum_{i=t-s}^{t} \|\boldsymbol{x}_i - \boldsymbol{x}_{i-1}\|$ is smaller than a threshold. In this case, we restart the nonlocal exploration. If the best state is not improved by restarting the nonlocal exploration, we accept the current best state as the final optimal state.

Algorithm 1 : The AdaDGS algorithm

Input: the initial guess $\boldsymbol{x}_0$, the initial covariance matrix $\mathbf{C}_0$, the initial smoothing radius σ_0, the initial orthogonal system $\boldsymbol{\Xi}_0$; shrinking factors τ_{local}, τ_{nonlocal}.

for $t = 0, \ldots,$ **do**
  Compute the DGS gradient $\widetilde{\nabla}^{\boldsymbol{M}_t}_{\sigma_t, \boldsymbol{\Xi}_t}[F](\boldsymbol{x}_t)$ using Eq. (5);
  Nonlocal and local line search to find λ_t satisfying the Armijo-Goldstein condition;
  Update $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \lambda_t \widetilde{\nabla}^{\boldsymbol{M}_t}_{\sigma_t, \boldsymbol{\Xi}_t}[F](\boldsymbol{x}_t)$;
  Update the smoothing radius σ_t using Eq. (9);
  Initialize $\mathbf{C}_t^0 = \mathbf{C}_t$;
  for $k = 1, \ldots, K$ **do**
    Draw samples $\{\boldsymbol{z}_i | i = 1, \ldots, \beta\}$ from $\mathcal{N}(\boldsymbol{x}_{t+1}, \sigma_t \mathbf{C}_t^k)$;
    Sort the samples to be $\{\boldsymbol{z}_{i:\beta} \,|\, i = 1, \ldots, \beta\}$ with $F(\boldsymbol{z}_{1:\beta}) \le \cdots \le F(\boldsymbol{z}_{\beta:\beta})$;
    Update the covariance matrix $\mathbf{C}_t^k$;
  end for
  Set $\mathbf{C}_{t+1} = \mathbf{C}_t^K$;
  Perform eigendecomposition of $\mathbf{C}_{t+1} = \mathbf{U}_{t+1}\mathbf{S}_{t+1}\mathbf{U}_{t+1}^T$;
  Update $\boldsymbol{\Xi}_{t+1} = \mathbf{U}_{t+1}$;
  Reduce the dimension of DGS by setting $m = 0$;

2.2.2 Numerical experiments on benchmark problems

We tested the AdaDGS method on 12 benchmark function. Details on the test can be found on our github page https://github.com/HoangATran/AdaDGS. The results are shown in Figure 6. The AdaDGS has the best performance overall. In particular, AdaDGS demonstrates significantly superior performance in optimizing the highly multimodal functions F1, F2, F4, F5, F7, F9, F10, F11, which is significant in global optimization. For the ill-conditioned functions F4 and F8, AdaDGS can at least match the performance of the best baseline method, e.g., IPop-CMA. For F12, all the methods fail to find the global minimum because it's highly multi-modal and there is no global structure to exploit, which makes it extremely challenging for all global optimization methods.

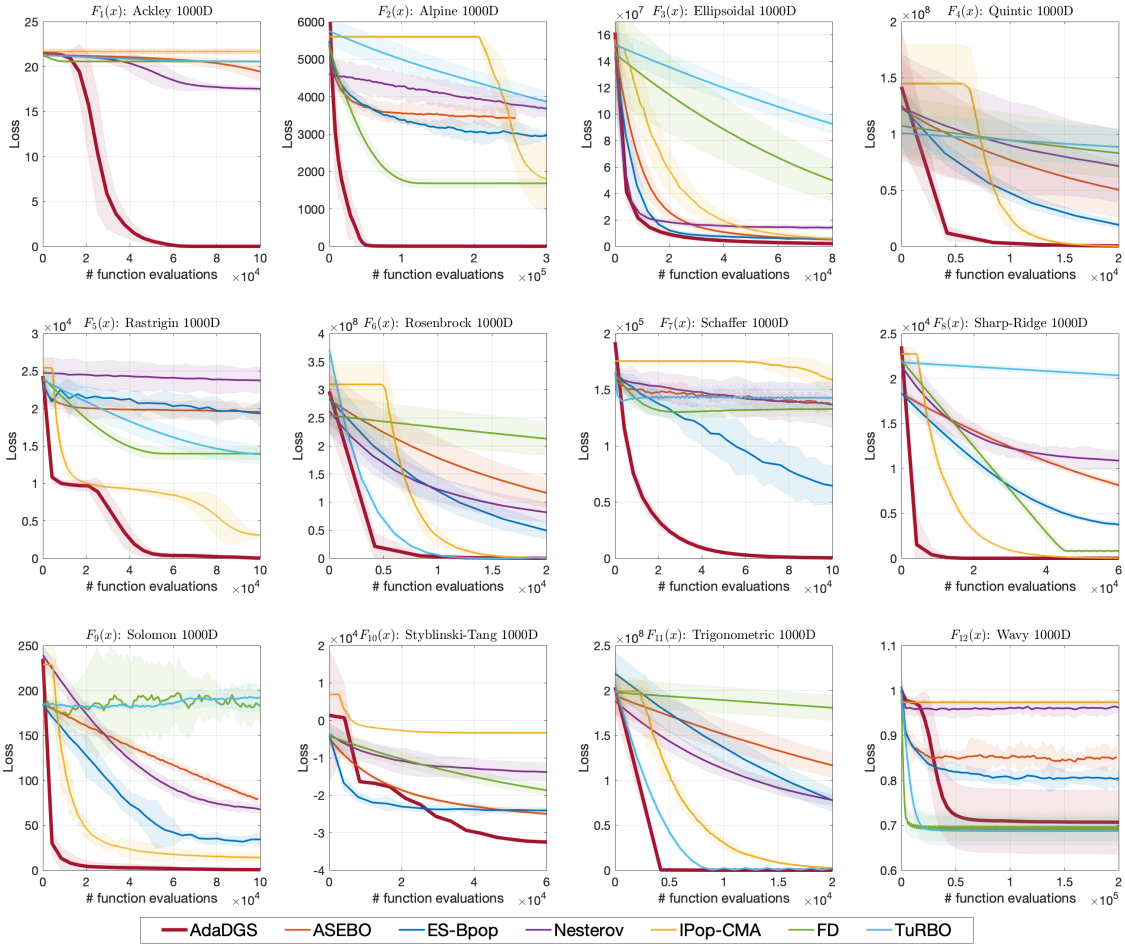


Figure 6. Comparison of the loss decay w.r.t. function evaluations for the 12 benchmark functions in 1000D. Each curve is the mean of 20 independent trials and the shaded areas represent [mean3std, mean+3std].

3. Scientific applications and impacts

3.1 Model calibration of the liquid mercury spallation target at SNS

This is a collaboration with a BES-funded project on machine learning for improving accelerator and neutron target performance. The mercury constitutive model predicting the strain and stress in the target vessel plays a central role in improving the lifetime prediction and future target designs of the mercury targets at the Spallation Neutron Source (SNS). We leverage the experiment strain data collected over multiple years

to improve the mercury constitutive model through a combination of large-scale simulations of the target behavior and the use of machine learning tools for parameter estimation. We developed two interdisciplinary approaches for surrogate-based model calibration of expensive simulations using evolutionary neural networks and sparse polynomial expansions, and then use the DGS gradient descent algorithm to find the optimal parameters. The experiments and results of the two methods show a very good agreement for the solid mechanics simulation of the mercury spallation target. The proposed methods are used to calibrate the tensile cutoff threshold, mercury density, and mercury speed of sound during intense proton pulse experiments. Using strain experimental data from the mercury target sensors, the newly calibrated simulations achieve 7% average improvement on the signal prediction accuracy and 8% reduction in mean absolute error compared to previously reported reference parameters, with some sensors experiencing up to 30% improvement. The proposed calibrated simulations can significantly aid in fatigue analysis to estimate the mercury target lifetime and integrity, which reduces abrupt target failure and saves a tremendous amount of costs.

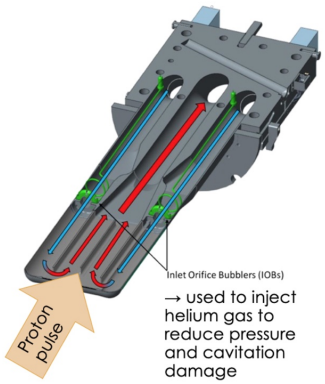


Figure 1: The liquid mercury spallation target.

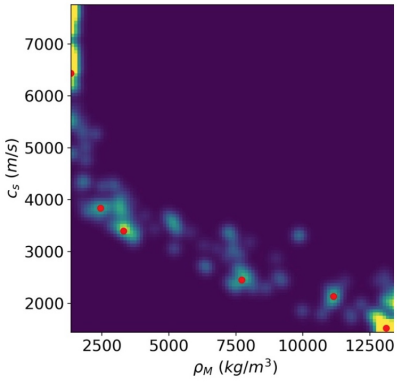


Figure 2: The distribution of the candidate parameters informed by polynomial surrogates. This distribution is multi-modal and the best candidate parameters are selected as the peak of each mode.

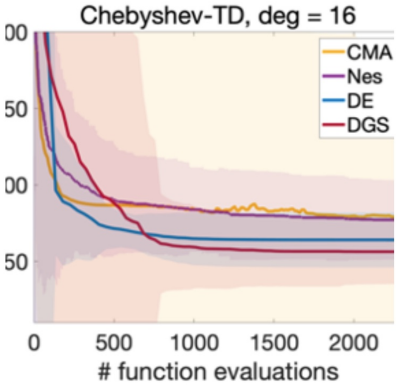


Figure 3: Performance of our method (DGS) compared to other baselines in optimizing on the surrogate. Each curve is the mean of 50 trials. DGS reliably achieves the lowest loss value.

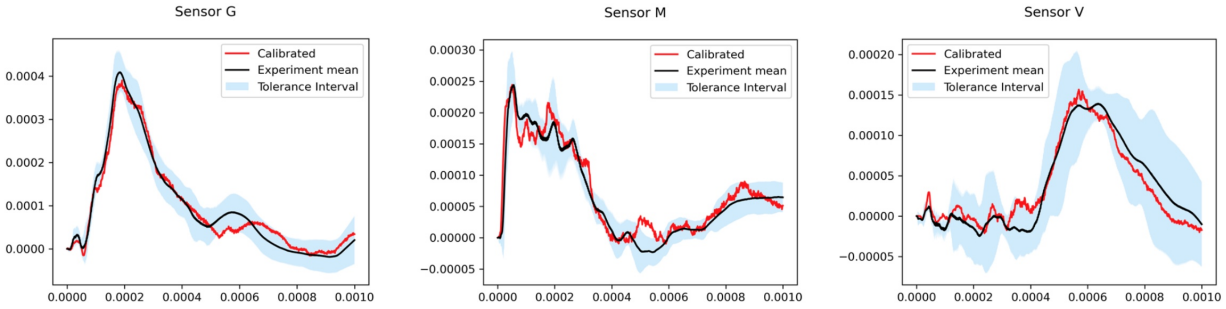
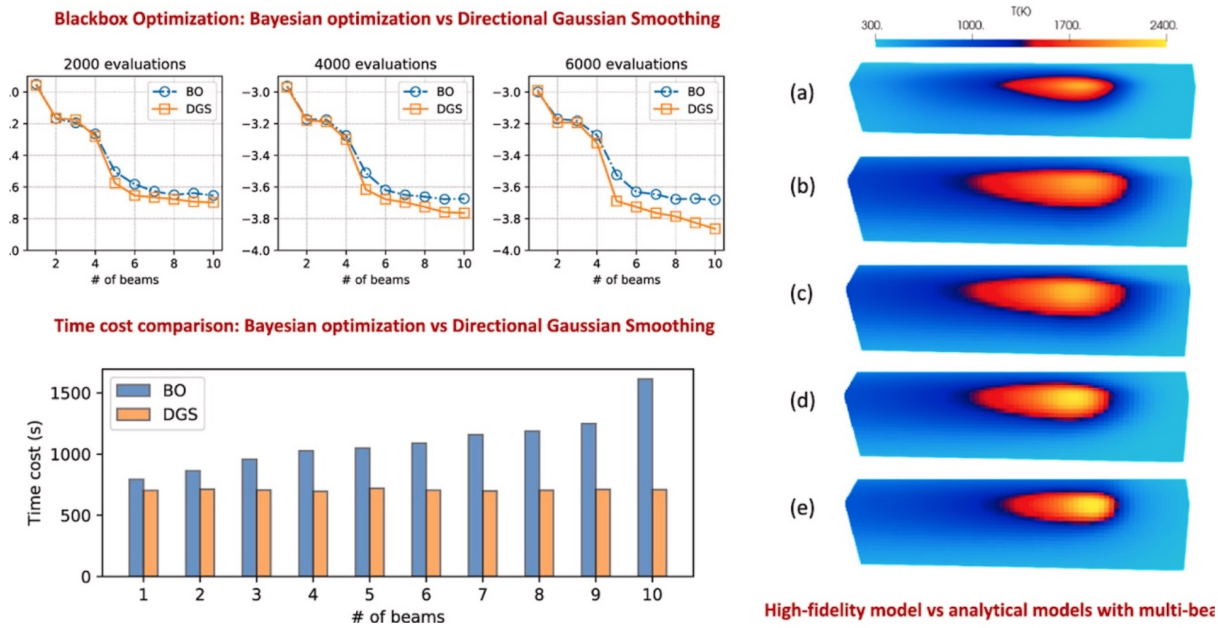


Figure 4: Results of the calibrated simulation against the experiment strain data.

3.2 Training heat conduction models for additive manufacturing simulation at MDF

This is a collaboration with the ExaAM project in the Exascale Computing Project. Heat transfer simulations play an important role in predicting solidification conditions that determine microstructure properties of additive manufactured products. High-fidelity heat transfer models, e.g., Truchas, provide a numerical

framework capable of resolving many desired physics (e.g., thermal radiation, elemental vaporization, gas-solid-liquid interactions, etc.), but they are often too computationally expensive to simulate the solidification conditions at the AM process-scale. Recently, Stump et al. proposed a low-fidelity analytical heat conduction model capable of simulating solidification conditions on the process-scale. This model makes the problem analytically tractable by neglecting many important physics that may have an impact on the accuracy of the predicted solidification conditions. An effective way of improving the accuracy of the analytical with little computational effort is to determine an optimal set of parameters that minimize the discrepancy between the low-fidelity analytical model and a high-fidelity numerical model. In this work, a model approximation problem is formulated which can be solved by leveraging the black-box methods. Multiple heat sources are used in the analytical model to improve the calibration performance in a flexible framework. Two blackbox optimization methods, i.e., Bayesian optimization and DGS gradient descent methods, are employed to address the challenge that the gradient of the loss function is inaccessible during optimization. The results show that the single-beam analytical model has limitations in fitting the high-fidelity model, but the proposed multi-beam analytical model provides satisfactory approximations to the high-fidelity temperature and melt pool fields.



Left-top Figure: Total loss performance comparison of different computational budget ranging from 2000 to 6000. For each budgets, we show the performance for one to ten beams. **Left-bottom Figure**: Computational time cost comparison of the BO and DGS on high-fidelity model approximation with analytical multi-beam model. **Right Figure**: Comparison of Ground truth results (a), using TruchasPBF and calibrated melt pool field using (b) one beam (c) two beams, (d) five beams and (e) ten beams.

3.3 Black-box adversarial attack against AI models

Black-box adversarial attack is used to test the robustness and find the vulnerability of AI models. A better attacking algorithm can stimulate the development of more advanced defense approaches. Black-box methods require a massive amount of queries to find a successful adversarial perturbation. Since each query to the target model costs time and money, query efficiency is a requisite for any practical black-box attack method. Recent years have seen the development of several black-box approaches with significant improved query efficiency. However, current black-box attacks access the target models only at perturbed samples and

completely rely on the queries there to update the perturbation at each iteration. To reduce the number of queries, it would be beneficial to be able to make use of these queries to extract more from the models, inferring the loss values and identifying candidate perturbations, where no model query was made. This is a challenging goal: since the landscapes of adversarial losses are often complicated and not well-understood, the accuracy of approximations of the loss values from available model queries is not guaranteed.

We develop a new l_2 black-box adversarial attack on frequency domain, which uses an interpolation scheme to approximate the loss value around the current state and use the DGS gradient to guide the state update. This algorithm is inspired by our observation that for many standard and robust image classifiers, the adversarial losses behave like parabolas with respect to perturbations of an image in the Fourier domain, thus can be captured with quadratic interpolation. We treat the adversarial attack problem as a constraint optimization on an l_2 sphere, and sample along geodesic curves on the sphere. Our method achieves significantly improved query efficiency because the perturbation updates are now informed not only directly from model queries (as in existing approaches), but also from an accurate quadratic approximation of the adversarial loss around the current state. The main contributions of this work can be summarized as follows:

- Theoretical justifications on the fact that the adversarial loss behaves like a parabola in the Fourier domain, but not like a parabola in the pixel domain.
- Development of a random-search-based black-box adversarial attack method that exploits the parabolic loss landscape to improve the query efficiency.
- Extensive evaluations of our method with targeted and untargeted attacks on MNIST, CIFAR-10 and ImageNet datasets with both standard and defended models.

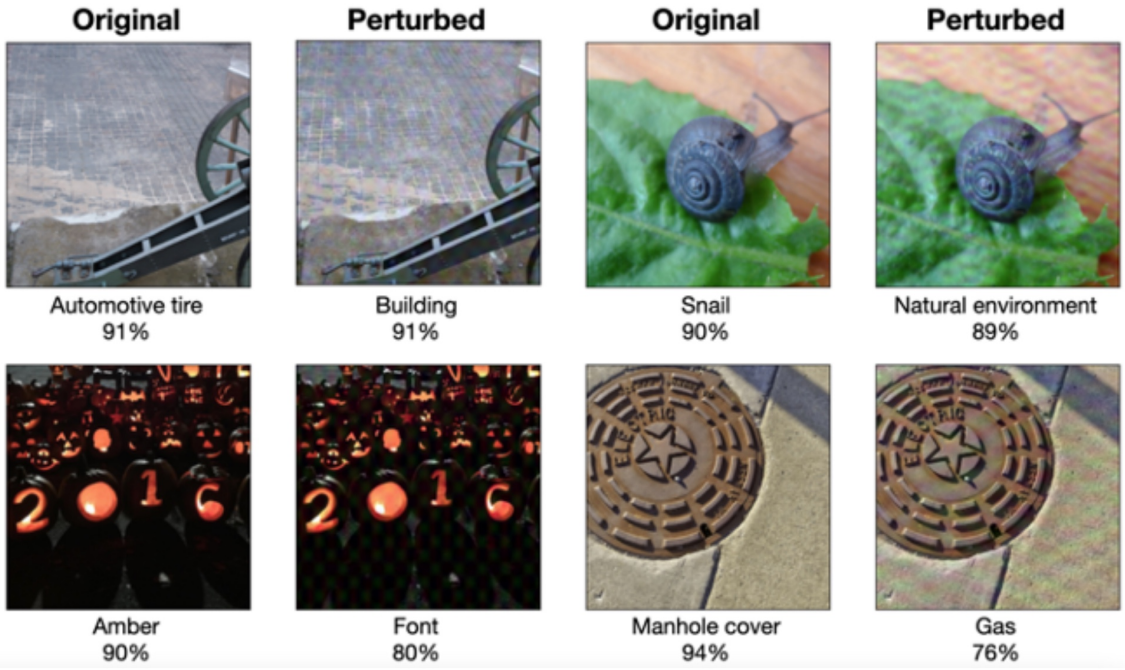


Figure 1: Examples of attacking the Google Cloud Vision API to remove top-3 labels . The original images and the perturbed images by our methods are shown with their top labels and probabilities.

4. Software

The source code of the AdaDGS algorithm has been uploaded to the github page: https://github.com/HoangATran/AdaDGS. The numerical experiments on benchmark functions in our papers can be repeated using one simple command. It is also straightforward for users to use our code by replacing the benchmark functions with other loss functions.

5. Outlook and future plan

There are several research paths we plan on exploring in the future. First, we will integrate the smoothing idea into stochastic gradient descent as a variance reduction approach. In large-scale neural network training problems, we cannot use a large mini-batch due to the GPU's memory limit. When the batch size is not sufficiently big, the stochasticity injected into the gradient is large, such that the loss decay becomes very fluctuating. The Gaussian smoothing can be used to reduce such fluctuation by taking the average of the gradient samples (with the same mini-batch) in the neighbourhood of the current parameter state. This strategy does not require moving more data to GPU's memory, i.e., no additional communication cost, so that it does not affect the scalability of the training algorithm. Second, we will explore the performance of the DGS gradient descent in ML surrogate-based optimization. When using a neural network to build a surrogate to the loss function, we cannot guarantee a good accuracy in the approximation of the first-order and second-order information (i.e., the Hessian). In this case, the second-order optimization methods using inaccurate Hessian approximation may fail to converge. We will investigate the performance of the DGS gradient with CMA preconditioning in this setting. Our observation is that the CMA approach can smooth out some approximation errors, which could make the DGS gradient descent converge faster to the optimum, especially for highly ill-conditioned problems.

6. List of publications

- H. Tran and G. Zhang, *An adaptive nonlocal gradient descent method for high-dimensional black-box optimization*, SIAM Journal on Scientific Computing, under review.
- J. Zhang, H. Tran, D. Lu, and G. Zhang, *Enabling long-range exploration in minimization of multimodal functions*, Proceedings of 37th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR 161: 1639-1649, 2021.
- H. Tran, D. Lu, and G. Zhang, *Exploiting the local parabolic landscapes of adversarial losses to accelerate black-box adversarial attack*, Proceedings of 17th European Conference on Computer Vision (ECCV 2022), pp 317–334, 2022.
- S. Bi, B. Stump, J. Zhang, Y. Lee, J. Coleman, M. Bement, G. Zhang, *Black-box optimization for approximating high-fidelity heat transfer calculations in metal additive manufacturing*, Results in Materials, 13, pp. 100258, 2022.
- M. Radaideh, H. Tran, L. Lin, H. Jiang, D. Winder, S. Gorti, G. Zhang, J. Mach, S. Cousineau, *Model Calibration of the Liquid Mercury Spallation Target using Evolutionary Neural Networks and Sparse Polynomial Expansions*, Nuclear Inst. and Methods in Physics Research B, 525(15), pp. 41-54, 2022.
- J. Zhang, S. Bi, and G. Zhang, *A directional Gaussian smoothing optimization method for computational inverse design in nanophotonics*, Materials Design, 197 (1), pp. 109213, 2021.

- J. Zhang, H. Tran, and G. Zhang, *Accelerating reinforcement learning with a directional-Gaussian-smoothing evolution strategy*, Electronic Research Archive, 29(6), pp. 4119-4135, 2021.
- H. Tran, D. Lu and G. Zhang, *Boosting black-box adversarial attack via exploiting loss smoothness*, Proceedings of ICLR Workshop on Security and Safety in Machine Learning Systems, 2021.
- J. Zhang, S. Bi, and G. Zhang, *A nonlocal-gradient descent method for inverse design in nanophotonics*, Proceedings of NeurIPS 2020 Workshop on Machine Learning for Engineering Modeling, Simulation and Design, 2020.
- Sirui Bi, Jiaxin Zhang and Guannan Zhang, *Towards efficient uncertainty estimation in deep learning for robust energy prediction in materials chemistry*, Proceedings of ICLR Workshop on Deep Learning for Simulation, 2021.

7. Conference and workshop presentations

- In Octorber 2022, H. Tran presented our work on "Exploiting the local parabolic landscapes of adversarial losses to accelerate black-box adversarial attack" at the 17th European Conference on Computer Vision (ECCV 2022).
- In July 2022, G. Zhang presented our work on "a nonlocal gradient for high-dimensional black-box optimization" at the SIAM Annual Meeting.
- In April 2022, H. Tran presented our work on "a nonlocal gradient descent method for high-dimensional black-box optimization" at the SIAM conference on UQ.
- In July 2021, G. Zhang presented our work on "a nonlocal gradient descent method for high-dimensional black-box optimization" at SIAM Annual Meeting.
- In May 2021, S. Bi presented the work on "A directional Gaussian smoothing optimization method for computational inverse design in nanophotonics" at the ICLR 2021.
- In May 2021, J. Zhang presented the work on "Towards efficient uncertainty estimation in deep learning for robust energy prediction in materials chemistry" at the ICLR Workshop on Deep Learning for Simulation, 2021.
- In April 2021, G. Zhang presented our work on "Enabling long-range exploration in minimization of multimodal functions" at the 37th Conference on Uncertainty in Artificial Intelligence (UAI).