Multi-fidelity Constrained Optimization for Stochastic Black-Box Simulators

Atul Agrawal*, **Phaedon-Stelios Koutsourelakis**
Professorship of Data-Driven Materials Modelling
Technical University of Munich
Munich, Germany
{atul.agrawal,p.s.koutsourelakis}@tum.de

Kislaya Ravi*, **Hans-Joachim Bungartz**
Chair of Scientific Computing
Technical University of Munich, Germany
{kislaya.ravi,bungartz}@tum.de

Abstract

Constrained optimization of the parameters of a simulator plays a crucial role in a design process. These problems become challenging when the simulator is stochastic, computationally expensive, and the parameter space is high-dimensional. One can efficiently perform optimization only by utilizing the gradient with respect to the parameters, but these gradients are unavailable in many legacy, black-box codes. We introduce the algorithm Scout-Nd (Stochastic Constrained Optimization for N dimensions) to tackle the issues mentioned earlier by efficiently estimating the gradient, reducing the noise of the gradient estimator, and applying multi-fidelity schemes to further reduce computational effort. We validate our approach on standard benchmarks, demonstrating its effectiveness in optimizing parameters highlighting better performance compared to existing methods.

1 Introduction

Physics-based simulators are used across fields of engineering and science to drive research [1], and more recently used to generate synthetic training data for machine learning related tasks [2]. A common challenge is finding optimum parameters given some objective subject to some constraints. High-dimensional parameter space and stochastic objective function make the optimization non-trivial.

Gradient-based methods have been shown to work well when the derivative is available [3, 4, 5, 6]. However, for optimization/inference tasks involving physics-based simulators, only black-box evaluations of the objective are often possible(e.g., legacy solvers). It is commonly called simulation based inference(SBI)/optimization [1, 7]. In such cases, one resort to gradient-free optimization [8], for example, genetic algorithms [9] or Bayesian Optimization and their extensions [10, 11]. The gradient-free methods perform poorly on high-dimensional parametric spaces [8]. More recently, stochastic gradient estimators [12] have been used to estimate gradients of black-box functions and, hence, perform gradient-based optimization [13, 14, 15, 16]. However, they do not account for the constraints.

This work introduces a novel approach for constrained stochastic optimization involving stochastic black-box simulators with high-dimensional parametric dependency. We draw inspiration from [17,

*equal contribution

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

18] to estimate the gradients, extended it to include constraints and employed multi-fidelity strategies to limit the number of function calls, as the cost of running the simulator can be high. We choose popular gradient-free constrained optimization methods like Constrained Bayesian Optimization (cBO)[19] and COBYLA [20] to compare our method on standard benchmark problems.

2 Methodology

Problem statement We are given a scalar valued function $f(\boldsymbol{x}, \boldsymbol{b})$ and a set of constraints $\mathcal{C}(\boldsymbol{x}, \boldsymbol{b}) = \{\mathcal{C}_1(\boldsymbol{x}, \boldsymbol{b}), \ldots, \mathcal{C}_I(\boldsymbol{x}, \boldsymbol{b})\}$, where $\boldsymbol{x} \in \mathbb{R}^d$ are the deterministic parameters and $\boldsymbol{b}$ represents a random vector [13]. The uncertainty may be caused by a lack of knowledge about the parameters or the inherent noise in the system. The objective f or the constraints $\mathcal{C}$ depend implicitly on the output of the black-box simulator. Our task is to minimize the function $f(\boldsymbol{x}, \boldsymbol{b})$ with respect to $\boldsymbol{x}$ subject to the constraints $\mathcal{C}(\boldsymbol{x}, \boldsymbol{b})$. Because of the stochastic nature of the problem, we will optimize the objective function with respect to a robustness measure [21, 22]. In this work, we will only consider the expectation as a robustness measure, in which case, the problem can be stated as follows:

$$\min_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{b}}[f(\boldsymbol{x}, \boldsymbol{b})], \quad \text{s.t} \quad \mathbb{E}_{\boldsymbol{b}}[\mathcal{C}_i(\boldsymbol{x}, \boldsymbol{b})] \leq 0, \ \ \forall i \in \{1, \ldots, I\} \tag{1}$$

In addition to that, the gradient of the objective function and the constraint is unavailable. Hence, one cannot directly apply gradient-based optimization methods.

2.1 Constraint augmentation

We cast the constrained optimization problem (Eq. (1)) to an unconstrained one using penalty-based methods [23, 24]. We define an augmented objective function $\mathcal{L}$ as follows:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{\lambda}) = f(\boldsymbol{x}, \boldsymbol{b}) + \sum_{i=i}^{I} \lambda_i \max\left(\mathcal{C}_i(\boldsymbol{x}, \boldsymbol{b}), 0\right) \tag{2}$$

where $\lambda_i > 0$ is the penalty parameter for the i^{th} constraint and the $\max(\cdot, \cdot)$ controls the magnitude of penalty applied. Incorporating the augmented objective (Eq. (2)) in the Eq. (1), one can arrive at the following optimization problem:

$$\min_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{b}}[\mathcal{L}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{\lambda})] \tag{3}$$

The expectation is approximated by Monte Carlo which induces noise and necessitates stochastic optimization methods. We alleviate the dependence on the penalty parameter $\boldsymbol{\lambda}$ by using the sequential unconstrained minimization technique (SUMT) algorithm [25] where one starts with a small penalty term and gradually increases its value.

2.2 Estimation of derivative

The direct computation of derivatives of $\mathcal{L}$ with respect to the optimization variables $\boldsymbol{x}$ is not feasible because of the unavailability of the gradients of the objective function and the constraints. One notes many active research threads across disciplines which are trying to tackle this bottleneck [1, 14, 26, 27, 4, 15, 15]. We draw inspiration from the Variational Optimization [17, 28, 16], which constructs an upper bound of the objective function as shown below:

$$\min \int \mathcal{L}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{\lambda}) p(\boldsymbol{b}) d\boldsymbol{b} \leq \int \mathcal{L}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{\lambda}) p(\boldsymbol{b}) q(\boldsymbol{x} \mid \boldsymbol{\theta}) d\boldsymbol{b} d\boldsymbol{x} = U(\boldsymbol{\theta}) \tag{4}$$

where $q(\boldsymbol{x} \mid \boldsymbol{\theta})$ is a density over the design variables $\boldsymbol{x}$ with parameters $\boldsymbol{\theta}$. If $\boldsymbol{x}^*$ yields the minimum of the objective $\mathbb{E}_{\boldsymbol{b}}[\mathcal{L}]$, then this can be achieved with a degenerate q that collapses to a Dirac-delta, i.e. if $q(\boldsymbol{x} \mid \boldsymbol{\theta}) = \delta(\boldsymbol{x} - \boldsymbol{x}^*)$. The inequality above would generally be strict for all other densities q or parameters $\boldsymbol{\theta}$. Hence, instead of minimizing $\mathbb{E}_{\boldsymbol{b}}[\mathcal{L}]$ with respect to $\boldsymbol{x}$, we can minimize the upper bound $U(\boldsymbol{\theta})$ with respect to the distribution parameters $\boldsymbol{\theta}$. Under mild restrictions outlined by [18], the bound $U(\boldsymbol{\theta})$ is differential w.r.t $\boldsymbol{\theta}$. One can evaluate the gradient of $U(\boldsymbol{\theta})$ as shown below:

$$\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}, \boldsymbol{b}}\left[\nabla_{\boldsymbol{\theta}} \log q(\boldsymbol{x} \mid \boldsymbol{\theta}) \mathcal{L}(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{\lambda})\right] \tag{5}$$

The Monte Carlo estimation of the expectation shown in Eq. (5) is as follows:

$$\frac{\partial U}{\partial \theta} \approx \frac{1}{S} \sum_{i=1}^{S} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{b}_i, \boldsymbol{\lambda}) \frac{\partial}{\partial \boldsymbol{\theta}} \log q\left(\boldsymbol{x}_i \mid \theta\right) \tag{6}$$

The Eq. (6) is known as the score function estimator [29], which also appears in the context of reinforcement learning [30]. The gradient estimation can be computationally expensive as each step will involve calling the simulator S number of times. This step is can be easily parallelized.

2.3 Variance reduction

To reduce the mean square error of the estimator in Eq. (6), we propose the use of baseline discussed in [31] as shown below:

$$\frac{\partial U}{\partial \theta} \approx \frac{1}{S} \sum_{i=1}^{S} \frac{\partial}{\partial \boldsymbol{\theta}} \log q\left(\boldsymbol{x}_i \mid \boldsymbol{\theta}\right) \left(\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{b}_i, \boldsymbol{\lambda}) - \frac{1}{S-1} \sum_{j=1, j \neq i}^{S} \mathcal{L}(\boldsymbol{x}_j, \boldsymbol{b}_j, \boldsymbol{\lambda}) \right) \tag{7}$$

The above is an unbiased estimator and implies no additional cost beyond the S samples.We also propose to use Quasi-Monte Carlo (QMC) sampling [32] for variance reduction. QMC replaces S randomly drawn samples by a pseudo-random sequence of samples of length S with low discrepancy. This sequence covers the underlying design space more evenly than the random samples, thereby reducing the variance of the gradient estimator. Fig. (1) numerically shows the advantage of using variance reduction technique. We observe that the variance of the gradient is decreased using the variance reduction methods, specifically in high dimensions where we observe $\sim 10\times$ benefit.

2.4 Multi-fidelity

The main computational bottleneck of the gradient estimation using Eq. (6) is the multiple evaluation of the objective function. This becomes a significant concern for computationally expensive simulators. We propose to solve this problem using the multi-fidelity (MF) method [33]. Suppose we are given a set of L functions modeling the same quantity and arranged in ascending order of accuracy and computational cost $\{f_1, f_2, \ldots, f_L\}$. We want to optimize the design parameter with respect to the highest-fidelity model (f_L). We can estimate the gradient of the corresponding upper bound using the method suggested in [34] as shown below:

$$\frac{\partial U_L}{\partial \theta} \approx \sum_{\ell=1}^{L} \frac{1}{S_\ell} \sum_{i=1}^{S_\ell} \left(\mathcal{L}_\ell(\boldsymbol{x}_i, \boldsymbol{b}_i, \boldsymbol{\lambda}) - \mathcal{L}_{\ell-1}(\boldsymbol{x}_i, \boldsymbol{b}_i, \boldsymbol{\lambda}) \right) \frac{\partial}{\partial \boldsymbol{\theta}} \log q\left(\boldsymbol{x}_i \mid \theta\right) \tag{8}$$

where S_ℓ is the number of samples used in the estimator at level ℓ and $\mathcal{L}_0(\cdot) = 0$. We want to use more samples from the low-fidelity model and lesser samples as we increase the fidelity. The method to calculate the number of samples is discussed in [34].

2.5 Implementation details

In the present study, we use `PyTorch` [35] to efficiently compute the gradient. After the gradient estimation, we use the ADAM optimizer [36] as the stochastic gradient descent method. In this work, $q(\boldsymbol{x} \mid \boldsymbol{\theta})$ takes the form of a Gaussian distribution with parameters $\boldsymbol{\theta} = \{\mu, \sigma\}$ representing mean and variance. Our proposed algorithms is summarized in Algorithm 1. The source code will be made available upon publication.

3 Numerical Illustrations

We discuss the numerical results of the proposed (MF)Scout-Nd algorithm on sphere-problem of varying dimensions ($d = \{2, 4, 8, 16, 32\}$). We use the data profiles proposed in [8] to compare (MF)Scout-Nd with cBO [19] and COBYLA [20], which are standard derivative-free optimization methods that can handle constraints. We consider the following optimization problem with *noisy* objective:

$$\min_{\boldsymbol{x}} \mathbb{E}_b \left[\sum_{i=1}^{d} x_i^2 + b \right]; \quad s.t. \quad \mathcal{C}(\boldsymbol{x}) \leq 0 \tag{9}$$

Algorithm 1 Stochastic constrained optimization for non-differentiable objective (Scout-Nd)

Input: Objective function(s), constraint(s), distribution $q(\boldsymbol{x} \mid \boldsymbol{\theta})$, stochastic gradient descent optimizer $\mathcal{G}$ and its hyper-parameters η, list of penalty terms $\{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_K\}$, $\boldsymbol{\lambda}_K \to \infty$
$\boldsymbol{\theta}_0^0 \leftarrow$ choose starting point, $k \leftarrow 1$
do
  $n \leftarrow 0$
  do
    $\boldsymbol{x}_i \sim q(\boldsymbol{x} \mid \boldsymbol{\theta}_k^n), \quad \boldsymbol{b}_i \sim p(\boldsymbol{b})$ ▷ Sampling step
    Evaluate augmented objectives $\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{b}_i, \boldsymbol{\lambda}_k)$ ▷ Eq. (2)
    Monte-Carlo gradient estimate $\nabla_{\boldsymbol{\theta}} U$ ▷ Eq. (7), 8
    $\boldsymbol{\theta}_k^{n+1} \leftarrow \mathcal{G}(\boldsymbol{\theta}_k^n, \eta, \nabla_{\boldsymbol{\theta}} U)$ and $n \leftarrow n + 1$ ▷ Stochastic Gradient Descent
  while $\|\boldsymbol{\theta}_k^n - \boldsymbol{\theta}_k^{n-1}\| > \varepsilon_\theta$
  $\boldsymbol{\theta}_{k+1}^0 \leftarrow \boldsymbol{\theta}_k^n$; $\{\mu, \sigma\} \leftarrow \boldsymbol{\theta}_k^n$ and $k \leftarrow k + 1$
while $\|\sigma\| > \varepsilon_\sigma$ ▷ Collapse to Dirac-delta
return $\{\mu, \sigma\}$

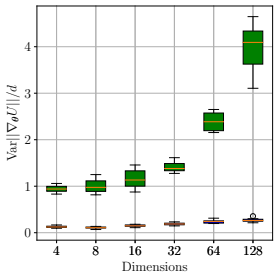

Figure 1: Box plot of the variance of the gradient estimator w.r.t the dimensions with 10 repeated runs for Eq. (9). Gradient estimated with 128 samples at $\boldsymbol{\theta} = \{\mathbf{1}_d, e^{\mathbf{1}_d}\}$. Green : no variance reduction, red : variance reduction

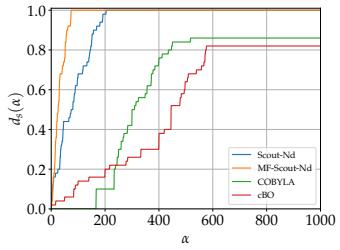

Figure 2: Data profiles plot for the set of optimizers. For the expectation estimation for each step, MF-Scout-Nd uses $S_1 = 10$ HF and $S_2 = 50$ LF evaluations. Every other method uses 50 black-box evaluations. $\epsilon_f = 0.1$

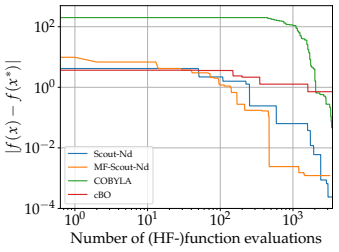

Figure 3: Evolution of the distance between the optimal objective values suggested by the optimizer $(f(x))$ and the theoretical optimum $(f(x^*))$ with respect to the number of (HF-)function evaluations for the first case of sphere problem 9 with $d = 8$.

with $b \sim \mathcal{N}(0, 0.1)$. We consider two different constraint cases. The first case is $\mathcal{C}(\boldsymbol{x}) = 1 - (x_1 + x_2)$ where the optimum lies on the constraint i.e. $\boldsymbol{x}^* = \{0.5, 0.5\} \cup \{0\}_{d-2}$. In the second case, the constraint surface is defined as $\mathcal{C}(\boldsymbol{x}) = \sum_{i=1}^{d} x_i - 1$ leading to the optimum at $\boldsymbol{x}^* = \{0\}_d$. Let $\mathcal{S}$ be the set of optimizers s and $\mathcal{P}$ be a set of benchmark problems p. Then the data profile $d_s(\alpha)$ [8] of a optimizer $s \in \mathcal{S}$ is given by

$$d_s(\alpha) = \frac{1}{|\mathcal{P}|} \left| \left\{ p \in \mathcal{P} : \frac{t_{p,s}}{d_p + 1} \leq \alpha \right\} \right| \tag{10}$$

where d_p in the number of design variables in p, α is the allowed number of functional evaluations scaled by the number of design variables and $t_{p,s}$ is the minimum number of function calls a solver s requires to reach the optimum of a problem p within accuracy level ϵ_f. We run each benchmark 5 times leading to $|\mathcal{P}| = 5(|d|) \times 5(\text{number of runs}) \times 2(\text{number of cases}) = 50$. For (MF)Scount-Nd, we consider two levels of multi-fidelity. The high-fidelity (HF) is given by the Eq. (9) and the low-fidelity (LF) is defined by down-scaling the x_i in Eq. (9) to have $x_i = x_i/1.05$.

We can observe from Fig. (2) that Scout-Nd performs better than cBO and COBYLA because it solved most of the benchmark problems $p \in \mathcal{P}$ for a given α. Our proposed algorithm outperforms the other two derivative-free methods because we use derivative approximation to move toward the optimum. This not only helped us to converge faster but also tackled high-dimensionality. MF-Scout-Nd performed better as it converged faster towards the optimum than Scout-Nd because it needs fewer

costly function evaluations. We can also observe from Fig. (3) that Scout-Nd and MF-Scout-Nd come closer to the actual optimal objective for a given computational budget.

4 Conclusions

We extended the method proposed by [18, 17] to account for constraints. We further employed a multi-fidelity strategy and gradient variance reduction schemes to reduce the number of costly simulator calls. We demonstrated on a classical benchmark problem that our method performs better than the chosen baselines in terms of quality of optimum and number of functional calls. As future work, we will test our method on real-world problems (for example: [15, 14]).

5 Broader Impact Statement

Many real-world systems in engineering and physics are modeled by complex simulators that might be parameterized by a high-dimensional random variable. Some notable examples include particle physics, fluid mechanics, molecular dynamics, protein folding, cosmology, material sciences, etc. Frequently, the simulators are black-box, making the task of optimization/inference challenging, specifically in high dimensions. The task can be further complicated with the inclusion of constraints. In the present work, we introduced an algorithm to approximate the gradients for an optimization/inference task involving these simulators. We demonstrated that the proposed method performs better than the state-of-the-art on a standard benchmark problem.

We do not see any direct ethical concerns associated with this research. The impact on society is primarily through the over-arching context of providing novel methods to approach optimization/inference involving complex simulators.

Acknowledgements

A.A. and P.S.K acknowledge the support of VDI Technologiezentrum Gmbh and the Federal Ministry for Education and Research (BMBF) in the context of the project "Probabilistic machine learning for the calibration and validation of physics-based models of concrete" (FKZ-13XP5125B). K.R and H.J.B acknowledge the support of Helmholtz Association under the research school Munich School for Data Science (MUDS).

References

[1] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. "The frontier of simulation-based inference". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30055–30062.

[2] German Ros et al. "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3234–3243.

[3] Jonas Degrave, Michiel Hermans, Joni Dambre, et al. "A differentiable physics engine for deep learning in robotics". In: *Frontiers in neurorobotics* (2019), p. 6.

[4] Atul Agrawal and Phaedon-Stelios Koutsourelakis. *A probabilistic, data-driven closure model for RANS simulations with aleatoric, model uncertainty*. 2023. arXiv: 2307.02432 [physics.flu-dyn].

[5] Filipe de Avila Belbute-Peres et al. "End-to-end differentiable physics for learning and control". In: *Advances in neural information processing systems* 31 (2018).

[6] Didier Lucor, Atul Agrawal, and Anne Sergent. "Simple computational strategies for more effective physics-informed neural networks modeling of turbulent natural convection". In: *Journal of Computational Physics* 456 (2022), p. 111022.

[7] Rajesh Ranganath, Sean Gerrish, and David Blei. "Black box variational inference". In: *Artificial intelligence and statistics*. PMLR. 2014, pp. 814–822.

[8] Jorge J Moré and Stefan M Wild. "Benchmarking derivative-free optimization algorithms". In: *SIAM Journal on Optimization* 20.1 (2009), pp. 172–191.

[9] Wolfgang Banzhaf et al. *Genetic programming: an introduction: on the automatic evolution of computer programs and its applications*. Morgan Kaufmann Publishers Inc., 1998.

[10] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical bayesian optimization of machine learning algorithms". In: *Advances in neural information processing systems* 25 (2012).

[11] Friedrich Menhorn et al. "A trust-region method for derivative-free nonlinear constrained stochastic optimization". In: *arXiv preprint arXiv:1703.04156* (2017).

[12] Shakir Mohamed et al. "Monte carlo gradient estimation in machine learning". In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 5183–5244.

[13] Georg Ch Pflug. *Optimization of stochastic models: the interface between simulation and optimization*. Vol. 373. Springer Science & Business Media, 2012.

[14] Gilles Louppe, Joeri Hermans, and Kyle Cranmer. "Adversarial Variational Optimization of Non-Differentiable Simulators". en. In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. ISSN: 2640-3498. PMLR, Apr. 2019, pp. 1438–1447. URL: https://proceedings.mlr.press/v89/louppe19a.html (visited on 11/22/2022).

[15] Sergey Shirobokov et al. "Black-box optimization with local generative surrogates". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 14650–14662.

[16] Nataniel Ruiz, Samuel Schulter, and Manmohan Chandraker. "Learning to simulate". In: *arXiv preprint arXiv:1810.02513* (2018).

[17] Thomas Bird, Julius Kunze, and David Barber. *Stochastic Variational Optimization*. arXiv:1809.04855 [cs, stat]. Sept. 2018. URL: http://arxiv.org/abs/1809.04855 (visited on 11/08/2022).

[18] Joe Staines and David Barber. *Variational Optimization*. arXiv:1212.4507 [cs, stat]. Dec. 2012. URL: http://arxiv.org/abs/1212.4507 (visited on 11/08/2022).

[19] Jacob R Gardner et al. "Bayesian optimization with inequality constraints." In: *ICML*. Vol. 2014. 2014, pp. 937–945.

[20] Michael JD Powell. *A direct search optimization method that models the objective and constraint functions by linear interpolation*. Springer, 1994.

[21] Aharon Ben-Tal and Arkadi Nemirovski. "Robust solutions of uncertain linear programs". In: *Operations research letters* 25.1 (1999), pp. 1–13.

[22] Dimitris Bertsimas, David B Brown, and Constantine Caramanis. "Theory and applications of robust optimization". In: *SIAM review* 53.3 (2011), pp. 464–501.

[23] I.-J. Wang and J.C. Spall. "Stochastic optimization with inequality constraints using simultaneous perturbations and penalty functions". en. In: *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*. Maui, HI, USA: IEEE, 2003, pp. 3808–3813. ISBN: 978-0-7803-7924-4. DOI: 10.1109/CDC.2003.1271742. URL: http://ieeexplore.ieee.org/document/1271742/ (visited on 10/28/2022).

[24] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[25] Anthony V Fiacco and Garth P McCormick. *Nonlinear programming: sequential unconstrained minimization techniques*. SIAM, 1990.

[26] Mark A Beaumont, Wenyang Zhang, and David J Balding. "Approximate Bayesian computation in population genetics". In: *Genetics* 162.4 (2002), pp. 2025–2035.

[27] Paul Marjoram et al. "Markov chain Monte Carlo without likelihoods". In: *Proceedings of the National Academy of Sciences* 100.26 (2003), pp. 15324–15328.

[28] Joe Staines and David Barber. "Optimization by Variational Bounding." In: *ESANN*. 2013.

[29] Peter W Glynn. "Likelihood ratio gradient estimation for stochastic systems". In: *Communications of the ACM* 33.10 (1990), pp. 75–84.

[30] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine learning* 8.3 (1992), pp. 229–256.

[31] Wouter Kool, Herke van Hoof, and Max Welling. "Buy 4 REINFORCE Samples, Get a Baseline for Free!" en. In: (July 2022). URL: https://openreview.net/forum?id=r1lgTGL5DE (visited on 11/11/2022).

[32] Josef Dick, Frances Y Kuo, and Ian H Sloan. "High-dimensional integration: the quasi-Monte Carlo way". In: *Acta Numerica* 22 (2013), pp. 133–288.

[33] Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. "Survey of multifidelity methods in uncertainty propagation, inference, and optimization". In: *Siam Review* 60.3 (2018), pp. 550–591.

[34] Michael B Giles. "Multilevel monte carlo methods". In: *Acta numerica* 24 (2015), pp. 259–328.

[35] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[36] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).