

# Scalable Acceleration for Classification-Based Derivative-Free Optimization

Tianyi Han\*, Jingya Li, Zhipeng Guo, Yuan Jin\*

Beijing Supreium Technology, Haidian District, Beijing, China  
{ty.han, jy.li, zp.guo, y.jin}@supreium.com

## Abstract

Derivative-free optimization algorithms play an important role in scientific and engineering design optimization problems, especially when derivative information is not accessible. In this paper, we study the framework of sequential classification-based derivative-free optimization algorithms. By introducing learning theoretic concept hypothesis-target shattering rate, we revisit the computational complexity upper bound of *SRACOS*. Inspired by the revisited upper bound, we propose an algorithm named *RACE-CARS*, which adds a random region-shrinking step compared with *SRACOS*. We further establish theorems showing the acceleration by region shrinking. Experiments on the synthetic functions as well as black-box tuning for language-model-as-a-service demonstrate empirically the efficiency of *RACE-CARS*. An ablation experiment on the introduced hyper-parameters is also conducted, revealing the mechanism of *RACE-CARS* and putting forward an empirical hyper-parameter tuning guidance.

## Introduction

In recent years, there has been a growing interest in the field of derivative-free optimization (DFO) algorithms, also known as zeroth-order optimization. These algorithms aim to optimize objective functions without relying on explicit gradient information, making them suitable for scenarios where obtaining derivatives is either infeasible or computationally expensive (Conn, Scheinberg, and Vicente 2009; Larson, Menickelly, and Wild 2019). For example, DFO techniques can be applied to hyperparameter tuning, which involves optimizing complex objective functions with unavailable first-order information (Falkner, Klein, and Hutter 2018; Akiba et al. 2019; Yang and Shami 2020). Moreover, DFO algorithms find applications in engineering design optimization, where the objective functions are computationally expensive to evaluate and lack explicit derivatives (Ray and Saini 2001; Liao 2010; Akay and Karaboga 2012).

Classical DFO methods such as Nelder-Mead method (Nelder and Mead 1965) or directional direct-search (DDS) method (Céa 1971; Yu 1979) have shown great success on convex problems. However, their performance is compromised when the objective is nonconvex, which is the

commonly-faced situation for black-box problems. In this paper, we focus on optimization problems reading as

$$\min_{x \in \Omega} f(x), \quad (1)$$

where the  $n$  dimensional compact cubic  $\Omega \subseteq \mathbb{R}^n$  is solution space. In addition, we will not stipulate any convexity, smoothness or separability assumption on  $f$ .

Recent decades, progresses have been made in the extensive exploration of DFO algorithms for nonconvex problems, and various kinds of algorithms have been established. For example, evolutionary algorithms (Bäck and Schwefel 1993; Fortin et al. 2012; Hansen 2016; Opara and Arabas 2019), Bayesian optimization (BO) methods (Snoek, Larochelle, and Adams 2012; Shahriari et al. 2015; Frazier 2018) and gradient approximation surrogate modeling algorithms (Nesterov and Spokoiny 2017; Chen et al. 2019; Ge et al. 2022; Ragonneau and Zhang 2023).

Given the *nonconvex* and *black-box* nature of the problem, the quest for enhanced sample efficiency inevitably presents the classic trade-off between exploration and exploitation. Similar to typical nonlinear/nonconvex numerical issues, the aforementioned algorithms are also susceptible to the curse of dimensionality (Bickel and Levina 2004; Hall, Marron, and Neeman 2005; Fan and Fan 2008; Shi et al. 2021; Scheinberg 2022; Yue et al. 2023). Improving scalability and efficiency can be distilled into a few key areas, such as function decomposition (Wang et al. 2018b), dimension reduction (Wang et al. 2016; Nayebi, Munteanu, and Poloczek 2019) and sample strategy refinement. Regarding the latter, previous algorithms have focused on preventing over-exploitation through, for instance, trying more efficient sampling dynamics (Ros and Hansen 2008; Hensman, Fusi, and Lawrence 2013; Yi et al. 2024), search space discretization (Sobol' 1967) or search region restriction (Eriksson et al. 2019; Wang, Fonseca, and Tian 2020).

Regardless of the improvements, many of these model-based DFO algorithms share the mechanism that, locally or globally, fits the objective function. The motivation of classification-based DFO algorithms is different: it learns a hypothesis (classifier) to fit the sub-level set  $\Omega_{\alpha_t} := \{x \in \Omega: f(x) - f^* \leq \alpha_t\}$  at each iteration  $t = 1, \dots, T$ , with which new candidates are generated for the successive round (Michalski 2000; Gotovos 2013; Bogunovic et al. 2016; Yu, Qian, and Hu 2016; Hashimoto, Yadowsky, and Duchi

2018). Sub-level sets are considered to contain less information than objective functions, since they are already obtained whenever the objective function is known. By this means, sample usage is more efficient. Moreover, it is less sensitive to function oscillation or noise (Liu et al. 2017) and easy to be parallelized (Liu et al. 2017, 2019b; Hashimoto, Yadlowsky, and Duchi 2018).

Training strategies of hypotheses have branched out in various directions. Hashimoto, Yadlowsky, and Duchi pursue accurate hypotheses using a *conservative* strategy raised by El-Yaniv and Wiener. They demonstrate that for a hypothesis class  $\mathcal{H}$  with a VC-dimension of  $VC(\mathcal{H})$ , an  $\epsilon$ -accurate hypothesis requires  $\mathcal{O}(\epsilon^{-1}(VC(\mathcal{H}) \log(\epsilon^{-1})))$  samples per batch. While a computationally efficient approximation has indeed been developed, its success was demonstrated only in the context of low-dimensional problems. The extent of its effectiveness in high-dimensional scenarios is yet to be determined. However, a contrasting approach successfully lead to the improvements in sample efficiency. Yu, Qian, and Hu propose to “RAnomly COordinate-wise Shrink” the solution region (*RACOS*), a more *radical* approach that diverges from the goal of producing accurate hypotheses. They prove *RACOS* converges to global minimum in polynomial time when the objective is locally Holder continuous, and experiments indicate success on high-dimensional problems ranging from hundreds to thousands of dimensions. An added benefit is the ease of sampling through the generated hypotheses, as their active regions are coordinate-orthogonal cubics. Building on this, the sequential counterpart of *RACOS*, known as *SRACOS*, takes an even more radical stance by sampling only once per iteration and learning from biased data. Despite this, *SRA-COS* has been shown to outperform *RACOS* under certain mild restriction (Hu, Qian, and Yu 2017).

## Outline and Contributions

1. Our research extends the groundwork laid by (Yu, Qian, and Hu 2016; Hu, Qian, and Yu 2017), yet we discover that the upper bound they proposed for the *SRACOS* does not fully encapsulate its behavior, potentially allowing for an overflow. On the other hand, we construct a counter-example (eq. (3)) illustrating the upper bound is not tight enough to delineate the convergence accurately.
2. We also identify a contentious assumption regarding the concept of error-target dependence that seems to underpin these limitations. In this paper, we introduce a novel learning theoretic concept termed the hypothesis-target  $\eta$ -shattering rate, which serves as a foundation for our reassessment of the upper bound on the computational complexity of *SRACOS*.
3. Inspired by the reanalysis, we recognize that the accuracy of the trained hypotheses is even less critical than previously thought. With this insight, we formulate a novel algorithm, *RACE-CARS* (an acronym for “RAnomized COordinatE Classifying And Region Shrinking”), which perpetuates the essence of *SRACOS* while promises a theoretical enhancement in convergence. We design experiments on both synthetic functions and black-box tuning

for LMaaS. These experiments juxtapose *RACE-CARS* against a selection of DFO algorithms, empirically highlighting the superior efficacy of *RACE-CARS*.

4. In discussion and appendix, we further substantiate the empirical prowess of *RACE-CARS* beyond the confines of continuity, also supported by theoretical framework. An ablation study is also conducted to demystify the selection of hyperparameters for our proposed algorithm.

The rest of the paper is organized in five sections, sequentially presenting the background, theoretical study, experiments, discussion and conclusion.

## Background

**Assumption 1.** In eq. (1), we presume  $f(x)$  is lower bounded within  $\Omega$ , with  $f^* := \min_{x \in \Omega} f(x)$ .

In our theoretical analysis, we refine the procedure as a stochastic process. Let  $\mathcal{F}$  denote the Borel  $\sigma$ -algebra on  $\Omega$ , and  $\mathbb{P}$  the probability measure defined on  $\mathcal{F}$ . For instance, if  $\Omega$  is continuous space,  $\mathbb{P}$  is derived from Lebesgue measure  $m$ , such that  $\mathbb{P}(B) := m(B)/m(\Omega)$  for all  $B \in \mathcal{F}$ .

**Assumption 2.** Define  $\Omega_\epsilon := \{x \in \Omega : f(x) - f^* \leq \epsilon\}$ , with the assumption  $|\Omega_\epsilon| := \mathbb{P}(\Omega_\epsilon) > 0$  for all  $\epsilon > 0$ .

A hypothesis (or classifier)  $h$  is a function mapping the solution space  $\Omega$  to  $\{0, 1\}$ . Define

$$D_h(x) := \begin{cases} \mathbb{P}(\{x \in \Omega : h(x) = 1\})^{-1}, & h(x) = 1 \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

a probability distribution in  $\Omega$ . Let  $\mathbf{X}_h$  be the random vector in  $(\Omega, \mathcal{F}, \mathbb{P})$  drawn from  $D_h$ , implying that  $\Pr(\mathbf{X}_h \in B) = \int_{x \in B} D_h(x) d\mathbb{P}$  for all  $B \in \mathcal{F}$ . Denoted by  $\mathbb{T} := \{1, 2, \dots, T\}$ , and let  $\mathbb{F} := (\mathcal{F}_t)_{t \in \mathbb{T}}$  be a filtration of  $\sigma$ -algebras on  $\Omega$  indexed by  $\mathbb{T}$ , satisfying  $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \dots \subseteq \mathcal{F}_T \subseteq \mathcal{F}$ . A typical classification-based optimization algorithm learns an  $\mathbb{F}$ -adapted stochastic process  $\mathbf{X} := (\mathbf{X}_t)_{t \in \mathbb{T}}$ , where each  $\mathbf{X}_t$  is induced by  $\mathbf{X}_{h_t}$  and  $h_t$  is the hypothesis updated at step  $t$ . Subsequently, it samples new data with a stochastic process  $\mathbf{Y} := (\mathbf{Y}_t)_{t \in \mathbb{T}}$  generated by  $\mathbf{X}$ . Typically, the new candidates at step  $t \geq r + 1$  are sampled from

$$\mathbf{Y}_t := \begin{cases} \mathbf{X}_t, & \text{with probability } \lambda \\ \mathbf{X}_\Omega, & \text{with probability } 1 - \lambda, \end{cases}$$

where  $\mathbf{X}_\Omega$  is random vector drawn from uniform distribution  $\mathcal{U}_\Omega$  and  $\lambda \in [0, 1]$  is the exploitation rate.

A simplified batch-mode classification-based optimization algorithm is outlined in Algorithm 1. At each step  $t$ , it selects a positive set  $\mathcal{S}_{positive}$  from  $\mathcal{S}$  containing the best  $m$  samples, with the remainder in the negative set  $\mathcal{S}_{negative}$ . Then it trains a hypothesis  $h_t$  distinguishing between the positive set and negative set, ensuring  $h_t(x_j) = 0$  for all  $(x_j, y_j) \in \mathcal{S}_{negative}$ . Finally, it samples  $r$  new solutions with the sampling random vector  $\mathbf{Y}_t$ . The sub-procedure  $h_t \leftarrow \mathcal{T}(\mathcal{S}_{positive}, \mathcal{S}_{negative})$  means hypothesis training.

*RACOS* is the abbreviation of “RAnomized COordinate Shrinking”. Literally, it trains the hypothesis by this means

---

Algorithm 1: Batch-Mode Classification-Based Optimization Algorithm

---

**Input:**  $T$ : Budget;  $r$ : Training size;  $m$ : Positive size.  
**Output:**  $(x_{best}, y_{best})$ .  
 Collect:  $\mathcal{S} = \{(x_1^0, y_1^0), \dots, (x_r^0, y_r^0)\}$  i.i.d. from  $\mathcal{U}_\Omega$ ;  
 $(x_{best}, y_{best}) = \arg \min\{y: (x, y) \in \mathcal{S}\}$ ;  
**for**  $t = 1, \dots, T/r$  **do**  
   Classify:  $(\mathcal{S}_{positive}, \mathcal{S}_{negative}) \leftarrow \mathcal{S}$ ;  
   Train:  $h_t \leftarrow \mathcal{T}(\mathcal{S}_{positive}, \mathcal{S}_{negative})$ ;  
   Sample:  $\{(x_1^t, y_1^t), \dots, (x_r^t, y_r^t)\}$  i.i.d. with  $\mathbf{Y}_t$ ;  
   Select:  $\mathcal{S} \leftarrow \mathcal{S} \cup \{(x_1^t, y_1^t), \dots, (x_r^t, y_r^t)\}$ ;  
    $(x_{best}, y_{best}) = \arg \min\{y: (x, y) \in \mathcal{S}\}$ .  
**end for**  
**Return:**  $(x_{best}, y_{best})$

---

Algorithm 2: RACOS

---

**Input:**  $\Omega$ : Boundary;  $(\mathcal{S}_{positive}, \mathcal{S}_{negative})$ : Binary sets;  
 $\mathbb{I} = \{1, \dots, n\}$ : Index of dimensions.  
**Output:**  $h$ : Hypothesis.  
 Randomly select:  $x_+ = (x_+^1, \dots, x_+^n) \leftarrow \mathcal{S}_{positive}$ ;  
 Initialize:  $h(x) \equiv 1$ ;  
**while**  $\exists x \in \mathcal{S}_{negative}$  s.t.  $h(x) = 1$  **do**  
   Randomly select:  $k \leftarrow \mathbb{I}$ ;  
   Randomly select:  $x_- = (x_-^1, \dots, x_-^n) \leftarrow \mathcal{S}_{negative}$ ;  
   **if**  $x_+^k \leq x_-^k$  **then**  
      $s \leftarrow \text{random}(x_+^k, x_-^k)$ ;  
     Shrink:  $h(x) = 0, \forall x \in \{x = (x^1, \dots, x^n) \in \Omega: x^k > s\}$ ;  
   **else**  
      $s \leftarrow \text{random}(x_-^k, x_+^k)$ ;  
     Shrink:  $h(x) = 0, \forall x \in \{x = (x^1, \dots, x^n) \in \Omega: x^k < s\}$ ;  
   **end if**  
**Return:**  $h$   
**end while**

---

(Yu, Qian, and Hu 2016), i.e., shrinking coordinates randomly such that all negative samples are excluded from active region of resulting hypothesis. Algorithm 2 shows a continuous version of *RACOS*.

Aside from sampling only once per iteration, another difference between sequential and batch mode classification-based DFO is that the sequential version will replace the training set with a new one under certain rules to finish step  $t$  (see in in Algorithm 3). In the rest of this paper, we will omit the details of *Replacing* sub-procedure which can be found in (Hu, Qian, and Yu 2017).

Classification-based DFO algorithms admit a bound on the query complexity (Yu and Qian 2014), quantifying the total number of function evaluations required to identify a solution that achieves an approximation level of  $\epsilon$  with a high probability of at least  $1 - \delta$ .

**Definition 1** ( $(\epsilon, \delta)$ -Query Complexity). Given  $f$ ,  $0 < \delta < 1$  and  $\epsilon > 0$ . The  $(\epsilon, \delta)$ -query complexity of an algorithm  $\mathcal{A}$  is the number of calls to  $f$  such that, with probability at least

---

Algorithm 3: Sequential-Mode Classification-Based Optimization Algorithm

---

**Input:**  $T$ : Budget;  $r$ : Training size;  $m$ : Positive size;  
*Replacing*: Replacing sub-procedure.  
**Output:**  $(x_{best}, y_{best})$ .  
 Collect  $\mathcal{S} = \{(x_1, y_1), \dots, (x_r, y_r)\}$  i.i.d. from  $\mathcal{U}_\Omega$ ;  
 $(x_{best}, y_{best}) = \arg \min\{y: (x, y) \in \mathcal{S}\}$ ;  
**for**  $t = r + 1, \dots, T$  **do**  
   Classify:  $(\mathcal{S}_{positive}, \mathcal{S}_{negative}) \leftarrow \mathcal{S}$ ;  
   Train:  $h_t \leftarrow \mathcal{T}(\mathcal{S}_{positive}, \mathcal{S}_{negative})$ ;  
   Sample:  $(x_t, y_t) \sim \mathbf{Y}_t$ ;  
   Replace:  $\mathcal{S} \leftarrow \text{Replacing}((x_t, y_t), \mathcal{S})$ ;  
    $(x_{best}, y_{best}) = \arg \min\{y: (x, y) \in \mathcal{S}\}$ ;  
**end for**  
**Return:**  $(x_{best}, y_{best})$

---

$1 - \delta$ ,  $\mathcal{A}$  finds at least one solution  $\tilde{x} \in \Omega$  satisfying

$$f(\tilde{x}) - f^* \leq \epsilon.$$

Definition 2 and 3 are given by (Yu, Qian, and Hu 2016). The first one characterizes the so-called dependence between classification error and target region, which is expected to be minimized to ensure that the efficiency. The second one characterizes the portion of active region of hypothesis, and is also expected to be as small as possible.

**Definition 2** (Error-Target  $\theta$ -Dependence). The error-target dependence  $\theta \geq 0$  of a classification-based optimization algorithm is its infimum such that, for any  $\epsilon > 0$  and any  $t = 1, \dots, T$ ,

$$|\Omega_\epsilon| \cdot \mathbb{P}(\mathcal{R}_t) - \mathbb{P}(\Omega_\epsilon \cap \mathcal{R}_t) \leq \theta |\Omega_\epsilon|,$$

where  $\mathcal{R}_t := \Omega_{\alpha_t} \Delta \{x \in \Omega: h_t(x) = 1\}$  denotes the relative error,  $\Omega_{\alpha_t}$  is the sub-level set at step  $t$  with  $\alpha_t := \min_{1 \leq i \leq t} f(x_i) - f^*$  and the operator  $\Delta$  is symmetric difference of two sets defined as  $A_1 \Delta A_2 = (A_1 \cup A_2) - (A_1 \cap A_2)$ .

**Definition 3** ( $\gamma$ -Shrinking Rate). The shrinking rate  $\gamma > 0$  of a classification-based optimization algorithm is its infimum such that  $\mathbb{P}(x \in \Omega: h_t(x) = 1) \leq \gamma |\Omega_{\alpha_t}|$ , for all  $t = 1, \dots, T$ .

## Theoretical Study

Previous studies gave a general bound of the query complexity of Algorithm 3 based on minor error-target  $\theta$ -dependence and  $\gamma$ -shrinking rate assumptions:

**Theorem 1.** (Hu, Qian, and Yu 2017) Given  $0 < \delta < 1$  and  $\epsilon > 0$ , if a sequential classification-based optimization algorithm has error-target  $\theta$ -dependence and  $\gamma$ -shrinking rate, then its  $(\epsilon, \delta)$ -query complexity is upper bounded by

$$\mathcal{O}\left(\max\left\{\frac{1}{|\Omega_\epsilon|}\left(\lambda + \frac{1-\lambda}{\gamma(T-r)} \sum_{t=r+1}^T \Phi_t\right)^{-1} \ln \frac{1}{\delta}, T\right\}\right),$$

where  $\Phi_t = \left(1 - \theta - \mathbb{P}(\mathcal{R}_{D_t}) - m(\Omega) \sqrt{\frac{1}{2} D_{\text{KL}}(D_t || \mathcal{U}_\Omega)}\right) \cdot |\Omega_{\alpha_t}|^{-1}$  with the notations  $D_t := \lambda D_{h_t} + (1 - \lambda) \mathcal{U}_\Omega$  and  $\mathbb{P}(\mathcal{R}_{D_t}) := \int_{\mathcal{R}_t} D_t d\mathbb{P}$ .

## Issues Introduced by Error-Target Dependence

### 1. Overflow of the Upper Bound

As assumptions entailed in (Yu, Qian, and Hu 2016; Hu, Qian, and Yu 2017), it can be observed that lower values of  $\theta$  or  $\gamma$  correlate with improved query complexity. However, this is not a hard-and-fast rule. Even with small values of these parameters, we can encounter scenarios where the expected performance does not materialize. Following the lemma given by (Yu, Qian, and Hu 2016):  $\mathbb{P}(\mathcal{R}_t) \leq \mathbb{P}(\mathcal{R}_{D_t}) + m(\Omega)\sqrt{\frac{1}{2}D_{\text{KL}}(D_t||\mathcal{U}_\Omega)}$ , we have the following inequality:

$$\Phi_t = \frac{1 - \theta - \mathbb{P}(\mathcal{R}_{D_t}) - m(\Omega)\sqrt{\frac{1}{2}D_{\text{KL}}(D_t||\mathcal{U}_\Omega)}}{|\Omega_{\alpha_t}|} \leq (1 - \theta - \mathbb{P}(\mathcal{R}_t)) \cdot |\Omega_{\alpha_t}|^{-1}.$$

The concept of error-target  $\theta$ -dependence reveals that a small  $\theta$  does not guarantee small relative error  $\mathbb{P}(\mathcal{R}_t)$ . Contrarily, a small  $\theta$  coupled with a large  $(\mathbb{P}(\Omega_\epsilon \cap \mathcal{R}_t))/|\Omega_\epsilon|$  can result in a significant  $\mathbb{P}(\mathcal{R}_t)$ , which can even be 1 as long as  $\Omega_{\alpha_t}$  is totally out of the active region of  $h_t$ , when the hypothesis  $h_t$  is completely wrong. Problematically, as a divisor in the proof of Theorem 1,  $\Phi_t \leq (1 - \theta - \mathbb{P}(\mathcal{R}_t)) \cdot |\Omega_{\alpha_t}|^{-1}$  is less or equal to 0, disrupting the established inequality. Nevertheless, in order that the inequality disrupt,  $\mathbb{P}(\mathcal{R}_t)$  is not necessary to be 1 as  $\theta$  is inherently nonnegative, highlighting that a series of inaccurate hypotheses can undermine the validity of the upper bound. This finding challenges the principle of *SRA-COS*'s radical training strategy.

### 2. Tightness of the Upper Bound

Consider an extreme but plausible situation where the hypotheses generated at each step are defined as follows:

$$h_t(x) = \begin{cases} 1, & x \in \Omega_\epsilon \\ 0, & x \notin \Omega_\epsilon. \end{cases} \quad (3)$$

In the context of sequential-mode classification-based optimization algorithms, where the training sets are not only small but also potentially biased, it's reasonable to expect large relative errors  $\mathbb{P}(\mathcal{R}_t)$ . This scenario could lead to a series of hypotheses that are inaccurate with respect to  $\Omega_{\alpha_t}$  but, by chance, accurate with respect to  $\Omega_\epsilon$ . Consequently, the error-target dependence  $\theta = \max_{1 \leq t \leq T} \mathbb{P}(\mathcal{R}_t)$  can be unexpectedly large. Even all  $\Phi_t$  values being positive, the query complexity bound given in Theorem 1 is therefore not optimistic. However, the probability of failing to identify an  $\epsilon$ -minimum:

$$\Pr\left(\min_{1 \leq t \leq T} f(x_t) - f^* \geq \epsilon\right) = (1 - |\Omega_\epsilon|)^r \left( (1 - \lambda)(1 - |\Omega_\epsilon|) \right)^{T-r},$$

is less than  $\delta$  for a reasonably sized  $T$ . This indicates that the upper bound may not be as tight as initially thought.

## Revisit of Query Complexity Upper Bound

It's evident that even minimal error-target dependence can not encapsulate issues arising from substantial relative error. This is because error-target dependence alone is insufficient to fully account for relative error. Intuitively, one might consider introducing an additional assumption to cap relative error. However, such an assumption would be impractical, given the inherently small and biased nature of training datasets in the process. To this end, we give a new concept that stands apart from the constraints of relative error:

**Definition 4** (Hypothesis-Target  $\eta$ -Shattering Rate). Given  $\eta \in [0, 1]$ , for a family of hypotheses  $\mathcal{H}$  defined on  $\Omega$ , we say  $\Omega_\epsilon$  is  $\eta$ -shattered by  $h \in \mathcal{H}$  if

$$\mathbb{P}(\Omega_\epsilon \cap \{x \in \Omega : h(x) = 1\}) \geq \eta|\Omega_\epsilon|,$$

and  $\eta$  is called hypothesis-target shattering rate.

The hypothesis-target shattering rate mirrors the error-target dependence in its relation to the hypothesis's target-accuracy. Importantly, it provides a limitation for error-target dependence in conjunction with relative error:

$$\theta \leq \max\{\mathbb{P}(\mathcal{R}_t), |1 - \mathbb{P}(\mathcal{R}_t) - \eta|\}.$$

This rate,  $\eta$ , measures the overlap between the target set  $\Omega_\epsilon$  and the active region of a hypothesis. Crucially, it also mitigates the influence of relative error on error-target dependence. Utilizing the concept hypothesis-target shattering rate, we reexamine the upper bound of  $(\epsilon, \delta)$ -query complexity in the subsequent theorem.

**Theorem 2.** For sequential-mode classification-based DFO Algorithm 3, let  $\mathbf{X}_t = \mathbf{X}_{h_t}$ ,  $\epsilon > 0$  and  $0 < \delta < 1$ . When  $\Omega_\epsilon$  is  $\eta$ -shattered by  $h_t$  for all  $t = r + 1 \dots, T$  and  $\max_{t=r+1, \dots, T} \mathbb{P}(\{x \in \Omega : h_t(x) = 1\}) \leq p \leq 1$ , the  $(\epsilon, \delta)$ -query complexity is upper bounded by

$$\mathcal{O}\left(\max\left\{\left(\lambda \frac{\eta}{p} + (1 - \lambda)\right)^{-1} \left(\frac{1}{|\Omega_\epsilon|} \ln \frac{1}{\delta} - r\right) + r, T\right\}\right).$$

### The Region-Shrinking Acceleration

In the analysis of  $(\epsilon, \delta)$ -query complexity for classification-based optimization, the focus shifts away from minimizing relative error  $\mathbb{P}(\mathcal{R}_t)$ , as our goal is to identify optima, not to develop a sequence of accurate hypotheses. The counterexample in equation (3), despite a potentially high relative error, represents an optimal hypothesis scenario. This realization allows for the consideration of more radical hypotheses, directing our attention to the overlap between  $\Omega_\epsilon$  and the active region of the hypotheses, which is quantified by the hypothesis-target shattering rate.

The  $\gamma$ -shrinking rate, as defined in Definition 3, measures the decay of  $\mathbb{P}(x \in \Omega : h_t(x) = 1)$ . However, the rapid decrease of  $|\Omega_{\alpha_t}|$  as  $\alpha_t$  approaches zero makes it impractical to sustain a series of hypotheses with a small  $\gamma$  through our training process. Thus, the  $\gamma$ -shrinking assumption is often not feasible for minimal  $\gamma$ .

Moving beyond the pursuit of minimal relative error and  $\gamma$ -shrinking relative to  $|\Omega_{\alpha_t}|$ , we introduce Algorithm 4,

---

**Algorithm 4: Accelerated Sequential-Mode Classification-Based Optimization Algorithm**


---

**Input:**  $\Omega$ : Boundary;  $T \in \mathbb{N}^+$ : Budget;  $r = m + k$ ; *Replacing*: Replacing sub-procedure;  $\gamma$ : Region shrinking rate;  $\rho$ : Region shrinking frequency.  
**Output:**  $(x_{best}, y_{best})$ .  
 Collect  $\mathcal{S} = \{(x_1, y_1), \dots, (x_r, y_r)\}$  i.i.d. from  $\mathcal{U}_\Omega$ ;  
 $(x_{best}, y_{best}) = \arg \min\{y: (x, y) \in \mathcal{S}\}$ ;  
 Initialize  $k = 1, \tilde{\Omega} = \Omega$ ;  
**for**  $t = r + 1, \dots, T$  **do**  
   Train:  $h_t \leftarrow \mathcal{T}(\mathcal{S}_{positive}, \mathcal{S}_{negative})$ ;  
    $s \leftarrow \text{random}(0, 1)$ ;  
   **if**  $s \leq \rho$  **then**  
     Shrink region:  $\tilde{\Omega} = \Omega \cap [x_{best} - \frac{1}{2}\gamma^k \|\Omega\|, x_{best} + \frac{1}{2}\gamma^k \|\Omega\|]$ ;  
      $k = k + 1$ ;  
   **end if**  
   Project:  $\mathbf{Y}_t \leftarrow \text{Proj}(h_t, \tilde{\Omega})$ ;  
   Sample:  $(x_t, y_t) \sim \mathbf{Y}_t$ ;  
   Replace:  $\mathcal{S} \leftarrow \text{Replacing}((x_t, y_t), \mathcal{S})$ ;  
    $(x_{best}, y_{best}) = \arg \min\{y: (x, y) \in \mathcal{S}\}$ ;  
**end for**  
**Return:**  $(x_{best}, y_{best})$

---

which adaptively shrinks the sampling random vector  $\mathbf{Y}_t$ 's active region through a *Projection* sub-procedure.

The operator  $\|\cdot\|$  returns a tuple represents the diameter of each dimension of the region. For instance, when  $\Omega = [\omega_1^1, \omega_2^1] \times [\omega_1^2, \omega_2^2]$ , we have  $\|\Omega\| = (\omega_2^1 - \omega_1^1, \omega_2^2 - \omega_1^2)$ . The projection operator  $\text{Proj}(h_t, \tilde{\Omega})$  generates a random vector  $\mathbf{X}_t$  with probability distribution  $D_{\tilde{h}_t} := \tilde{h}_t / \mathbb{P}(\{x \in \Omega: \tilde{h}_t(x) = 1\})$ , with  $\tilde{h}_t(x) = 1$  whenever  $h_t(x) = 1$  for  $x \in \tilde{\Omega}$ . The sampling random vector  $\mathbf{Y}_t$  is induced by  $\mathbf{X}_t$ . The subsequent theorem presents the upper bound of query complexity for Algorithm 4.

**Theorem 3.** For Algorithm 4 with region shrinking rate  $0 < \gamma < 1$  and region shrinking frequency  $0 < \rho < 1$ . Let  $\epsilon > 0$  and  $0 < \delta < 1$ . When  $\Omega_\epsilon$  is  $\eta$ -shattered by  $\tilde{h}_t$  for all  $t = r + 1 \dots, T$ , the  $(\epsilon, \delta)$ -query complexity is upper bounded by

$$\mathcal{O}\left(\max\left\{\frac{\gamma^{-\rho} + \gamma^{-(T-r)\rho}}{2} \lambda \eta + (1 - \lambda)\right\}^{-1} \left(\frac{1}{|\Omega_\epsilon|} \ln \frac{1}{\delta} - r\right) + r, T\right).$$

The condition  $\gamma \in (0, 1)$  ensures that the term  $2p/(\gamma^{-\rho} + \gamma^{-(T-r)\rho})$  is significantly less than 1. According to Theorem 3, the  $(\epsilon, \delta)$ -query complexity of the Algorithm 4 is lower than that of Algorithm 3 providing  $\eta > 0$ .

Theorem 3 establishes an upper bound on the  $(\epsilon, \delta)$ -query complexity that is applicable to a wide range of scenarios, assuming only that the objective function  $f$  is lower-bounded. Building on this, we identify a sufficient condition for acceleration that applies to dimensionally local Holder

continuity functions (Definition 5), detailed in the appendix. Within this context, the *SRACOS* algorithm, which utilizes *RACOS* for *Training* phase in Algorithm 3, exhibit polynomial convergence (Hu, Qian, and Yu 2017). We adopt the same *RACOS* approach for *Training* sub-procedure in Algorithm 4, introducing "RAnomized CoordinatE Classifyng And Region Shrinking" (*RACE-CARS*) algorithm.

**Definition 5** (Dimensionally local Holder continuity). Assume that  $x_* = (x_*^1, \dots, x_*^n)$  is the unique global minimum such that  $f(x_*) = f^*$ . We call  $f$  dimensionally local Holder continuity if for all  $i = 1, \dots, n$ ,

$$\begin{aligned} & L_1^i |x^i - x_*^i|^{\beta_1^i} \\ & \leq |f(x_*^1, \dots, x^i, \dots, x_*^n) - f^*| \\ & \leq L_2^i |x^i - x_*^i|^{\beta_2^i}, \end{aligned}$$

for all  $x = (x^1, \dots, x^n)$  in the neighborhood of  $X_*$ , where  $\beta_1^i, \beta_2^i, L_1^i, L_2^i$  are positive constants for  $i = 1, \dots, n$ .

## Experiments

In this section, we design experiments to test *RACE-CARS* on synthetic functions, and language model tasks respectively. We use same budget to compare *RACE-CARS* with a selection of DFO algorithms, including *SRACOS* (Hu, Qian, and Yu 2017), zeroth-order adaptive momentum method (*ZO-Adam*) (Chen et al. 2019), differential evolution (*DE*) (Opara and Arabas 2019) and covariance matrix adaptation evolution strategies (*CMA-ES*) (Hansen 2016). All the baseline algorithms are fine-tuned, and the essential hyperparameters of *RACE-CARS* can be found in Appendix.

### On Synthetic Functions

We commence our empirical experiments with four benchmark functions: Ackley, Levy, Rastrigin and Sphere. Their analytic forms and 2-dimensional illustrations are detailed in Appendix. Characterized by extreme non-convexity and numerous local minima and saddle points—with the exception of the Sphere function—each is minimized within the boundary  $\Omega = [-10, 10]^n$ , with a global minimum value of 0. We choose the dimension of solution space  $n$  to be 50 and 500, with corresponding function evaluation budgets of 5000 and 50000. Notably, as indicates, the convergence of the *RACE-CARS* requires only a fraction of this budget.

Region shrinking rate is configured to be  $\gamma = 0.9$  and 0.95, with shrinking frequency of  $\rho = 0.01$  and 0.001 for  $n = 50, 500$ , respectively. Each algorithm is executed over 30 trials, and the mean convergence trajectories of the best-so-far values under  $100n$  function evaluations are depicted in Figure 1 and Figure 2. The numerical values adjacent to the algorithm names in the legends represent the mean of the attained minima. It is evident that that *RACE-CARS* performs the best on both convergence speed and optimal value, with a slight edge to *CMA-ES* on the strongly convex Sphere function. Yet this comes at the cost of scalability due to *CMA-ES*'s reliance on an  $n$ -dimensional covariance matrix, which is significantly more computationally intensive compared to the other algorithms.

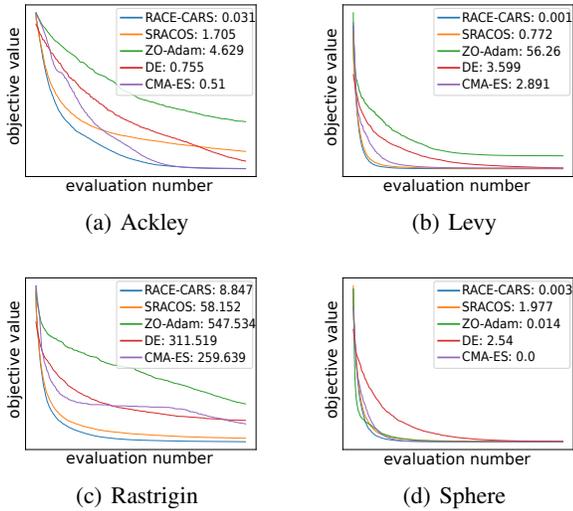


Figure 1: Comparison of synthetic functions with  $n = 50$ .

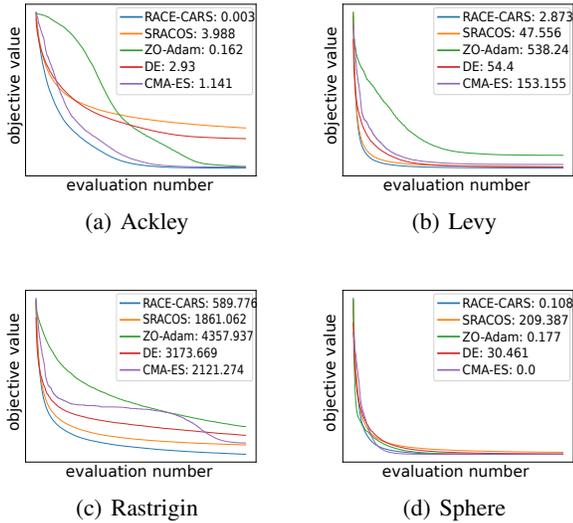


Figure 2: Comparison of synthetic functions with  $n = 500$ .

## On Black-Box Tuning for LMaaS

Prompt tuning for extremely large pre-trained language models (PTMs) has shown great power. PTMs such as GPT-3 (Brown et al. 2020) are usually released as a service due to commercial considerations and the potential risk of misuse, allowing users to design individual prompts to query the PTMs through black-box APIs. This scenario is called Language-Model-as-a-Service (LMaaS) (Diao et al. 2022; Sun et al. 2022). In this part we follow the experiments designed by (Sun et al. 2022)<sup>1</sup>, where language understanding task is formulated as a classification task predicting for

<sup>1</sup>Code can be found in <https://github.com/txsun1997/Black-Box-Tuning>

a batch of PTM-modified input texts  $X$  the labels  $Y$  in the PTM vocabulary, namely we need to tune the prompt such that the black-box PTM inference API  $f$  takes a continuous prompt  $\mathbf{p}$  satisfying  $Y = f(\mathbf{p}; X)$ . Moreover, to handle the high-dimensional prompt  $\mathbf{p}$ , (Sun et al. 2022) proposed to randomly embed the  $D$ -dimensional prompt  $\mathbf{p}$  into a lower dimensional space  $\mathbb{R}^d$  via random projection matrix  $\mathbf{A} \in \mathbb{R}^{D \times d}$ . Therefore, the objective becomes:

$$\min_{z \in \mathcal{Z}} \mathcal{L}(f(\mathbf{A}z + \mathbf{p}_0; X), Y),$$

where  $\mathcal{Z} = [-50, 50]^d$  is the search space and  $\mathcal{L}(\cdot)$  is cross entropy loss.

In our experimental setup, we configure the search space dimension to  $d = 500$  and the prompt length to 50, with RoBERTa (Liu et al. 2019a) serving as the backbone model. We evaluate performance on datasets SST-2 (Socher et al. 2013), Yelp polarity and AG’s News (Zhang, Zhao, and LeCun 2015), and RTE (Wang et al. 2018a). With a fixed API call budget of  $T = 8000$ , we pit RACE-CARS against SRACOS and the default DFO algorithm CMA-ES utilized in (Sun et al. 2022)<sup>2</sup>.

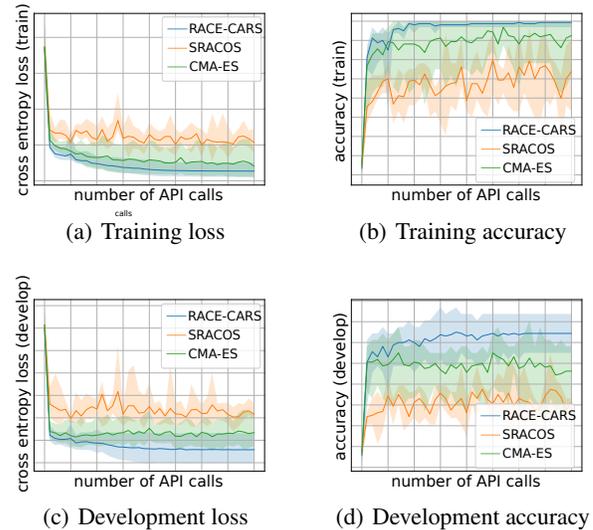


Figure 3: Comparisons on SST-2.

For our tests, the shrinking rate is  $\gamma = 0.7$ , with shrinking frequency of  $\rho = 0.002$ . Each algorithm is repeated 5 times independently with unique seeds. We assess the algorithms based on the mean and deviation of training loss, training accuracy, development loss and development accuracy. The SST-2 dataset results are highlighted in Figure 3, with additional findings for Yelp Polarity, AG’s News, and RTE detailed in the appendix. The results indicate that RACE-CARS consistently accelerates the convergence of SRACOS. While CMA-ES shows superior performance on Yelp Polarity, AG’s News, and RTE, it also exhibits signs of overfitting.

<sup>2</sup>Our choice to exclude ZO-Adam and DE is based on their sub-optimal performance with high-dimensional nonconvex black-box functions, as demonstrated in the last section.

*RACE-CARS* achieves comparable performance to *CMA-ES*, despite the latter’s hyperparameters being finely tuned. Notably, the hyperparameters for *RACE-CARS* were empirically adjusted based on the SST-2 dataset and then applied to the other three datasets without further tuning.

## Discussion

### Beyond Continuity

#### 1. For discontinuous objective functions.

Dimensionally local Holder continuity in Definition 5, imposes certain constraints on the objective through a set of continuous envelopes, whereas the objective is not supposed to be continuous. Beyond the continuous cases discussed in the previous section, *RACE-CARS* remains applicable to discontinuous objectives as well. Refer to appendix for a comprehensive understanding.

#### 2. For discrete optimization.

Similar to *SRACOS* algorithm, *RACE-CARS* retains the capability to tackle discrete optimization problems. The convergence theorems, presented in Theorem 2 and Theorem 3, encompass this situation by altering the measure of probability space to be for example, induced by counting measure. We extend experiments on mixed-integer programming problems, substantiating the acceleration of *RACE-CARS* empirically. See appendix for details.

### On the Concept Hypothesis-Target Shattering

The concept hypothesis-target shattering, central to our discussion, draws inspiration from the established learning theory notion of *shatter* and its deep ties to the Vapnik-Chervonenkis (VC) theory (Vapnik et al. 1998). At the heart of VC theory lies the VC dimension, a measure of a hypothesis family’s capacity to distinguish among data points based on their labels. Specifically, for a collection of data points with binary labels,  $S$ , a subset  $S' \subseteq S$  is said to be shattered by a hypothesis family  $\mathcal{H}$  if there exists a hypothesis  $h \in \mathcal{H}$  that perfectly aligns with the labels of points in  $S'$  and contrasts with those outside:

$$h(x) = \begin{cases} 1, & x \in S' \\ 0, & x \notin S'. \end{cases}$$

The shattering coefficient,  $\mathcal{S}(\mathcal{H}, n)$ , signifies the variety of point-label combinations that  $\mathcal{H}$  can produce for  $n$  points. The VC dimension is then defined as  $VC(\mathcal{H}) := \{\sup n : \mathcal{S}(\mathcal{H}, n) = 2^n\}$ , indicating the maximum number of points that can be distinctly labeled by  $\mathcal{H}$ .

In the context of classification-based DFO, we represent the target-representative capability of a family of hypotheses through hypothesis-target shattering, measuring the overlap of hypothesis’s active region and the target. Therefore the quintessence of algorithm design hinges on maximizing this quantity. However, discerning the target-representative capacity within the intricate landscape of *nonconvex, black-box* optimization problems is nontrivial. Nonetheless, the target-representative capability of hypotheses family generated by *RACOS*, although proved under the previous framework, empirically suggests sufficient efficacy in scenarios

where the objective function exhibits locally Holder continuity. Looking ahead, altering the *Training* and *Replacing* sub-procedures inherited from *SRACOS*, which may ideally lead to a bigger shattering rate and maintain the easy-to-sample characterization, will be another extension direction of the current study.

### Ablation Experiments

While it’s an appealing goal to develop a universally effective DFO algorithm for black-box functions without the need for hyperparameter tuning, this remains an unrealistic aspiration. Our proposed algorithm, *RACE-CARS*, introduces two hyperparameters: shrinking-rate  $\gamma$  and shrinking-frequency  $\rho$ . For an  $n$ -dimensional optimization problem, we call  $\gamma^{n\rho}$  shrinking factor of *RACE-CARS*. We take Ackley for a case study, design ablation experiments on the two hyperparameters of *RACE-CARS* to reveal the mechanism. We stipulate that we do not aim to identify a optimal combination of hyperparameters for maximizing the overlap with the target hypothesis. Instead, our aim is to provide empirical guidance for tuning these hyperparameters effectively. For further details, the reader is directed to the appendix.

## Conclusion

In this paper, we refine the framework of classification-based DFO as a stochastic process, and propose a novel learning concept named hypothesis-target shattering rate. Our research delves into the convergence properties of sequential-mode classification-based DFO algorithms and provides a fresh perspective on their query complexity upper bound. Delighted by the computational complexity upper bound under the new framework, we propose a theoretically grounded region-shrinking technique to accelerate the convergence. In empirical analysis, we study the scalability performance of *RACE-CARS* on both synthetic functions and black-box tuning for LMaaS, showing its superiority over *SRACOS*.

## References

- Akay, B.; and Karaboga, D. 2012. Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of intelligent manufacturing*, 23: 1001–1014.
- Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; and Koyama, M. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631.
- Bäck, T.; and Schwefel, H.-P. 1993. An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, 1(1): 1–23.
- Bickel, P. J.; and Levina, E. 2004. Some theory for Fisher’s linear discriminant function, naive Bayes’, and some alternatives when there are many more variables than observations. *Bernoulli*, 10(6): 989–1010.
- Bogunovic, I.; Scarlett, J.; Krause, A.; and Cevher, V. 2016. Truncated variance reduction: A unified approach to bayesian optimization and level-set estimation. *Advances in neural information processing systems*, 29.

- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.
- Céa, J. 1971. *Optimisation: théorie et algorithmes*. Jean Céa. Dunod.
- Chen, X.; Liu, S.; Xu, K.; Li, X.; Lin, X.; Hong, M.; and Cox, D. 2019. Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization. *Advances in neural information processing systems*, 32.
- Conn, A. R.; Scheinberg, K.; and Vicente, L. N. 2009. *Introduction to derivative-free optimization*. SIAM.
- Diao, S.; Huang, Z.; Xu, R.; Li, X.; Lin, Y.; Zhou, X.; and Zhang, T. 2022. Black-box prompt learning for pre-trained language models. *arXiv preprint arXiv:2201.08531*.
- El-Yaniv, R.; and Wiener, Y. 2012. Active Learning via Perfect Selective Classification. *Journal of Machine Learning Research*, 13(2).
- Eriksson, D.; Pearce, M.; Gardner, J.; Turner, R. D.; and Poloczek, M. 2019. Scalable global optimization via local Bayesian optimization. *Advances in neural information processing systems*, 32.
- Falkner, S.; Klein, A.; and Hutter, F. 2018. BOHB: Robust and efficient hyperparameter optimization at scale. In *International conference on machine learning*, 1437–1446. PMLR.
- Fan, J.; and Fan, Y. 2008. High dimensional classification using features annealed independence rules. *Annals of statistics*, 36(6): 2605.
- Fortin, F.-A.; De Rainville, F.-M.; Gardner, M.-A. G.; Parizeau, M.; and Gagné, C. 2012. DEAP: Evolutionary algorithms made easy. *The Journal of Machine Learning Research*, 13(1): 2171–2175.
- Frazier, P. I. 2018. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.
- Ge, D.; Liu, T.; Liu, J.; Tan, J.; and Ye, Y. 2022. SOLNP+: A Derivative-Free Solver for Constrained Nonlinear Optimization. *arXiv preprint arXiv:2210.07160*.
- Gotovos, A. 2013. *Active learning for level set estimation*. Master’s thesis, Eidgenössische Technische Hochschule Zürich, Department of Computer Science,.
- Hall, P.; Marron, J. S.; and Neeman, A. 2005. Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(3): 427–444.
- Hansen, N. 2016. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*.
- Hashimoto, T.; Yadlowsky, S.; and Duchi, J. 2018. Derivative free optimization via repeated classification. In *International Conference on Artificial Intelligence and Statistics*, 2027–2036. PMLR.
- Hensman, J.; Fusi, N.; and Lawrence, N. D. 2013. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*.
- Hu, Y.-Q.; Qian, H.; and Yu, Y. 2017. Sequential classification-based optimization for direct policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Larson, J.; Menickelly, M.; and Wild, S. M. 2019. Derivative-free optimization methods. *Acta Numerica*, 28: 287–404.
- Liao, T. W. 2010. Two hybrid differential evolution algorithms for engineering design optimization. *Applied Soft Computing*, 10(4): 1188–1199.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019a. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Liu, Y.-R.; Hu, Y.-Q.; Qian, H.; Qian, C.; and Yu, Y. 2017. Zoopt: Toolbox for derivative-free optimization. *arXiv preprint arXiv:1801.00329*.
- Liu, Y.-R.; Hu, Y.-Q.; Qian, H.; and Yu, Y. 2019b. Asynchronous classification-based optimization. In *Proceedings of the First International Conference on Distributed Artificial Intelligence*, 1–8.
- Michalski, R. S. 2000. Learnable evolution model: Evolutionary processes guided by machine learning. *Machine learning*, 38: 9–40.
- Nayebi, A.; Munteanu, A.; and Poloczek, M. 2019. A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*, 4752–4761. PMLR.
- Nelder, J. A.; and Mead, R. 1965. A simplex method for function minimization. *The computer journal*, 7(4): 308–313.
- Nesterov, Y.; and Spokoiny, V. 2017. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17: 527–566.
- Opara, K. R.; and Arabas, J. 2019. Differential Evolution: A survey of theoretical analyses. *Swarm and evolutionary computation*, 44: 546–558.
- Ragonneau, T. M.; and Zhang, Z. 2023. PDFO—A Cross-Platform Package for Powell’s Derivative-Free Optimization Solver. *arXiv preprint arXiv:2302.13246*.
- Ray, T.; and Saini, P. 2001. Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Engineering Optimization*, 33(6): 735–748.
- Ros, R.; and Hansen, N. 2008. A simple modification in CMA-ES achieving linear time and space complexity. In *International conference on parallel problem solving from nature*, 296–305. Springer.
- Scheinberg, K. 2022. Finite Difference Gradient Approximation: To Randomize or Not? *INFORMS Journal on Computing*, 34(5): 2384–2388.
- Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; and De Freitas, N. 2015. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175.

- Shi, H.-J. M.; Xuan, M. Q.; Oztoprak, F.; and Nocedal, J. 2021. On the numerical performance of derivative-free optimization methods based on finite-difference approximations. *arXiv preprint arXiv:2102.09762*.
- Snoek, J.; Larochelle, H.; and Adams, R. P. 2012. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.
- Sobol', I. M. 1967. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4): 784–802.
- Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.
- Sun, T.; Shao, Y.; Qian, H.; Huang, X.; and Qiu, X. 2022. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, 20841–20855. PMLR.
- Vapnik, V.; et al. 1998. Statistical learning theory.
- Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Wang, L.; Fonseca, R.; and Tian, Y. 2020. Learning search space partition for black-box optimization using monte carlo tree search. *Advances in Neural Information Processing Systems*, 33: 19511–19522.
- Wang, Z.; Gehring, C.; Kohli, P.; and Jegelka, S. 2018b. Batched large-scale Bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, 745–754. PMLR.
- Wang, Z.; Hutter, F.; Zoghi, M.; Matheson, D.; and De Freitas, N. 2016. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55: 361–387.
- Yang, L.; and Shami, A. 2020. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415: 295–316.
- Yi, Z.; Wei, Y.; Cheng, C. X.; He, K.; and Sui, Y. 2024. Improving sample efficiency of high dimensional Bayesian optimization with MCMC. *arXiv preprint arXiv:2401.02650*.
- Yu, W.-C. 1979. Positive basis and a class of direct search techniques. *Scientia Sinica, Special Issue of Mathematics*, 1(26): 53–68.
- Yu, Y.; and Qian, H. 2014. The sampling-and-learning framework: A statistical view of evolutionary algorithms. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, 149–158. IEEE.
- Yu, Y.; Qian, H.; and Hu, Y.-Q. 2016. Derivative-free optimization via classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.
- Yue, P.; Yang, L.; Fang, C.; and Lin, Z. 2023. Zeroth-order Optimization with Weak Dimension Dependency. In *The Thirty Sixth Annual Conference on Learning Theory*, 4429–4472. PMLR.
- Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.