Global Optimization with a Power-Transformed Objective and Gaussian Smoothing

Chen Xu [1]

Abstract

We propose a novel method, namely Gaussian Smoothing with a Power-Transformed Objective (GS-PowerOpt), that solves global optimization problems in two steps: (1) perform a (exponential) power-N transformation to the not necessarily differentiable objective $f : \mathbb{R}^d \to \mathbb{R}$ and get f_N, and (2) optimize the Gaussian-smoothed f_N with stochastic approximations. Under mild conditions on f, for any $\delta > 0$, we prove that with a sufficiently large power N_δ, this method converges to a solution in the δ-neighborhood of f's global optimum point, at the iteration complexity of $O(d^4\varepsilon^{-2})$. If we require that f is differentiable and further assume the Lipschitz condition on f and its gradient, the iteration complexity reduces to $O(d^2\varepsilon^{-2})$, which is significantly faster than the standard homotopy method. In most of the experiments performed, our method produces better solutions than other algorithms that also apply the smoothing technique.

1. Introduction

In this work, we consider the global optimization problem of

$$\max_{\boldsymbol{x}\in\mathcal{S}\subset\mathbb{R}^d} f(\boldsymbol{x}), \tag{1}$$

where $\mathcal{S}$ is a compact set and $f : \mathcal{S} \to \mathbb{R}$ is a continuous and possibly non-concave function with a *unique* global maximum point $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathcal{S}} f(\boldsymbol{x})$. The minimize-version of this problem is often encountered in machine learning, such as model trainings and adversarial attacks in computer vision. The gradient-based algorithms are commonly used, such as the (stochastic) gradient descent. In general, these methods only guarantee to approximate a locally optimal solution (Mertikopoulos et al., 2020; Lei et al., 2019; Choromanska et al., 2015).

Gaussian smoothing (e.g., Section 3 in (Nesterov & Spokoiny, 2017)) refers to convolving f with a Gaussian density to obtain a surrogate objective

$$\hat{f}_\sigma(\boldsymbol{\mu}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I_d)}[f(\boldsymbol{x})], \tag{2}$$

where $\mathcal{N}$ denotes a multivariate Gaussian distribution, $\sigma > 0$ is called the scaling parameter, and I_d denotes a $d \times d$ identity matrix. The smoothing effect possibly eliminates certain local extremes of the objective, and this technique is applied by the widely used homotopy methods for global optimizations.

Homotopy (e.g., (Mobahi & Fisher, 2015)), also called the graduated continuation, is a class of methods that aim to find a global solution to (1), with many applications in machine learning (Xu et al., 2016; Iwakiri et al., 2022). It converts the original problem to

$$\max_{\boldsymbol{\mu}\in\mathbb{R}^d,\sigma\geq 0} \mathbb{E}_{\boldsymbol{\xi}}[f(\boldsymbol{\mu}+\sigma\boldsymbol{\xi})], \tag{3}$$

where $\boldsymbol{\xi}$ is a random vector with a pre-selected distribution, such as a standard multivariate Gaussian distribution (Gaussian Homotopy, GH) or a uniform distribution in a unit sphere. Based on the observation that $\boldsymbol{\mu}^*_\sigma := \arg\max_{\boldsymbol{\mu}} \mathbb{E}[f(\boldsymbol{\mu}+\sigma\xi)]$ approaches[1] $\boldsymbol{x}^*$ as σ decreases to 0, the standard homotopy method adopts a double-loop mechanism: the outer loop iteratively decreases σ, and for each fixed value of σ, the inner loop solves $\max_{\boldsymbol{\mu}} \mathbb{E}[f(\boldsymbol{\mu}+\sigma\xi)]$, with the solution found in the current inner loop as the starting search point in the next inner loop.

The double-loop mechanism is costly in time. To tackle this issue, (Iwakiri et al., 2022) proposes a single-loop Gaussian homotopy (SLGH) method that updates both $\boldsymbol{\mu}$ and σ in each iteration. Although SLGH aims at the global optimums, in theory, it only guarantees to approximate a local optimum[2], which is not necessarily a global one. A time-efficient algorithm that aims at the global maximum is still to be found.

[1] Department of Engineering, Shenzhen MSU-BIT University, Shenzhen, P. R. China. Correspondence to: Chen Xu <xuchen@smbu.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

[1] Note that $\mathbb{E}[f(\boldsymbol{\mu}+\sigma\boldsymbol{\xi})] = f(\boldsymbol{\mu})$ if $\sigma = 0$.

[2] Theorem 4.1 in (Iwakiri et al., 2022) shows that SLGH approximates a solution $\hat{\boldsymbol{x}}$ such that $\mathbb{E}[\nabla f(\hat{\boldsymbol{x}})] = 0$.

Therefore, in this work, we propose a new method, namely the Gaussian Smoothing with a Power-Transformed Objective (GS-PowerOpt), which adopts a single-loop mechanism and aims at the global optimum. Specifically, with $\sigma > 0$ and a sufficiently large $N > 0$, it solves the surrogate problem of $\max_{\boldsymbol{\mu}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I_d)}[f_N(\boldsymbol{x})]$, where $f_N(\boldsymbol{x})$ modifies f by putting more weight to its global maximum point $\boldsymbol{x}^*$. The functional form of $f_N(\boldsymbol{x})$ can be either $f^N(\boldsymbol{x})$ or $e^{Nf(\boldsymbol{x})}$. According to our theory (see Subsection 3.1 for a preview), GS-PowerOpt converges to a neighborhood of the solution $\boldsymbol{x}^*$ to (1) at the iteration complexity of $O(d^4\varepsilon^{-2})$. This complexity can be improved to $O(d^2\varepsilon^{-2})$ if the commonly assumed Lipschitz conditions hold. The majority of experiments in Section 5 show that the GS-PowerOpt-based algorithm (e.g., EPGS, introduced later) outperforms other algorithms that also apply the smoothing technique.

Related Work. The homotopy methods, firstly proposed in (Blake & Zisserman, 1987), are intensively studied in the field of machine learning for global optimization problems. (Mobahi & Fisher, 2015) derives a bound for the worst scenario of the GH algorithm in a deterministic setting (i.e., the expectation $\mathbb{E}$ is computed accurately), while (Hazan et al., 2016) provides a convergence analysis in a stochastic setting (i.e., $\mathbb{E}$ is estimated with samples). (Gao & Sener, 2022) changes the distribution of the perturbation $\boldsymbol{\xi}$ from the commonly used Gaussian or uniform to the distribution that minimizes the estimation error of the gradient $\nabla_{\boldsymbol{\mu}}\mathbb{E}_{\boldsymbol{\xi}}[f(\boldsymbol{\mu}+\boldsymbol{\xi})]$. (Lin et al., 2023) proposes an algorithm for learning the whole solution path produced by the homotopy. Specifically, their algorithm learns a model $x_\phi(\sigma)$ that predicts (for any $\sigma > 0$) the solution to $\min_{\boldsymbol{\mu}} \nabla_{\boldsymbol{\mu}}\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(\boldsymbol{0},I_d)}[f(\boldsymbol{\mu}+\sigma\boldsymbol{\xi})]$, where ϕ is the set of model parameters to be trained.

The smoothing technique and the homotopy method have a large number of successful applications in machine learning, such as neural network training (Hazan et al., 2016), adversarial attack on image classification models (Iwakiri et al., 2022), solving L_1-regularized least-square problems ((Xiao & Zhang, 2012)), neural combinatorial optimization (Gao & Sener, 2022), improving the optimization algorithms of stochastic gradient descent and Adam (Starnes & Webster, 2024), and so on.

There are a few existing studies (Dvijotham et al., 2014; Roulet et al., 2020; Chen et al., 2024) that replace the original f with a surrogate objective that also involves the exponential transformation $e^{Nf(\boldsymbol{\mu}+\xi)}$ before smoothing. But their works are different from ours. (Dvijotham et al., 2014) proposes to minimize the surrogate objective of $G(\boldsymbol{\mu}) := \frac{1}{N}\log\left(\mathbb{E}_{\xi\sim\mathcal{N}(0,\Sigma)}[e^{Nf(\boldsymbol{\mu}+\xi)}]\right) + \frac{1}{2}\boldsymbol{\mu}^T R\boldsymbol{\mu}$. The theory[3] that justifies this surrogate objective requires that N, R, and Σ be selected so that $NR - \Sigma^{-1}$ is positive semi-definite. This indicates that our EPGS (see Section 2), for which $R = \boldsymbol{0}$ and $\Sigma = \sigma I_d$, is not a special case of theirs, since $-\sigma^{-1}I_d$ is negative definite and violates their requirement. Moreover, their theory on the distance between the optimum point of the new surrogate and $\boldsymbol{x}^*$ is incomplete (see Section 3.2 in (Dvijotham et al., 2014)). For optimal-control problems, (Roulet et al., 2020) study the surrogate objective that is similar to $G(\boldsymbol{\mu})$, and provide a theoretical analysis on the corresponding algorithm's convergence to a stationary point. However, the relation between this stationary point and the global optimum point $\boldsymbol{x}^*$ is not revealed. The proposed surrogate objective in (Chen et al., 2024) is $(1-N)f(\boldsymbol{\mu}) + \log\left(\mathbb{E}_{\xi\sim\mathcal{N}(0,\Sigma)}[e^{Nf(\boldsymbol{\mu}+\xi)}]\right)$, where $N \in [0,1]$. This is very different from our requirement that N is sufficiently large (see Theorem 2.1). Also, their theory (i.e., Theorem 10 in (Chen et al., 2024)) bounds $|f(\boldsymbol{x}^*) - f(\boldsymbol{\mu}^*)|$ with $O(N\sigma^2) + G(N,\sigma)$ where $G(N,\sigma)$ is in general nonlinear in N, which does not imply an improvement for increasing the value of N.

As shown later in Eq. (5), GS-PowerOpt iteratively generates solution candidates and update its internal state (i.e., $\boldsymbol{\mu}_t$) using candidates' evaluations. This pattern is shared by many evolutionary algorithms (EA), such as simulated annealing (Van Laarhoven et al., 1987), particle swarm optimization (PSO, e.g., (Miranda, 2018) and Section 3.1.5 in (Locatelli & Schoen, 2013)), the cross-entropy method for optimization (Boer et al., 2005), and the covariance matrix adaptation evolution strategy (CMA-ES, (Hansen & Ostermeier, 2001)). However, GS-PowerOpt is distinct from the EA algorithms in its construction of the surrogate objective $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I_d)}[f_N(\boldsymbol{x})]$ and its use of the stochastic gradient method for optimization.

Contribution. This paper introduces a novel zeroth-order method, GS-PowerOpt, for solving global optimization problems, with the contributions summarized as follows.

1. To our knowledge, this is the first work that proposes the idea[4] of putting sufficiently large weight on the global maximum values of the objective, to decrease the distance between the optimum point before and after Gaussian smoothing (i.e., $\|\boldsymbol{x}^* - \boldsymbol{\mu}^*\|$). It provides a theoretical foundation (Theorem 2.1 and Lemma 3.4), as well as motivations, for future studies to find better ways of weight shifting than power transforms.

2. In theory, compared to the iteration complexity of $O(d^2/\varepsilon^4)$ of the standard zeroth-order (ZO) homotopy method analyzed in Theorem 5.1 of (Hazan et al., 2016), GS-PowerOpt is much faster at the iteration complexity of $O(d^2\varepsilon^{-2})$ under the Lipschitz condi-

[3] According to Theorem 3.1 in (Dvijotham et al., 2014), $\min_\mu G(\boldsymbol{\mu},\Sigma)$ is a convex problem given that $NR - \Sigma^{-1}$ is positive definite and $\mathcal{C}$ is convex.

[4] (Dvijotham et al., 2014; Roulet et al., 2020; Chen et al., 2024), which involve power transforms, have not mentioned this idea.

tions (see Section 3.5) and does not require the strong assumption of "σ-nice". Compared to the ZO methods in (Ghadimi & Lan, 2013; Chen et al., 2019) and ZO-SLGH, the main advantage of GS-PowerOpt lies in its ability to approximate the global solution. The majority of our experiments show that it ranks highest among all these algorithms that apply smoothing (see Section 5.4).

3. Our convergence analysis in Corollary 3.9 does not require the Lipschitz condition on the original objective f, which is assumed in the theoretical analysis of homotopy methods in other studies (Hazan et al., 2016; Iwakiri et al., 2022). Therefore, our analysis applies to more situations.

4. The theory derived in this work is on the distance between the found solution and the optimal one, while the convergence analysis in other studies on homotopy (Hazan et al., 2016; Iwakiri et al., 2022) is on the objective value of the found solution. Therefore, our theory has a wider range of applications (e.g., for problems that concern the distance between the found solution and the true one, such as inverse problems (Arridge et al., 2019) and adversarial attacks in image recognitions).

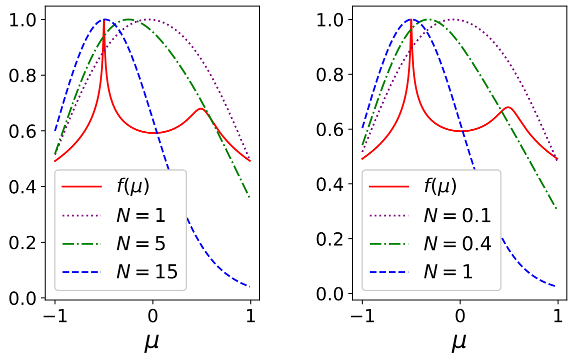
Road Map. Section 2 describes GS-PowerOpt and the intuition behind it. Its theoretical convergence analysis is performed in Section 3 and 4 (see Subsection 3.1 for a preview). Numerical experiments are included in Section 5. Section 6 provides a guidance on selecting values of the hyper-parameters, and Section 7 concludes. The appendix includes a table of notations, *proofs to all our theoretical results (lemma, theorem, corollary and proposition)*, details of certain compared algorithms, and hyper-parameter values used in our experiments. All our codes can be found at `http://github.com/chen-research/GS-PowerTransform`.

2. GS-PowerOpt: The Proposed Method

2.1. Motivation

Intuitively, if we modify $f(\boldsymbol{x})$ to put sufficiently large weight on its global maximum point $\boldsymbol{x}^*$, then $\boldsymbol{\mu}^* := \arg\max_{\boldsymbol{\mu}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I_d)}[f(\boldsymbol{x})]$ should get close enough to $\boldsymbol{x}^*$. One way of such modification is by taking (exponential) powers of f, if $f(\boldsymbol{x}^*) > 1$. The difference $f^N(\boldsymbol{x}^*) - f^N(\boldsymbol{x})$ is positively related with the power N, which indicates that more weight is put on $\boldsymbol{x}^*$ as N increases. Figure 1 verifies this intuition with an example. As shown in these two toy examples, $\boldsymbol{\mu}^*$ approaches $\boldsymbol{x}^*$ as N increases.

From the above intuition, we propose GS-PowerOpt for solving the global optimization problem (1). It is a new method that places more weight on the objective f's maximum value (by increasing the gap between $f(\boldsymbol{x}^*)$ and f-values at other points) before performing Gaussian smoothing. Based on GS-PowerOpt, we design two algorithms[5], Power Gaussian Smoothing (PGS) and Exponential Power Gaussian Smoothing (EPGS), which are featured with replacing the original objective $f(\boldsymbol{x})$ with a (exponential) power transformation. Specifically, with σ and N as two hyper-parameters, PGS solves $\max_{\boldsymbol{\mu}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I_d)}[f^N(X)]$ and EPGS solves $\max_{\boldsymbol{\mu}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I_d)}[e^{Nf(\boldsymbol{x})}]$, both using a stochastic gradient ascent algorithm derived in this paper, which does not require the differentiability of f.



(a) Gauss. Smooth of f^N.

(b) Gauss. Smooth of e^{Nf}.

Figure 1: Graph of approximated $F_{N,\sigma}(\mu)$, where $F_{N,\sigma}(\mu)$ is defined as (a) $\mathbb{E}_{x\sim\mathcal{N}(\mu,\sigma^2)}[f^N(x)]$ or (b) $\mathbb{E}_{x\sim\mathcal{N}(\mu,\sigma^2)}[e^{Nf(x)}]$, $\sigma = 0.5$, and $f(\mu) = -\log((\mu+0.5)^2+10^{-5}) - \log((\mu-0.5)^2+10^{-2}) + 10$ for $|\mu| \leq 1$ and $f(\mu) = 0$ for $|\mu| > 1$. All function graphs are scaled to have a maximum value of 1 for easier comparisons.

2.2. Theoretical Justification of the Motivation

In Theorem 2.1, we lay the ground for justifying the intuition that motivates GS-PowerOpt: Given $\sigma > 0$, for any $\delta > 0$, there exists a threshold such that whenever N exceeds this threshold, the global maximum point $\arg\max_{\boldsymbol{\mu}} F_N(\boldsymbol{\mu}, \sigma)$ lies within a δ-neighborhood of $\boldsymbol{x}^*$, where $F_N(\boldsymbol{\mu}, \sigma) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma I_d)}[f_N(\boldsymbol{x})]$ and

$$
\begin{aligned}
f_N(\boldsymbol{x}) &:= \begin{cases} f^N(\boldsymbol{x}), & \boldsymbol{x} \in \mathcal{S}; \\ 0, & \text{otherwise}, \end{cases} \quad \text{(PGS setting)}; \\
f_N(\boldsymbol{x}) &:= \begin{cases} e^{Nf(\boldsymbol{x})}, & \boldsymbol{x} \in \mathcal{S}; \\ 0, & \text{otherwise}. \end{cases} \quad \text{(EPGS setting)}.
\end{aligned} \tag{4}
$$

Theorem 2.1. *Let $f : \mathcal{S} \subset \mathbb{R}^d \to \mathbb{R}$ be a continuous function that is possibly non-concave (and non-negative only for the case of PGS), where $\mathcal{S}$ is compact. Assume that f has a global maximum $\boldsymbol{x}^*$ such that* $\sup_{\boldsymbol{x}:\|\boldsymbol{x}-\boldsymbol{x}^*\|\geq\delta} f(\boldsymbol{x}) <$

[5] See Algorithm 1 for the two algorithms.

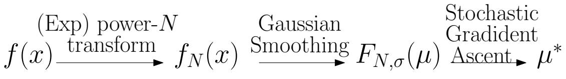


Figure 2: A Flowchart of GS-PowerOpt

$f(\boldsymbol{x}^*)$ for any $\delta > 0$. For any $\sigma > 0$ and any $N > 0$, define

$$F_{N,\sigma}(\boldsymbol{\mu}) := \frac{1}{(\sqrt{2\pi}\sigma)^d} \int_{\boldsymbol{x}\in\mathbb{R}^d} f_N(\boldsymbol{x}) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x}, \boldsymbol{\mu} \in \mathbb{R}^d$$

where f_N is defined in (4) for either PGS or EPGS. Then, for any $M > 0$ and $\delta > 0$ such that $\bar{B}(\boldsymbol{x}^*; \delta) := \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \delta\} \subset \mathcal{S}$, there exists $N_{\delta,\sigma,M} > 0$, such that whenever $N > N_{\delta,\sigma,M}$, for any $\|\boldsymbol{\mu}\| \leq M$ and any $i \in \{1, 2, ..., d\}$ we have that: $\frac{\partial F_N(\boldsymbol{\mu},\sigma)}{\partial \mu_i} > 0$ if $\mu_i < x_i^* - \delta$, and $\frac{\partial F_N(\boldsymbol{\mu},\sigma)}{\partial \mu_i} < 0$ if $\mu_i > x_i^* + \delta$. Here, μ_i and x_i^* denote the i^{th} entry of $\boldsymbol{\mu}$ and $\boldsymbol{x}^*$, respectively.

Remark 2.2. The inequalities in Theorem 2.1 imply that any stationary point $\boldsymbol{\mu}'$ (if any) of $F_{N,\sigma}(\boldsymbol{\mu})$ within the region $\|\boldsymbol{\mu}\| \leq M$ satisfies $|\mu_i' - x_i^*\| \leq \delta$ for all $i \in \{1, 2, ..., d\}$.

Proposition 2.3. *Given $\sigma > 0$ and $N > 0$, $F_{N,\sigma}(\boldsymbol{\mu})$ in Theorem 2.1 attains a global max at some $\boldsymbol{\mu}^* \in \mathbb{R}^d$.*

2.3. Solution Updating Rule of GS-PowerOpt

For the optimization problem (1), based on Theorem 2.1, with the pre-selected hyper-parameters N and $\sigma > 0$, GS-PowerOpt follows a stochastic gradient ascent scheme to solve $\max_{\boldsymbol{\mu}} F_{N,\sigma}(\boldsymbol{\mu})$. Specifically, the rule for updating the solution candidate is

$$\text{GS-PowerOpt :} \qquad \boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \alpha_t \hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu}_t), \tag{5}$$

where $\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu}_t) := \frac{1}{K}\sum_{k=1}^{K}(\boldsymbol{x}_k - \boldsymbol{\mu}_t) f_N(\boldsymbol{x}_k)$, $\{\boldsymbol{x}_k\}_{k=1}^K$ are independently sampled from the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_t, \sigma^2 I_d)$, and $f_N(\boldsymbol{x}_k)$ is defined in (4). Note that $\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu}_t)$ is a sample estimate of $\sigma^2 \nabla F_{N,\sigma}(\boldsymbol{\mu})$, since

$$\begin{aligned} &\nabla F_{N,\sigma}(\boldsymbol{\mu}) = \nabla_{\boldsymbol{\mu}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}_t,\sigma^2 I_d)}[f_N(\boldsymbol{x})] \\ &= (\sqrt{2\pi})^{-d} \sigma^{-(d+2)} \int_{\boldsymbol{x}\in\mathbb{R}^d} (\boldsymbol{x} - \boldsymbol{\mu}_t) f_N(\boldsymbol{x}) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}_t\|^2}{2\sigma^2}} d\boldsymbol{x} \\ &= \sigma^{-2} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}_t,\sigma^2 I_d)}[(\boldsymbol{x} - \boldsymbol{\mu}_t) f_N(\boldsymbol{x})], \end{aligned} \tag{6}$$

where the interchange of differentiation and integral in the second line can be justified by Lebesgue's dominated convergence theorem. A flowchart of the algorithm is given in Figure 2.

Based on GS-PowerOpt, PGS and EPGS are designed in Algorithm 1. They normalize the gradient before updating the solution, which is a common practice to stabilize results. An effective method to avoid computation overflows caused by a large N-value can be found in Appendix B.

Algorithm 1 PGS/EPGS for Solving (1)

```
Input: The power $N > 0$, the scaling parameter $\sigma > 0$, the objective $f$, the initial value $\boldsymbol{\mu}_0$, the number $K$ of sampled points for gradient approximation, the total number $T$ of $\boldsymbol{\mu}$-updates, and the learning rate schedule $\{\alpha_t\}_{t=0}^{T-1}$.
for $t$ from 0 to $T-1$ do
    Independently sample from the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_t, \sigma I_d)$ and obtain $\{\boldsymbol{x}_k\}_{k=1}^K$.
    $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \alpha_t \hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu}_t) / \|\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu}_t)\|$, where $\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu}_t)$ is defined below (5).
end for
Return $\{\boldsymbol{\mu}_t\}_{t=1}^T$, from which $\boldsymbol{\mu}^*$ is selected to approximate $\boldsymbol{x}^*$ (e.g., $\boldsymbol{\mu}^* := \arg\max_{\boldsymbol{\mu}\in\{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,...,\boldsymbol{\mu}_T\}} f(\boldsymbol{\mu})$).
```

3. Convergence Analysis

3.1. Preview of the Main Theoretical Results

In this section, we prove our main theoretical results, Lemma 3.4 and Corollary 3.9, which indicate that GS-PowerOpt converges to an arbitrarily small neighborhood of $\boldsymbol{x}^* := \arg\max_{\boldsymbol{x}\in\mathcal{S}} f(\boldsymbol{x})$. Specifically, we show that, for any *fixed* $\delta > 0$ and any *fixed* $\sigma > 0$, given N larger than some (δ, σ)-dependent threshold, all the stationary points of $F_{N,\sigma}(\boldsymbol{\mu})$ lie in a δ-neighborhood of $\boldsymbol{x}^*$ (Lemma 3.4), and the stochastic gradient ascent in Eq. (5) converges to one of these stationary points with the iteration complexity of $O_N(d^4\varepsilon^{-2})$ (Corollary 3.9 and Remark 3.10). This complexity is derived from the bound given in Theorem 3.7, whose proof requires Lemma 3.5 and 3.6.

Furthermore, with additional conditions on f, we prove that the complexity is improved to $O_N(d^2\varepsilon^{-2})$ in Subsection 3.5, and that the dependence of the O-factor on N can be removed (Corollary 4.2).

3.2. Notations in Section 3

In this section, let $\delta > 0$ and $\sigma > 0$ be fixed, and δ satisfies

$$\mathcal{S}_{\boldsymbol{x}^*,\delta} := \{\boldsymbol{\mu} \in \mathbb{R}^d : |\mu_i - x_i^*| \leq \delta, i \in \{1, 2, ..., d\}\} \subset \mathcal{S}, \tag{7}$$

where i denotes the i^{th} entry, $\boldsymbol{x}^* := \arg\max_{\boldsymbol{x}\in\mathcal{S}} f(\boldsymbol{x})$, and $\mathcal{S}$ is f's compact domain as specified in Assumption 3.1. Let $N > 0$ be such that all the stationary points (if any) of $F_{N,\sigma}(\boldsymbol{\mu})$ within the region $\|\boldsymbol{\mu}\| < \sqrt{d}M$ lie in $\mathcal{S}_{\boldsymbol{x}^*,\delta}$, where M is specified in Assumption 3.1. Such an N exists because of Theorem 2.1 (see Remark 2.2), and its value depends on δ, σ, and $\sqrt{d}M$. Unless needed for clarity, we omit the subscripts in $F_{N,\sigma}(\boldsymbol{\mu})$ as they remain fixed throughout the rest of Section 3, and write $F(\boldsymbol{\mu})$ instead. $\nabla F(\boldsymbol{\mu})$ refers to the gradient of F with respect to $\boldsymbol{\mu}$.

3.3. Assumptions and Lemmas

Assumption 3.1. Assume that $f(\boldsymbol{x}) : \mathcal{S} \to \mathbb{R}$ is a function satisfying the conditions specified in Theorem 2.1, where $\mathcal{S} \subset \mathcal{S}_M := \{\boldsymbol{x} \in \mathbb{R}^d : |x_i| \leq M, i \in \{1, 2, ..., d\}\}$, for some $M > 0$.

Assumption 3.2. Assume that the learning rate α_t satisfies

$$\alpha_t > 0, \ \sum_{t=0}^{\infty} \alpha_t = +\infty, \text{ and } \sum_{t=0}^{\infty} \alpha_t^2 < +\infty.$$

Lemma 3.3. *Under Assumption 3.1, $F_{N,\sigma}(\boldsymbol{\mu})$ and $\nabla F_{N,\sigma}(\boldsymbol{\mu})$ are well-defined for all $\boldsymbol{\mu} \in \mathbb{R}^d$.*

Lemma 3.4. *Under Assumption 3.1, any stationary point of $F_{N,\sigma}(\boldsymbol{\mu})$ in $\mathbb{R}^d$ belongs to $\mathcal{S}_{\boldsymbol{x}^*,\delta}$ (defined in (7)).*

Lemma 3.5. *Under Assumption 3.1, for any $\sigma > 0$, the objective function $F_{N,\sigma}(\boldsymbol{\mu})$ is Lipschitz Smooth. That is, for any $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^d$,*

$$\|\nabla F_{N,\sigma}(\boldsymbol{\mu}_1) - \nabla F_{N,\sigma}(\boldsymbol{\mu}_2)\| \leq L\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|,$$

where $L = 2d\sigma^{-2} f_N(\boldsymbol{x}^*)$, and $f_N(\boldsymbol{x}^*) = f^N(\boldsymbol{x}^*)$ for PGS and $f_N(\boldsymbol{x}^*) = e^{Nf(\boldsymbol{x}^*)}$ for EPGS.

Lemma 3.6. *Under Assumption 3.1, for any $\sigma > 0$, let $\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu})$ be defined in (5). Then, $\mathbb{E}[\|\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu})\|^2] < G$, where*

$$G = \begin{cases} d\sigma^2 f^{2N}(\boldsymbol{x}^*), & \text{for PGS;} \\ d\sigma^2 e^{2Nf(\boldsymbol{x}^*)}, & \text{for EPGS.} \end{cases}$$

3.4. Iteration Complexity

Theorem 3.7. *Let $\{\boldsymbol{\mu}_t\}_{t=0}^{T} \subset \mathbb{R}^d$ be produced by following the iteration rule of (5), with a pre-selected and deterministic $\boldsymbol{\mu}_0$ and all the involved terms defined as in Section 3.2. Then, under Assumption 3.1 and 3.2, we have that*

$$\sum_{t=0}^{T-1} \alpha_t \sigma^2 \mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t)\|^2] \leq f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + LG\sum_{t=0}^{\infty} \alpha_t^2,$$

where L and G are as defined in Lemma 3.5 and Lemma 3.6, respectively.

Remark 3.8. The inequality in Theorem 3.7 implies that $\min_{t\in\{0,1,...,T-1\}} \mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t)\|^2]$ approaches 0 as $T \to \infty$, since the right-hand side is finite and $\sum_{t=0}^{\infty} \alpha_t = \infty$. Furthermore, the iteration complexity of this convergence is given in Corollary 3.9 when $\alpha_t := (t+1)^{-(1/2+\gamma)}$.

Corollary 3.9. *Suppose Assumption 3.1 and 3.2 hold. Let $\{\boldsymbol{\mu}_t\}$ be produced by the stochastic gradient ascent rule (5) of GS-PowerOpt, with a pre-selected and deterministic $\boldsymbol{\mu}_0$. Then, for any $\varepsilon \in (0, 1)$, after*

$$T > (C_1 C_2 d^2 \varepsilon^{-1})^{2/(1-2\gamma)} = O_N((d^2\varepsilon^{-1})^{2/(1-2\gamma)})$$

times of $\boldsymbol{\mu}_t$-updating by (5), we have that $\min_{t\in\{0,1,...,T\}} \mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t)\|^2] < \varepsilon$. Here, $\gamma \in (0, 1/2)$ is a parameter in the learning rate $\alpha_t := (t+1)^{-(1/2+\gamma)}$, $C_0 = f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + 2f_N^3(\boldsymbol{x}^*)\sum_{t=1}^{\infty} t^{-(1+2\gamma)}$, $C_1 = C_0(1-2\gamma)\sigma^{-2}$, and $C_2 = \max\{1, 2/C_1\}$.

Remark 3.10. The iteration complexity is approximately $O_N(d^4\varepsilon^{-2})$ if γ is close to 0. Also, the dependence of the big-O factor on N can be removed with an additional assumption. See Corollary 4.2 for this point.

3.5. Improved Iteration Complexity under Lipschitz Conditions

In Proposition 3.11, we re-derive the coefficient L in Lemma 3.5 under the Lipschitz condition on the objective f and its first derivative ∇f. The new value of L does not depend on the dimension d. With this result, the iteration complexity in Corollary 3.9 becomes $O_N(d^2\varepsilon^{-2})$ when γ is close to 0.

Proposition 3.11. *Assume Assumption 3.1, $f(\boldsymbol{x}) = 0$ on the boundary of f's domain, the Lipschitz condition on f: $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq L_0\|\boldsymbol{x} - \boldsymbol{y}\|$, and the Lipschitz condition on ∇f: $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L_1\|\boldsymbol{x} - \boldsymbol{y}\|$. Then, the following two statements are true.*

1. *$\|\nabla F_N(\boldsymbol{\mu}_1, \sigma) - \nabla F_N(\boldsymbol{\mu}_2, \sigma)\| \leq L\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|$ holds for any $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^d$, where $L = (Nf^{N-1}(\boldsymbol{x}^*)L_1 + N(N-1)L_0^2 f^{N-2}(\boldsymbol{x}^*))$ under the PGS setting, and $L = (NL_1 e^{Nf(\boldsymbol{x}^*)} + N^2 L_0^2 e^{Nf(\boldsymbol{x}^*)})$ under the EPGS setting.*

2. *The iteration complexity in Corollary 3.9 reduces to $O_N((d\varepsilon^{-1})^{2/(1-2\gamma)})$.*

Remark 3.12. "$f(\boldsymbol{x}) = 0$ on the boundary of f's domain" is assumed without loss of generality, since we can continuously extend f to a larger domain to make it true.

The Lipschitz condition is common in literature. For example, the convergence analysis for the standard homotopy method requires the Lipschiz assumption on f (see Theorem 5.1 in (Hazan et al., 2016)), and the convergence analysis for ZOSLGH requires the Lipschitz condition on both f and ∇f (see Assumption A1 in (Iwakiri et al., 2022)).

4. Removing the Iteration Complexity's Dependence on N

Lemma 3.4 and Corollary 3.9 imply that GS-PowerOpt converges to a δ-neighborhood of $\boldsymbol{x}^*$ at the iteration complexity of $O_N(d^4\varepsilon^{-2})$. To ensure the convergence to a smaller neighborhood of $\boldsymbol{x}^*$, we need to decrease δ. However, this possibly requires a larger N for Lemma 3.4 and Corollary 3.9 to hold (recall N's definition in Subsection 3.2), and in turn increases the iteration complexity of $O_N(d^4\varepsilon^{-2})$. The following additional assumption removes the complexity's dependence on N, and guarantees an iteration complexity of $O(d^4\varepsilon^{-2})$ regardless of how close the positive δ is to 0.

Assumption 4.1. $f(\boldsymbol{x}) \in [0, 1)$ for the PGS case, and $f(\boldsymbol{x}) < 0$ for the EPGS case.

In practice, this assumption can be realized if we know an upper bound of the objective $f(\cdot)$. For example, if it is known that $f(\cdot) < B$, then we can proceed EPGS with the new objective function $f_1 := f - B$ (f and f_1 share the same global maximum point $\boldsymbol{x}^*$). Also, at least shown by our experiments (Table 1 - 4), GS-PowerOpt works well for objectives that do not necessarily satisfy this assumption.

Corollary 4.2. *Under Assumption 3.1 and 4.1, the iteration complexity in Corollary 3.9 becomes $O((d^2\varepsilon^{-1})^{2/(1-2\gamma)})$, and the iteration complexity in Point 2 of Proposition 3.11 becomes $O((d\varepsilon^{-1})^{2/(1-2\gamma)})$, both of which are independent from N.*

Remark 4.3. Note that the two complexities approach $O(d^4\varepsilon^{-2})$ and $O(d^2\varepsilon^{-2})$, respectively, as $\gamma \to 0$.

5. Experiments

5.1. Effects of Increasing Powers

We illustrate the improvements made by increasing N for PGS/EPGS through an example problem of $\max_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$, where

$$f(\boldsymbol{x}) = -\log(\|\boldsymbol{x}-\boldsymbol{m}_1\|^2+10^{-5})-\log(\|\boldsymbol{x}-\boldsymbol{m}_2\|^2+10^{-2}), \tag{8}$$

the global maximum point $\boldsymbol{m}_1 \in \mathbb{R}^d$ has all its entries equal to -0.5, and the local maximum point $\boldsymbol{m}_2 \in \mathbb{R}^d$ has all its entries equal to 0.5. The graph of its 2D-version is plotted in Figure 3 (a).

With each value of N, we perform both the PGS and EPGS in Algorithm 1 to solve this problem. The experiments are done in two settings, one is two-dimensional ($d = 2$) and the other is five-dimensional ($d = 5$). More details can be found in Appendix E.

The results, plotted in Figure 4, show that, as N increases, the distance between the produced solution $\boldsymbol{\mu}^*$ and the global maximum point $\boldsymbol{x}^*$ approaches zero (see the decreasing MSE curve in the plot), which is consistent with Theorem 2.1 and the idea that $F(\boldsymbol{\mu})$'s maximum $\boldsymbol{\mu}^*$ approaches the global maximum point $\boldsymbol{x}^*$ of f as we put more weight on $f(\boldsymbol{x}^*)$.

5.2. Performance on Benchmark Objective Functions

In this subsection, we test the performance of PGS and EPGS on solving (1), with the objective f being either of the two popular benchmark objective functions, the Ackley and the Rosenbrock (maximize-version).

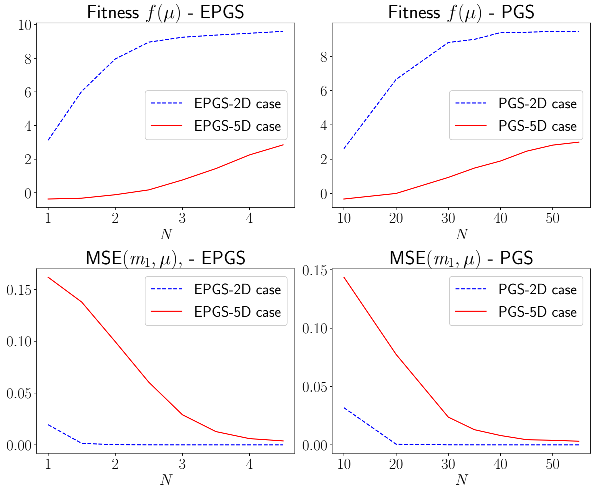


Figure 4: Effects of Increasing N. For each N, we perform the algorithm 100 times to stabilize results, and obtain $\{\boldsymbol{\mu}_k\}_{k=1}^{100}$. The average fitness $\sum_{k=1}^{100} f(\boldsymbol{\mu}_k)/100$ and $\sum_{k=1}^{100}$ MSE$(\boldsymbol{m}_1, \boldsymbol{\mu}_k)/100$ are plotted, where MSE$(\boldsymbol{m}_1, \boldsymbol{\mu}_k) := \sum_{i=1}^{d}(\mu_{ki} + 0.5)^2/d$, $\sigma = 1.0$, and f is defined in (8). Note that $\boldsymbol{x}^* = \boldsymbol{m}_1$ has all its entries equal to -0.5.

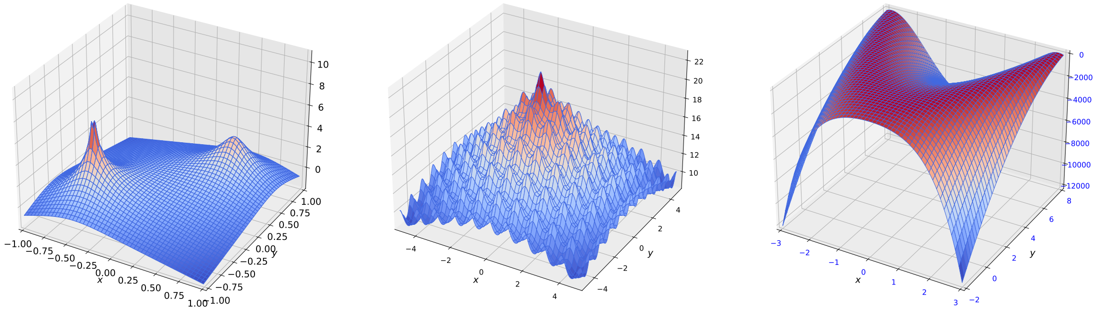
The performances of other popular global algorithms (maximize-version) are reported for comparison, including (1) a standard homotopy method STD-Homotopy; (2) two zeroth-order single-loop Gaussian homotopy algorithms, ZOSLGHd and ZOSLGHr (see the determinsitic version[6] of Algorithm 3 in (Iwakiri et al., 2022)); (3) the gradient-ascent version of the zeroth-order algorithm of ZOSGD (see Equation (1) in (Chen et al., 2019) and Section 2.1 in (Ghadimi & Lan, 2013)) and ZOAdaMM (Algorithm 1 in (Chen et al., 2019))[7], which were also used for comparisons in (Iwakiri et al., 2022); (4) as well as CMA-ES (Hansen & Ostermeier, 2001; Hansen et al., 2019), one of the state-of-the-art evolutionary algorithms.

More details and discussions on the compared algorithms can be found in Appendix D. The selected hyper-parameter values of all algorithms can be found in Appendix F.

Ackley. The Ackley objective function is characterized by numerous local optima and a single global optimum. We solve the max-version of the corresponding problem, which is $\max_{(x,y)\in\mathbb{R}^2} f(x, y)$ and $f(x, y) = 20e^{-0.2\sqrt{0.5(x^2+y^2)}} + e^{0.5(\cos(2\pi x)+\cos(2\pi y))}$.The graph of this function is plotted in Figure 3(b). From both the func-

[6] A deterministic version of SLGH is for solving the optimization problem of $\max_{\boldsymbol{x}} f(\boldsymbol{x})$, with f being a deterministic function.

[7] ZOSGD and ZOAdaMM were originally designed to solve (3). We take their solutions to (3) as solutions to (1), and treat the scaling parameter in (3) as a hyper-parameter.



(a) Two-log Objective (b) Ackley Func. (Max-Version) (c) Rosenbrock Func. (Max-Version).

Figure 3: Graph of the Benchmark Objective Functions.

Table 1: Performances on Maximizing Ackley. “Iter. Taken” refers to the number of iterations taken to reach the best found solution. All numbers are rounded to keep at most 3 decimal places.

Algorithm	Iter. Taken	Best Solution Found ($\boldsymbol{\mu}^*$)	$f(\boldsymbol{\mu}^*)$
CMA-ES	116	$(0.0, 0.0)$	22.718
EPGS ($N = 1$)	143	$(0.001, 0.0)$	22.683
PGS ($N = 20$)	141	$(0.001, 0.002)$	22.678
ZOSLGHr	131	$(0.001, -0.002)$	22.621
ZOAdaMM	158	$(0.005, 0.001)$	22.613
ZOSLGHd	123	$(-0.003, -0.001)$	22.61
ZOSGD	174	$(0.005, 0.007)$	22.596
STD-Homotopy	194	$(0.962, 0.946)$	17.58

tional form and the graph, it is not difficult to see that $f(x, y)$ attains its maximum at $\boldsymbol{x}^* = (0, 0)$.

The solutions and their f-values found by each of the compared algorithms are reported in Table 1. For each algorithm, the total number T of iterations is set as 200, and $K = 100$ samples are used for each solution update. The hyper-parameters are selected by trials. For each set of hyper-parameter candidates, we perform 100 experiments and *take the average* to stabilize the results (so all the reported numbers are averages). The initial solution value for each experiment is drawn from a multivariate Gaussian with mean $\boldsymbol{\mu}_0 = [5.0, 5.0]$ and covariance $0.01I_2$.

The results in the table show that all the algorithms perform well and are close to each other, except STD-Homotopy.

Rosenbrock. The Rosenbrock objective is known to be difficult to optimize, since its global optimum point $\boldsymbol{x}^* = (1.0, 1.0)$ is surrounded by a flat curved plane (see Figure 3(c)). The problem to be solved is $\max_{(x,y)\in\mathbb{R}^2} f(x, y)$, where $f(x, y) = -100(y - x^2)^2 - (1 - x)^2$.

Table 2: Performances on Maximizing Rosenbrock. The differences between this experiment and that in Table 1 are: The number T of iterations is set as 1000, and the initial solution candidate is taken as $\boldsymbol{\mu}_0 := [-3.0, 2.0]$. For PGS, the Rosenbrock is added by $20,000$ to ensure the search agent only encounter positive values. The global maximum point of the Rosenbrock function is (1,1).

Algorithm	Iter. Taken	Best Solution Found ($\boldsymbol{\mu}^*$)	$f(\boldsymbol{\mu}^*)$
CMA-ES	72	$(1.0, 1.0)$.	0.0
EPGS ($N = 3$)	487.	$(0.999, 1.000)$	-0.017
STD-Homotopy	624	$(0.903, 0.885)$	-2.401
PGS ($N = 1$)	513	$(0.773, 1.025)$	-22.84
ZOAdaMM	852	$(0.004, 0.618)$.	-39.206
ZOSLGHr	148	$(0.105, 0.938)$.	-88.477
ZOSGD	45	$(0.272, 1.173)$.	-121.14
ZOSLGHd	471	$(-0.447, 1.991)$.	-137.016

The performance of each algorithm is recorded in Table 2, which shows that the performances of CMA-ES and EPGS are close, and are significantly better than those of other algorithms.

5.3. Performance on the Black-box Targeted Adversarial Attack

Let $\mathcal{C}$ be a black-box[8] image classifier. The targeted adversarial attack on $\mathcal{C}$ refers to the task of modifying the pixels of a given image $\boldsymbol{a}$ so that $\mathcal{C}(\boldsymbol{a} + \boldsymbol{x})$ is equal to a pre-specified target label $\mathcal{T}$. We perform attacks in the most difficult case - the target label $\mathcal{T}$ is pre-selected to be the one with the smallest predicted probability. Another goal of this task is to minimize the perturbation size $\|\boldsymbol{x}\|$. Hence, we set the

[8] A black-box classifier refers to a classification model whose parameters are not accessible.

loss as

$$L(\boldsymbol{x}) := \max(\max_{i \neq \mathcal{T}} \mathcal{C}(\boldsymbol{a}+\boldsymbol{x})_i - \mathcal{C}(\boldsymbol{a}+\boldsymbol{x})_{\mathcal{T}},\ \kappa) + \lambda \|\boldsymbol{x}\|,$$

where $\mathcal{C}(\boldsymbol{a}+\boldsymbol{x})_i$ denotes the predicted logit (i.e., log probability) for the i^{th} class, κ is a hyper-parameter that controls the certainty level of the attack, λ is a regularization coefficient (we set $\lambda = 1$ in our experiments), $\mathcal{T} := \arg\min_i \mathcal{C}(\boldsymbol{a})_i$, and $\|\boldsymbol{x}\|$ denotes the L_2 norm of $\boldsymbol{x}$ (i.e., the square root of the sum of squares of entries in $\boldsymbol{x}$). This loss function resembles the popular one designed in (Carlini & Wagner, 2017).

With EPGS and other compared algorithms, we perform adversarial attacks on 100 randomly selected images from each of two image datasets, the set of MNIST handwritten digits (LeCun et al., 1998) and the CIFAR-10 set (Krizhevsky & Hinton, 2009). Specifically, the goal is to solve:

$$\max_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x}) := -L(\boldsymbol{x}),$$

where d is the total number of pixels in each image. For Figure-MNIST images, $d = 28 \times 28$, and for CIFAR-10 images, $d = 32 \times 32 \times 3$. We call f as the fitness function.

The classifier is a robust convolutional neural network (CNN) trained using the technique of defensive distillation[9]. The distillation temperature is set at 100, which leads to a high level of robustness (Carlini & Wagner, 2017). In the MNIST attacks, our trained classifier $\mathcal{C}$ has a classification accuracy of 97.4% on the testing images. In the CIFAR-10 attacks, the trained $\mathcal{C}$ has a test accuracy of 86.2%.

The hyper-parameters of all the tested algorithms are selected by trials and can be found in Appendix F. For ZO-SLGHd, ZO-SLGHr, and ZO-AdaMM, the hyper-parameters reported in Appendix E.1 of (Iwakiri et al., 2022) are included in our candidate set, since they performed similar tasks.

We choose EPGS over PGS for this task since the fitness function $f(\boldsymbol{x})$ can be negative, which makes EPGS more convenient to apply. But note that we can modify $f(\boldsymbol{x})$ by adding a large positive constant to facilitate PGS.

Table 3: Targeted Adversarial Attack on 100 MNIST images (per-image). For each image attack, we set the initial perturbation $\boldsymbol{\mu}_0 = \boldsymbol{0}$, and $T_{total} = 1,500$. The success rate (SR) is the portion of successful attacks out of the 100 attacks. $\bar{R}^2$, $\overline{\|\boldsymbol{\mu}^*\|}$, and $\bar{T}$ are the averages defined below in the main text. The numbers in the parentheses are sample standard deviations. STD-Htp is STD-Homotopy for short. For EPGS, N is selected to be 0.05. See Table 10 in the Appendix for the produced adversarial images for four randomly selected images.

Algorithm	SR	$\bar{R}^2$	$\overline{\|\boldsymbol{\mu}^*\|}$	$\bar{T}$
CMA-ES	100%	89%(4%)	2.81(0.61)	1489(12)
EPGS	100%	87%(5%)	3.01(0.60)	397(101)
ZOSGD	100%	85%(5%)	3.14(0.61)	1427(242)
ZOSLGHd	100%	74%(9%)	4.21(0.71)	1490(24)
ZOSLGHr	100%	65%(13%)	4.86(0.81)	476(658)
ZOAdaMM	100%	29%(27%)	6.88(1.15)	45(15)
STD-Htp	97%	−4%(37%)	8.25(1.09)	530(264)

Experiments on MNIST. For each image $\boldsymbol{a}_m$ that is randomly drawn from the dataset, where $m \in \{1, 2, ..., 100\}$, and each algorithm, we perform an attack (i.e., experiment) of T_{total} iterations. Let $\{\boldsymbol{\mu}_{m,t}\}_{t=0}^{T_{total}-1}$ denote all the perturbations (solutions) produced in these T_{total} iterations. We say that a perturbation $\boldsymbol{\mu}$ is successful if the predicted log probability of the target label is at least $\kappa = 0.001$ greater than that of other classes (i.e., $\mathcal{C}(\boldsymbol{a}+\boldsymbol{\mu})_{\mathcal{T}} - \max_{i \neq \mathcal{T}} \mathcal{C}(\boldsymbol{a}+\boldsymbol{\mu})_i > \kappa$). We say that an attack is successful if the produced $\{\boldsymbol{\mu}_{m,t}\}_{t=0}^{T_{total}-1}$ contain at least one successful perturbation. If the attack is successful, let $\boldsymbol{\mu}_m^*$ denote the successful perturbation with the largest R^2-value among $\{\boldsymbol{\mu}_{m,t}\}_{t=0}^{T_{total}-1}$, and let T_m denote the number of iterations taken by the algorithm to produce $\boldsymbol{\mu}_m^*$. We use $R^2(\boldsymbol{a}, \boldsymbol{a}+\boldsymbol{\mu}) := 1 - \frac{\sum_{i=1}^d \mu_i^2}{\sum_{i=1}^d (a_i - \bar{a})^2}$ to measure the similarity between $\boldsymbol{a}$ and the perturbed image $\boldsymbol{a}+\boldsymbol{\mu}$, where a_i and μ_i denote the i^{th} pixel (entry) of $\boldsymbol{a}$ and $\boldsymbol{\mu}$, respectively, and $\bar{a}$ denotes the mean of $\{a_i\}$.

With the above notations, we construct four measures on the performances of an algorithm. One is the success rate, which refers to the ratio of successful image attacks out of the total number of attacks (100). The second measure is the average R^2, which equals $\bar{R}^2 := \sum_{m \in \mathbb{S}} R^2(\boldsymbol{a}_m, \boldsymbol{a}_m + \boldsymbol{\mu}_m^*)/|\mathbb{S}|$, where $\mathbb{S}$ denotes the set of indices of the successful attacks and $\boldsymbol{\mu}_m^*$ denotes the optimal perturbation for the m^{th} figure. The third one is the average $\overline{\|\boldsymbol{\mu}^*\|}$ of $\{\|\boldsymbol{\mu}_m^*\|\}_{m \in \mathbb{S}}$, where $\|\cdot\|$ denotes the L_2-norm. The last measure is the average $\bar{T}$ of $\{T_m\}_{m \in \mathbb{S}}$.

The results are reported in Table 3, from which we see that EPGS has a high $\bar{R}^2$-score of 87% (ranks among the top 2), indicating that the perturbed image is closest to the original one ($R^2 = 100\%$ implies that $\boldsymbol{a}$ and $\boldsymbol{a}+\boldsymbol{\mu}$ are identical). Also, the average number of iterations taken by EPGS to reach the optimal perturbation is 397, which is among the two fastest algorithms.

Experiments on CIFAR-10. With 100 randomly drawn

[9] We borrow the code by (Carlini & Wagner, 2017) for training the classifier from `https://github.com/carlini/nn_robust_attacks`, which applies TensorFlow (Abadi et al., 2015) for model training. We changed the layer structure in the neural network to either increase accuracy or to decrease computation complexity while maintaining the accuracy.

Table 4: Targeted Adversarial Attack on 100 CIFAR-10 images (per-image). For each image attack, we set $\boldsymbol{\mu}_0 = \mathbf{0}$ and $T_{total} = 1,500$. For EPGS, N is selected as 0.03. See Table 11 in the Appendix for the produced adversarial images for four randomly selected images.

Algorithm	SR	$\bar{R}^2$	$\overline{\|\boldsymbol{\mu}^*\|}$	$\bar{T}$
ZOSLGHd	98%	99%(1%)	1.72(0.32)	1290(411)
ZOSLGHr	98%	98%(3%)	2.66(0.68)	456(345)
EPGS	98%	98%(2%)	3.05(0.57)	748(248)
CMA-ES	99%	75%(25%)	10.06(2.35)	158(399)
ZOAdaMM	100%	58%(39%)	13.13(2.71)	58(31)
ZOSGD	62%	99%(1%)	1.19(0.19)	764(349)
STD-Htp	52%	87%(13%)	7.54(1.57)	566(396)

images from the CIFAR-10 test set, we repeat the per-image targeted adversarial attacks. Their results are reported in Table 4. These results are in general better than those in the MNIST attacks, which we believe is because of the lower accuracy of the trained CIFAR-10 CNN (86.0% vs 97.8%).

The results show that EPGS produces a success rate of 98% and scores in the top three with respect to $\bar{R}^2$. Specifically, the $\bar{R}^2$-value of 98% indicates that the perturbed image produced by EPGS is very close to the original image, which is consistent with our goal that the perturbation size $\|\boldsymbol{\mu}^*\|$ should be small.

5.4. Summary on Experiment Results

Comparing to algorithms that use the smoothing technique (which **excludes CMA-ES**), PGS and EPGS ranked among the tops in all the tasks we performed. Specifically, in the tasks of Ackley, Rosenbrock, and MNIST-attack (Table 1, 2 and 3), EPGS has the highest fitness value $f(\boldsymbol{\mu}^*)$ than other algorithms except CMA-ES. For CIFAR-10 attacks (Table 4), EPGS's $\bar{R}^2$ score equals that of the best (ZOSLGHd). In sum, EPGS outperforms other algorithms that also apply smoothing techniques in the majority of the performed experiments.

While EPGS underperforms CMA-ES in experiments of Ackley, Rosenbrock, and MNIST, it beats CMA-ES in the CIFAR-10 task with a comparable success rate and a significantly smaller perturbation norm (see Table 4). Moreover, to the best of our knowledge, the theoretical convergence guarantee for CMA-ES on general non-convex objectives is less developed, providing a competitive edge to GS-PowerOpt.

5.5. The Case of Multiple Global Maxima

Although the convergence theory (e.g., Theorem 2.1) requires the condition that the objective f has a unique global maximum, it is a sufficient rather than a necessary condition for GS-PowerOpt to work. To illustrate this point, we perform an additional experiment with EPGS on an objective with two global maxima: $f(\boldsymbol{x}) = -\log(\|\boldsymbol{x} - \boldsymbol{m}_1\|^2 + 10^{-5}) - \log(\|\boldsymbol{x} - \boldsymbol{m}_2\|^2 + 10^{-5})$, where $\boldsymbol{m}_1 = [-.5, -.5]$ and $\boldsymbol{m}_2 = [.5, .5]$. A hundred trials are performed, where $N = 1$ and each $\boldsymbol{\mu}_0$ is randomly sampled around the origin. The mean square error between the found solution and one of the two global maximum point is close to 0, with its average (over the 100 trials) equal to 1.4×10^{-5} and its sample standard deviation equal to 1.6×10^{-5}. This example indicates that GS-PowerOpt is capable of locating at least one of the multiple global optima. We leave the corresponding theoretical analysis in our future works.

6. Guidance on Choosing Hyper-parameters

Although a larger N guarantees a convergence of GS-PowerOpt to a smaller neighborhood of $\boldsymbol{x}^*$ in theory (see Subsection 3.1), it also leads to a higher variance of of the update term $\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu}_t)$ in Eq. (5), which in turn decreases the efficiency of the algorithm in practice. Therefore, we recommend to start from a moderate N and incrementally increase its value during tuning. Although the proper starting value of N may vary for different problems, based on our experience, 5 for PGS and 0.1 for EPGS are good choices.

While a large σ increases the exploration range for each $\boldsymbol{\mu}$ update, it also increases the sample complexity of the algorithm. Our experiments show that a σ-value of 10% of the search radius for $\boldsymbol{x}$ is a good starting value for tuning.

7. Conclusion and Future Work

The convergence analysis and numerical results show that the easily implemented optimization method of GS-PowerOpt stands out among its peers that also apply smoothing techniques. Our work provides a foundation for future studies to explore more efficient ways to increase the gap between $f(\boldsymbol{x}^*)$ and f values at other points before smoothing.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions for improving this paper.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., H., A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2015. URL https://www.tensorflow.org/.

Arridge, S., Maass, P., Öktem, O., and Schönlieb, C.-B. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

Blake, A. and Zisserman, A. *Visual Reconstruction*. MIT press, 1987.

Boer, P. D., Kroese, D. P., Mannor, S., and Rubinstein, R. Y. A tutorial on the cross-entropy method. *Ann. Oper. Res.*, 134(1):19–67, 2005. doi: 10.1007/s10479-005-5724-z.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017. doi: 10.1109/SP.2017.49.

Chen, J., Guo, Z., Li, H., and Chen, C. P. Regularizing scale-adaptive central moment sharpness for neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(5), 2024.

Chen, X., Liu, S., Xu, K., Li, X., Lin, X., Hong, M., and Cox, D. Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization. *Advances in neural information processing systems*, 32, 2019.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pp. 192–204. PMLR, 2015.

Dvijotham, K., Fazel, M., and Todorov, E. Universal convexification via risk-aversion. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, 2014.

Gao, K. and Sener, O. Generalizing Gaussian smoothing for random search. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 7077–7101. PMLR, 2022.

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

Hansen, N. and Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

Hansen, N., Akimoto, Y., and Baudis, P. CMA-ES/pycma on Github, 2019. URL https://doi.org/10.5281/zenodo.2559634.

Hazan, E., Levy, K. Y., and Shalev-Shwartz, S. On graduated optimization for stochastic non-convex problems. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 1833–1841, 2016.

Iwakiri, H., Wang, Y., Ito, S., and Takeda, A. Single loop gaussian homotopy method for non-convex optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 7065–7076, 2022.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lei, Y., Hu, T., Li, G., and Tang, K. Stochastic gradient descent for nonconvex learning without bounded gradient assumptions. *IEEE transactions on neural networks and learning systems*, 31(10):4394–4400, 2019.

Lin, X., Yang, Z., Zhang, X., and Zhang, Q. Continuation path learning for homotopy optimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 21288–21311. PMLR, 2023.

Locatelli, M. and Schoen, F. *Global Optimization: Theory, Algorithms, and Applications*. SIAM, Philadelphia, PA, 2013. doi: 10.1137/1.9781611972672.

Magnus, J. R. and Neudecker, H. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons Ltd, 2019.

Mertikopoulos, P., Hallak, N., Kavis, A., and Cevher, V. On the almost sure convergence of stochastic gradient descent in non-convex problems. *Advances in Neural Information Processing Systems*, 33:1117–1128, 2020.

Miranda, L. J. V. PySwarms, a research-toolkit for particle swarm optimization in python. *Journal of Open Source Software*, 3, 2018. doi: 10.21105/joss.00433.

Mobahi, H. and Fisher, J. W. A theoretical analysis of optimization by gaussian continuation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, pp. 1205–1211, 2015. doi: 10.1609/aaai.v29i1.9356.

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

Roulet, V., Fazel, M., Srinivasa, S., and Harchaoui, Z. On the convergence of the iterative linear exponential quadratic gaussian algorithm to stationary points. In *2020 American Control Conference (ACC)*, pp. 132–137. IEEE, 2020.

Starnes, A. and Webster, C. Improved performance of stochastic gradients with gaussian smoothing, 2024. URL https://arxiv.org/abs/2311.00531.

Van Laarhoven, P. J., Aarts, E. H., van Laarhoven, P. J., and Aarts, E. H. *Simulated annealing*. Springer, 1987.

Xiao, L. and Zhang, T. A proximal-gradient homotopy method for the l1-regularized least-squares problem. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 839–846, 2012.

Xu, Y., Yan, Y., Lin, Q., and Yang, T. Homotopy smoothing for non-smooth problems with lower complexity than $o(1/\epsilon)$. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

A. Table of Notations

Notation	Description	Reference
$\mathcal{S}$	A compact set in $\mathbb{R}^d$.	Theorem 2.1
f	The real-valued objective function, which is further assumed to be non-negative for PGS.	Eq. (1) or Theorem 2.1
$\boldsymbol{x}^*$	The global maximum point of f.	
x_i^*	The i^{th} entry of $\boldsymbol{x}^*$, where $i \in \{1, 2, ..., d\}$.	(7)
μ_i	The i^{th} entry of $\boldsymbol{\mu} \in \mathbb{R}^d$, where $i \in \{1, 2, ..., d\}$.	(7)
$\mathcal{N}(\boldsymbol{\mu}, \sigma I_d)$	A multivariate standard Gaussian distribution, where $\boldsymbol{\mu} \in \mathbb{R}^d$, $\sigma > 0$, and I_d denotes the $d \times d$ identity matrix.	Eq. (2)
$\hat{f}_\sigma(\boldsymbol{\mu})$	$\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I_d)}[f(\boldsymbol{x})]$	(2)
N	$N > 0$ and $N \in \mathbb{R}$. In Section 3, it is assumed to be greater than a threshold (see Section 3.2 for details).	
$f_N(\boldsymbol{x})$	The (exponential) power of f, extended to $\mathbb{R}^d$.	Eq. (4)
$F_{N,\sigma}(\boldsymbol{\mu})$	$F_{N,\sigma}(\boldsymbol{\mu}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I_d)}[f_N(\boldsymbol{x})]$.	Theorem 2.1
$F(\boldsymbol{\mu})$	The abbreviation of $F_{N,\sigma}(\boldsymbol{\mu})$, where N and σ are fixed.	Section 3.2.
$\nabla F(\boldsymbol{\mu})$, $\nabla F_N(\boldsymbol{\mu}, \sigma)$	The gradient of $F_{N,\sigma}(\boldsymbol{\mu})$ with respect to $\boldsymbol{\mu}$.	Eq. (6).
$\hat{F}_{N,\sigma}(\boldsymbol{\mu})$	A sample estimate of $\sigma^2 \nabla F_{N,\sigma}(\boldsymbol{\mu})$.	Eq. (5)
$\mathcal{S}_{\boldsymbol{x}^*,\delta}$	A δ-neighborhood of $\boldsymbol{x}^*$ in $\mathbb{R}^d$, where $\delta > 0$.	Eq. (7)
L	The Lipschitz coefficient of $\nabla F_{N,\sigma}(\boldsymbol{\mu})$.	Lemma 3.5.
G	The upper bound of $\mathbb{E}[\|\hat{\nabla} F(\boldsymbol{\mu})\|^2]$.	Lemma 3.6.
$\|\boldsymbol{x}\|$	$\|\boldsymbol{x}\| = \sqrt{\sum_{i=1}^d x_i^2}$, where $\boldsymbol{x} = [x_1, x_2, ..., x_d] \in \mathbb{R}^d$.	

B. PGS/EGPS with A Baseline to Avoid Computation Overflows

An effective method to avoid computation overflows caused by a large N-value in Algorithm 1 is to modify the gradient estimate as $\hat{\nabla}_{\boldsymbol{\mu}} F(\boldsymbol{\mu}_t) = \frac{1}{K}\sum_{k=1}^K (\boldsymbol{x}_k - \boldsymbol{\mu}_t) f_N^{(b)}(\boldsymbol{x}_k)$, where $f_N^{(b)}(\boldsymbol{x}_k) = 0$ if $\boldsymbol{x}_k \notin \mathcal{S}$, and

$$f_N^{(b)}(\boldsymbol{x}_k) = \begin{cases} f^N(\boldsymbol{x}_k)/f^N(\boldsymbol{\mu}_t), & \text{for PGS;} \\ e^{N(f(\boldsymbol{x}_k) - f(\boldsymbol{\mu}_t))}, & \text{for EPGS,} \end{cases}$$

if $\boldsymbol{x}_k \in \mathcal{S}$. This modification produces the same $\boldsymbol{\mu}$-updates as in Algorithm 1 because of the gradient-normalization step. We call this algorithm as *PGS (EPGS) with a baseline*.

C. Proofs to Theoretical Results

C.1. Proof to Theorem 2.1 for EPGS

Theorem 2.1 *Let $f : \mathcal{S} \subset \mathbb{R}^d \to \mathbb{R}$ be a continuous function that is possibly non-concave (and non-negative only for the case of PGS), where $\mathcal{S}$ is compact. Assume that f has a global maximum $\boldsymbol{x}^*$ such that $\sup_{\boldsymbol{x}:\|\boldsymbol{x}-\boldsymbol{x}^*\|\geq\delta} f(\boldsymbol{x}) < f(\boldsymbol{x}^*)$ for any $\delta > 0$. For $\sigma > 0$ and any $N > 0$, define*

$$F_{N,\sigma}(\boldsymbol{\mu}) := (\sqrt{2\pi}\sigma)^{-d} \int_{\boldsymbol{x}\in\mathbb{R}^k} f_N(\boldsymbol{x}) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x}, \quad \boldsymbol{\mu} \in \mathbb{R}^d,$$

where f_N is defined in (4) for either PGS or EPGS. Then, for any $M > 0$ and $\delta > 0$ such that $\bar{B}(\boldsymbol{x}^*; \delta) := \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \delta\} \subset \mathcal{S}$, there exists $N_{\delta,\sigma,M} > 0$, such that whenever $N > N_{\delta,\sigma,M}$, for any $\|\boldsymbol{\mu}\| < M$ and any $i \in \{1, 2, ..., d\}$ we have that: $\frac{\partial F_N(\boldsymbol{\mu},\sigma)}{\partial \mu_i} > 0$ if $\mu_i < x_i^* - \delta$, and $\frac{\partial F_N(\boldsymbol{\mu},\sigma)}{\partial \mu_i} < 0$ if $\mu_i > x_i^* + \delta$. Here, μ_i and x_i^* denote the i^{th} entry of $\boldsymbol{\mu}$ and $\boldsymbol{x}^*$, respectively.

Here, we provides the proof for the EPGS setting, and the proof for the PGS setting is similar.

Proof. Recall that for EPGS, $f_N(\boldsymbol{x}_k) := \begin{cases} e^{Nf(\boldsymbol{x}_k)}, & \boldsymbol{x} \in \mathcal{S}; \\ 0, & \text{otherwise.} \end{cases}$ For any given $\delta > 0$, define $V_\delta := \sup_{\boldsymbol{x}:\|\boldsymbol{x}-\boldsymbol{x}^*\| \geq \delta} f(\boldsymbol{x})$ and $D_\delta := (V_\delta + f(\boldsymbol{x}^*))/2$. Using this symbol, we re-write $F_{N,\sigma}(\boldsymbol{\mu})$ as

$$F_{N,\sigma}(\boldsymbol{\mu}) = e^{D_\delta N} G_{N,\sigma}(\boldsymbol{\mu}) = e^{D_\delta N}(H_{N,\sigma}(\boldsymbol{\mu}) + R_{N,\sigma}(\boldsymbol{\mu})), \tag{9}$$

where

$$\begin{aligned} G_{N,\sigma}(\boldsymbol{\mu}) :=& (\sqrt{2\pi}\sigma)^{-d} \int_{\boldsymbol{x}\in\mathbb{R}^d} e^{-ND_\delta} f_N(\boldsymbol{x}) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x}, \\ H_{N,\sigma}(\boldsymbol{\mu}) :=& (\sqrt{2\pi}\sigma)^{-d} \int_{\boldsymbol{x}\in B(\boldsymbol{x}^*;\delta)} e^{-ND_\delta} f_N(\boldsymbol{x}) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x}, \\ R_{N,\sigma}(\boldsymbol{\mu}) :=& (\sqrt{2\pi}\sigma)^{-d} \int_{\boldsymbol{x}\notin B(\boldsymbol{x}^*;\delta)} e^{-ND_\delta} f_N(\boldsymbol{x}) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x}, \end{aligned}$$

where $B(\boldsymbol{x}^*;\delta) := \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}-\boldsymbol{x}^*\| < \delta\}$. The main idea of the proof is to first show that $\left|\frac{\partial H_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right|$ dominates $\left|\frac{\partial R_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right|$ if N is sufficiently large, and then that the sign of $\frac{\partial H_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}$ satisfies the declared property in the theorem.

We derive an upper bound for $\left|\frac{\partial R_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right|$. For any $\boldsymbol{\mu} \in \mathbb{R}^d$,

$$\begin{aligned} \left|\frac{\partial R_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right| &\leq \frac{1}{(\sqrt{2\pi})^d \sigma^{d+2}} \int_{\boldsymbol{x}\notin B(\boldsymbol{x}^*;\delta)} |x_i - \mu_i| e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} e^{-ND_\delta} f_N(\boldsymbol{x}) d\boldsymbol{x} \\ &\leq \frac{1}{(\sqrt{2\pi})^d \sigma^{d+2}} \int_{\boldsymbol{x}\notin B(\boldsymbol{x}^*;\delta)} |x_i - \mu_i| e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} e^{N(f(\boldsymbol{x})-D_\delta)} d\boldsymbol{x} \\ &\leq \frac{1}{(\sqrt{2\pi})^d \sigma^{d+2}} \int_{\boldsymbol{x}\notin B(\boldsymbol{x}^*;\delta)} |x_i - \mu_i| e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} e^{N(V_\delta-D_\delta)} d\boldsymbol{x} \\ &\leq \frac{e^{N(V_\delta-D_\delta)}}{(\sqrt{2\pi})^d \sigma^{d+2}} \int_{\boldsymbol{x}\in\mathbb{R}^d} |x_i - \mu_i| e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x} \\ &\leq e^{N(V_\delta-D_\delta)} \left(\Pi_{j\neq i} \frac{1}{\sqrt{2\pi}\sigma} \int_{x_j\in\mathbb{R}} e^{-\frac{(x_j-\mu_j)^2}{2\sigma^2}} dx_j \right) \\ &\qquad \cdot \frac{1}{\sqrt{2\pi}\sigma^3} \int_{x_i\in\mathbb{R}} |x_i - \mu_i| e^{-\frac{(x_i-\mu_i)^2}{2\sigma^2}} dx_i \\ &= e^{N(V_\delta-D_\delta)} \frac{1}{\sqrt{2\pi}\sigma^3} \int_{y\in\mathbb{R}} \sqrt{2}\sigma |y| e^{-y^2} d(\sqrt{2}\sigma y), \quad y := \frac{x_i - \mu_i}{\sqrt{2}\sigma}, \\ &= e^{N(V_\delta-D_\delta)} \frac{\sqrt{2}}{\sqrt{\pi}\sigma} \cdot 2 \int_0^\infty y e^{-y^2} dy, \\ &= e^{N(V_\delta-D_\delta)} \frac{\sqrt{2}}{\sqrt{\pi}\sigma} \cdot \int_0^\infty e^{-y^2} dy^2, \\ &= e^{N(V_\delta-D_\delta)} \frac{\sqrt{2}}{\sqrt{\pi}\sigma} \cdot \int_0^\infty e^{-z} dz, \\ &= \frac{\sqrt{2} e^{N(V_\delta-D_\delta)}}{\sqrt{\pi}\sigma} \end{aligned} \tag{10}$$

where the third inequality sign is because $\|\boldsymbol{x}-\boldsymbol{x}^*\| \geq \delta \Rightarrow f(\boldsymbol{x}) \leq V_\delta$, and the fifth inequality sign is from the separability of a multivariate integral.

Since f is continuous, for $\epsilon_\delta := f(\boldsymbol{x}^*) - D_\delta > 0$ (because $V_\delta < f(\boldsymbol{x}^*)$), there exists $\delta' \in (0, \delta)$ such that whenever $\|\boldsymbol{x}-\boldsymbol{x}^*\| \leq \delta'$,

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*) - \epsilon_\delta = D_\delta > V_\delta. \tag{11}$$

Using this result, we derive a lower bound for $\left|\frac{\partial H_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right|$ when $\|\boldsymbol{\mu}\| \leq M$ and $|\mu_i - x_i^*| > \delta$.

$$\begin{aligned}
\left|\frac{\partial H_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right| &= \frac{1}{(\sqrt{2\pi})^d \sigma^{d+2}} \int_{\boldsymbol{x}\in B(\boldsymbol{x}^*;\delta)} |x_i - \mu_i| e^{-ND_\delta} f_N(\boldsymbol{x}) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x} \\
&= \frac{1}{(\sqrt{2\pi})^d \sigma^{d+2}} \int_{\boldsymbol{x}\in B(\boldsymbol{x}^*;\delta)} |x_i - \mu_i| e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} e^{N(f(\boldsymbol{x})-D_\delta)} d\boldsymbol{x}, \text{ since } B(\boldsymbol{x}^*;\delta) \subset \mathcal{S}, \\
&\geq \frac{1}{(\sqrt{2\pi})^d \sigma^{d+2}} \int_{\boldsymbol{x}\in B(\boldsymbol{x}^*;\delta')} |x_i - \mu_i| e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} e^{N(f(\boldsymbol{x})-D_\delta)} d\boldsymbol{x} \\
&\geq \frac{1}{(\sqrt{2\pi})^d \sigma^{d+2}} \int_{\boldsymbol{x}\in B(\boldsymbol{x}^*;\delta')} (\delta - \delta') e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x} \\
&\geq \frac{1}{(\sqrt{2\pi})^d \sigma^{d+2}} \int_{\boldsymbol{x}\in B(\boldsymbol{x}^*;\delta')} (\delta - \delta') e^{-\frac{M^2}{\sigma^2}} e^{-\frac{\|\boldsymbol{x}\|^2}{\sigma^2}} d\boldsymbol{x}, \quad \|\boldsymbol{x}-\boldsymbol{\mu}\|^2 \leq 2(\|\boldsymbol{x}\|^2 + \|\boldsymbol{\mu}\|^2), \\
&\geq (\delta - \delta') e^{-\frac{M^2}{\sigma^2}} V(\delta', d, \sigma)
\end{aligned} \tag{12}$$

where the fourth line is implied by

- $e^{N(f(\boldsymbol{x})-D_\delta)} \geq 1$ because of (11);
- $|x_i - \mu_i| = |(x_i - x_i^*) + (x_i^* - \mu_i)| \geq |x_i^* - \mu_i| - |x_i - x_i^*| > \delta - \delta'$.

In the last line of (12),

$$V(\delta', d, \sigma) := \frac{1}{(\sqrt{2\pi})^d \sigma^{d+2}} \int_{\boldsymbol{x}\in B(\boldsymbol{x}^*;\delta')} e^{-\frac{\|\boldsymbol{x}\|^2}{\sigma^2}} d\boldsymbol{x}.$$

The positive number $N_{\delta,\sigma,M}$ is constructed by solving the following inequality for N, which involves the two bounds in (10) and (12).

$$\frac{\sqrt{2} e^{N(V_\delta - D_\delta)}}{\sqrt{\pi}\sigma} < (\delta - \delta') e^{-\frac{M^2}{\sigma^2}} V(\delta', d, \sigma).$$

The solution of this inequality is

$$N > \frac{\ln\left(\frac{\sqrt{\pi}\sigma}{\sqrt{2}}(\delta - \delta') e^{-\frac{M^2}{\sigma^2}} V(\delta', d, \sigma)\right)}{V_\delta - D_\delta},$$

where $V_\delta - D_\delta < 0$ and the numerator is negative for sufficiently large $M > 0$. Therefore, whenever $\|\boldsymbol{\mu}\| \leq M$, $|\mu_i - x_i^*| > \delta$, and

$$N > N_{\delta,\sigma,M} := \max\left\{0, \frac{\ln\left(\frac{\sqrt{\pi}\sigma}{\sqrt{2}}(\delta - \delta') e^{-\frac{M^2}{\sigma^2}} V(\delta', d, \sigma)\right)}{V_\delta - D_\delta}\right\},$$

we have

$$\left|\frac{\partial R_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right| \leq \frac{\sqrt{2} e^{N(V_\delta - D_\delta)}}{\sqrt{\pi}\sigma} < (\delta - \delta') e^{-\frac{M^2}{\sigma^2}} V(\delta', d, \sigma) \leq \left|\frac{\partial H_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right|. \tag{13}$$

When $N > N_{\delta,\sigma,M}$, $\|\boldsymbol{\mu}\| \leq M$, and $\mu_i > x_i^* + \delta$,

$$\begin{aligned}
\frac{\partial G_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i} &= \frac{\partial H_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i} + \frac{\partial R_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i} \\
&= \frac{1}{(\sqrt{2\pi})^k \sigma^{k+2}} \int_{\boldsymbol{x}\in B(\boldsymbol{x}^*;\delta)} (x_i - \mu_i) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} e^{N(f(\boldsymbol{x})-D_\delta)} d\boldsymbol{x} + \frac{\partial R_N(\boldsymbol{\mu},\sigma)}{\partial \mu_i} \\
&= -\left|\frac{\partial H_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right| + \frac{\partial R_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i} \\
&<^{\text{by (13)}} -\left|\frac{\partial R_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right| + \left|\frac{\partial R_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right| \\
&= 0,
\end{aligned} \tag{14}$$

where the third line is because the integrand of the first term is always negative in the integration region.

On the other hand, when $N > N_{\delta,\sigma,M}$, $\|\boldsymbol{\mu}\| \le M$, and $\mu_i < x_i^* - \delta$,

$$\begin{aligned}\frac{\partial G_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i} =& \frac{\partial H_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i} + \frac{\partial R_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i} \\ &= \left|\frac{\partial H_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right| + \frac{\partial R_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i} \\ &>^{\text{by (13)}} \left|\frac{\partial R_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right| - \left|\frac{\partial R_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}\right| \\ &= 0.\end{aligned} \tag{15}$$

Then, (14) and (15) imply the result in the theorem since $\frac{\partial G_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}$ and $\frac{\partial F_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i}$ share the same sign (see Eq. (9)). □

C.2. Proof to Proposition 2.3

Proposition 2.3 *Given $\sigma > 0$ and $N > 0$, $F_{N,\sigma}(\boldsymbol{\mu})$ in Theorem 2.1 has a global max point $\boldsymbol{\mu}^*$.*

Proof. Since $\sigma > 0$ and $N > 0$ are fixed, we denote $F_{N,\sigma}(\boldsymbol{\mu})$ by $F(\boldsymbol{\mu})$ for convenience. Recall that $F(\boldsymbol{\mu}) \ge 0$ because of the definition of f_N in (4). Our proof applies the following two results, whose proofs will be given later.

1. $F(\boldsymbol{\mu})$ is Lipschitz, which indicates the continuity.
2. $\lim_{\|\boldsymbol{\mu}\|\to\infty} F(\boldsymbol{\mu}) = 0$.

From F's definition, it is trivial to see that there exists some $\boldsymbol{\mu}' \in \mathbb{R}^d$ such that $F(\boldsymbol{\mu}') > 0$. Define $\epsilon' := F(\boldsymbol{\mu}')$. From point 2, there exists $M > 0$ such that, whenever $\|\boldsymbol{\mu}\| > M$, $F(\boldsymbol{\mu}) < \epsilon'$.

Now, consider the closed ball $B(\mathbf{0}; M') := \{\boldsymbol{\mu} \in \mathbb{R}^d : \boldsymbol{\mu} \le M'\}$, where $M' > M$ and $\boldsymbol{\mu}' \in B(\mathbf{0}; M')$. From the extreme value theorem, the continuity of F and the compactness of $B(\mathbf{0}; M')$ indicate that F attains a maximum $\boldsymbol{\mu}^*$ on $B(\mathbf{0}; M')$. That is, $\boldsymbol{\mu}^* \in \mathbb{R}^d$, and $F(\boldsymbol{\mu}^*) \ge F(\boldsymbol{\mu})$ for any $\boldsymbol{\mu} \in B(\mathbf{0}; M')$. Therefore,

$$F(\boldsymbol{\mu}^*) \ge F(\boldsymbol{\mu}') = \epsilon' > F(\boldsymbol{\mu}), \text{ for any } \boldsymbol{\mu} \notin B(\mathbf{0}; M').$$

This indicates that $\boldsymbol{\mu}^*$ is the global maximum point of F in $\mathbb{R}^d$.

Finally, it remains to prove Point 1 and 2.

Proof to Point 1. From Lemma 3.3, the gradient ∇F exists. Hence, from the mean value theorem (e.g., Theorem 5.10 in (Magnus & Neudecker, 2019)), for any $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^d$, there exists $\boldsymbol{v}$ that lies on the line segment connecting $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ that

$$\begin{aligned}|F(\boldsymbol{\mu}_1) - F(\boldsymbol{\mu}_2)| &= |\nabla F(\boldsymbol{v})(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)| \\ &\le \|\nabla F(\boldsymbol{v})\|\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|, \text{ Cauchy Schwarz Inequality,} \\ &= \sigma^{-2}\|\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{v},\sigma^2 I_d)}[(\boldsymbol{x}-\boldsymbol{\mu})f_N(\boldsymbol{x})]\|\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|, \text{ proof same as (6),} \\ &\le \frac{\sqrt{2d}}{\sigma\sqrt{\pi}} f_N(\boldsymbol{x}^*)\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|,\end{aligned}$$

where the last inequality is derived using the last inequality derived in the proof for Lemma 3.3. This proves Point 1.

Proof to Point 2. Since $\mathcal{S} \in \mathbb{R}^d$ is compact, it is bounded. Assume that $\mathcal{S} \subset B(\mathbf{0}; M)$ for some $M > 0$. Then, $f_N(\boldsymbol{x}) = 0$ if $\|\boldsymbol{x}\| > M$. Hence,

$$\begin{aligned}F(\boldsymbol{\mu}) &= \frac{1}{(\sqrt{2\pi}\sigma)^d}\int_{\|\boldsymbol{x}\|\le M} f_N(\boldsymbol{x}) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x} \\ &\le \frac{f_N(\boldsymbol{x}^*)}{(\sqrt{2\pi}\sigma)^d}\int_{\|\boldsymbol{x}\|\le M} e^{-\frac{(\|\boldsymbol{\mu}\|-M)^2}{2\sigma^2}} d\boldsymbol{x}, \text{ whenever } \|\boldsymbol{\mu}\| > M, \\ &\to 0, \quad \text{as } \|\boldsymbol{\mu}\| \to \infty,\end{aligned}$$

where the second line is because $\|\boldsymbol{x}-\boldsymbol{\mu}\| \geq \|\boldsymbol{\mu}\| - \|\boldsymbol{x}\| \geq \|\boldsymbol{\mu}\| - M$, which further implies $-\|\boldsymbol{x}-\boldsymbol{\mu}\|^2 \leq -(\|\boldsymbol{\mu}\|-M)^2$ if $\|\boldsymbol{\mu}\| > M$. □

C.3. Proof to Lemma 3.3

Lemma 3.3 *Under Assumption 3.1, $F_{N,\sigma}(\boldsymbol{\mu})$ and $\nabla F_{N,\sigma}(\boldsymbol{\mu})$ are well-defined for all $\boldsymbol{\mu} \in \mathbb{R}^d$.*

Proof. By the definition of $F_{N,\sigma}(\boldsymbol{\mu})$ in Theorem 2.1, $F_{N,\sigma}(\boldsymbol{\mu}) = \mathbb{E}[f_N(\boldsymbol{x})]$ and $\nabla F_{N,\sigma}(\boldsymbol{\mu}) = \mathbb{E}[(\boldsymbol{x}-\boldsymbol{\mu})f_N(\boldsymbol{x})]$, where the expectation is taken with respect to $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I_d)$. To prove that $F_{N,\sigma}(\boldsymbol{\mu})$ and $\nabla F_{N,\sigma}(\boldsymbol{\mu})$ are well-defined, it sufficies to prove that $\mathbb{E}|f_N(\boldsymbol{x})| < +\infty$ and $\mathbb{E}|(x_i-\mu_i)f_N(\boldsymbol{x})| < +\infty$, for each $i \in \{1,2,...,d\}$. The former inequality holds since $|f_N(\boldsymbol{x})|$ is bounded by Assumption 3.1 (the condition that f is continuous on its compact domain implies the boundedness of $|f_N(\boldsymbol{x})|$). The latter inequality holds because

$$\begin{aligned}
&\mathbb{E}|(x_i-\mu_i)f_N(\boldsymbol{x})| \\
&\leq f_N(\boldsymbol{x}^*)\mathbb{E}|x_i-\mu_i| \\
&= \frac{f_N(\boldsymbol{x}^*)}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}|x_i-\mu_i|e^{-\frac{(x_i-\mu_i)^2}{2\sigma^2}}dx_i \\
&= \frac{f_N(\boldsymbol{x}^*)}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}|y|e^{-\frac{y^2}{2\sigma^2}}dy, \quad y := x_i-\mu_i, \\
&= \frac{f_N(\boldsymbol{x}^*)}{\sqrt{2\pi}\sigma}\int_{0}^{\infty}2ye^{-\frac{y^2}{2\sigma^2}}dy, \\
&= \frac{f_N(\boldsymbol{x}^*)}{\sqrt{2\pi}\sigma}\int_{0}^{\infty}e^{-\frac{y^2}{2\sigma^2}}dy^2, \\
&= \frac{f_N(\boldsymbol{x}^*)}{\sqrt{2\pi}\sigma}\int_{0}^{\infty}e^{-\frac{t}{2\sigma^2}}dt, \quad t := y^2, \\
&= \frac{f_N(\boldsymbol{x}^*)}{\sqrt{2\pi}\sigma}\int_{0}^{\infty}(-2\sigma^2)de^{-\frac{t}{2\sigma^2}}dt \\
&= \sqrt{2}f_N(\boldsymbol{x}^*)\sigma/\sqrt{\pi} < +\infty.
\end{aligned}$$

This finishes the proof for Lemma 3.3. □

C.4. Proof to Lemma 3.4

Lemma 3.4 *Under Assumption 3.1, any stationary point of $F(\boldsymbol{\mu})$ in $\mathbb{R}^d$ belongs to $\mathcal{S}_{\boldsymbol{x}^*,\delta}$, which is defined in (7).*

Remark C.1. Proposition 2.3 shows that $F(\boldsymbol{\mu})$ has at least one stationary point.

Proof. For any point $\boldsymbol{\mu} \notin \mathcal{S}_{\boldsymbol{x}^*,\delta}$, we show that $\nabla F(\boldsymbol{\mu}) \neq 0$.

On one hand, $\boldsymbol{\mu} \in \mathcal{S}_M - \mathcal{S}_{\boldsymbol{x}^*,\delta}$, then $\|\boldsymbol{\mu}\| \leq \sqrt{d}M$ and $\nabla F(\boldsymbol{\mu}) \neq 0$ because of the definition of N in Section 3.2.

On the other hand, if $\boldsymbol{\mu} \notin \mathcal{S}_M$, there is at least one j such that $|\mu_j| > M$. Then,

$$\begin{aligned}
\frac{\partial F(\boldsymbol{\mu})}{\partial \mu_j} &= \frac{\int_{\boldsymbol{x}\in\mathbb{R}^d}(x_j-\mu_j)e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}}f_N(\boldsymbol{x})d\boldsymbol{x}}{(\sqrt{2\pi})^d\sigma^{d+2}} \\
&= \frac{\int_{\boldsymbol{x}\in\mathcal{S}}(x_j-\mu_j)e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}}f_N(\boldsymbol{x})d\boldsymbol{x}}{(\sqrt{2\pi})^d\sigma^{d+2}}, \text{ by (4)}, \\
&= \begin{cases} negative, & \text{if } \mu_j > M; \\ positive, & \text{if } \mu_j < -M, \end{cases}
\end{aligned}$$

where the last equality is because $x_j \in \mathcal{S} \Rightarrow |x_j| \leq M$ by Assumption 3.1. In sum, for any point $\boldsymbol{\mu} \notin \mathcal{S}_{\boldsymbol{x}^*,\delta}$, $\nabla F(\boldsymbol{\mu}) \neq 0$, which further implies that any stationary point of $F(\boldsymbol{\mu})$ belongs to $\mathcal{S}_{\boldsymbol{x}^*,\delta}$. □

C.5. Proof to Lemma 3.5

Lemma 3.5 *Under Assumption 3.1, for any $\sigma > 0$, the objective function $F_{N,\sigma}(\boldsymbol{\mu})$ is Lipschitz Smooth. That is, for any $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^d$,*

$$\|\nabla F_{N,\sigma}(\boldsymbol{\mu}_1) - \nabla F_{N,\sigma}(\boldsymbol{\mu}_2)\| \leq L\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|,$$

where $L = 2d\sigma^{-2} f_N(\boldsymbol{x}^*)$, and $f_N(\boldsymbol{x}^*) = f^N(\boldsymbol{x}^*)$ for the case of PGS and $f_N(\boldsymbol{x}^*) = e^{Nf(\boldsymbol{x}^*)}$ for the case of EPGS.

Proof. The idea is to bound each entry of the difference, a vector in $\mathbb{R}^d$, on the left-hand side (LHS). For each $i \in \{1, 2, ..., d\}$, define $g_i(\boldsymbol{\mu})$ as the i^{th} entry of $\nabla F_{N,\sigma}(\boldsymbol{\mu}) \in \mathbb{R}^d$. Then, for the i^{th} entry of the difference on LHS,

$$\begin{aligned}
|g_i(\boldsymbol{\mu}_1) - g_i(\boldsymbol{\mu}_2)| &= |\nabla g_i(\boldsymbol{v})(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)|, \text{ by Mean Value Theorem (e.g., Theorem 5.10 in (Magnus \& Neudecker, 2019))},\\
&\leq \|\nabla g_i(\boldsymbol{v})\|\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|, \text{ by Cauchy Schwarz Inequality},\\
&\leq 2\sqrt{d}\sigma^{-2} f_N(\boldsymbol{x}^*)\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|, \text{ proof given later},
\end{aligned}$$

where $\boldsymbol{v}$ is some point on the line segment connecting $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, and $\nabla g_i(\boldsymbol{v})(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$ denotes the inner product of the two vectors in $\mathbb{R}^d$. This further implies

$$\|\nabla F_{N,\sigma}(\boldsymbol{\mu}_1) - \nabla F_{N,\sigma}(\boldsymbol{\mu}_2)\| \leq 2d\sigma^{-2} f_N(\boldsymbol{x}^*)\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|,$$

which is the desired result.

Now, it remains to prove for each $i \in \{1, 2, ..., d\}$ and each $\boldsymbol{\mu} \in \mathbb{R}^d$ that $\|\nabla g_i(\boldsymbol{\mu})\| \leq 2\sqrt{d}\sigma^{-2} f_N(\boldsymbol{x}^*)$. It suffices to derive a bound for each entry of $\nabla g_i(\boldsymbol{\mu}) \in \mathbb{R}^d$. Specifically, for its j^{th} entry:

$$\begin{aligned}
\left|\frac{\partial g_i(\boldsymbol{u})}{\partial \mu_j}\right| &=^{\text{by (6)}} \left|\frac{1}{(\sqrt{2\pi})^d \sigma^{d+2}} \frac{\partial \int_{\boldsymbol{x}\in\mathbb{R}^d} (x_i - \mu_i) f_N(\boldsymbol{x}) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x}}{\partial \mu_j}\right|\\
&= \left|\frac{-\delta_{ij} \int_{\boldsymbol{x}\in\mathbb{R}^d} f_N(\boldsymbol{x}) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x} + \sigma^{-2} \int_{\boldsymbol{x}\in\mathbb{R}^d} (x_i - \mu_i)(x_j - \mu_j) f_N(\boldsymbol{x}) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x}}{(\sqrt{2\pi})^d \sigma^{d+2}}\right|\\
&\leq \sigma^{-2}\delta_{ij}\mathbb{E}[f_N(\boldsymbol{x})] + \frac{\sigma^{-2} f_N(\boldsymbol{x}^*) \int_{\boldsymbol{x}\in\mathbb{R}^d} |x_i - \mu_i||x_j - \mu_j| e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x}}{(\sqrt{2\pi})^d \sigma^{d+2}}, \; f_N(\boldsymbol{x}) \geq 0 \text{ by (4)},\\
&\leq \sigma^{-2}\delta_{ij} f_N(\boldsymbol{x}^*) + \frac{\sigma^{-2} f_N(\boldsymbol{x}^*) \int_{\boldsymbol{x}\in\mathbb{R}^d} ((x_i - \mu_i)^2 + (x_j - \mu_j)^2) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x}}{2(\sqrt{2\pi})^d \sigma^{d+2}}\\
&= \sigma^{-2}\delta_{ij} f_N(\boldsymbol{x}^*) + \frac{\sigma^{-2} f_N(\boldsymbol{x}^*) 2 \int_{x_j\in\mathbb{R}} (x_j - \mu_j)^2 e^{-\frac{(x_j-\mu_j)^2}{2\sigma^2}} dx_j}{2\sqrt{2\pi}\sigma^3}\\
&= \sigma^{-2}\delta_{ij} f_N(\boldsymbol{x}^*) + f_N(\boldsymbol{x}^*)\sigma^{-4} Var(x_j)\\
&= \sigma^{-2}(\delta_{ij} + 1) f_N(\boldsymbol{x}^*)\\
&\leq 2\sigma^{-2} f_N(\boldsymbol{x}^*),
\end{aligned}$$

where the interchange of differentiation and integral in the second line is by Lebesgue's dominated convergence theorem, and the fifth line is because of the separability of the multivariate integral. Here, $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ if $i \neq j$. The above bound for $\left|\frac{\partial g_i(\boldsymbol{u})}{\partial \mu_j}\right|$ implies $\|\nabla g_i(\boldsymbol{\mu})\| \leq 2\sqrt{d}\sigma^{-2} f_N(\boldsymbol{x}^*)$. This finishes the proof for Lemma 3.5. □

C.6. Proof to Lemma 3.6

Lemma 3.6 *Under Assumption 3.1, for any $\sigma > 0$, let $\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu})$ be defined in (5). Then, $\mathbb{E}[\|\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu})\|^2] < G$, where*

$$G = \begin{cases} d\sigma^2 f^{2N}(\boldsymbol{x}^*), & \textit{for PGS};\\ d\sigma^2 e^{2Nf(\boldsymbol{x}^*)}, & \textit{for EPGS}.\end{cases}$$

Proof.

$$\begin{aligned}\mathbb{E}\left[\|\hat{\nabla}F_{N,\sigma}(\boldsymbol{\mu}_t)\|^2\right] &= \frac{1}{K^2}\sum_{k=1}^{K}\sum_{l=1}^{K}\mathbb{E}[(\boldsymbol{x}_k-\boldsymbol{\mu}_t)(\boldsymbol{x}_l-\boldsymbol{\mu}_t)f_N(\boldsymbol{x}_k)f_N(\boldsymbol{x}_l)]\\ &\le f_N^2(\boldsymbol{x}^*)\frac{1}{K^2}\sum_{k=1}^{K}\sum_{l=1}^{K}\mathbb{E}[|(\boldsymbol{x}_k-\boldsymbol{\mu}_t)(\boldsymbol{x}_l-\boldsymbol{\mu}_t)|]\\ &\le f_N^2(\boldsymbol{x}^*)\frac{1}{K^2}\sum_{k=1}^{K}\sum_{l=1}^{K}\mathbb{E}[\|(\boldsymbol{x}_k-\boldsymbol{\mu}_t)\|\|(\boldsymbol{x}_l-\boldsymbol{\mu}_t)\|],\\ &\le f_N^2(\boldsymbol{x}^*)\frac{1}{K^2}\sum_{k=1}^{K}\sum_{l=1}^{K}\sqrt{\mathbb{E}[\|\boldsymbol{x}_k-\boldsymbol{\mu}_t\|^2]\mathbb{E}[\|\boldsymbol{x}_l-\boldsymbol{\mu}_t)\|^2]}\\ &= f_N^2(\boldsymbol{x}^*)\sigma^2 d, \quad d \text{ denotes the number of dimensions,}\end{aligned}$$

where $(\boldsymbol{x}_k-\boldsymbol{\mu}_t)(\boldsymbol{x}_l-\boldsymbol{\mu}_t)$ denotes the inner product of the two vectors, the expectation $\mathbb{E}$ is over the random vectors $\boldsymbol{x}_k, \boldsymbol{x}_l \sim \mathcal{N}(\boldsymbol{\mu}_t, \sigma^2 I_d)$, and the fourth line is from the application of the Cauchy-Schwarz inequality to expectations. Replacing $f_N^2(\boldsymbol{x}^*) = e^{2Nf(\boldsymbol{x}^*)}$ for EPGS and $f_N^2(\boldsymbol{x}^*) = f^{2N}(\boldsymbol{x}^*)$ for PGS gives the desired result. □

C.7. Proof to Theorem 3.7

Theorem 3.7 *Let $\{\boldsymbol{\mu}_t\}_{t=0}^T \subset \mathbb{R}^d$ be produced by following the iteration rule of (5), with a pre-selected and deterministic $\boldsymbol{\mu}_0$ and all the involved terms defined as in Section 3.2. Then, under Assumption 3.1 and 3.2, we have that*

$$\sum_{t=0}^{T-1}\alpha_t\sigma^2\mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t)\|^2] \le f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + LG\sum_{t=0}^{\infty}\alpha_t^2,$$

where L and G are as defined in Lemma 3.5 and Lemma 3.6, respectively.

Proof. By the Gradient Mean Value Theorem, there exists $\boldsymbol{\nu}_t \in \mathbb{R}^d$ such that $\nu_{t,i}$ lies between $\mu_{t+1,i}$ and $\mu_{t,i}$ for each of the ith entry, and

$$\begin{aligned}F(\boldsymbol{\mu}_{t+1}) =&F(\boldsymbol{\mu}_t) + (\nabla F(\boldsymbol{\nu}_t))'(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t),\\ =&F(\boldsymbol{\mu}_t) + (\nabla F(\boldsymbol{\mu}_t))'(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t) + (\nabla F(\boldsymbol{\nu}_t)-\nabla F(\boldsymbol{\mu}_t))'(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t)\\ =&F(\boldsymbol{\mu}_t) + \alpha_t(\nabla F(\boldsymbol{\mu}_t))'(\hat{\nabla}F(\boldsymbol{\mu}_t)) - (\nabla F(\boldsymbol{\mu}_t)-\nabla F(\boldsymbol{\nu}_t))'(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t)\\ \ge&F(\boldsymbol{\mu}_t) + \alpha_t(\nabla F(\boldsymbol{\mu}_t))'(\hat{\nabla}F(\boldsymbol{\mu}_t)) - L\|\boldsymbol{v}_t-\boldsymbol{\mu}_t\|\cdot\|\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t\|, \text{ by Lemma 3.5},\\ \ge&F(\boldsymbol{\mu}_t) + \alpha_t(\nabla F(\boldsymbol{\mu}_t))'(\hat{\nabla}F(\boldsymbol{\mu}_t)) - L\|\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t\|^2,\\ =&F(\boldsymbol{\mu}_t) + \alpha_t(\nabla F(\boldsymbol{\mu}_t))'(\hat{\nabla}F(\boldsymbol{\mu}_t)) - \alpha_t^2 L\|\hat{\nabla}F_N(\boldsymbol{\mu}_t)\|^2.\end{aligned}$$

where $'$ denotes the vector transpose, and the second inequality is because $v_{t,i}$ is between $\mu_{t+1,i}$ and $\mu_{t,i}$. Taking the expectation of the left-end and right-end of the above derived inequality gives

$$\begin{aligned}\mathbb{E}[F(\boldsymbol{\mu}_{t+1})] \ge&\mathbb{E}[F(\boldsymbol{\mu}_t)] + \alpha_t\sigma^2\mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t))\|^2] - \alpha_t^2 L\mathbb{E}[\|\hat{\nabla}F_N(\boldsymbol{\mu}_t)\|^2]\\ \ge&\mathbb{E}[F(\boldsymbol{\mu}_t)] + \alpha_t\sigma^2\mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t))\|^2] - \alpha_t^2 LG,\end{aligned} \tag{16}$$

where the second inequality is because of Lemma 3.6, and for the first inequality, note that

$$\mathbb{E}[(\nabla F(\boldsymbol{\mu}_t))'(\hat{\nabla}F(\boldsymbol{\mu}_t))] =\mathbb{E}\left[\mathbb{E}[(\nabla F(\boldsymbol{\mu}_t))'(\hat{\nabla}F(\boldsymbol{\mu}_t))|\boldsymbol{\mu}_t]\right] =^{\text{by (6)}} \sigma^2\mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t))\|^2].$$

Taking the sum from $t=0$ to $t=T-1$ on both sides of (16) gives

$$\mathbb{E}[F(\boldsymbol{\mu}_T)] \ge \mathbb{E}[F(\boldsymbol{\mu}_0)] + \sum_{t=0}^{T-1}\alpha_t\sigma^2\mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t))\|^2] - LG\sum_{t=0}^{T-1}\alpha_t^2.$$

Re-organizing the terms gives

$$\begin{aligned}\sum_{t=0}^{T-1} \alpha_t \sigma^2 \mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t)\|^2] &\le \mathbb{E}[F(\boldsymbol{\mu}_T)] - \mathbb{E}[F(\boldsymbol{\mu}_0)] + LG\sum_{t=0}^{T-1}\alpha_t^2 \\ &\le f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + LG\sum_{t=0}^{\infty}\alpha_t^2\end{aligned}$$

This finishes the proof for Theorem 3.7. □

C.8. Proof to Corollary 3.9

Corollary 3.9 *Suppose Assumption 3.1 and 3.2 hold. Let $\{\boldsymbol{\mu}_t\}$ be produced by the stochastic gradient ascent rule (5) of GS-PowerOpt, with a pre-selected and deterministic $\boldsymbol{\mu}_0$ Then, for any $\varepsilon \in (0,1)$, after*

$$T > (C_1 C_2 d^2 \varepsilon^{-1})^{2/(1-2\gamma)} = O_N((d^2\varepsilon^{-1})^{2/(1-2\gamma)})$$

times of $\boldsymbol{\mu}_t$-updating by (5), we have that $\min_{t\in\{0,1,\dots,T\}} \mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t)\|^2] < \varepsilon$. Here, $\gamma \in (0, 1/2)$ is a parameter in the learning rate $\alpha_t := (t+1)^{-(1/2+\gamma)}$, $C_0 = f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + 2f_N^3(\boldsymbol{x}^*)\sum_{t=1}^{\infty} t^{-(1+2\gamma)}$, $C_1 = C_0(1-2\gamma)\sigma^{-2}$, and $C_2 = \max\{1, 2/C_1\}$.

Proof. For any non-negative integer t, define

$$\nu_t := \min_{\tau\in\{0,1,\dots,t\}} \mathbb{E}[\|\nabla F(\boldsymbol{\mu}_\tau)\|^2]. \tag{17}$$

Then,

$$\begin{aligned}\sum_{t=0}^{T-1}\alpha_t\sigma^2\nu_T &\le \sum_{t=0}^{T-1}\alpha_t\sigma^2\nu_t, \quad \text{since } \nu_{t+1}\le\nu_t \text{ for each } t\ge 0, \\ &\le \sum_{t=0}^{T-1}\alpha_t\sigma^2\mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t)\|^2], \quad \text{since } \nu_t \le \mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t)\|^2] \text{ from (17)}, \\ &\le f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + LG\sum_{t=0}^{\infty}\alpha_t^2, \quad \text{from Theorem 3.7}, \\ &\le f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + 2d^2 f_N^3(\boldsymbol{x}^*)\sum_{t=1}^{\infty} t^{-(1+2\gamma)}, \quad d \text{ comes from } L \text{ in Lemma 3.5 and } G \text{ in Lemma 3.6}, \\ &\le \left(f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + 2f_N^3(\boldsymbol{x}^*)\sum_{t=1}^{\infty} t^{-(1+2\gamma)}\right)d^2, \quad \text{since } d>1 \text{ and } f_N(\boldsymbol{x}^*)\ge F(\boldsymbol{\mu}_0), \\ &= C_0 d^2, \quad \text{where } C_0 := f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + 2f_N^3(\boldsymbol{x}^*)\sum_{t=1}^{\infty} t^{-(1+2\gamma)} < \infty.\end{aligned} \tag{18}$$

Therefore, we have

$$\sum_{t=0}^{T-1}\alpha_t\sigma^2\nu_T \le C_0 d^2.$$

Since $\alpha_t = (t+1)^{-(1/2+\gamma)}$, dividing both the left-end and the right-end of the above inequality by $\sum_{t=0}^{T-1} \alpha_t \sigma^2$ gives

$$\begin{aligned}
\nu_T &\le \frac{C_0 d^2}{\sigma^2 \sum_{t=1}^{T} t^{-(1/2+\gamma)}} \\
&< \frac{C_0 d^2}{\sigma^2 \int_1^T t^{-(1/2+\gamma)} dt} \\
&= \frac{C_0 d^2}{\sigma^2 (T^{\frac{1}{2}-\gamma} - 1)/(\frac{1}{2}-\gamma)} \\
&< \frac{C_0 d^2}{\sigma^2 (T^{\frac{1}{2}-\gamma}/2)/(\frac{1}{2}-\gamma)}, \quad \text{when } T > 2^{2/(1-2\gamma)}, \\
&= C_1 \frac{d^2}{T^{\frac{1}{2}-\gamma}}, \quad \text{where } C_1 := \frac{C_0}{\sigma^2 (1/2)/(\frac{1}{2}-\gamma)}.
\end{aligned}$$

In sum, we have

$$\nu_T \le C_1 \frac{d^2}{T^{\frac{1}{2}-\gamma}}, \qquad \text{whenever } T > 2^{2/(1-2\gamma)}. \tag{19}$$

Define $C_2 := \max\{1, 2/C_1\}$. Given any $\varepsilon \in (0,1)$, whenever $T > (C_2 C_1 d^2 \varepsilon^{-1})^{2/(1-2\gamma)} = O((d^2\varepsilon^{-1})^{2/(1-2\gamma)})$, we have

$$T > (C_2 C_1 d^2 \varepsilon^{-1})^{2/(1-2\gamma)} > (C_1 C_2)^{2/(1-2\gamma)} \ge 2^{2/(1-2\gamma)},$$

and

$$\nu_T \le^{\text{from (19)}} C_1 \frac{d^2}{T^{\frac{1}{2}-\gamma}} < C_1 \frac{d^2}{(C_2 C_1 d^2 \varepsilon^{-1})^{\frac{2}{1-2\gamma}(\frac{1}{2}-\gamma)}} = \frac{\varepsilon}{C_2} \le \varepsilon.$$

This implies that after $T = O((d^2\varepsilon^{-1})^{2/(1-2\gamma)})$ times of updating $\boldsymbol{\mu}_t$ by GS-PowerOpt, $v_T = \min_{t \in \{0,1,2,\ldots,T\}} \mathbb{E}[\|\nabla F(\boldsymbol{\mu}_t)\|^2] < \varepsilon$. □

C.9. Proof to Proposition 3.11

Proposition 3.11 *Assume Assumption 3.1, $f(\boldsymbol{x}) = 0$ on the boundary of f's domain, the Lipschitz condition on f: $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \le L_0\|\boldsymbol{x}-\boldsymbol{y}\|$, and the Lipschitz condition on ∇f: $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \le L_1\|\boldsymbol{x}-\boldsymbol{y}\|$. Then, the following two statements are true.*

1. *$\|\nabla F_{N,\sigma}(\boldsymbol{\mu}_1) - \nabla F_{N,\sigma}(\boldsymbol{\mu}_2)\| \le L\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|$ holds for any $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^d$, where $L = (Nf^{N-1}(\boldsymbol{x}^*)L_1 + N(N-1)L_0^2 f^{N-2}(\boldsymbol{x}^*))$ under the PGS setting, and $L = (NL_1 e^{Nf(\boldsymbol{x}^*)} + N^2 L_0^2 e^{Nf(\boldsymbol{x}^*)})$ under the EPGS setting.*
2. *The iteration complexity in Corollary 3.9 reduces to $O_N((d\varepsilon^{-1})^{2/(1-2\gamma)})$.*

Proof to Point 1. We extend the domain of f from $\mathcal{S}$ to $\mathbb{R}^d$ such that $f(\boldsymbol{x}) = 0$ for any $\boldsymbol{x} \neq \mathcal{S}$. Clearly, after the domain extension, the two Lipschitz conditions hold in $\mathbb{R}^d$. From (4), with the extended f,

$$f_N(\boldsymbol{x}) = \begin{cases} f^N(\boldsymbol{x}), & \text{(PGS setting)}; \\ e^{Nf(\boldsymbol{x})}, & \text{(EPGS setting)}. \end{cases}$$

By definition,

$$\begin{aligned}
F_{N,\sigma}(\boldsymbol{\mu}) &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma I_d)}[f_N(\boldsymbol{x})] \\
&= (\sqrt{2\pi}\sigma)^{-d} \int_{\boldsymbol{x}\in\mathbb{R}^d} f_N(\boldsymbol{x}) e^{\frac{-\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x} \\
&= (\sqrt{2\pi})^{-d} \int_{\boldsymbol{\xi}\in\mathbb{R}^d} f_N(\boldsymbol{\mu}+\sigma\boldsymbol{\xi}) e^{\frac{-\|\boldsymbol{\xi}\|^2}{2}} d\boldsymbol{\xi}, \quad \boldsymbol{x} := \boldsymbol{\mu}+\sigma\boldsymbol{\xi}, \\
&= \mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(\boldsymbol{0},I_d)}[f_N(\boldsymbol{\mu}+\sigma\boldsymbol{\xi})].
\end{aligned}$$

Let ∇ denotes the first derivative with respect to $\boldsymbol{\mu}$. Under the PGS setting, $\nabla f_N(\boldsymbol{\mu}+\sigma\boldsymbol{\xi}) = Nf^{N-1}(\boldsymbol{\mu}+\sigma\boldsymbol{\xi})\nabla f(\boldsymbol{\mu}+\sigma\boldsymbol{\xi})$. Then,

$$
\begin{aligned}
&\|\nabla F_{N,\sigma}(\boldsymbol{\mu}_1) - \nabla F_{N,\sigma}(\boldsymbol{\mu}_2)\| \\
&= \|\mathbb{E}[\nabla f_N(\boldsymbol{\mu}_1+\sigma\boldsymbol{\xi}) - \nabla f_N(\boldsymbol{\mu}_2+\sigma\boldsymbol{\xi})]\| \\
&= \|\mathbb{E}[Nf^{N-1}(\boldsymbol{\mu}_{1,\xi})\nabla f_N(\boldsymbol{\mu}_{1,\xi}) - Nf^{N-1}(\boldsymbol{\mu}_{2,\xi})\nabla f(\boldsymbol{\mu}_{2,\xi})]\| \\
&= N\|\mathbb{E}[f^{N-1}(\boldsymbol{\mu}_{1,\xi})(\nabla f_N(\boldsymbol{\mu}_{1,\xi}) - \nabla f_N(\boldsymbol{\mu}_{2,\xi})) + (f^{N-1}(\boldsymbol{\mu}_{1,\xi}) - f^{N-1}(\boldsymbol{\mu}_{2,\xi}))\nabla f(\boldsymbol{\mu}_{2,\xi})]\| \\
&\le Nf^{N-1}(\boldsymbol{x}^*)\mathbb{E}\|\nabla f_N(\boldsymbol{\mu}_{1,\xi}) - \nabla f_N(\boldsymbol{\mu}_{2,\xi})\| + NL_0\mathbb{E}|f^{N-1}(\boldsymbol{\mu}_{1,\xi}) - f^{N-1}(\boldsymbol{\mu}_{2,\xi})| \\
&\le Nf^{N-1}(\boldsymbol{x}^*)L_1\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| + N(N-1)f^{N-2}(\boldsymbol{x}^*)L_0\mathbb{E}|f(\boldsymbol{\mu}_{1,\xi}) - f(\boldsymbol{\mu}_{2,\xi})| \\
&\le Nf^{N-1}(\boldsymbol{x}^*)L_1\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| + N(N-1)f^{N-2}(\boldsymbol{x}^*)L_0^2\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| \\
&\le (Nf^{N-1}(\boldsymbol{x}^*)L_1 + N(N-1)L_0^2 f^{N-2}(\boldsymbol{x}^*))\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|,
\end{aligned}
$$

where $\boldsymbol{\mu}_{1,\xi} := \boldsymbol{\mu}_1 + \sigma\boldsymbol{\xi}$, $\boldsymbol{\mu}_{2,\xi} := \boldsymbol{\mu}_2 + \sigma\boldsymbol{\xi}$, the first inequality sign is derived using the formula[10] of $\|\mathbb{E}\boldsymbol{x}\| \le \mathbb{E}\|\boldsymbol{x}\|$ and the result that $\|\nabla f\| \le L_0$ from the Lipschitz condition on f, and the second inequality sign is derived using the formula $a^{N-1} - b^{N-1} = (a-b)\sum_{n=0}^{N-2} a^n b^{N-2-n}$.

Under the EPGS setting, $\nabla f_N(\boldsymbol{\mu}+\sigma\boldsymbol{\xi}) = Ne^{Nf(\boldsymbol{\mu}+\sigma\boldsymbol{\xi})}\nabla f(\boldsymbol{\mu}+\sigma\boldsymbol{\xi})$. Then,

$$
\begin{aligned}
&\|\nabla F_{N,\sigma}(\boldsymbol{\mu}_1) - \nabla F_{N,\sigma}(\boldsymbol{\mu}_2)\| \\
&= \|\mathbb{E}[\nabla f_N(\boldsymbol{\mu}_1+\sigma\boldsymbol{\xi}) - \nabla f_N(\boldsymbol{\mu}_2+\sigma\boldsymbol{\xi})]\| \\
&= \|\mathbb{E}[Ne^{Nf(\boldsymbol{\mu}_{1,\xi})}\nabla f_N(\boldsymbol{\mu}_{1,\xi}) - Ne^{Nf(\boldsymbol{\mu}_{2,\xi})}\nabla f(\boldsymbol{\mu}_{2,\xi})]\| \\
&= N\|\mathbb{E}[e^{Nf(\boldsymbol{\mu}_{1,\xi})}(\nabla f_N(\boldsymbol{\mu}_{1,\xi}) - \nabla f_N(\boldsymbol{\mu}_{2,\xi})) + (e^{Nf(\boldsymbol{\mu}_{1,\xi})} - e^{Nf(\boldsymbol{\mu}_{2,\xi})})\nabla f(\boldsymbol{\mu}_{2,\xi})]\| \\
&\le Ne^{Nf(\boldsymbol{x}^*)}\mathbb{E}\|\nabla f_N(\boldsymbol{\mu}_{1,\xi}) - \nabla f_N(\boldsymbol{\mu}_{2,\xi})\| + NL_0\mathbb{E}|e^{Nf(\boldsymbol{\mu}_{1,\xi})} - e^{Nf(\boldsymbol{\mu}_{2,\xi})}| \\
&\le Ne^{Nf(\boldsymbol{x}^*)}L_1\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| + NL_0Ne^{Nf(\boldsymbol{x}^*)}L_0\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|, \text{ by mean-value theorem,} \\
&\le (NL_1e^{Nf(\boldsymbol{x}^*)} + N^2L_0^2e^{Nf(\boldsymbol{x}^*)})\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|.
\end{aligned}
$$

□

Proof to Point 2. Point 2 can be proved from a modified version of the proof to Corollary 3.9. Specifically, when deriving (18), replacing Lemma 3.5 by Point 1 of Proposition 3.11 removes the dependence of L on d, which decreases the power of d by 1 on the right-end of (18). This in turn replaces d^2 by d in the rest of the proof to Corollary 3.9, and finally leads to the desired iteration complexity in Point 2 of Proposition 3.11. Note that C_0 needs to be redefined accordingly in (18). □

C.10. Proof to Corollary 4.2

Corollary 4.2 *Under Assumption 3.1 and 4.1, the iteration complexity in Corollary 3.9 becomes $O((d^2\varepsilon^{-1})^{2/(1-2\gamma)})$, and the iteration complexity in Point 2 of Proposition 3.11 becomes $O((d\varepsilon^{-1})^{2/(1-2\gamma)})$, both of which are independent from N.*

Proof. The dependence of the iteration complexity $O_N((d^2\varepsilon^{-1})^{2/(1-2\gamma)})$ in Corollary 3.9 on N comes from $C_0 = f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + 2f_N^3(\boldsymbol{x}^*)\sum_{t=1}^{\infty} t^{-(1+2\gamma)}$. Assumption 4.1 guarantees that Corollary 3.9 still holds true if C_0 is redefined as

$$C_0 = 1 + 2\sum_{t=1}^{\infty} t^{-(1+2\gamma)}.$$

This makes the iteration complexity in Corollary 3.9 become $O((d^2\varepsilon^{-1})^{2/(1-2\gamma)})$, which is independent from N.

To prove that Corollary 3.9 still holds true with the above new definition of C_0, we only need to modify the last line of (18) in the proof to Corollary 3.9:

$$\left(f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + 2f_N^3(\boldsymbol{x}^*)\sum_{t=1}^{\infty} t^{-(1+2\gamma)}\right)d^2 \le \left(1 + 2\sum_{t=1}^{\infty} t^{-(1+2\gamma)}\right)d^2 = C_0d^2,$$

[10] This inequality can be derived from Jensen's inequality and the fact that the Euclidean norm $\|\cdot\|$ is convex.

where the inequality can be justified by $f_N(\boldsymbol{x}^*) \in [0,1)$ (a result immediately implied by Assumption 4.1).

Next, we prove that the iteration complexity in Point 2 of Proposition 3.11 becomes $O((d\varepsilon^{-1})^{2/(1-2\gamma)})$ under Assumption 4.1. Let ν_t be defined as in (17).

$$\begin{aligned}
\sum_{t=0}^{T-1} \alpha_t \sigma^2 \nu_T &\le f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + LG \sum_{t=0}^{\infty} \alpha_t^2, \quad \text{from the third line in (18)},\\
&\le f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + Ld\sigma^2 f_N^2(\boldsymbol{x}^*) \sum_{t=0}^{\infty} \alpha_t^2, \quad \text{from Lemma 3.6},\\
&\le f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + Ld\sigma^2 \sum_{t=0}^{\infty} \alpha_t^2, \quad \text{since } f_N(\boldsymbol{x}^*) \in [0,1),\\
&\le f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + B(L_1 + L_0^2) d\sigma^2 \sum_{t=0}^{\infty} \alpha_t^2, \quad \text{from (20) given below},\\
&\le (1 + B(L_1 + L_0^2)\sigma^2 \sum_{t=0}^{\infty} \alpha_t^2) d,\\
&= C_0 d, \quad \text{where } C_0 := 1 + B(L_1 + L_0^2)\sigma^2 \sum_{t=0}^{\infty} \alpha_t^2.
\end{aligned}$$

In sum, we have $\sum_{t=0}^{T-1} \alpha_t \sigma^2 \nu_T \le C_0 d$, where $C_0 = 1 + B(L_1 + L_0^2)\sigma^2 \sum_{t=0}^{\infty} \alpha_t^2$ is independent from N. This result, and the rest of the proof (where the term d^2 is replaced by d) to Corollary 3.9 after Eq. (18), gives the desired iteration complexity of $O((d\varepsilon^{-1})^{2/(1-2\gamma)})$.

Finally, it remains to prove for some constant $B > 0$,

$$L \le B(L_1 + L_0^2). \tag{20}$$

Under the PGS setting, from Point 1 of Proposition 3.11, $L = (Nf^{N-1}(\boldsymbol{x}^*)L_1 + N(N-1)L_0^2 f^{N-2}(\boldsymbol{x}^*))$. Since $f(\boldsymbol{x}^*) \in [0,1)$,

$$\lim_{N\to+\infty} Nf^{N-1}(\boldsymbol{x}^*) = \lim_{N\to+\infty} N(N-1)f^{N-2}(\boldsymbol{x}^*) = 0,$$

which implies both $Nf^{N-1}(\boldsymbol{x}^*)$ and $N(N-1)f^{N-2}(\boldsymbol{x}^*)$ are upper bounded by a constant B for all $N \ge 0$. This implies $L \le B(L_1 + L_0^2)$.

Under the EPGS setting, from Point 1 of Proposition 3.11, $L = (NL_1 e^{Nf(\boldsymbol{x}^*)} + N^2 L_0^2 e^{Nf(\boldsymbol{x}^*)})$. Since $f(\boldsymbol{x}^*) < 0$,

$$\lim_{N\to+\infty} Ne^{Nf(\boldsymbol{x}^*)} = \lim_{N\to+\infty} N^2 e^{Nf(\boldsymbol{x}^*)} = 0,$$

which implies both $Ne^{Nf(\boldsymbol{x}^*)}$ and $N^2 e^{Nf(\boldsymbol{x}^*)}$ are upper bounded by a constant B for all $N \ge 0$. This implies $L \le B(L_1 + L_0^2)$.

□

D. More Details and Discussions on Compared Algorithms

D.1. Zeroth-order Gradient Algorithms

The two zeroth-order gradient algorithms, ZO-SGD (Ghadimi & Lan, 2013)) and ZO-AdaMM (Chen et al., 2019), aim to solve $\max_{\boldsymbol{x}} \mathbb{E}[\boldsymbol{x} + \sigma\boldsymbol{\xi}]$. Here, $\boldsymbol{\xi}$ denotes a random noise and we select it to be a standard multi-variate Gaussian vector. For non-concave maximization problems, they are theoretically guaranteed to converge to stationary points of f. In our experiments, we take the scaling parameter σ as a hyper-parameter.

D.2. STD-Homotopy

STD-Homotopy is a standard homotopy algorithm for optimization. It has a double-loop mechanism. The inner loop updates the solution $\boldsymbol{\mu}_t$ with a fixed scaling parameter σ until no improvements of $f(\boldsymbol{\mu}_t)$ are made, and the outer loop decays σ iteratively. In this algorithm, the term $\widehat{\nabla F}(\boldsymbol{\mu}, \sigma)$ is an estimate of $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma I_d)}[(\boldsymbol{x}-\boldsymbol{\mu})f(\boldsymbol{x})] = \sigma^2\nabla_{\boldsymbol{\mu}}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma I_d)}[f(\boldsymbol{x})]$, which is used to update $\boldsymbol{\mu}$.

Algorithm 2 STD-Homotopy

```
**Input:** The maximum of iteration number $T_{total} > 0$, the initial scaling parameter $\sigma > 0$, the objective $f$, the initial value $\boldsymbol{\mu}_0$, the number $K$ of sampled points for gradient approximation, the maximum number $N_\sigma$ of times $\sigma$ gets updated, the maximum $T_{\boldsymbol{\mu}}$ of the number of times $\boldsymbol{\mu}$ gets updated for each value of $\sigma$, the tolerance number $\tau$ for no improvements of $f(\boldsymbol{\mu}_t)$ for any fixed $\sigma$, the decay factor $\gamma \in (0, 1)$, and the learning rate $\alpha_t > 0$.
Initialize $t_1 = 0$, $n_\sigma = 0$.
**while** $t_1 \leq T_{total}$ and $n_\sigma < N_\sigma$ **do**
  $t = 0, I = True$.
  **while** $t < T_{\boldsymbol{\mu}}$ and $I == True$ and $t_1 \leq T_{total}$ **do**
    Independently and uniformly sample $K$ points $\{\boldsymbol{v}_k\}_{k=1}^K$ from the uniform sphere in $\mathbb{R}^d$. Compute $\boldsymbol{x}_k := \boldsymbol{\mu}_t + \sigma\boldsymbol{v}_k$, for each $k$.
    Compute the gradient estimate $\widehat{\nabla F}(\boldsymbol{\mu}_t, \sigma) = \frac{1}{K}\sum_{k=1}^K(\boldsymbol{x}_k - \boldsymbol{\mu}_t)f(\boldsymbol{x}_k)$.
    $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \alpha_t\widehat{\nabla F_N}(\boldsymbol{\mu}_t, \sigma)/\|\widehat{\nabla F_N}(\boldsymbol{\mu}_t, \sigma)\|$.
    **if** $\max\{f(\boldsymbol{\mu}_{t+1}), f(\boldsymbol{\mu}_t), ..., f(\boldsymbol{\mu}_{t-\tau+1})\} \leq f(\boldsymbol{\mu}_{t-\tau})$ **then**
      $I = False$.
    **end if**
    $t_1 = t_1 + 1, t = t + 1$.
  **end while**
  $\sigma = \gamma\sigma, n_\sigma = n_\sigma + 1$.
**end while**
Return($\boldsymbol{\mu}_{t_1}$).
```

D.3. ZO-SGD

The zeroth-order stochastic gradient ascent (ZO-SGD) is a maximize-version of Equation (1) in (Chen et al., 2019), whose gradient estimate method is from (Nesterov & Spokoiny, 2017). The gradient-ascent version of this algorithm is in Algorithm 3.

Algorithm 3 ZO-SGD

```
**Input:**: The scaling parameter $\sigma > 0$, the objective $f$, the initial value $\boldsymbol{\mu}_0 \in \mathbb{R}^d$, the number $K$ of sampled points for gradient approximation, the total number $T$ of $\boldsymbol{\mu}$-updates, and the learning rate schedule $\{\alpha_t\}_{t=1}^T$.
**for** $t$ from 0 to $T - 1$ **do**
  Independently and uniformly sample $K$ points $\{\boldsymbol{v}_k\}_{k=1}^K$ from the uniform sphere in $\mathbb{R}^d$.
  Compute the gradient estimate
      $$\widehat{\nabla F}(\boldsymbol{\mu}_t, \sigma) = \frac{1}{K}\sum_{k=1}^{K}\frac{(f(\boldsymbol{\mu}_t + \sigma\boldsymbol{v}_k) - f(\boldsymbol{\mu}_t))\boldsymbol{v}_k d}{\sigma}.$$
  $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \alpha_t\widehat{\nabla F_N}(\boldsymbol{\mu}_t, \sigma)$.
**end for**
Return($\boldsymbol{\mu}_T$).
```

E. Details on the Experiments for Figure 4

The set of N-values is $\{10, 20, ..., 65\}$ for PGS and $\{1.0, 1.5, 2.0, ..., 4.5\}$ for EGS. For each N value and each algorithm, we do 100 trials to stabilize the result. In each trial, the initial solution candidate $\boldsymbol{\mu}_0$ is uniformly sampled from $C := \{\boldsymbol{x} \in \mathbb{R}^d | x_i \in [-1.0, 1.0], i \in \{1, 2, ..., d\}\}$, where x_i represents the i^{th} entry of $\boldsymbol{x}$. We set the initial learning rate as 0.1, the scaling parameter σ as 0.5, and the total number of solution updates as 1000. The objective for PGS is modified to be $f_1(\boldsymbol{x}) := f(\boldsymbol{x}) + 10$ to ensure that the PGS agent will not encounter negative fitness values during the 1000 updates.

F. Hyper-parameters

For experiments on the benchmark test functions (Table 1 and 2), the set of hyper-parameter values with the smallest mean square error (averaged over the 100 experiments) between the true and estimated solutions are selected. The set of candidate values, as well as the selected values, are listed in Table 6 and 7.

For the image attacks (Table 3 and 4), for each set of the hyper-parameter candidate values, we randomly choose 10 images to attack. The set with the highest average fitness value (average over the 10 image attacks) will be selected. Table 8 and 9 reports the candidate set and the selected values.

Table 6: Hyper-parameters for Optimizing Ackley. The candidate set for learning rates is $\mathcal{L} := \{.1, .001\}$. The candidate set for smoothing parameters is $\mathcal{S} := \{.1, 1.0, 2.0\}$. t_1 in ZO-SLGHd and ZO-SLGHr is the initial scaling parameter. μ in ZO-AdaMM is the scaling parameter. The set of candidate values that lead to the minimum mean square error between the true solution and the found solution will be selected. For CMA-ES, σ_0 denotes the initial standard deviation.

	Selected Values	Candidates
CMA-ES	$\sigma_0 = 1.5$	$\sigma_0 \in \{.5, 1.0, 1.5\}$.
EPGS	$\alpha_0 = .1$, $N = 1$, $\sigma = 1.0$.	$N \in \{1, 2, 3\}$, $\alpha_0 \in \mathcal{L}$, $\sigma \in \mathcal{S}$.
PGS	$\alpha_0 = .1$, $N = 20$, $\sigma = 1.0$.	$N \in \{5, 10, 20, 30\}$, $\alpha_0 \in \mathcal{L}$, $\sigma \in \mathcal{S}$.
STD-Homotopy	$\alpha = .1$, $\gamma = .5$, $\sigma = 2.0$, $T_{\boldsymbol{\mu}} = 500$, $\tau = 100$, $N_\sigma = 10$.	$\gamma \in \{.2, .5, .8\}$, $\alpha \in \mathcal{L}$, $\sigma \in \mathcal{S}$.
ZO-SLGHd	$\beta = .001$, $\eta = .001$, $t_1 = 2.0$, $\gamma = .99$	$\beta \in \{.1, .001\}$, $\eta \in \{.1, .01, .001\}$, $t_1 \in \mathcal{S}$, $\gamma \in \{.99, .95\}$.
ZO-SLGHr	$\beta = .001$, $t_1 = 0.1$, $\gamma = .995$	$\beta \in \mathcal{L}$, $\gamma \in \{.999, .995\}$, $t_1 \in \mathcal{S}$.
ZO-AdaMM	$\beta_1 = .5$, $\beta_2 = .5$, $\alpha = 0.1$, $\mu = 1.0$.	$\alpha \in \mathcal{L}$, $\beta_1 \in \{.5, .7, .9\}$, $\beta_2 \in \{.1, .3, .5\}$, $\mu \in \mathcal{S}$
ZO-SGD	$\alpha = .1$, $\sigma = 1.0$.	$\alpha \in \mathcal{L}$, $\mu \in \mathcal{S}$.

G. Adversarial Images

We randomly select four images $\{\boldsymbol{a}_m\}_{m=1}^4$ from the test image set and perform a per-image adversarial attack on each of them, using each of the compared algorithms. Table 10 and 11 report the adversarial image $\boldsymbol{a}_m + \boldsymbol{\mu}_m^*$ with the highest R^2 score, given that the attack is successful. That is, $\boldsymbol{\mu}_m^* = \arg\max_{\boldsymbol{\mu}_{m,t} \in \{\boldsymbol{\mu}_{m,t}\}_{t=0}^{T-1}} R^2(\boldsymbol{a}_m, \boldsymbol{a}_m + \boldsymbol{\mu}_{m,t})$, where $\{\boldsymbol{\mu}_{m,t}\}_{t=0}^{T-1}$ denote the sequence of perturbations generated during the m^{th} attack, and T is set at the same value (1,500) as in Table 3 and 4. According to the results in Table 10 and 11, the performances of the compared algorithms are consistent with the statistics reported in Table 3 and 4.

Table 7: Hyper-parameters for Optimizing Rosenbrock. The candidate set for learning rates is $\mathcal{L} := \{.2, .1, .01, .001, .0001\}$. The candidate set for smoothing parameters is $\mathcal{S} := \{1.0, 2.0\}$. t_1 in ZO-SLGHd and ZO-SLGHr is the initial scaling parameter. μ in ZO-AdaMM is the scaling parameter. For EPGS and PGS, $\alpha_t = \frac{1{,}000\alpha_0}{1{,}000+t}$. The set of candidate values that lead to the minimum mean square error between the true solution and the found solution will be selected.

	Selected Values	Candidates
CMA-ES	$\sigma_0 = .1$.	$\sigma_0 \in \{.1, .5, 1.0\}$.
EPGS	$\alpha_0 = .1, N = 1, \sigma = 1.0$.	$N \in \{1, 2, 3\}, \alpha_0 \in \mathcal{L}, \sigma \in \mathcal{S}$.
PGS	$\alpha_0 = .1, N = 1, \sigma = 1.0$.	$N \in \{1, 3, 5\}, \alpha_0 \in \mathcal{L}, \sigma \in \mathcal{S}$.
STD-Homotopy	$\alpha = .2$, $\gamma = .2$, $\sigma = 2.0$,, $T_{\boldsymbol{\mu}} = 500$, $\tau = 100$, $N_\sigma = 10$.	$\gamma \in \{.2, .5, .8\}, \alpha \in \mathcal{L}, \sigma \in \mathcal{S}$.
ZO-SLGHd	$\beta = .0001$, $\eta = .001$, $t_1 = 2.0$, $\gamma = .999$	$\beta \in \mathcal{L}$, $\eta \in \{.1, .01, .001\}$, $t_1 \in \mathcal{S}$, $\gamma \in \{.99, .995, .999\}$.
ZO-SLGHr	$\beta = .0001$, $t_1 = 2.0$, $\gamma = .999$.	$\beta \in \mathcal{L}, \gamma \in \{.999, .995\}, t_1 \in \mathcal{S}$.
ZO-AdaMM	$\beta_1 = .5$, $\beta_2 = .5$, $\alpha = .2$, $\mu = 2.0$.	$\alpha \in \mathcal{L}, \beta_1 \in \{.5, .7, .9\}, \beta_2 \in \{.1, .3, .5\}, \mu \in \mathcal{S}$
ZO-SGD	$\alpha = .001, \sigma = 2.0$.	$\alpha \in \mathcal{L}, \mu \in \mathcal{S}$.

Table 8: Hyper-parameters for MNIST Attack. The candidate set for (initial) smoothing parameters is $\mathcal{S} := \{1.0, .1\}$. The hyper-parameter symbols for each algorithm are the same as their source publications. For example, t_1 in ZO-SLGHd and ZO-SLGHr denotes the initial scaling parameter, μ in ZO-AdaMM is the scaling parameter, and α denotes a constant learning rate. The number 784 equals the dimensional number d. It appears in the candidate set since it is taken from (Iwakiri et al., 2022), who performed similar experiments. The set of candidate values that lead to the highest fitness (averaged over the 10 image attackes) are be selected. For CMA-ES, σ_0 denotes the initial standard deviation.

	Selected Values	Candidates ($\boldsymbol{\mu}^*$)
CMA-ES	$\sigma_0 = .05$.	$\sigma_0 \in \{.05, .1, .5\}$.
EPGS	$\alpha = .1, N = .05, \sigma = .1$.	$N \in \{.02, .03, .04, .05\}, \alpha_t \in \{.1, .05\}, \sigma \in \mathcal{S}$.
STD-Homotopy	$\alpha = .5$, $\gamma = .5$, $\sigma = 1.0$,, $T_{\boldsymbol{\mu}} = 500$, $\tau = 100$, $N_\sigma = 10$.	$\gamma \in \{.5, .8\}, \alpha \in \{.5, .1\}, \sigma \in \mathcal{S}$.
ZO-SLGHd	$\beta = 10^{-4}$, $\eta = .1/784$, $t_1 = .1, \gamma = .995$	$\beta \in \{1/784, 10^{-4}, .1\}, \eta \in \{.1/784, 10^{-3}\}, t_1 \in \mathcal{S}, \gamma \in \{.999, .995\}$.
ZO-SLGHr	$\beta = 10^{-4}$, $t_1 = .1$, $\gamma = .995$.	$\beta \in \{10^{-4}, 1/784, .1\}, \gamma \in \{.999, .995\}, t_1 \in \mathcal{S}$.
ZO-AdaMM	$\beta_1 = .9$, $\beta_2 = .1$, $\alpha = .1$, $\mu = .1$.	$\alpha \in \{100/784, .001, .1\}$, $\beta_1 \in \{.5, .9\}$, $\beta_2 \in \{.1, .3\}, \mu \in \mathcal{S}$
ZO-SGD	$\alpha = 10^{-4}, \mu = .1$.	$\alpha \in \{10^{-4}, .1, 1/784\}, \mu \in \mathcal{S}$.

Table 9: Hyper-parameters for CIFAR-10 Attack. The candidate set for (initial) smoothing parameters is $\mathcal{S} := \{1.0, .1\}$. The symbols are the same as those in Table 8. The set of candidate values that lead to the highest fitness (averaged over the 10 image attackes) are be selected.

	Selected Values	Candidates ($\boldsymbol{\mu}^*$)
CMA-ES	$\sigma_0 = .05$.	$\sigma_0 \in \{.05, .1, .5\}$.
EPGS	$\alpha = .1$, $N = .03$, $\sigma = .1$.	$N \in \{.02, .03, .04\}$, $\alpha_t \in \{.1, .05\}$, $\sigma \in \mathcal{S}$.
STD-Homotopy	$\alpha = .5$, $\gamma = .8$, $\sigma = .1$, $T_{\boldsymbol{\mu}} = 300$, $\tau = 100$, $N_\sigma = 10$.	$\gamma \in \{.5, .8\}$, $\alpha \in \{.1, .5\}$, $\sigma \in \mathcal{S}$.
ZO-SLGHd	$\beta = .01/3072$, $\eta = 10^{-5}$, $t_1 = 1.0$, $\gamma = .999$	$\beta \in \{.01/3072, .1\}$, $\eta \in \{10^{-4}/784, 10^{-5}\}$, $t_1 \in \mathcal{S}$, $\gamma \in \{.995, .999\}$.
ZO-SLGHr	$\beta = .01/3072$, $t_1 = .1$, $\gamma = .995$.	$\beta \in \{.01/3072, .1\}$, $\gamma \in \{.999, .995\}$, $t_1 \in \mathcal{S}$.
ZO-AdaMM	$\beta_1 = .9$, $\beta_2 = .1$, $\alpha = .1$, $\mu = .1$.	$\alpha \in \{.5/3072, .001, .1\}$, $\beta_1 \in \{.5, .9\}$, $\beta_2 \in \{.1, .3\}$, $\mu \in \mathcal{S}$
ZO-SGD	$\alpha = .01/3072$, $\sigma = .1$.	$\alpha \in \{.01/3072, 10^{-4}, .05\}$, $\sigma \in \mathcal{S}$.

Table 10: Adversarial images for MNIST, produced by algorithms that apply the smoothing techniques. The adversarial target label $\mathcal{T} := \arg\min_i \left(\mathcal{C}(\boldsymbol{a})\right)_i$ is the label with the minimum predicted probability. All attacks are successful.

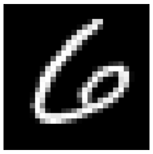
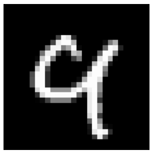
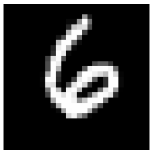
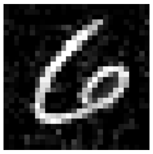
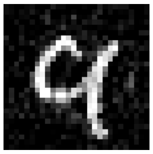
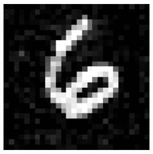
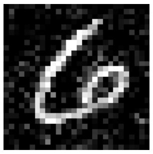
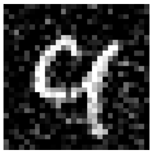
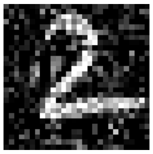
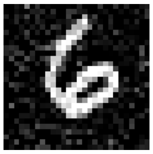
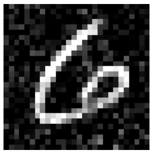
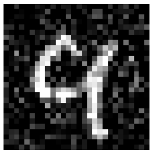
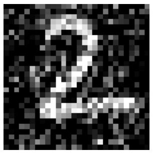
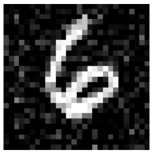
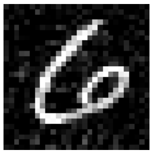
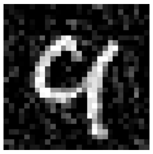
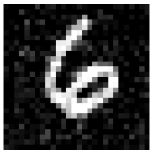
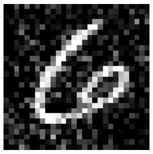
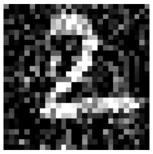
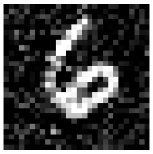
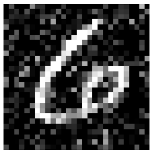
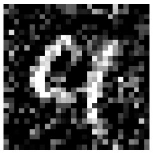
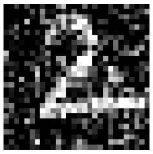
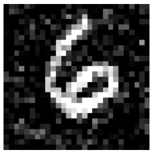
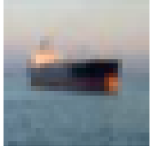
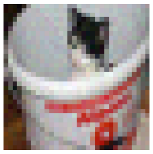
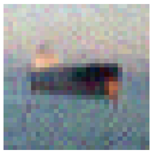
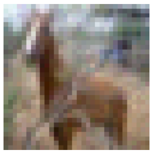
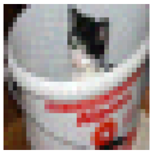
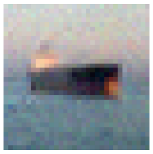
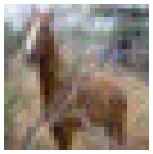
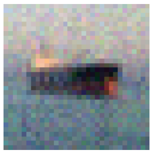
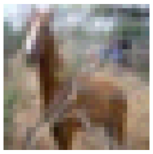
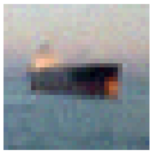
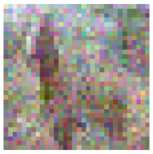
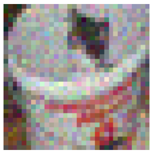
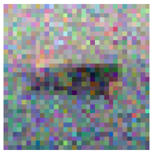
Test Image ID	9953	3850	4962	3886
Original Image **Adversarial Target Label** $\mathcal{T}$	 9	 0	 9	 3
EPGS $R^2(\boldsymbol{a}, \boldsymbol{a} + \boldsymbol{\mu}^*)$	 90%	 88%	 78%	 93%
ZO-SLGHd $R^2(\boldsymbol{a}, \boldsymbol{a} + \boldsymbol{\mu}^*)$	 70%	 61%	 41%	 76%
ZO-SLGHr $R^2(\boldsymbol{a}, \boldsymbol{a} + \boldsymbol{\mu}^*)$	 68%	 55%	 14%	 74%
ZO-SGD $R^2(\boldsymbol{a}, \boldsymbol{a} + \boldsymbol{\mu}^*)$	 85%	 76%	 61%	 87%
ZO-AdaMM $R^2(\boldsymbol{a}, \boldsymbol{a} + \boldsymbol{\mu}^*)$	 50%	 36%	 15%	 60%
STD-Homotopy $R^2(\boldsymbol{a}, \boldsymbol{a} + \boldsymbol{\mu}^*)$	 21%	 −15%	 −19%	 61%

Table 11: Adversarial images for CIFAR-10, produced by algorithms that apply the smoothing techniques. The adversarial target label $\mathcal{T} := \arg\min_i \left(\mathcal{C}(\boldsymbol{a})\right)_i$ is the label with the minimum predicted probability. Unsuccessful attacks (i.e., predicted label is different from the adversarial target) are marked with 'Unsuccessful'.

Test Image ID	9953	3850	4962	3886
Original Image **True Label $\mathcal{T}$** **Adversarial Target Label $\mathcal{T}$**	 7 (i.e., Horse) 1(i.e., Automobile)	 9 (i.e., Truck) 4 (i.e., Deer)	 3 (i.e., Cat) 4 (i.e., Deer)	 8 (i.e., Ship) 7 (i.e., Horse)
EPGS $R^2(\boldsymbol{a}, \boldsymbol{a}+\boldsymbol{\mu}^*)$	 96.0%	 99.3%	 99.0%	 96.1%
ZO-SLGHd $R^2(\boldsymbol{a}, \boldsymbol{a}+\boldsymbol{\mu}^*)$	 99.2%	 99.6%	 99.6%	 98.3%
ZO-SLGHr $R^2(\boldsymbol{a}, \boldsymbol{a}+\boldsymbol{\mu}^*)$	 97.7%	 99.5%	 98.8%	 94.3%
ZO-SGD $R^2(\boldsymbol{a}, \boldsymbol{a}+\boldsymbol{\mu}^*)$	 99.6%	 99.8%	Unsuccessful.	 99.1%
ZO-AdaMM $R^2(\boldsymbol{a}, \boldsymbol{a}+\boldsymbol{\mu}^*)$	 39.6%	 92.0%	 84.9%	 39.7%
STD-Homotopy $R^2(\boldsymbol{a}, \boldsymbol{a}+\boldsymbol{\mu}^*)$	Unsuccessful.	 93.2%	 92.8%	Unsuccessful.