Power Homotopy for Zeroth-Order Non-Convex Optimizations

Chen Xu[1]

[1]Department of Engineering, Shenzhen MSU-BIT University, China.
Email: xuchen@smbu.edu.cn

Abstract

We introduce GS-PowerHP, a novel zeroth-order method for non-convex optimization problems of the form $\max_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$. Our approach leverages two key components: a power-transformed Gaussian-smoothed surrogate $F_{N,\sigma}(\boldsymbol{\mu}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I_d)}[e^{Nf(\boldsymbol{x})}]$ whose stationary points cluster near the global maximizer $\boldsymbol{x}^*$ of f for sufficiently large N, and an incrementally decaying σ for enhanced data efficiency. Under mild assumptions, we prove convergence in expectation to a small neighborhood of $\boldsymbol{x}^*$ with the iteration complexity of $O(d^2\varepsilon^{-2})$. Empirical results show our approach consistently ranks among the top three across a suite of competing algorithms. Its robustness is underscored by the final experiment on a substantially high-dimensional problem ($d = 150,528$), where it achieved first place on least-likely targeted black-box attacks against images from ImageNet, surpassing all competing methods.

Keywords: Gaussion Smoothing, Homotopy for Optimizations, Adversarial Image Attacks.

1 Introduction

Zeroth-order (ZO) optimization methods minimize an objective function $f : \mathbb{R}^d \to \mathbb{R}$ without requiring its gradient. This makes them particularly useful for non-differentiable or non-convex problems, which are prevalent in machine learning and computer vision. Prominent applications include training neural networks [24, 27, 31], hyperparameter tuning [3], and generating image adversarial attacks [6, 23, 28].

A powerful category of zeroth-order optimization methods, namely homotopy for optimization (e.g., [22, 14]), are characterized by a surrogate objective, which is constructed as the Gaussian smoothed transform of the original: $F_\sigma(\boldsymbol{\mu}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I_d)}[f(\boldsymbol{x})]$, where $\mathcal{N}$ denotes a Gaussian distribution, I_d denotes an identity matrix of shape $d \times d$, and $\sigma > 0$ is called the smoothing radius (or scaling parameter). This transformation smooths the landscape of F compared to the original f, which can remove sharp local minima and facilitate locating the global optimum. The gradient of $F_\sigma(\boldsymbol{\mu})$ can be efficiently estimated (e.g., using the method in [25]) with f's value on points randomly sampled near $\boldsymbol{\mu}$, which enables stochastic gradient methods for optimizing F_σ. However, the optimum point $\boldsymbol{\mu}^*$ of F_σ is in general away from the global optimum $\boldsymbol{x}^*$ of f, unless σ is close to 0. Therefore, the standard homotopy applies a double-loop mechanism, where the outer loop incrementally decreases the smoothing radius σ and the inner loop performs a stochastic gradient algorithm to find the maximizer of F_σ under the current σ value ([14]).

The double-loop mechanism is time-consuming. The iteration complexity for the standard ZO homotopy is $O(d^2\varepsilon^{-4})$ [14, Theorem 5.1] and selecting an efficient σ-decreasing schedule is difficult. Therefore, [17] proposes an efficient single-loop method (ZOSLGH) that updates the solution candidate and σ at the same iteration. However, ZOSLGH is only theoretically guaranteed to converge a local optimum of f.

Another smoothing-based method, named GS-

PowerOpt ([30]), proposes to power transform the objective before performing Gaussian smoothing, rather than adopting a σ-decreasing schedule. As illustrated in their Section 2.1, increasing the power will drag the surrogate F's global optimum to $\boldsymbol{x}^*$ (global optimum of f). In fact, the author proves that with a sufficiently large power, all the stationary points of the surrogate objective fall in a close neighborhood of the global optimum point $\boldsymbol{x}^*$ of f, given that f has a unique global optimum. Despite of its good empirical performances, a potential issue lies in its fixed scaling parameter σ. Although a larger σ increases the method's exploration capacity, it also reduces the algorithm's sample efficiency, as discussed in their Section 6.

In this work, we propose a new smoothing-based optimization method, named Power-Transformed Gaussian Homotopy (GS-PowerHP), that tackles the issue associated with the recent methods. Compared to ZOSLGH, the advantage of GS-PowerHP lies in its adoption of the power transformed objective, which leads to the theoretical guarantee of a convergence to a small neighborhood of the global optimum point $\boldsymbol{x}^*$. Compared to GS-PowerOpt, the advantage of GS-PowerHP lies in its adoption of the σ-decreasing mechanism which provides significant improvements, as illustrated in simple example problems and the complex experimental tasks of image adversarial attacks.

Contributions. Our contribution is threefold. (1) Based on our current understanding, GS-PowerHP is the first algorithm that combines the power-transformed objective with a σ-decaying mechanism, which significantly increases the algorithm's capacity for non-convex optimizations, as shown in our experiments. (2) We provide a theoretical analysis for GS-PowerHP, which shows a convergence in expectation to an arbitrarily small neighborhood of the global optimum $\boldsymbol{x}^*$, with an iteration complexity of $O(d^2\varepsilon^{-2})$. (3) To the best of our knowledge, this is the first published work to specifically evaluate the performance of optimization-based algorithms on the least-likely targeted black-box adversarial attack scenario using the ImageNet dataset. While black-box perturbation-generating attacks against ImageNet are abundant, they typically address the less challenging untargeted setting ([16]), or, in the case of targeted attacks, they do not explicitly employ the most challenging least-likely target selection ([12, 7, 5, 1]).

2 Related Works

Homotopy is a popular method for black-box non-convex optimizations problems in machine learning. It is initially designed with a double-loop mechanism and its theoretical convergence property is derived in both the deterministic and stochastic setting ([22, 14]). Owing to its strong performance, homotopy for optimization has attracted significant research interest in recent years, leading to the development of several notable variants. These include: a shift to a more efficient single-loop mechanism [17]; the adaptation of the sampling distribution from Gaussian to a problem-specific distribution [10]; and the introduction of a homotopy-inspired path-learning method [20].

GS-PowerOpt [30] is another type of smoothing-based method closely related to our work. To solve $\max_{\boldsymbol{x}} f(\boldsymbol{x})$, it optimizes the surrogate objective of $F_{N,\sigma}(\boldsymbol{\mu}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I_d)}[e^{Nf(\boldsymbol{x})}]$ with stochastic gradient ascent. The primary distinction between the GS-PowerOpt approach and our proposed method lies in the mechanism governing the noise magnitude σ. GS-PowerOpt employs a fixed σ value throughout the optimization, whereas our method strategically adopts a decreasing mechanism for σ to enhance convergence. To our knowledge, there are a few other studies that also incorporates the power-transformed objective with Gaussian smoothing [9, 26, 4]. However, these studies do not establish a formal relationship between the power size and the geometric shift–the distance between the global optimum of the original objective function (f) and that of the smoothed surrogate $F_{N,\sigma}(\boldsymbol{\mu})$. Furthermore, their methodologies do not contain an incremental σ-decreasing mechanism.

Our method can be categorized as an Evolutionary Algorithm (EA) (e.g., [13, 21]), characterized by an iterative process of internal state updates and random sampling. However, the rule governing our in-

ternal state updates is distinct from conventional EA approaches. As demonstrated by our experimental results, this design yields powerful performance in image adversarial attacks. This research field helps building safe and robust classification models. Furthermore, unlike domain-specific methods such as the Square Attack [1] and SimBA [12], our method exhibits greater universality as a general-purpose black-box zeroth-order optimization technique, making it applicable to a broader range of gradient-free problems.

3 Motivation for the σ-Decaying Mechanism

The σ-decaying mechanism in our approach is inspired from our experimental finding on GS-PowerOpt [30]: when N is large, a smaller σ amplifies $\|\nabla F_{N,\sigma}(\boldsymbol{\mu})\|$ within the region $\mathcal{S}$ near f's global maximizer(s), while rendering $\|\nabla F_{N,\sigma}(\boldsymbol{\mu})\|$ negligible in regions outside $\mathcal{S}$. This is illustrated with a simple example in Figure 1. Therefore, in the solution update iterations of $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_{t+1} + \alpha_t \hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu}_t)$, we propose initially setting σ to a relatively large value and then gradually decreasing it. This ensures that the unbiased gradient estimator $\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu}_t)$ remains sufficiently large even when $\boldsymbol{\mu}_t$ is far from f's global maximizer(s). In this way, the gradient magnitude is maintained at an appropriate scale throughout the optimization process.

We provide an intuitive explanation on the finding, under the assumption that f has a unique global maximum $\boldsymbol{x}^*$. However, as evidenced by the experimental results in Subsection 6.3, our approach remains effective for objectives with potentially many global optimums.

The gradient's formula $\nabla F_{N,\sigma}(\boldsymbol{\mu}) = \frac{1}{(\sqrt{2\pi}\sigma)^d \sigma^2} \int_{\boldsymbol{x}\in\mathbb{R}^d} (\boldsymbol{x} - \boldsymbol{\mu}) e^{Nf(\boldsymbol{x})} e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}} d\boldsymbol{x}$ shows that it is a weighted average of $(\boldsymbol{x} - \boldsymbol{\mu}) e^{Nf(\boldsymbol{x})}$ over $\boldsymbol{x} \in \mathbb{R}^d$. Our analysis focuses on the exponential terms $e^{Nf(\boldsymbol{x})}$ and $e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}}$, as their growth and decay rates dominate those of the polynomial factors $(\boldsymbol{x} - \boldsymbol{\mu})$ and σ^{-d-2}, respectively. Intuitively, for a fixed sufficiently large N, the value of $e^{Nf(\boldsymbol{x})}$ in regions distant from the global maximizer $\boldsymbol{x}^*$ become negligible compared to those in $\boldsymbol{x}^*$'s immediate neighborhood $\mathcal{S}_{\boldsymbol{x}^*}$. Meanwhile, when $\boldsymbol{\mu}$ is distant from $\boldsymbol{x}^*$, a decaying σ causes the Gaussian weight $e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}}$ to decay exponentially for points $\boldsymbol{x} \in \mathcal{S}_{\boldsymbol{x}^*}$. Together, these effects make the weighted average—and thus the gradient norm $\|F_{N,\sigma}(\boldsymbol{\mu})\|$—significantly smaller when $\boldsymbol{\mu}$ lies far from $\boldsymbol{x}^*$.

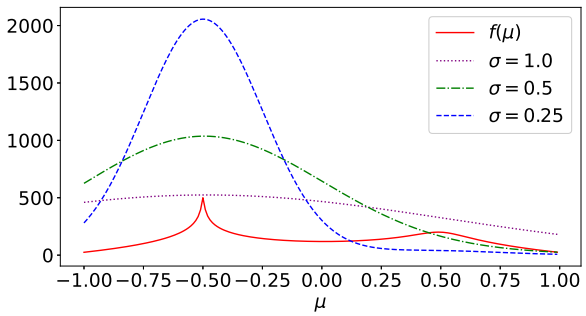

Figure 1: The graphs of an example f and its Gaussian smoothed function $F_{N,\sigma}$, with $N = 1$ and σ assuming different values. In this example, $x^* = \arg\max_{x\in\mathbb{R}} f(x) = -0.5$.

The above discussion motivates an incrementally decreasing schedule for σ as $\{\boldsymbol{\mu}_t\}$ is updated, which is expected to be more efficient than using a fixed σ.

4 Zeroth-Order Single-Loop Power Homotopy (GS-PowerHP)

GS-PowerHP aims to solve the following deterministic optimization problem.

$$\max_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w}). \tag{1}$$

With the pre-selected N, at each iteration t, GS-PowerHP performs a one-step stochastic gradient ascent to solve

$$\max_{\boldsymbol{\mu}} F_{N,\sigma_t}(\boldsymbol{\mu}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma_t^2 I_d)}[f_N(\boldsymbol{x})]. \tag{2}$$

Specifically, the rule for updating the solution candidate is, for all $t \geq 1$,

$$\text{GS-PowerHP}: \quad \begin{aligned} \sigma_{t+1} &= \beta^{t+1}\sigma_0 + b, \\ \boldsymbol{\mu}_{t+1} &= \boldsymbol{\mu}_t + \alpha_t \hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t), \end{aligned} \tag{3}$$

where $\beta \in (0,1)$ is the discounter factor, $b > 0$ is a lower bound for σ_t, $\hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t) := \frac{1}{K}\sum_{k=1}^{K}(\boldsymbol{w}_k - \boldsymbol{\mu}_t) f_N(\boldsymbol{w}_k)$, $\{\boldsymbol{w}_k\}_{k=1}^K$ are independently sampled from the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_t, \sigma_{t+1}^2 I_d)$, and $f_N(\boldsymbol{w}_k)$ is defined as

$$f_N(\boldsymbol{w}) = e^{N f(\boldsymbol{w})}. \tag{4}$$

The values of β, σ_0, b, $\boldsymbol{\mu}_0 \in \mathbb{R}^d$, and K are preselected. Note that $\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu}_t)$ is an unbiased sample estimate of $\sigma^2 \nabla F_{N,\sigma}(\boldsymbol{\mu}_t)$, since

$$\begin{aligned} &\sigma^2 \nabla F_{N,\sigma}(\boldsymbol{\mu}_t) = \sigma^2 \nabla_{\boldsymbol{\mu}} \mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{\mu}_t,\sigma^2 I_d)}[f_N(\boldsymbol{w})] \\ &= \frac{1}{(\sqrt{2\pi}\sigma)^d} \int_{\boldsymbol{x}\in\mathbb{R}^d} (\boldsymbol{w} - \boldsymbol{\mu}_t) f_N(\boldsymbol{w}) e^{-\frac{\|\boldsymbol{w}-\boldsymbol{\mu}_t\|^2}{2\sigma^2}} d\boldsymbol{x} \\ &= \mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{\mu}_t,\sigma^2 I_d)}[(\boldsymbol{w} - \boldsymbol{\mu}_t) f_N(\boldsymbol{w})] = \mathbb{E}[\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu}_t)], \end{aligned} \tag{5}$$

where the interchange of differentiation and integral in the second line can be justified by Lebesgue's dominated convergence theorem.

Remark 1. *To prevent computational overflow potentially caused by large exponents, we can replace $e^{N(f(\boldsymbol{x}_k))}$ with $e^{N(f(\boldsymbol{x}_k - \boldsymbol{\mu}_t))}$ for size control. It does not affect the value of the normalized gradient. Also, we can choose to output the optimal sample $\boldsymbol{x}$ found in the whole process instead of $\boldsymbol{\mu}^*$.*

The only difference between GS-PowerHP and GS-PowerOpt [30] is that the former iteratively decreases the scaling parameter σ.

Algorithm 1 GS-PowerHP for Solving $\max_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$.

Require: Power $N > 0$, initial smoothing radius $\sigma_0 > 0$, the minimum smoothing radius $b > 0$, decay rate $\beta \in (0,1)$, objective f, initial point $\boldsymbol{\mu}_0 \in \mathbb{R}^d$, samples per iteration K, total iterations T, learning rates $\{\alpha_t\}_{t=0}^{T-1}$.

1: **for** $t = 0$ to $T-1$ **do**

2: Set $\sigma_{t+1} = \sigma_0 \beta^{t+1} + b$

3: Sample i.i.d. $\{\boldsymbol{x}_k\}_{k=1}^K$ from $\mathcal{N}(\boldsymbol{\mu}_t, \sigma_{t+1}^2 I_d)$

4: Compute the gradient estimator:

$$\hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t) := \frac{1}{K}\sum_{k=1}^{K} (\boldsymbol{x}_k - \boldsymbol{\mu}_t) e^{N f(\boldsymbol{x}_k)}$$

5: Update:

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \alpha_t \cdot \frac{\hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)}{\|\hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|}$$

6: **Return** $\boldsymbol{\mu}^* = \arg\max_{\boldsymbol{\mu}\in\{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_T\}} f(\boldsymbol{\mu})$

5 Convergence Analysis

In this section, we establish in Corollary 2 that, under mild assumptions (Assumptions 1–4), GS-PowerHP converges in expectation to an arbitrarily small neighborhood of the global maximizer $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$, provided that the power $N > 0$ is sufficiently large. With an appropriately chosen learning rate, the iteration complexity approaches $O(d^2 b^{-2} \varepsilon^{-2})$.

The proof consists of two parts. The first one shows in Corollary 1 that the solutions $\{\boldsymbol{\mu}_t\}$ produced by GS-PowerHP's iteration process (3) converges in expectation to a stationary point of $F_{N,\sigma}$. The second part shows in Subsection 5.3 that, for any given positive δ and b, there exists a threshold $N_{\delta,b}$ such that for any $N > N_{\delta,b}$, all the stationary points of $F_{N,\sigma}$ lies in the $\boldsymbol{x}^*$-neighborhood of $\mathcal{S}_{\boldsymbol{x}^*,\delta} := \{\boldsymbol{\mu} \in \mathbb{R}^d : |\mu_i - x_i^*| \leq \delta, i \in \{1,2,...,d\}\}$, as long as $\sigma > b$. Here, μ_i and x_i^* denote the i^{th} entry of $\boldsymbol{\mu}$ and $\boldsymbol{x}^*$, respectively. Finally, our stated convergence result is implied by the two parts collectively. The full proofs for all our theoretical results, if not provided in this section, can be found in our appendix (Section 8.1).

5.1 Assumptions

We make two assumptions that are standard in optimization researches for machine learning.

Assumption 1. *Assume that the maximization objective $f : \mathbb{R}^d \to \mathbb{R}$ is continuous, $\lim_{\|\boldsymbol{x}\|\to+\infty} f(\boldsymbol{x}) = -\infty$, and has a unique global maximum point $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$.*

Remark 2. *The counterpart, $\lim_{\|\boldsymbol{x}\|\to+\infty} f(\boldsymbol{x}) = +\infty$ for $\min_{\boldsymbol{x}} f(\boldsymbol{x})$ is the commonly seen coercivity assumption.*

Assumption 2. *Assume that f is Lipschitz in $\mathbb{R}^d$ with its Lipschitz constant equal to L_f.*

Assumption 3. $\alpha_t > 0$, $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$, *and* $\sum_{t=0}^{\infty} \alpha_t = \infty$.

5.2 Convergence of μ_t to a stationary point of $F_{N,\sigma}$

The differentiability and the Lipschitz constant of the Gaussian-smoothed objective $F_{N,\sigma}$ is given in the following lemma.

Lemma 1. *Under Assumption 1, given any $N > 0$ and $\sigma > 0$, (1) both $F_{N,\sigma}(\boldsymbol{\mu})$ and $\nabla F_{N,\sigma}(\boldsymbol{\mu})$ are well-defined and Lipschitz in $\mathbb{R}^d$; (2) The Lipschitz constant for $\nabla F_{N,\sigma}$ is $L = 2d\sigma^{-2}e^{Nf(\boldsymbol{x}^*)}$, (3) $F_{N,\sigma}$ has at least one global maximum $\boldsymbol{\mu}^* \in \mathbb{R}^d$, and (4) $\|\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu})\|^2 \le G = d\sigma^2 e^{2Nf(\boldsymbol{x}^*)}$ for all $\boldsymbol{\mu} \in \mathbb{R}^d$.*

With the Lipschitz constant given in Lemma 1, we bound the expected sum of gradients, $\sum_{t=0}^{T-1} \alpha_t \sigma_{t+1}^2 \mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|^2]$. Since $\sum_{t=0}^{T-1} \alpha_t$ approaches ∞ as $T \to \infty$ by Assumption 3, the boundedness implies $\lim_{T\to\infty} \mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|^2] = 0$.

Theorem 1. *For any deterministic $\boldsymbol{\mu}_0 \in \mathbb{R}^d$, $N, b, \sigma_0 > 0$, and positive integer T, let $\{(\boldsymbol{\mu}_t, \sigma_t)\}_{t=1}^{T}$ be generated by the GS-PowerHP iterations (3). Under Assumption 1 and 2, we have*

$$\sum_{t=0}^{T-1} \alpha_t \sigma_{t+1}^2 \mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|^2] \le e^{Nf(\boldsymbol{x}^*)} - F_{N,\sigma_0}(\boldsymbol{\mu}_0) + 2d^2 e^{3Nf(\boldsymbol{x}^*)} \sum_{t=0}^{T-1} \alpha_t^2 + \sigma_0 C_{N,d,f}(1-\beta) \sum_{t=0}^{T-1} \beta^t,$$

where $C_{N,d,f} = \sqrt{2}\frac{\Gamma((d+1)/2)}{\Gamma(d/2)} N e^{Nf(\boldsymbol{x}^*)} L_f$.

Proof. (Sketch) We bound $\alpha_t \sigma_{t+1}^2 \mathbb{E}\big[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|^2\big]$ by applying the mean value theorem to the smoothed function update and leveraging Lipschitz properties of $F_{N,\sigma}$. This yields an inequality in expectation, augmented by a bounded error term from the zeroth-order gradient estimator and a controlled drift due to decaying smoothing radius σ_t. Summing over T iterations and using the boundedness $F_{N,\sigma_T} \le e^{Nf(\boldsymbol{x}^*)}$, we obtain the finite-sum bound in Theorem 1. □

With the bound in Theorem 1, we derive the iteration complexity for $\boldsymbol{\mu}_t$ converging to stationary points of $F_{N,\sigma}$.

Corollary 1. *Assume the conditions in Theorem 1 and Assumption 3. Let $\alpha_t = (t+1)^{-(1/2+\gamma)}$ for some $\gamma \in (0, 1/2)$. For any $\varepsilon \in (0,1)$, after $T > (C_2 C_1 d^2 \varepsilon^{-1})^{2/(1-2\gamma)} = O_N((d^2\varepsilon^{-1}b^{-2})^{2/(1-2\gamma)})$ times of parameter updating, we have*

$$\min_{t\in\{0,1,2,\ldots,T\}} \mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|^2] < \varepsilon,$$

where $C_0 := f_N(\boldsymbol{x}^*) - F_{N,\sigma_0}(\boldsymbol{\mu}_0) + 2f_N^3(\boldsymbol{x}^*)\sum_{t=1}^{\infty} t^{-(1+2\gamma)} + \sqrt{2}\sigma_0 N f_N(\boldsymbol{x}^*) L_f$, $C_1 = b^{-2}C_0(1-2\gamma)$, and $C_2 = \max\{1, 1/C_1\}$. Furthermore, the dependence of the O_N-factor on N can be removed under the additional assumption of $f(\boldsymbol{x}) \le 0$ for all $\boldsymbol{x} \in \mathbb{R}^d$.

Remark 3. *The additional assumption of $f(\boldsymbol{x}) \le 0$ holds true for a wide range of optimization problems in machine learning. For example, the loss function L for training a regression model (including neural nets) in supervised learning is commonly defined as the sum of squared prediction errors, which corresponds to a non-positive objective $-L$ if the loss minization is converted to a maximization problem.*

5.3 Convergence to the Global Maximum of f

In this section, under an additional mild assumption (i.e., Assumption 4), we show that for any given $\delta >$

0 and $\sigma > 0$, there exists a threshold $N_{\delta,\sigma} > 0$ such that, whenever $N > N_{\delta,\sigma}$ all the stationary points of $F_{N_\delta,\sigma}$ lies in the δ-neighborhood $\mathcal{S}_{\boldsymbol{x}^*,\delta}$, as long as $\sigma > b$.

Theorem 2. *Suppose Assumption 1 holds. Given any positive numbers N, σ, δ, and M such that $\delta < M$, there exists $N_{\delta,\sigma,M} > 0$ such that whenever $N > N_{\delta,\sigma,M}$ the following statement holds true: if $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\|\boldsymbol{\mu}\| < M$, then for any $i \in \{1, 2, ..., d\}$*

- $\mu_i > x^* + \delta \Rightarrow \frac{\partial F_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i} < 0$*; and*
- $\mu_i < x^* - \delta \Rightarrow \frac{\partial F_{N,\sigma}(\boldsymbol{\mu})}{\partial \mu_i} > 0$,

where μ_i and x_i^* denotes the i^{th} entry of $\boldsymbol{\mu}$ and $\boldsymbol{x}^*$, respectively.

Proof. Despite of some small difference between the assumptions for this theorem and those for [30, Theorem 2.1], the two theorems share the same proof, which can be found in their appendix. □

Assumption 4. *Given any $\delta, b > 0$ and any $\boldsymbol{\mu}_0 \in \mathbb{R}^d$, assume that there is some $N_{\delta,b,\boldsymbol{\mu}_0} > 0$ such that for all $N > N_{\delta,b,\boldsymbol{\mu}_0}$, $\{\boldsymbol{\mu}_t\}$ generated by the iteration process (3) are bounded by some constant $M_{(\delta,\sigma,\boldsymbol{\mu}_0)}$.*

Remark 4. *Assumption 4 is justified for large N: Theorem 2 ensures a persistent inward drift toward $\mathcal{S}_{\boldsymbol{x}^*,\delta}$ when $\boldsymbol{\mu}_t$ is far away, which—coupled with decreasing σ_t—prevents divergence and ensures bounded trajectories.*

Corollary 2. *Suppose Assumption 1–4 hold. For any $\delta > 0$, $b > 0$, and $\boldsymbol{\mu}_0 \in \mathbb{R}^d$, let $N_{\delta,b,\boldsymbol{\mu}_0}$ and $M_{(\delta,b,\boldsymbol{\mu}_0)}$ be the threshold and $\|\boldsymbol{\mu}\|$-bound in Assumption 4, respectively. Let $N_{\delta,b,M_{(\delta,b,\boldsymbol{\mu}_0)}}$ be the threshold stated in Theorem 2. Then, whenever $N > \max\{N_{\delta,b,\boldsymbol{\mu}_0}, N_{\delta,b,M_{(\delta,b,\boldsymbol{\mu}_0)}}\}$, $\{\boldsymbol{\mu}_t\}$ generated by GS-PowerHP (3) converges in expectation to $\mathcal{S}_{\boldsymbol{x}^*,\delta}$ with the iteration complexity of $O((d^2\varepsilon^{-1}b^{-2})^{2/(1-2\gamma)})$, given the learning rate of $\alpha_t = (t+1)^{-(1/2+\gamma)}$ for some $\gamma \in (0, 1/2)$.*

Proof. Corollary 1 implies that $\boldsymbol{\mu}_t$ converges in expectation to a stationary point $\boldsymbol{\mu}^*$ of $F_{N,b}(\boldsymbol{\mu})$. Since $\|\boldsymbol{\mu}_t\| \le M_{(\delta,b,\boldsymbol{\mu}_0)}$, $\|\boldsymbol{\mu}^*\| \le M_{(\delta,b,\boldsymbol{\mu}_0)}$. Also, from Theorem 2, all the stationary points of $F_{N,b}$ within the region of $\|\boldsymbol{\mu}\| \le M_{(\delta,b,\boldsymbol{\mu}_0)}$ lie in $\mathcal{S}_{\boldsymbol{x}^*;\delta}$. This completes the proof. □

6 Experiments

In this section, we test the capacity of GS-PowerHP with extensive experiments. The compared algorithms can be divided to three kinds, the smoothing-based zeroth-order optimization methods of ZOSGD ([11]), ZO-AdaMM ([6]), and GS-PowerOpt[1] ([30]), the homotopy methods of standard homotopy and ZOSLGH ([17]), the state-of-the-art evolutionary algorithm of CMA-ES ([13]), and the powerful method of square attack ([1]) in the field of adversarial image attacks. A brief introduction to some of these methods can be found in Appendix D in [30].

6.1 Advantage of the σ-Decreasing Mechanism

In this section, we illustrates the effects of the σ-decreasing mechanism with the maximization objective

$$f(\boldsymbol{x}) = -\log(\|\boldsymbol{x}-\boldsymbol{m}_1\|^2+10^{-5})-\log(\|\boldsymbol{x}-\boldsymbol{m}_2\|^2+10^{-2}), \tag{6}$$

where all entries of $\boldsymbol{m}_1 \in \mathbb{R}^d$ equal -0.5 and all entries of $\boldsymbol{m}_2 \in \mathbb{R}^d$ equal 0.5. Note that $\boldsymbol{x}^* = \boldsymbol{m}_1$ for this objective.

The illustration is done through comparing the performances of GS-PowerOpt from [30] and GS-PowerHP. The only difference between these two algorithms is, the former is characterized by a fixed σ value while the latter features with an iterative σ-decreasing mechanism. For each σ from a pre-selected set $\{0.1, 0.5, 1.0, 2.0, 3.0\}$, we perform 100 trials of GS-PowerOpt to maximize $f(\boldsymbol{x})$ in (6). Then, in Table 1, we report the average of $\{f(\boldsymbol{\mu}_n^*)\}_{n=1}^{100}$ and the average of $\{\text{MSE}(\boldsymbol{\mu}_n^*)\}_{n=1}^{100}$, where $\text{MSE}(\boldsymbol{\mu}^*)$ denotes the L_2 norm $\|\boldsymbol{\mu}^* - \boldsymbol{x}^*\|^2/d$. For GS-PowerHP, we perform the same experiment,

[1] GS-PowerOpt has two formats. One is PGS, which transforms the objective f to f^N. The other one is EPGS, which transforms f to e^{Nf}. In this work, we refer GS-PowerOpt as EPGS.

Table 1: The performances of GS-PowerOpt and GS-PowerHP on optimizing $f(\boldsymbol{x})$ in (6). We set $N = 1$ and $T = 1,000$. The reported σ value for GS-PowerHP is the initial σ value (i.e., $\boldsymbol{\mu}_0$).

Algo.	$d=3$			$d=5$		
	σ	$f(\boldsymbol{\mu}^*)$	MSE	σ	$f(\boldsymbol{\mu}^*)$	MSE
GS-PowerOpt	3.0	1.18	0.07	3.0	−0.28	0.12
	2.0	1.33	0.05	2.0	−0.24	0.11
	1.0	3.37	0.01	1.0	0.24	0.06
	0.5	5.47	0.26	0.5	2.39	0.31
	0.1	4.82	0.39	0.1	3.20	0.29
Our Algo.	3.0	**7.68**	**0.00**	0.1	**4.20**	**0.03**

Table 2: Ackley Maximization. $\bar{t}^*$, $\bar{\boldsymbol{\mu}}^*$, and $\bar{f}(\boldsymbol{\mu}^*)$ are defined in the second paragraph in Section 6.2. All numbers are rounded to keep at most 3 decimal places. Note that $\boldsymbol{x}^* = (0, 0)$.

Algorithm	$\bar{t}^*$	$\bar{\boldsymbol{\mu}}^*$	$\bar{f}(\boldsymbol{\mu}^*)$
CMA-ES	115	$(0.0, 0.0)$	22.718
Our Algo.	140	$(0.001, 0.002)$	22.683
GS-PowerOpt	145	$(0.002, -0.001)$	22.682
ZOSLGHr	130	$(-0.001, 0.001)$	22.622
ZOAdaMM	153	$(0.001, -0.003)$	22.615
ZOSGD	172	$(-0.001, 0.0)$	22.608
ZOSLGHd	131	$(0.003, -0.001)$	22.585
STD-Htp	195	$(0.983, 0.911)$	17.561

with $\sigma_0 = 3$, $b = 0$, and the decaying factor β being the solution to $3\beta^{1000} = 0.1$.

From the table, we see that GS-PowerHP is able to produce a solution $\boldsymbol{\mu}^*$ that is closer to $\boldsymbol{x}^*$ than all the solutions found by GS-PowerOpt, indicating the advantage of the σ-decreasing mechanism of GS-PowerHP over the fixed-σ method of GS-PowerOpt.

Note that although $\sigma = 0.1$ is associated with a comparatively large MSE, the corresponding average fitness is good. This is because GS-PowerOpt's performance is volatile with this σ value. Some trials produces $\boldsymbol{\mu}^*$ close to $\boldsymbol{x}^*$ and in turn lead to a high fitness.

6.2 Performances on Optimizing Benchmark Objectives

We test GS-PowHP on maximizing the 2D-version of two popular benchmark objectives for non-convex optimization methods, the Ackley and Rosenbrock functions, whose functional forms are

$$\begin{aligned} &\text{Ackley: } f(x,y) = 20e^{-\frac{\sqrt{0.5(x^2+y^2)}}{5}} + e^{\frac{\cos(2\pi x)+\cos(2\pi y)}{2}}; \\ &\text{Rosenbrock: } f(x,y) = -100(y-x^2)^2 - (1-x)^2. \end{aligned} \tag{7}$$

These two functions are non-trivial to maximize because Ackley has a large number of local maximums and the global maximum of Rosenbrock is surrounded by a plateau. Their graphs can be found in [30, Figure 3].

We follow the same procedures of the experiments in [30, Section 5.2 & 5.3] to test and compare GS-PowHP's performance with other algorithms. Specifically, for each algorithm and each candidate set of hyperparameters, we perform 100 trials to optimize the objective function. Let $\boldsymbol{\mu}_n^*$ denote the candidate solution in the n^{th} trial closest to the global maximum point $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} f(\boldsymbol{x})$ and let t_n^* denote the number of solution updates taken to achieve $\boldsymbol{\mu}_n^*$. Then, for each algorithm, we select the hyperparameter set[2] that produces the smallest $\bar{\boldsymbol{\mu}}^* := \sum_{n=1}^{100} \|\boldsymbol{\mu}_n^* - \boldsymbol{x}^*\|^2/n$, and report $\bar{t}^* := \sum_{n=1}^{100} t_n^*)/100$, $\bar{f}(\boldsymbol{\mu}^*) := \sum_{n=1}^{100} f(\boldsymbol{\mu}_n^*)/100$, and $\bar{t}^* := \sum_{n=1}^{100} t_n^*/100$ in Table 2 or Table 3.

6.3 Least-likely Targeted Black-Box Image Adversarial Attacks (IAA)

Given an image classifier $\mathcal{G}(\cdot)$ that outputs a probability distribution and an image $\boldsymbol{a}$, let $\mathcal{T}$ denotes the least-likely class according to $\mathcal{G}(\boldsymbol{a})$. The goal of

[2]The selected hyper-parameters for GS-PowHP are $(N, \sigma_0, b, \text{learning rate}) = (2, 1.0, 0, 0.1)$ for Ackley and $(N, \sigma_0, b, \text{learning rate}) = (3, 1.0, 0, 0.1)$ for Rosenbrock. The hyper-parameters selected for other methods are the same as those in [30, Table 6-7, Appendix F].

Table 3: Rosenbrock Maximization. $\bar{t}^*$, $\bar{\boldsymbol{\mu}}^*$, and $\bar{f}(\boldsymbol{\mu}^*)$ are defined in the second paragraph in Section 6.2. Note that $\boldsymbol{x}^* = (1, 1)$.

Algorithm	$\bar{t}^*$	$\boldsymbol{\mu}^*$	$f(\boldsymbol{\mu}^*)$
CMA-ES	71	$(1.0, 1.0)$.	0.0
Our Algo.	608.	$(0.999, 1.001)$	-0.009
GS-PowerOpt	487.	$(0.998, 1.001)$	-0.017
STD-Htp	663	$(0.916, 0.889)$	-1.476
ZOAdaMM	872	$(0.004, 0.617)$.	-39.029
ZOSLGHr	133	$(0.1, 0.976)$.	-95.685
ZOSGD	45	$(0.278, 1.201)$.	-125.70
ZOSLGHd	475	$(-0.45, 1.995)$.	-139.143

the least-likely targeted black-box IAA is to find a minimal perturbation $\boldsymbol{x}$ such that $\mathcal{G}(\boldsymbol{x} + \boldsymbol{a})$ assigns the largest probability to $\mathcal{T}$.

This problem can be formulated as an optimization problem. Suppose the pixels in the image $\boldsymbol{a}$ are normalized to $[-1, 1]$ and the perturbation is transformed to $\boldsymbol{y} = \tanh(\boldsymbol{x})$ to restrict size. We define the loss in the following way, which is similar to the one used in [2].

$$L(\boldsymbol{x}) := \max(\max_{i \neq \mathcal{T}} \mathcal{G}(\boldsymbol{a}+\boldsymbol{y})_i - \mathcal{G}(\boldsymbol{a}+\boldsymbol{y})_{\mathcal{T}},\ \kappa) + \lambda \|\boldsymbol{y}\|, \tag{8}$$

where $\mathcal{G}(\boldsymbol{a} + \boldsymbol{y})_i$ denotes the probability assigned to class i, and $\kappa < 0$ is a hyper-parameter used to prevent excessive effort on attack confidence. The hyper-parameter κ, which we set to 0.001, helps to reduce perturbation size for successful attacks. Here, an attack is called *successful* if it produces a successful perturbation, while a perturbation $\boldsymbol{x}$ is called successful if $\max_{i \neq \mathcal{T}} \mathcal{G}(\boldsymbol{a} + \boldsymbol{y})_i - \mathcal{G}(\boldsymbol{a} + \boldsymbol{y})_{\mathcal{T}} < \kappa$.

To test its capacity for large-dimension problems, we apply GS-PowerPH and other compared algorithms to solve

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

on images randomly selected from three popular image sets, the MNIST hand-written figures [19], CIFAR-10 [18], and ImageNet [8]. The images from MNIST, CIFAR-10, and ImageNet have $28 \times 28 = 784$, $32 \times 32 \times 3 = 3,072$, and $224 \times 224 \times 3 = 150,528$ pixels, respectively. These values correspond to the dimensionality d of the input space for each dataset. We normalize the pixel range to $[-1, 1]$ for all involved images before performing experiments.

For each algorithm and each dataset, we randomly pick 100 images to attack. If the attack on the m^{th} image $\boldsymbol{a_m}$ is successful, let $\boldsymbol{\mu}_m^*$ denote the one with the least L_2 norm among all the successful perturbations produced during the attack process, let T_m denote the number of solution updates (i.e., number of iterations) to reach $\boldsymbol{\mu}_m^*$, and let $R_m^2 := 1 - \frac{\sum_{j=1}^d (\mu_j^* - a_j)}{(a_j - \bar{a})^2}$, where μ_j^* and a_j^* denote the j^{th} entry of $\boldsymbol{\mu}_m^*$ and $\boldsymbol{a_m}$, respectively.

For each algorithm, let $\mathcal{S}$ denote the set of indices corresponding to successful attacks out of the 100 performed, and let $|\mathcal{S}|$ denote the number of successful attacks. The performance statistics we will report are the sucess rate SR:$= |\mathcal{S}|/100$, the average R^2 score $\overline{R^2} := \sum_{m \in \mathcal{S}} R_m^2 / |\mathcal{S}|$, the average perturbation L_2 norm $\overline{\|\boldsymbol{\mu}^*\|_2} := \sum_{m \in \mathcal{S}} \|\boldsymbol{\mu}_m^*\| / |\mathcal{S}|$, and the average number of iterations taken to reach the optimal $\overline{T} := \sum_{m \in \mathcal{S}} T_m / |\mathcal{S}|$.

The hyperparameters for the compared algorithms (except square attack) are set to the values specified in [30, Appendix, Table 7-8] for the MNIST and CIFAR-10 experiments, as the experimental procedures are identical to ours and the loss functions are similar. The hyperparameters used in the ImageNet experiments are reported in the Appendix. These values were selected through experimental trials on a small set of 10 images. The learning rate α used for GS-PowerPH is fixed for convenience. For each solution update, we use $K = 10$ random samples. This also corresponds to 11 inquiries to the classifier (10 for random samples and 1 for the current solution value $\boldsymbol{\mu}_t$). The Square Attack employs a different attack mechanism, and we set its maximum iteration number to ensure it uses the same total number of queries as the other algorithms. To maintain a consistent comparison scale for the average query count ($\overline{T}$), the raw value obtained for the Square Attack is normalized by dividing it by 11. This aligns its measure with that of the other algorithms.

The classifiers for MNIST and CIFAR-10 are neu-

Table 4: Per-image Attack on 100 Images from MNIST. For each image attack, we set the initial perturbation $\boldsymbol{\mu}_0 = \mathbf{0}$, and $T_{total} = 2,500$ (it is $25,000$ for Square Attack). The hyper-parameters for GS-PowerHP are selected as $N = 0.5, \sigma_0 = 0.05$, $\alpha = 0.07$, and $\beta = 0.999$. Square A. denotes square attack, which uses one sample in each iteration, so we report

Algorithm	SR	$\bar{R}^2$	$\overline{\|\boldsymbol{\mu}^*\|_2}$	$\bar{T}$
CMAES	100%	93%(3%)	4.37(0.98)	2485(22)
GS-PowerOpt	100%	92%(3%)	4.77(0.91)	1189(462)
Our Algo.	**100%**	**92%**(3%)	**4.79**(0.92)	**1166**(277)
ZOSGD	100%	90%(5%)	5.25(0.88)	2318(460)
ZOSLGHd	100%	86%(4%)	6.09(0.97)	2461(85)
ZOSLGHr	100%	83%(5%)	6.74(1.03)	2374(532)
ZOAdaMM	100%	71%(9%)	8.84(1.31)	92(42)
STD-Htp	48%	68%(11%)	8.58(1.02)	917(589)
Square A.	86%	−50%(50%)	19.77(1.35)	34(200)

Table 5: Per-image Attack on 100 Images from CIFAR-10. For each image attack, we set $\boldsymbol{\mu}_0 = \mathbf{0}$ and $T_{total} = 1,500$ ($15,000$ for Square Attack). The hyper-parameters for GS-PowerHP are selected as $N = 0.1, \sigma_0 = 0.05$, $\alpha = 0.08$, and $\beta = 0.999$.

Algorithm	SR	$\bar{R}^2$	$\overline{\|\boldsymbol{\mu}^*\|_2}$	$\bar{T}$
ZOSLGHd	100%	99%(1%)	1.76(0.32)	1237(467)
Our Algo.	**100%**	**99%**(1%)	**2.06**(0.38)	**538**(179)
ZOSLGHr	100%	98%(2%)	2.63(0.69)	418(362)
GS-PowerOpt	94%	97%(3%)	3.20(0.60)	819(274)
CMAES	100%	70%(29%)	10.65(2.26)	195(436)
ZOAdaMM	100%	62%(31%)	12.16(1.99)	60(32)
ZOSGD	61%	99.5%(1%)	1.20(0.18)	709(317)
STD-Htp	54%	88%(11%)	7.66(1.25)	588(317)
Square A.	80%	−48%(1.83)	24.79(7.79)	17(72)

ral networks trained using the distillation technique, for increasing prediction robustness ([2]). The test scores is 98% for the MNIST classifer and 86% for the CIFAR-10 classifier. For ImageNet images, we use a pretrained ResNet-50 ([15, 29]) as the classifier, which is a common choice in this research field ([1]).

MNIST The experimental results on MNIST are reported in Table 4. They show that GS-PowerHP achieves a 100% success rate. Furthermore, its perturbation size is among the three smallest and is significantly smaller than those of the other algorithms.

CIFAR-10 Table 5 documents the experiments for CIFAR-10. Our algorithm achieves a 100% success rate and its average perturbation size is inferior to that of ZOSLGHd but significantly better than all other algorithms.

IMAGENET The results on attacking images from the ImageNet are reported in Table 6. We do not test CMA-ES since it requires too much more computing resource than other algorithms for this huge-dimension task. [3] The results show that GS-PowerHP significantly outperforms other algorithms if both the success rate and perturbation size are taken into account. Figure 2 plots example perturbed images generated by GS-PowerHP.

7 Conclusion

In this work, we proposed a new zeroth-order method, namely GS-PowerHP, for general non-convex optimization problems. It is characterized by a power-transformed Gaussian smoothed objective and an incrementally decreasing scaling parameter σ, which produces theoretical and empirical advantage over other smoothing-based methods. Experimental results show that our method outperforms other compared algorithms overall, including the extremely-high dimension task of image adversarial attacks on ImageNet.

In this work, we introduce GS-PowerHP, a novel zeroth-order method for general non-convex opti-

[3] According to the reported Python message, CMAES requires a RAM of 165 GB each iteration, which is beyond the capacity of our computer.

Table 6: Per-image Attack on 100 Images from ImageNet. For each image attack, we set $\boldsymbol{\mu}_0 = \mathbf{0}$ and $T_{total} = 3,000$. The hyper-parameters for GS-PowerHP are selected as $N = 8, \sigma_0 = 0.1$, $\alpha = 1.0$, and $\beta = 0.999$.

Algorithm	SR	$\bar{R}^2$	$\overline{\|\boldsymbol{\mu}^*\|}$	$\bar{T}$
Our Algo.	**78**%	**95**%(7%)	**35.8**(6.5)	**1389**(542)
ZOSLGHd	67%	97%(4%)	28.1(3.3)	2330(933)
GS-PowerOpt	47%	95%(3%)	36.5(2.8)	1780(644)
ZOSLGHr	43%	69%(16%)	92.8(3.4)	2167(801)
ZOAdaMM	61%	61%(27%)	104.9(13.6)	2896(454)
ZOSGD	80%	−25%(1.46)	180(22)	3000(0)
Square A.	98%	6%(98%)	169.6(31.3)	106(102)

mization. By combining a power-transformed Gaussian smoothed objective with an incrementally decreasing scaling parameter σ, our approach delivers superior theoretical convergence guarantees and empirical performance compared to existing smoothing-based methods. Extensive experiments demonstrate that GS-PowerHP consistently outperforms competing algorithms, even in extremely high-dimensional settings such as adversarial image attacks on ImageNet.

(a) Clean Image A from ImageNet.

(b) Perterbed Image GS-PowerHP

(c) Clean Image B from ImageNet

(d) Perturbed Image GS-PowerHP

Figure 2: Attacks by GS-PowerHP on two randomly selected images from ImageNet.

References

[1] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *Computer Vision – ECCV 2020*, pages 484–501, Cham, 2020. Springer International Publishing.

[2] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[3] J. Chen, H. Chen, B. Gu, and H. Deng. Fine-grained theoretical analysis of federated zeroth-order optimization. *Advances in Neural Information Processing Systems*, 36:54496–54508, 2023.

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

[5] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

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

[7] F. Croce, M. Andriushchenko, N. D. Singh, N. Flammarion, and M. Hein. Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 6437–6445, 2022.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

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

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

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

[12] C. Guo, J. Gardner, Y. You, A. G. Wilson, and K. Weinberger. Simple black-box adversarial attacks. In *International conference on machine learning*, pages 2484–2493. PMLR, 2019.

[13] N. Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

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

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[16] A. Ilyas, L. Engstrom, and A. Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *International Conference on Learning Representations*, pages 2484–2493. PMLR, 2019.

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

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

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

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

[21] L. J. V. Miranda. PySwarms, a research-toolkit for particle swarm optimization in python. *Journal of Open Source Software*, 3, 2018.

[22] H. Mobahi and J. F. III. A theoretical analysis of optimization by gaussian continuation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, pages 1205–1211, 2015.

[23] H. Mohaghegh Dolatabadi, S. Erfani, and C. Leckie. Advflow: Inconspicuous black-box adversarial attacks using normalizing flows. *Advances in Neural Information Processing Systems*, 33:15871–15884, 2020.

[24] B. Mukhoty, V. Bojkovic, W. de Vazelhes, X. Zhao, G. De Masi, H. Xiong, and B. Gu. Direct training of snn using local zeroth order method. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 18994–19014. Curran Associates, Inc., 2023.

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

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

[27] H. Sawada, K. Aoyama, and Y. Hikima. Natural perturbations for black-box training of neural networks by zeroth-order optimization. In A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 53063–53079. PMLR, 13–19 Jul 2025.

[28] Y. Shu, Q. Zhang, K. He, and Z. Dai. Refining adaptive zeroth-order optimization at ease. In A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 55403–55426. PMLR, 13–19 Jul 2025.

[29] TensorFlow Team. Keras documentation: Resnet50, 2024.

[30] C. Xu. Global optimization with a power-transformed objective and Gaussian smoothing. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 69189–69216. PMLR, 2025.

[31] Y. Zhang, P. Li, J. Hong, J. Li, Y. Zhang, W. Zheng, P.-Y. Chen, J. D. Lee, W. Yin, M. Hong, Z. Wang, S. Liu, and T. Chen. Revisiting zeroth-order optimization for memory-efficient LLM fine-tuning: A benchmark. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 59173–59190. PMLR, 21–27 Jul 2024.

8 Appendix

8.1 Proof Details

Lemma 1. *Under Assumption 1, given any $N > 0$ and $\sigma > 0$, (1) both $F_{N,\sigma}(\boldsymbol{\mu})$ and $\nabla F_{N,\sigma}(\boldsymbol{\mu})$ are well-defined and Lipschitz in $\mathbb{R}^d$; (2) The Lipschitz constant for $\nabla F_{N,\sigma}$ is $L = 2d\sigma^{-2}e^{Nf(\mathbf{x}^*)}$, (3) $F_{N,\sigma}$ has at least one global maximum $\boldsymbol{\mu}^* \in \mathbb{R}^d$, and (4) $\|\hat{\nabla} F_{N,\sigma}(\boldsymbol{\mu})\|^2 \leq G = d\sigma^2 e^{2Nf(\mathbf{x}^*)}$ for all $\boldsymbol{\mu} \in \mathbb{R}^d$.*

Proof. The proofs for Property (1), (2), and (4) are the same as those to their counterpart in [30], although in Assumption 1 we assume that the objective f is defined over the whole space $\mathbb{R}^d$. Specifically, the proof for (1) is the same as that for [30, Lemma 3.3]. For (2), the proof that $F_{N,\sigma}(\boldsymbol{\mu})$ is Lipschitz is the same as that for Point 1 in the proof to [30, Proposition 2.3]. The proof for the Lipschitz property of $\nabla F_{N,\sigma}$ and $L = 2d\sigma^{-2}e^{Nf(\mathbf{x})}$ is the same as the proof to [30, Lemma 3.5]. The proof to (4) is the same as that to [30, Lemma 3.6].

Property (3) shares the same proof as [30, Proposition 2.3], except the part of showing $\lim_{\|\boldsymbol{\mu}\|\to+\infty} F_{N,\sigma} = 0$, which is given below. Specifically, for any $\epsilon > 0$, let $M > 0$ be such that whenever

$\|\boldsymbol{x}\| > M$, $f(\boldsymbol{x}) < (\ln\epsilon)/N$. Then,

$$\begin{aligned}
F_{N,\sigma}(\boldsymbol{\mu}) =& \frac{1}{(\sqrt{2\pi}\sigma)^d}\int_{\boldsymbol{x}\in\mathbb{R}^d} e^{Nf(\boldsymbol{x})}e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}}d\boldsymbol{x}\\
=& \frac{1}{(\sqrt{2\pi}\sigma)^d}\int_{\|\boldsymbol{x}\|\le M} e^{Nf(\boldsymbol{x})}e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}}d\boldsymbol{x}\\
&+ \frac{1}{(\sqrt{2\pi}\sigma)^d}\int_{\|\boldsymbol{x}\|> M} e^{Nf(\boldsymbol{x})}e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{2\sigma^2}}d\boldsymbol{x}\\
\le& \frac{1}{(\sqrt{2\pi}\sigma)^d}\int_{\|\boldsymbol{x}\|\le M} e^{Nf(\boldsymbol{x})}e^{-\frac{(\|\boldsymbol{\mu}\|-M)^2}{2\sigma^2}}d\boldsymbol{x} + \epsilon\\
\le& 2\epsilon, \quad \text{when } \|\boldsymbol{\mu}\| \text{ is larger than some } b_\epsilon > 0.
\end{aligned}$$

In sum, for any $\epsilon > 0$, there exists $b_\epsilon > 0$ such that whenever $\|\boldsymbol{\mu}\| > b_\epsilon$, $F(\boldsymbol{\mu}) < 2\epsilon$. Hence, $\lim_{\|\boldsymbol{\mu}\|\to+\infty} F_{N,\sigma}(\boldsymbol{\mu}) = 0$. □

Theorem 1: *For any deterministic $\boldsymbol{\mu}_0 \in \mathbb{R}^d$, $N, b, \sigma_0 > 0$, and positive integer T, let $\{(\boldsymbol{\mu}_t, \sigma_t)\}_{t=1}^T$ be generated by the GS-PowerHP iterations (3). Under Assumption 1 and 2, we have*

$$\begin{aligned}
\sum_{t=0}^{T-1}\alpha_t\sigma_{t+1}^2\mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|^2] \le& e^{Nf(\boldsymbol{x}^*)} - F_{N,\sigma_0}(\boldsymbol{\mu}_0)\\
&+ 2d^2e^{3Nf(\boldsymbol{x}^*)}\sum_{t=0}^{T-1}\alpha_t^2 + \sigma_0 C_{N,d,f}(1-\beta)\sum_{t=0}^{T-1}\beta^t,
\end{aligned}$$

where $C_{N,d,f} = \sqrt{2}\frac{\Gamma((d+1)/2)}{\Gamma(d/2)}Ne^{Nf(\boldsymbol{x}^*)}L_f$.

Proof. From the Gradient Mean Value Theorem, there exists $\boldsymbol{\nu}_t \in \mathbb{R}^d$ on the line segment joining $\boldsymbol{\mu}_{t+1}$ and $\boldsymbol{\mu}_t$ such that

$$\begin{aligned}
F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_{t+1}) =& F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\\
&+ (\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\nu}_t))'(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t)\\
=& F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t) + (\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))'(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t)\\
&+ (\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\nu}_t) - \nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))'(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t)\\
=& F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t) + \alpha_t(\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))'(\hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))\\
&- (\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\nu}_t) - \nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))'(\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t)\\
\ge& F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t) + \alpha_t(\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))'(\hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))\\
&- L\|\boldsymbol{v}_t - \boldsymbol{\mu}_t\|\cdot\|\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t\|\\
\ge& F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t) + \alpha_t(\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))'(\hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))\\
&- L\|\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t\|^2,\\
=& F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t) + \alpha_t(\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))'(\hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))\\
&- \alpha_t^2 L\|\hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|^2,
\end{aligned}$$

where $'$ denotes vector transpose, the third equality is from (3), and the first inequality is because of the Cauchy-Schwarz inequality and (2) in Lemma 1. Taking the expectation of the left-end and right-end gives

$$\begin{aligned}
&\mathbb{E}[F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_{t+1})] \ge \mathbb{E}[F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)] +\\
&\quad \alpha_t\sigma_{t+1}^2\mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))\|^2] - \alpha_t^2 L\mathbb{E}[\|\hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|^2]\\
&\ge \mathbb{E}[F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)] + \alpha_t\sigma_{t+1}^2\mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))\|^2]\\
&\quad - \alpha_t^2 LG, \ \text{ by Lemma 1(4)},\\
&= \mathbb{E}[F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)] + \alpha_t\sigma_{t+1}^2\mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))\|^2]\\
&\quad - \alpha_t^2 2d^2e^{3Nf(\boldsymbol{x}^*)}, \quad \text{by Lemma 1},
\end{aligned}$$

where the first inequality holds because

$$\begin{aligned}
&\mathbb{E}[(\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))'(\hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))]\\
=& \mathbb{E}\left[\mathbb{E}[(\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))'(\hat{\nabla} F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))|\boldsymbol{\mu}_t]\right]\\
=& {}^{\text{by (5)}}\sigma_{t+1}^2\mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))\|^2].
\end{aligned}$$

In sum, we arrived at an inequality which will be used later:

$$\begin{aligned}
&\mathbb{E}[F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_{t+1})] \ge \mathbb{E}[F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)]\\
&+ \alpha_t\sigma_{t+1}^2\mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))\|^2] - \alpha_t^2 2d^2e^{3Nf(\boldsymbol{x}^*)}.
\end{aligned} \tag{9}$$

Next, we derive the relation between $F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)$ and $F_{N,\sigma_t}(\boldsymbol{\mu}_t)$. Let $f_N(\boldsymbol{x}) = e^{Nf(\boldsymbol{x})}$. For any $\boldsymbol{\mu} \in \mathbb{R}^d$,

$$
\begin{aligned}
&|F_{N,\sigma_{t+1}}(\boldsymbol{\mu}) - F_{N,\sigma_t}(\boldsymbol{\mu})| \\
=&\left|\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},I_d)}[f_N(\boldsymbol{\mu}+\sigma_{t+1}\boldsymbol{\epsilon})]\right. \\
&\left.-\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},I_d)}[f_N(\boldsymbol{\mu}+\sigma_t\boldsymbol{\epsilon})]\right| \\
=&\left|(2\pi)^{-d/2}\int_{\boldsymbol{\epsilon}\in\mathbb{R}^d}(f_N(\boldsymbol{\mu}+\sigma_{t+1}\boldsymbol{\epsilon})\right. \\
&\left.-f_N(\boldsymbol{\mu}+\sigma_t\boldsymbol{\epsilon}))e^{-\|\boldsymbol{\epsilon}\|^2/2}d\boldsymbol{\epsilon}\right| \\
\leq&(2\pi)^{-d/2}\int_{\boldsymbol{\epsilon}\in\mathbb{R}^d}|f_N(\boldsymbol{\mu}+\sigma_{t+1}\boldsymbol{\epsilon}) \\
&-f_N(\boldsymbol{\mu}+\sigma_t\boldsymbol{\epsilon})|\,e^{-\|\boldsymbol{\epsilon}\|^2/2}d\boldsymbol{\epsilon} \\
=&(2\pi)^{-d/2}\int_{\boldsymbol{\epsilon}\in\mathbb{R}^d}\left|e^{Nf(\boldsymbol{\mu}+\sigma_{t+1}\boldsymbol{\epsilon})}\right. \\
&\left.-e^{Nf(\boldsymbol{\mu}+\sigma_t\boldsymbol{\epsilon})}\right|e^{-\|\boldsymbol{\epsilon}\|^2/2}d\boldsymbol{\epsilon} \\
\leq&(2\pi)^{-d/2}\int_{\boldsymbol{\epsilon}\in\mathbb{R}^d}Ne^{Nf(\boldsymbol{x}^*)}|f(\boldsymbol{\mu}+\sigma_{t+1}\boldsymbol{\epsilon}) \\
&-f(\boldsymbol{\mu}+\sigma_t\boldsymbol{\epsilon})|e^{-\|\boldsymbol{\epsilon}\|^2/2}d\boldsymbol{\epsilon} \\
\leq&\frac{1}{(2\pi)^{d/2}}\int_{\boldsymbol{\epsilon}\in\mathbb{R}^d}Ne^{Nf(\boldsymbol{x}^*)}L_f|\sigma_{t+1}-\sigma_t|\|\boldsymbol{\epsilon}\|e^{-\|\boldsymbol{\epsilon}\|^2/2}d\boldsymbol{\epsilon} \\
\leq&\frac{Ne^{Nf(\boldsymbol{x}^*)}}{(2\pi)^{d/2}}L_f|\sigma_{t+1}-\sigma_t|\int_0^\infty e^{-r^2/2}r^d dr\int_{\boldsymbol{\theta}\in S_d}d\boldsymbol{\theta} \\
\leq&\frac{Ne^{Nf(\boldsymbol{x}^*)}}{(2\pi)^{d/2}}L_f|\sigma_{t+1}-\sigma_t|\int_0^\infty e^{-r^2/2}r^d dr\frac{2\pi^{d/2}}{\Gamma(d/2)} \\
\leq&Ne^{Nf(\boldsymbol{x}^*)}L_f|\sigma_{t+1}-\sigma_t|\frac{2}{2^{d/2}\Gamma(d/2)}\int_0^\infty e^{-r^2/2}r^d dr \\
\leq&Ne^{Nf(\boldsymbol{x}^*)}L_f|\sigma_{t+1}-\sigma_t|\frac{2}{2^{d/2}\Gamma(d/2)}2^{(d-1)/2}\Gamma\left(\frac{d+1}{2}\right) \\
\leq&Ne^{Nf(\boldsymbol{x}^*)}L_f|\sigma_{t+1}-\sigma_t|\frac{\sqrt{2}\Gamma((d+1)/2)}{\Gamma(d/2)} \\
=&Ne^{Nf(\boldsymbol{x}^*)}L_f\frac{\sqrt{2}\Gamma((d+1)/2)}{\Gamma(d/2)}\sigma_0(1-\beta)\beta^t \\
=&\sigma_0 C_{N,d,f}(1-\beta)\beta^t,
\end{aligned}
$$

where $C_{N,d,f} = \sqrt{2}\frac{\Gamma((d+1)/2)}{\Gamma(d/2)}Ne^{Nf(\boldsymbol{x}^*)}L_f$, the fifth relation is because of mean value theorem, the sixth relation is because of Assumption 2, S_d denotes the unit sphere surface in $\mathbb{R}^d$, and Γ denotes the Gamma function. In sum, we have

$$|F_{N,\sigma_{t+1}}(\boldsymbol{\mu}) - F_{N,\sigma_t}(\boldsymbol{\mu})| \leq \sigma_0 C_{N,d,f}(1-\beta)\beta^t.$$

This result further implies

$$F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t) \geq F_{N,\sigma_t}(\boldsymbol{\mu}_t) - \sigma_0 C_{N,d,f}(1-\beta)\beta^t.$$

Plugging this result to (9) gives

$$
\begin{aligned}
\mathbb{E}[F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_{t+1})] \geq& \mathbb{E}[F_{N,\sigma_t}(\boldsymbol{\mu}_t)] \\
&+\alpha_t\sigma_{t+1}^2\mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t))\|^2] \\
&-\alpha_t^2 2d^2e^{3Nf(\boldsymbol{x}^*)} - \sigma_0 C_{N,d,f}(1-\beta)\beta^t.
\end{aligned}
$$

Summing both sides over $t \in \{0,1,...,T-1\}$ gives

$$
\begin{aligned}
\mathbb{E}[F_{N,\sigma_T}(\boldsymbol{\mu}_T)] \geq& \mathbb{E}[F_{N,\sigma_0}(\boldsymbol{\mu}_0)] \\
&+\sum_{t=0}^{T-1}\alpha_t\sigma_{t+1}^2\mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|^2] \\
&-2d^2e^{3Nf(\boldsymbol{x}^*)}\sum_{t=0}^{T-1}\alpha_t^2 - \sigma_0 C_{N,d,f}(1-\beta)\sum_{t=0}^{T-1}\beta^t.
\end{aligned}
$$

After re-organizing the terms, we arrive at

$$
\begin{aligned}
&\sum_{t=0}^{T-1}\alpha_t\sigma_{t+1}^2\mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|^2] \\
\leq&\mathbb{E}[F_{N,\sigma_T}(\boldsymbol{\mu}_T)] - \mathbb{E}[F_{N,\sigma_0}(\boldsymbol{\mu}_0)] \\
&+2d^2e^{3Nf(\boldsymbol{x}^*)}\sum_{t=0}^{T-1}\alpha_t^2 + \sigma_0 C_{N,d,f}(1-\beta)\sum_{t=0}^{T-1}\beta^t \\
\leq&f_N(\boldsymbol{x}^*) - F_{N,\sigma_0}(\boldsymbol{\mu}_0) + 2d^2e^{3Nf(\boldsymbol{x}^*)}\sum_{t=0}^{T-1}\alpha_t^2 \\
&+\sigma_0 C_{N,d,f}(1-\beta)\sum_{t=0}^{T-1}\beta^t
\end{aligned}
$$

This finishes the proof for Theorem 1. □

Corollary 1. *Assume the conditions in Theorem 1 and Assumption 3. Let* $\alpha_t = (t+1)^{-(1/2+\gamma)}$ *for some* $\gamma \in (0,1/2)$. *For any* $\varepsilon \in (0,1)$, *after* $T > (C_2C_1d^2\varepsilon^{-1})^{2/(1-2\gamma)} = O_N((d^2\varepsilon^{-1}b^{-2})^{2/(1-2\gamma)})$ *times of parameter updating, we have*

$$\min_{t\in\{0,1,2,...,T\}}\mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|^2] < \varepsilon,$$

where $C_0 := f_N(\boldsymbol{x}^*) - F_{N,\sigma_0}(\boldsymbol{\mu}_0) + 2f_N^3(\boldsymbol{x}^*)\sum_{t=1}^{\infty} t^{-(1+2\gamma)} + \sqrt{2}\sigma_0 N f_N(\boldsymbol{x}^*)L_f$, $C_1 = b^{-2}C_0(1-2\gamma)$, *and* $C_2 = \max\{1, 1/C_1\}$. *Furthermore, the dependence of the* O_N*-factor on* N *can be removed under the additional assumption of* $f(\boldsymbol{x}) \le 0$ *for all* $\boldsymbol{x} \in \mathbb{R}^d$.

Proof. For any non-negative integer t, define

$$\nu_t := \min_{\tau \in \{0,1,\ldots,t\}} \mathbb{E}[\|\nabla F_{N,\sigma_{\tau+1}}(\boldsymbol{\mu}_\tau)\|^2]. \tag{10}$$

Then,

$$\begin{aligned}
&\sum_{t=0}^{T-1} \alpha_t \sigma_T^2 \nu_T \le \sum_{t=0}^{T-1} \alpha_t \sigma_{t+1}^2 \nu_t \\
&\le^{by (10)} \sum_{t=0}^{T-1} \alpha_t \sigma_{t+1}^2 \mathbb{E}[\|\nabla F_{N,\sigma_{t+1}}(\boldsymbol{\mu}_t)\|^2] \\
&\le^{\text{Theorem 1}} f_N(\boldsymbol{x}^*) - F_{N,\sigma_0}(\boldsymbol{\mu}_0) + 2d^2 f_N^3(\boldsymbol{x}^*) \sum_{t=0}^{T-1} \alpha_t^2 \\
&\quad + \sigma_0 C_{N,d,f}(1-\beta)\sum_{t=0}^{T-1} \beta^t, \\
&\le f_N(\boldsymbol{x}^*) - F_{N,\sigma_0}(\boldsymbol{\mu}_0) + 2d^2 f_N^3(\boldsymbol{x}^*) \sum_{t=1}^{\infty} t^{-(1+2\gamma)} \\
&\quad + \sigma_0 C_{N,d,f} \\
&\le \left(f_N(\boldsymbol{x}^*) - F_{N,\sigma_0}(\boldsymbol{\mu}_0) + 2f_N^3(\boldsymbol{x}^*) \sum_{t=1}^{\infty} t^{-(1+2\gamma)} \right. \\
&\quad \left. + \sqrt{2}\sigma_0 N f_N(\boldsymbol{x}^*) L_f \right) d^2, \\
&= C_{0,N} d^2,
\end{aligned} \tag{11}$$

where $C_{0,N} := f_N(\boldsymbol{x}^*) - F(\boldsymbol{\mu}_0) + 2f_N^3(\boldsymbol{x}^*)\sum_{t=1}^{\infty} t^{-(1+2\gamma)} + \sqrt{2}\sigma_0 N f_N(\boldsymbol{x}^*)L_f < \infty$, the second-to-last relation is because $d > 1$ and $f_N(\boldsymbol{x}^*) \ge F_{N,\sigma_0}(\boldsymbol{\mu}_0)$, $C_{0,N} = \sqrt{2}\frac{\Gamma((d+1)/2)}{\Gamma(d/2)} N f_N(\boldsymbol{x}^*) L_f$, and $\frac{\Gamma((d+1)/2)}{\Gamma(d/2)} \le d$ for any $d \ge 1$. Therefore, we have

$$\sum_{t=0}^{T-1} \alpha_t \sigma_T^2 \nu_T \le C_{0,N} d^2.$$

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

$$\begin{aligned}
\nu_T &\le \frac{C_{0,N} d^2}{\sigma_T^2 \sum_{t=1}^{T} t^{-(1/2+\gamma)}} \\
&< \frac{C_{0,N} d^2}{b^2 \int_1^T t^{-(1/2+\gamma)} dt}, \quad \text{recall } \sigma_T = \sigma_0 \beta^T + b, \\
&< \frac{C_{0,N} d^2}{b^2 \int_1^T t^{-(1/2+\gamma)} dt} \\
&= \frac{C_{0,N} d^2}{b^2 (T^{\frac{1}{2}-\gamma} - 1)/(\frac{1}{2}-\gamma)} \\
&< \frac{C_{0,N} d^2}{b^2 (T^{\frac{1}{2}-\gamma}/2)/(\frac{1}{2}-\gamma)}, \quad \text{when } T > 2^{2/(1-2\gamma)}, \\
&= C_{1,N} \frac{d^2}{T^{\frac{1}{2}-\gamma}}, \quad \text{where } C_{1,N} := \frac{C_{0,N}}{b^2(1/2)/(\frac{1}{2}-\gamma)}.
\end{aligned}$$

In sum, we have

$$\nu_T \le C_{1,N} \frac{d^2}{T^{\frac{1}{2}-\gamma}}, \quad \text{whenever } T > 2^{2/(1-2\gamma)}. \tag{12}$$

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

$$\begin{aligned}
T &> (C_{2,N} C_{1,N} d^2 \varepsilon^{-1})^{2/(1-2\gamma)} \\
&> (C_{2,N} C_{1,N})^{2/(1-2\gamma)} \\
&\ge 2^{2/(1-2\gamma)},
\end{aligned}$$

and

$$\begin{aligned}
\nu_T &\le^{\text{from (12)}} C_{1,N} \frac{d^2}{T^{\frac{1}{2}-\gamma}} \\
&< C_{1,N} \frac{d^2}{(C_{2,N} C_{1,N} d^2 \varepsilon^{-1})^{\frac{2}{1-2\gamma}(\frac{1}{2}-\gamma)}} \\
&= \frac{\varepsilon}{C_{2,N}} \le \varepsilon.
\end{aligned}$$

This finishes the proof for the first result.

Under the additional assumption that $f(\boldsymbol{x}) \le 0$ for all $\boldsymbol{x} \in \mathbb{R}^d$, we have $f_N(\boldsymbol{x}) = e^{N f(\boldsymbol{x})} \le 1$. Hence, $C_{0,N} \le C_0$, where $C_0 := 1 - F(\boldsymbol{\mu}_0) + 2(\boldsymbol{x}^*)\sum_{t=1}^{\infty} t^{-(1+2\gamma)} + \sqrt{2}\sigma_0 N L_f$. The above

proof still holds if we replace $C_{0,N}$ with C_0, $C_{1,N}$ with $C_1 := \frac{C_0}{b^2(1/2)/(\frac{1}{2}-\gamma)}$, and $C_{2,N}$ with $C_2 := \max\{1, 1/C_1\}$. Then, the O_N factor's dependence on N is removed. This finishes the proof for Cororllary 1. □

8.2 Hyper-Parameters

For the experiment of MNIST and CIFAR-10, we report the hyper-parameter candidates and selections of GS-PowerHP and Square Attack in Table 7 and 8, respectively. The hyper-parameters for other algorithms are chosen the same as those used in [30, Table 8-9] since the same experiments are performed there. For the experiment on ImageNet, the hyper-parameters are reported in Table 9. Note that b for GS-PowerHP is set as 0 for all experiments.

Table 7: Hyper-parameters for Attacks against MNIST.

	Selected Values	Candidates ($\boldsymbol{\mu}^*$)
GS-PowerHP	$\alpha_0 = 0.07$, $N = .5$, $\sigma = .05$.	$N \in \{0.05, 0.1, 0.5\}$, $\alpha_t \in \{.05, .07, .1\}$, $\sigma \in \{.05, .1\}$
Square Attack	$\varepsilon = 20$, $p = 0.1$	$\varepsilon \in \{1, 5, 10, 15, 20, 40, 100\}$, $p \in \{0.1, 0.5, 0.9\}$

Table 8: Hyper-parameters for Attacks against CIFAR-10.

	Selected Values	Candidates ($\boldsymbol{\mu}^*$)
GS-PowerHP	$\alpha = .08$, $N = .1$, $\sigma_0 = .05$, $\beta = .999$.	$N \in \{0.05, 0.1, 0.2\}$, $\alpha_t \in \{.05, .07.08\}$, $\sigma \in \{.05, .1\}$, $\beta \in \{.995, .999\}$.
Square Attack	$\varepsilon = 15$, $p = 0.05$	$\varepsilon \in \{1, 5, 10, 15, 20, 40, 100\}$, $p \in \{0.05, 0.1, 0.5, 0.9\}$.

Table 9: Hyper-parameters for ImageNet Attack. The candidate set for (initial) smoothing parameters is $\mathcal{S} := \{0.1, .05\}$. 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 set of candidate values that lead to the highest success rate and minimal perturbation size (averaged over the 10 image attackes) are be selected.

	Selected Values	Candidates ($\boldsymbol{\mu}^*$)
GS-PowerHP	$\alpha = 1.0$, $N = 8$, $\sigma = .1$.	$N \in \{5, 8\}$, $\alpha_t \in \{1.0, 1.5\}$, $\sigma_0 \in \mathcal{S}$.
GS-PowerGS	$\alpha = 1.5$, $N = 5$, $\sigma = .1$.	$N \in \{5, 8\}$, $\alpha_t \in \{1.0, 1.5\}$, $\sigma \in \mathcal{S}$.
ZO-SLGHd	$\beta = 1/3072$, $\eta = 10^{-5}$, $t_1 = .1$, $\gamma = .995$	$\beta \in \{.1, 1/3072, .01/3072\}$, $\eta \in \{.0001/3072, 10^{-5}\}$, $t_1 \in \mathcal{S}$, $\gamma \in \{.999, .995\}$.
ZO-SLGHr	$\beta = 1/3072$, $t_1 = .1$, $\gamma = .999$.	$\beta \in \{.1, 1/3072, .01/3072\}$, $\gamma \in \{.999, .995\}$, $t_1 \in \mathcal{S}$.
ZO-AdaMM	$\beta_1 = .9$, $\beta_2 = .1$, $\alpha = .1$, $\mu = .05$.	$\alpha \in \{0.5/3072, 1/3072\}$, $\beta_1 \in \{.5, .9\}$, $\beta_2 \in \{.1, .3\}$, $\mu \in \mathcal{S}$
ZO-SGD	$\alpha = 0.01$, $\mu = .1$.	$\alpha \in \{.01, 1/3072, .01/3072\}$, $\mu \in \mathcal{S}$.
Square Attack	$\varepsilon = 250$, $p = 0.001$	$\varepsilon \in \{20, 60, 100, 150, 200, 250, 300\}$, $p \in \{0.0001, 0.001, 0.005, 0.01\}$