FASTER GRADIENT-FREE METHODS FOR ESCAPING SADDLE POINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Escaping from saddle points has become an important research topic in non-convex optimization. In this paper, we study the case when calculations of explicit gradients are expensive or even infeasible, and only function values are accessible. Currently, there have two types of gradient-free (zeroth-order) methods based on random perturbation and negative curvature finding proposed to escape saddle points efficiently and converge to an ϵ-approximate second-order stationary point. Nesterov's accelerated gradient descent (AGD) method can escape saddle points faster than gradient descent (GD) which have been verified in first-order algorithms. However, whether AGD could accelerate the gradient-free methods is still unstudied. To unfold this mystery, in this paper, we propose two accelerated variants for the two types of gradient-free methods of escaping saddle points. We show that our algorithms can find an ϵ-approximate second-order stationary point with $\tilde{\mathcal{O}}(1/\epsilon^{1.75})$ iteration complexity and $\tilde{\mathcal{O}}(d/\epsilon^{1.75})$ oracle complexity, where d is the problem dimension. Thus, our methods achieve a comparable convergence rate to their first-order counterparts and have fewer oracle complexity compared to prior derivative-free methods for finding second-order stationary points.

1 INTRODUCTION

Non-convex optimization has received increasing attention in recent years because lots of modern machine learning (ML) and deep learning (DL) tasks can be formulated as optimizing models with non-convex loss functions. In this paper, we consider non-convex optimization with the following general form:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}), \tag{1}$$

where $f(\mathbf{x})$ is differentiable and has Lipschitz continuous gradient and Hessian.

In this paper, we focus on situations when first-order information (gradient) is not always directly accessible. Many machine learning and deep learning applications often encounter settings where the calculation of explicit gradients is expensive or even infeasible, such as black-box adversarial attack on deep neural networks (Papernot et al., 2017; Madry et al., 2018; Chen et al., 2017; Bhagoji et al., 2018; Tu et al., 2019), policy search in reinforcement learning (Salimans et al., 2017; Choromanski et al., 2018; Jing et al., 2021), hyper-parameter optimization (Bergstra & Bengio, 2012). Therefore, zeroth-order optimization, which utilizes only the zeroth-order information (function value) to optimize the non-convex problem 1, has gained increasing attention in machine learning.

In general, the goal of a non-convex optimization problem (1) is to find an ϵ-approximate first-order stationary point (FOSP, see Definition 3), since finding the global minimum is NP-hard. Gradient descent is proven to be an optimal first-order algorithm for finding an ϵ-approximate FOSP of non-convex problem 1 under the gradient Lipschitz assumption (Carmon et al., 2020; 2021), which needs a gradient query complexity of $\mathcal{O}(\frac{1}{\epsilon^2})$. However, for non-convex functions, FOSPs can be local minima, global minima and saddle points. The ubiquity of saddle points makes high-dimensional non-convex optimization problems extremely difficult and will lead to highly suboptimal solutions (Jain et al., 2017; Sun et al., 2018). Therefore, many recent research works have focused on escaping saddle points and studying properties of converging to an ϵ-approximate second-order stationary point (SOSP, see Definition 4) using first-order methods.

A recent line of work showed that first-order methods can efficiently escape saddle points and converge to SOSPs. Specifically, Jin et al. (2017) proposed the perturbed gradient descent (PGD) algorithm by adding uniform random perturbation into the standard gradient descent algorithm that can find an ϵ-approximate SOSP in $\tilde{\mathcal{O}}(\log^4 d/\epsilon^2)$ gradient queries. Under the zeroth-order setting, Jin et al. (2018a) proposed a zeroth-order perturbed stochastic gradient descent (ZPSGD) method, which studied the power of Gaussian smoothing and stochastic perturbed gradient for finding local minima. The role of Gaussian smoothing is to reduce zeroth-order optimization to a stochastic first-order optimization of a Gaussian smoothed function of problem (1). They proved their method can find an ϵ-approximate SOSP with a function query complexity of $\tilde{\mathcal{O}}\left(d^2/\epsilon^5\right)$. Vlatakis-Gkaragkounis et al. (2019) proposed the perturbed approximate gradient descent (PAGD) method using the forward difference of the coordinate-wise gradient estimators, which finds an ϵ-approximate SOSP in $\tilde{\mathcal{O}}\left(d\log^4 d/\epsilon^2\right)$ function queries. Recently, Lucchi et al. (2021) proposed a random search power iteration (RSPI) method, which alternatively runs the random search step and zeroth-order power iteration step, and can find an $(\epsilon, \epsilon^{2/3})$-approximate SOSP ($\|\nabla f(\mathbf{x})\| \leq \epsilon$, $\lambda_{\min}(\nabla^2 f(\mathbf{x})) \geq -\epsilon^{2/3}$) in $\mathcal{O}(d\log d/\epsilon^{\frac{8}{3}})$ function queries.

Table 1: Comparison of different zeroth-order methods for finding ϵ-approximate second-order stationary points. (CoordGE, GaussGE are abbreviations of "coordinate-wise gradient estimator", "Gaussian random gradient estimator", respectively.)

Algorithm	Reference	Main Technique	Function Queries
ZPSGD	Jin et al. (2018a)	Random perturbation	$\tilde{\mathcal{O}}\left(\frac{d^2}{\epsilon^5}\right)$
PAGD	Vlatakis-Gkaragkounis et al. (2019)	Random perturbation	$\mathcal{O}\left(\frac{d\log^4 d}{\epsilon^2}\right)$
RSPI	Lucchi et al. (2021)	Negative curvature finding	$\mathcal{O}(\frac{d\log d}{\epsilon^{8/3}})$ ‡
Algorithm 1	**Theorem 1**	Random perturbation	$\mathcal{O}\left(\frac{d\log^6 d}{\epsilon^{7/4}}\right)$
Algorithm 3	**Theorem 2**	Negative curvature finding	$\mathcal{O}\left(\frac{d\log d}{\epsilon^{7/4}}\right)$

‡ guarantees $(\epsilon, \epsilon^{2/3})$-approximate second-order stationary points.

Although gradient descent has achieved an optimal convergence rate for finding FOSPs under gradient Lipschitz assumption, potential improvements are achievable under additional Hessian Lipschitz assumption (Carmon et al., 2021). Nesterov's AGD combined with some special mechanisms, has been proved to be able to find ϵ-approximate FOSPs with less query complexity. Carmon et al. (2017) proposed a variant of Nesterov's AGD with a "convex until guilty" mechanism, which can find an ϵ-approximate FOSP with gradient query complexity $\mathcal{O}(\frac{1}{\epsilon^{7/4}}\log\frac{1}{\epsilon})$. Recently, Li & Lin (2022) proposed a restarted accelerated gradient descent method that can find an ϵ-approximate FOSP in gradient query complexity $\mathcal{O}(\frac{1}{\epsilon^{7/4}})$, which adds a restart mechanism to Nesterov's AGD.

On finding SOSPs, Nesterov's AGD is also proved to be more efficient than GD. Jin et al. (2018b) studied a variant of Nesterov's AGD named perturbed AGD, and proved that it can find an ϵ-approximate SOSP in $\tilde{\mathcal{O}}(\log^6 d/\epsilon^{7/4})$ gradient queries. Their method added two algorithmic features to Nesterov's AGD: random perturbation and negative curvature exploitation, to ensure the monotonic decrease of the Hamiltonian function (see Eq.4). Allen-Zhu & Li (2018) proposed a first-order Negative curvature finding framework named Neon2 that can find the most negative curvature direction efficiently. Combining Neon2 with CDHS method of Carmon et al. (2018) can find an ϵ-approximate SOSPs in $\tilde{\mathcal{O}}(\log d/\epsilon^{7/4})$ gradient queries, which improved the complexity of perturbed AGD method by a factor of $\mathrm{poly}(\log d)$ due to the use of negative curvature finding subroutine. Recently, Zhang & Li (2021) proposed a single-loop algorithm that also achieves the same function query complexity, which replaced the random perturbation step in perturbed AGD with accelerated negative curvature finding.

Given the advantages of Nesterov's AGD in finding SOSPs in first-order optimization, it is then natural to design AGD based zeroth-order methods for finding SOSPs with fewer function query complexity. To the best of our knowledge, it is still a vacancy in zeroth-order optimization.

Contributions The main contributions of this paper are summarized as follows,

- We study the complexity of two AGD based zeroth-order methods for finding ϵ-approximate SOSPs. We first study a zeroth-order version of the perturbed AGD method (Algorithm 1) using the central finite difference version of the coordinate-wise gradient estimator, which can be proved to have a lower approximation error compared to its forward counterpart. The total function query complexity of Algorithm 1 for finding an ϵ-approximate SOSP is $\tilde{\mathcal{O}}(d\log^6 d/\epsilon^{\frac{7}{4}})$.
- Due to the efficiency of the negative curvature finding for finding the most negative curvature direction near a saddle. We further study a zeroth-order version of the perturbed AGD with accelerated negative curvature finding subroutine (Algorithm 3), which uses the finite difference of the two coordinate-wise gradient estimators to approximate the Hessian-vector product. We show that Algorithm 3 can further improve the function query complexity of Algorithm 1 by a factor of $\mathrm{poly}(\log d)$.
- Finally, we conduct several empirical experiments to verify the efficiency and effectiveness of our methods in escaping saddle points.

2 PRELIMINARIES

2.1 NOTATIONS

Throughout this paper, we use bold uppercase letters $\mathbf{A}, \mathbf{B}$ to denote matrices and bold lowercase letters $\mathbf{x}, \mathbf{y}$ to denote vectors. We use $\|\cdot\|$ to denote the Euclidean norm of a vector and the spectral norm of a matrix. We use $\mathbb{B}_{\mathbf{x}}(r)$ to denote the ℓ_2 ball with radius r centered at point $\mathbf{x}$. We use $\tilde{\mathcal{O}}(\cdot)$ to hide absolute constants and log factors.

2.2 DEFINITIONS

Definition 1. *For a differentiable nonconvex function $f : \mathbb{R}^d \to \mathbb{R}$, f is ℓ-Lipschitz smooth if*

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq \ell\|\mathbf{x} - \mathbf{y}\|.$$

Definition 2. *For a twice differentiable nonconvex function $f : \mathbb{R}^d \to \mathbb{R}$, f is ρ-Hessian Lipschitz if*

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\| \leq \rho\|\mathbf{x} - \mathbf{y}\|.$$

Definition 3. *For a differentiable function f, we say $\mathbf{x}$ is an ϵ-approximate first-order stationary point if $\|\nabla f(\mathbf{x})\| \leq \epsilon$.*

Definition 4. *For a twice differentiable function f, we say $\mathbf{x}$ is an ϵ-approximate second-order stationary point if*

$$\|\nabla f(\mathbf{x})\| \leq \epsilon \quad \textit{and} \quad \lambda_{\min}(\nabla^2 f(\mathbf{x})) \geq -\sqrt{\rho\epsilon}.$$

2.3 ZEROTH-ORDER GRADIENT ESTIMATOR

In this subsection, we introduce a central difference coordinate-wise gradient estimator, which is widely studied in literature of zeroth-order optimization (Ji et al., 2019; Vlatakis-Gkaragkounis et al., 2019; Lucchi et al., 2021),

$$\hat{\nabla} f(\mathbf{x}) = \sum_{i=1}^{d} \frac{f(\mathbf{x} + \mu\mathbf{e}_i) - f(\mathbf{x} - \mu\mathbf{e}_i)}{2\mu}\mathbf{e}_i, \tag{2}$$

where $\mathbf{e}_i$ is the i-th standard basis vector with 1 at its i-th coordinate and 0 otherwise. When analyzing the approximation error of the above gradient estimator, previous work only exploited the smoothness property of the gradient of f, not the property of Hessian Lipschitz (which is a basic assumption for analyzing the second-order convergence properties). To fill this gap, we establish the following lemma,

Lemma 1. *For a twice differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, assume that f is ρ-Hessian Lipschitz, then for any given smoothing parameter μ and any $x \in \mathbb{R}^d$, we have*

$$\|\hat{\nabla} f(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \frac{1}{36}\rho^2 d\mu^4.$$

Note that, under the Hessian Lipschitz assumption, the central difference has a lower approximation error than that of $\mathcal{O}(\ell^2 d\mu^2)$ error under the ℓ-smooth assumption (Ji et al., 2019).

2.4 ZEROTH-ORDER HESSIAN-VECTOR PRODUCT ESTIMATOR

In this subsection, we show how to approximate the Hessian-vector product under the setting that we only have access to the zeroth-order information. By the property Hessian Lipschitz, it is easy to check that the Hessian-vector product $\nabla^2 f(\mathbf{x}) \cdot \mathbf{v}$ can be approximated by the difference of two gradients $\nabla f(\mathbf{x}+\mathbf{v}) - \nabla f(\mathbf{x})$ with approximation error up to $\mathcal{O}(\|\mathbf{v}\|^2)$ for some $\mathbf{v}$ with small magnitude. On the other hand, by Lemma 1, $\nabla f(\mathbf{x}+\mathbf{v}), \nabla f(\mathbf{x})$ can be approximated by the central difference coordinate-wise gradient estimator with high accuracy. Then we define the following zeroth-order Hessian-vector product estimator as follows, which was previously studied in (Ye et al., 2018; Lucchi et al., 2021):

$$\begin{aligned}\mathcal{H}_f(\mathbf{x})\mathbf{v} &= \hat{\nabla} f(\mathbf{x}+\mathbf{v}) - \hat{\nabla} f(\mathbf{x}) \\ &= \sum_{i=1}^{d} \frac{f(\mathbf{x}+\mathbf{v}+\mu\mathbf{e}_i) - f(\mathbf{x}+\mathbf{v}-\mu\mathbf{e}_i)}{2\mu}\mathbf{e}_i - \sum_{i=1}^{d} \frac{f(\mathbf{x}+\mu\mathbf{e}_i) - f(\mathbf{x}-\mu\mathbf{e}_i)}{2\mu}\mathbf{e}_i\end{aligned} \tag{3}$$

Above, the notation $\mathcal{H}_f(x)$ can be seen as the Hessian matrix of f at point $\mathbf{x}$ with small perturbations and we don't need to know the explicit expression since we only need to study the approximation error of it, which is established in the following lemma. Lucchi et al. (2021) proved that the approximation error of the zeroth-order Hessian-vector product can be upper bounded by $\mathcal{O}(\rho\|\mathbf{v}\|^2 + \ell\sqrt{d}\mu)$. However, by using Lemma 1, it can be further improved to:

Lemma 2. *For a twice differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, assume that f is ρ-Hessian Lipschitz, then for any smoothing parameter μ and $\mathbf{x} \in \mathbb{R}^d$, we have*

$$\|\mathcal{H}_f(\mathbf{x})\mathbf{v} - \nabla^2 f(\mathbf{x})\mathbf{v}\| \le \rho\left(\frac{\|\mathbf{v}\|^2}{2} + \frac{\sqrt{d}\mu^2}{3}\right).$$

2.5 HAMILTONIAN

The following function, which takes the form of Hamiltonian, was proposed by Jin et al. (2018b) to tackle the problem of monotonic decrease of the function value for the momentum-based algorithms in the nonconvex setting,

$$E_t = f(\mathbf{x}_t) + \frac{1}{2\eta}\|\mathbf{v}_t\|^2, \tag{4}$$

where $\mathbf{v}_t = \mathbf{x}_t - \mathbf{x}_{t-1}$ is the momentum.

3 ALGORITHM DESCRIPTION

In this section, we propose two novel Nesterov's accelerated method based algorithms that can escape saddle points and converge to an ϵ-approximate SOSP using only zeroth-order oracles.

3.1 ZEROTH-ORDER PERTURBED ACCELERATED GRADIENT DESCENT

In this subsection, we introduce the zeroth-order perturbed accelerated gradient descent method in Algorithm 1. The algorithms consist of three part: the random perturbation steps, the accelerated gradient descent steps and the negative curvature exploitation steps. The random perturbation step is called when the gradient is small and no perturbation is added over the past $\mathscr{T}$ iterations. Let $\kappa = \frac{\ell}{\sqrt{\rho\epsilon}}$, and set the parameters of Algorithm 1 as follows,

$$\begin{aligned}\eta &= \frac{1}{4\ell}, & \theta &= \frac{1}{4\sqrt{\kappa}}, & \gamma &= \frac{\theta^2}{\eta}, \\ s &= \frac{\gamma}{4\rho}, & \mathscr{T} &= \sqrt{\kappa}\chi c, & r &= \eta\epsilon\chi^{-5}c^{-8},\end{aligned} \tag{5}$$

where $\chi = \max\{1, \log\frac{d\ell\Delta_f}{\rho\epsilon\delta}\}$.

Algorithm 1 Zeroth-Order Perturbed Accelerated Gradient Descent

$\mathbf{v}_0 \leftarrow 0$, $t_{\text{perturb}} \leftarrow 0$
for $t = 0, 1, \ldots$ **do**
  if $\|\hat{\nabla} f(\mathbf{x}_t)\| \leq \frac{3}{4}\epsilon$ and $t - t_{\text{perturb}} > \mathscr{T}$ **then**
    $\mathbf{x}_t \leftarrow \mathbf{x}_t + \xi_t$, $\xi_t \sim \text{Unif}\,(\mathbb{B}_0(r))$, $t_{\text{perturb}} \leftarrow t$
  $\mathbf{y}_t \leftarrow x_t + (1-\theta)\mathbf{v}_t$
  $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t - \eta\hat{\nabla} f(\mathbf{y}_t)$
  $\mathbf{v}_{t+1} = \mathbf{x}_{t+1} - \mathbf{x}_t$
  if $f(\mathbf{x}_t) \leq f(\mathbf{y}_t) + \left\langle \hat{\nabla} f(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t \right\rangle - \frac{\gamma}{2}\|\mathbf{y}_t - \mathbf{x}_t\|^2$ **then**
    $(\mathbf{x}_{t+1}, \mathbf{v}_{t+1}) \leftarrow \text{NCE}(\mathbf{x}_t, \mathbf{v}_t, s)$

Algorithm 2 Negative Curvature Exploitation $(\mathbf{x}_t, \mathbf{v}_t, s)$

if $\|\mathbf{v}_t\| \geq s$ **then**
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$
else
  $\delta = s \cdot \mathbf{v}_t / \|\mathbf{v}_t\|$
  $\mathbf{x}_{t+1} \leftarrow \arg\min_{\mathbf{x} \in \{\mathbf{x}_t + \delta, \mathbf{x}_t - \delta\}} f(\mathbf{x})$
Return $(\mathbf{x}_{t+1}, 0)$

Since we only have access to the zeroth-order information, we can verify if a point $\mathbf{x}$ is an ϵ-approximate FOSP by using the coordinate-wise gradient estimator based on the following fact:

Proposition 1. *Assume that f is ρ-Hessian Lipschitz, with choice of the smoothing parameter μ in Eq.2 such that $\mu \leq \sqrt{\frac{3\epsilon}{2\rho\sqrt{d}}}$, we can conclude that*

- *if $\|\hat{\nabla} f(\mathbf{x})\| \leq \frac{3\epsilon}{4}$, then we have $\|\nabla f(x)\| \leq \epsilon$,*
- *if $\|\hat{\nabla} f(\mathbf{x})\| > \frac{3\epsilon}{4}$, then we have $\|\nabla f(x)\| \geq \frac{\epsilon}{2}$.*

The proof of this proposition directly follows from Lemma 1. The random perturbation is uniformly randomly selected from the ℓ_2-ball with radius r. The second part of the Algorithm 1 is the Nesterov's accelerated gradient descent steps with its gradients estimated by 2.

The negative curvature exploitation step is called when the following condition holds:

$$f(\mathbf{x}_t) \leq f(\mathbf{y}_t) + \left\langle \hat{\nabla} f(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t \right\rangle - \frac{\gamma}{2}\|\mathbf{y}_t - \mathbf{x}_t\|^2. \tag{6}$$

If this condition hold, then the function have an approximate large negative curvature between $\mathbf{x}_t$ and $\mathbf{y}_t$. In this case, the accelerated gradient step may not decrease the function value of the Hamiltonian. Then we call the negative curvature exploitation step to further decrease the Hamiltonian. Specifically, when 6 doesn't hold, we have the following lemma:

Lemma 3. *Assume that $f(\cdot)$ is ℓ-smooth, ρ-Hessian Lipschitz and set the learning rate $\eta \leq \frac{1}{4\ell}, \theta \in [2\eta\gamma, \frac{1}{2}]$. Then, for each iteration t where 6 does not holds, we have:*

$$E_{t+1} \leq E_t - \frac{\theta}{2\eta}\|\mathbf{v}_t\|^2 - \frac{\eta}{4}\|\nabla f(\mathbf{y}_t)\|^2 + \eta \cdot \frac{\rho^2 d\mu^4}{48}.$$

On the other hand, when 6 holds, *i.e.*, a negative curvature direction is observed, then we have the following lemma:

Lemma 4. *Assume that $f(\cdot)$ is ℓ-smooth and ρ-Hessian Lipschitz. Then, for each iteration t where 6 holds, we have:*

$$E_{t+1} \leq E_t - \min\{\frac{s^2}{2\eta}, \frac{1}{2}\gamma s^2 - \rho s^3 - \frac{\rho^2 d\mu^4}{9\gamma}\}.$$

Remark 1. *The results in Lemma 3 and 4 are similar to the ones in Jin et al. (2018b) while with additional system error terms induced by the smoothing parameter μ. Lemma 3 and 4 together ensure the monotonic decrease of the Hamiltonian in each iteration as long as the smoothing parameter μ is sufficient small.*

Then we set $\mathscr{T} = \sqrt{\kappa}\chi c = \tilde{\Theta}(\sqrt{\kappa})$ and denote $\mathscr{E} := \sqrt{\frac{\epsilon^3}{\rho}}\chi^{-5}c^{-7} = \tilde{\Theta}(\sqrt{\frac{\epsilon^3}{\rho}})$. Based on Lemma 3 and Lemma 4, we can further prove that when the current approximate gradient is large, i.e., $\|\hat{\nabla} f(\mathbf{x}_t)\| \geq \frac{3\epsilon}{4}$ (or equivalently, $\|\nabla f(\mathbf{x}_t)\| \geq \frac{\epsilon}{2}$, according to Lemma 1). We have the following average decrease lemma:

Lemma 5 (**Large gradient**). *If $\|\hat{\nabla} f(\mathbf{x}_\tau)\| \geq \frac{3\epsilon}{4}$ with $\mu \leq \mathcal{O}((\frac{3\epsilon}{2\rho\sqrt{d}})^{1/2})$ in Line 3 of Algorithm 1 for all $\tau \in [0, \mathscr{T}]$, then by running Algorithm 1 with $\mu \leq \tilde{\mathcal{O}}(\frac{\epsilon^{5/8}}{d^{1/4}})$ in Line 6 and $\mu \leq \tilde{\mathcal{O}}(\frac{\epsilon^{1/2}}{d^{1/4}})$ in Line 8, we have $E_{\mathscr{T}} - E_0 \leq -\mathscr{E}$.*

On the other hand, when the current approximate gradient is small and no perturbation is added over the past $\mathscr{T}$ iterations, then we add a uniform random perturbation in $\mathbb{B}_0(r)$. If there exist a large negative curvature direction of the current point, we have

Lemma 6 (**Negative curvature**). *Suppose $\|\hat{\nabla} f(\mathbf{x}_t)\| \leq \frac{3\epsilon}{4}$ (thus $\|\hat{\nabla} f(\mathbf{x}_t)\| \leq \epsilon$), $\lambda_{\min}(\nabla^2 f(\mathbf{x}_t)) \leq -\sqrt{\rho\epsilon}$ and no perturbation is added in iterations $[t - \mathscr{T}, t)$. Then by running Algorithm 1, we have $E_{\mathscr{T}} - E_0 \leq -\mathscr{E}$ with probability at least $1 - \frac{\delta\mathscr{E}}{2\Delta_f}$.*

Utilizing the above lemmas, we finally get the following main result.

Theorem 1. *Assume that $f : \mathbb{R}^d \to \mathbb{R}$ is ℓ-smooth and ρ-Hessian Lipschitz. For any $\delta > 0, \epsilon \leq \frac{\ell^2}{\rho}$, $f(\mathbf{x}_0) - f^* \leq \Delta_f$, if we set the hyperparameters as in 5 and choose $\mu = \tilde{\mathcal{O}}(\frac{\epsilon^{1/2}}{d^{1/4}})$ in Line 3 and 8, $\mu = \tilde{\mathcal{O}}(\frac{\epsilon^{13/8}}{d^{1/2}})$ in Line 6 of Algorithm 1, respectively, then with probability at least $1 - \delta$, one of the iterates will be an ϵ-approximate SOSP. The total number of function query is no more than*

$$\mathcal{O}\left(\frac{d\Delta_f \ell^{1/2}\rho^{1/4}}{\epsilon^{7/4}} \log^6(\frac{d\ell\Delta_f}{\rho\epsilon\delta})\right).$$

Remark 2. *Note that, Theorem 1 only ensures that with high probability, one of the iterates will be an ϵ-approximate SOSP. It is then natural to add a termination condition to make the algorithm more practical: Once the pre-condition of random perturbation step is reached, record the current iterate point $\mathbf{x}_{t_0}$ and the current function value of the Hamiltonian E_{t_0} before adding the random perturbation. If the decrease of the Hamiltonian is less than $\mathscr{E}$ after $\mathscr{T}$ iterations, then, with high probability $\mathbf{x}_{t_0}$ is an ϵ-approximate SOSP according to Lemma 6.*

3.2 Zeroth-Order Perturbed Accelerated Gradient Descent with Accelerated Negative Curvature Finding

In this subsection, we introduce how to utilize the negative curvature finding to accelerate escaping saddle points. The main task of the negative curvature finding is to find the approximate most negative eigenvector direction near a saddle point. Then adding a perturbation in this direction will obtain a more efficient decrease of the function value.

Classical methods for computing the most negative eigenvector direction like the power method and Lanczos method require the computations of the Hessian-vector products. Since we have only access to the zeroth-order information, an efficient way to approximate the Hessian-vector product is to utilize the zeroth-order Hessian-vector product estimator in 3. The accelerated negative curvature finding subroutine is self-contained in Line 11-13 of Algorithm 3 when $\zeta \neq \mathbf{0}$. The following lemma states that the accelerated negative curvature finding using zeroth-order Hessian-vector estimator can find a negative curvature direction in almost the same iteration complexity as the Lanczos method.

Lemma 7. *Suppose $\|\hat{\nabla} f(\mathbf{x}_t)\| \leq \frac{3\epsilon}{4}$, $\lambda_{\min}(\nabla^2 f(\mathbf{x}_t))$ and no perturbation is added in iterations $[t - \mathscr{T}', t]$. For any $0 < \delta_0 < 1$, let $\kappa = \frac{\ell}{\sqrt{\rho\epsilon}}$, and set the parameters as follows,*

$$\eta = \frac{1}{4\ell}, \qquad \theta = \frac{1}{4\sqrt{\kappa}}, \qquad \gamma = \frac{\theta^2}{\eta}, \qquad s = \frac{\gamma}{4\rho},$$

Algorithm 3 Zeroth-Order Perturbed Accelerated Gradient Descent with Accelerated Negative Curvature Finding

1: $t_{\text{perturb}} \leftarrow -\mathscr{T}' - 1, \mathbf{y}_0 \leftarrow \mathbf{x}_0, \tilde{\mathbf{x}} \leftarrow \mathbf{x}_0, \zeta \leftarrow \mathbf{0}$
2: **for** $t = 0, 1, \ldots,$ **do**
3: **if** $\|\hat{\nabla} f(\mathbf{x}_t)\| \leq \frac{3\epsilon}{4}$ and $t - t_{\text{perturb}} > \mathscr{T}'$ **then**
4: $\tilde{\mathbf{x}} = \mathbf{x}_t$
5: $\mathbf{x}_t = \tilde{\mathbf{x}} + \xi_t, \xi_t \sim \text{Unif}(\mathbb{B}_0(r'))$
6: $\mathbf{y}_t = \mathbf{x}_t, \zeta = \hat{\nabla} f(\tilde{\mathbf{x}}), t_{\text{perturb}} \leftarrow t$
7: **if** $t_{\text{perturb}} \neq -\mathscr{T}' - 1$ and $t - t_{\text{perturb}} = \mathscr{T}'$ **then**
8: $\hat{\mathbf{e}} \leftarrow \frac{\mathbf{x}_t - \tilde{\mathbf{x}}}{\|\mathbf{x}_t - \tilde{\mathbf{x}}\|}$
9: $\mathbf{x}_t \leftarrow \arg\min_{\mathbf{x} \in \{\tilde{\mathbf{x}} - \frac{1}{4}\sqrt{\frac{\epsilon}{\rho}}\hat{\mathbf{e}}, \tilde{\mathbf{x}} + \frac{1}{4}\sqrt{\frac{\epsilon}{\rho}}\hat{\mathbf{e}}\}} f(\mathbf{x})$
10: $\mathbf{y}_t = \mathbf{x}_t, \zeta = \mathbf{0}$
11: $\mathbf{x}_{t+1} = \mathbf{y}_t - \eta(\hat{\nabla} f(\mathbf{y}_t) - \zeta)$
12: $\mathbf{v}_{t+1} = \mathbf{x}_{t+1} - \mathbf{x}_t$
13: $\mathbf{y}_{t+1} = \mathbf{x}_{t+1} + (1 - \theta)\mathbf{v}_{t+1}$
14: **if** $t_{\text{perturb}} \neq -\mathscr{T}' - 1$ and $t - t_{\text{perturb}} < \mathscr{T}'$ **then**
15: $(\mathbf{y}_{t+1}, \mathbf{x}_{t+1}) = \tilde{\mathbf{x}} + r' \cdot (\frac{\mathbf{y}_{t+1} - \tilde{\mathbf{x}}}{\|\mathbf{y}_{t+1} - \tilde{\mathbf{x}}\|}, \frac{\mathbf{x}_{t+1} - \tilde{\mathbf{x}}}{\|\mathbf{x}_{t+1} - \tilde{\mathbf{x}}\|})$
16: **else if** $f(\mathbf{x}_{t+1}) \leq f(\mathbf{y}_{t+1}) + \left\langle \hat{\nabla} f(\mathbf{y}_{t+1}), \mathbf{x}_{t+1} - \mathbf{y}_{t+1} \right\rangle - \frac{\gamma}{2}\|\mathbf{y}_{t+1} - \mathbf{x}_{t+1}\|^2$ **then**
17: $(\mathbf{x}_{t+1}, \mathbf{v}_{t+1}) \leftarrow \text{NCE}(\mathbf{x}_{t+1}, \mathbf{v}_{t+1}, s)$
18: $\mathbf{y}_{t+1} \leftarrow \mathbf{x}_{t+1} + (1 - \theta)\mathbf{v}_{t+1}$

$$\mathscr{T}' = 32\sqrt{\kappa}\log(\frac{\ell\sqrt{d}}{\delta_0\sqrt{\rho\epsilon}}), \qquad r' = \frac{\delta_0\epsilon}{32}\sqrt{\frac{\pi}{\rho d}}. \tag{7}$$

Then by running Algorithm 3 for $\mathscr{T}'$ iterations after adding the random perturbation in Line 5, with probability at least $1 - \delta_0$, we have

$$\hat{\mathbf{e}}^\top \nabla^2 f(\mathbf{x}_t)\hat{\mathbf{e}} \leq -\frac{\sqrt{\rho\epsilon}}{4}.$$

Then moving along the direction of $\hat{\mathbf{e}}$, the function value of f will make further decrease according to the following lemma:

Lemma 8 (Zhang & Li (2021), Lemma 6)**.** *Suppose the function $f : \mathbb{R}^d \to \mathbb{R}$ is ℓ-smooth and ρ-Hessian Lipschitz. Then for any point $\mathbf{x}_0 \in \mathbb{R}^d$, if there exist a unit vector $\hat{\mathbf{e}}$ satisfying $\hat{\mathbf{e}}\nabla^2 f(\mathbf{x}_0)\hat{\mathbf{e}} \leq -\frac{\sqrt{\rho\epsilon}}{4}$, then we have*

$$f(\mathbf{x}_0 - \frac{f'_{\hat{\mathbf{e}}}(\mathbf{x}_0)}{4|f'_{\hat{\mathbf{e}}}(\mathbf{x}_0)|}\sqrt{\frac{\epsilon}{\rho}}\hat{\mathbf{e}}) \leq f(\mathbf{x}_0) - \frac{1}{384}\sqrt{\frac{\epsilon^3}{\rho}},$$

where $f'_{\hat{\mathbf{e}}}(\mathbf{x}_0)$ is the directional derivative along the direction $\hat{\mathbf{e}}$.

Remark 3. *In the first-order setting, $f'_{\hat{\mathbf{e}}}(\mathbf{x}_0) = \langle \nabla f(\mathbf{x}_0), \hat{e} \rangle$. However, in the zeroth-order setting, the directional derivative cannot be computed directly. To tackle this problem, one can simply compare the function value of two opposite directions,* i.e.*, Line 9 of Algorithm 3.*

Theorem 2. *Assume that $f(\cdot)$ is ℓ-smooth and ρ-Hessian Lipschitz. For any $\delta > 0, \epsilon \leq \frac{\ell^2}{\rho}, f(\mathbf{x}_0) - f^* \leq \Delta_f$, if set set the hyperparameters as in 7 with $\delta_0 = \frac{\delta}{384\Delta_f}\sqrt{\frac{\epsilon^3}{\rho}}$ and choose $\mu = \tilde{\mathcal{O}}(\frac{\epsilon^{1/2}}{d^{1/4}})$ in Line 3 and 16, $\mu = \tilde{\mathcal{O}}(\frac{\epsilon^{13/8}}{d^{1/2}})$ in Line 11 of Algorithm 3. Then with probability at least $1 - \delta$, one of the iterates of Algorithm 3 will be an ϵ-approximate SOSP. The total number of function query is no more than*

$$\mathcal{O}\left(\frac{d\Delta_f\ell^{1/2}\rho^{1/4}}{\epsilon^{7/4}}\log(\frac{\ell\sqrt{d}}{\delta_0\sqrt{\rho\epsilon}})\right).$$

Remark 4. *Similar to Algorithm 1, we can also add an termination condition for Algorithm 3: Once the pre-condition of random perturbation step is reached, record the current iterate point $\mathbf{x}_{t_0}$*

and the current function value $f(\mathbf{x}_{t_0})$ before adding the random perturbation. If the decrease of the function is less than $\frac{1}{384}\sqrt{\frac{\epsilon^3}{\rho}}$ after $\mathscr{T}'$ iterations, then, with high probability $\mathbf{x}_{t_0}$ is an ϵ-approximate SOSP according to Lemma 8.

4 Numerical Experiments

In this section, we conduct several numerical experiments to verify the effectiveness of the proposed methods for escaping saddle points and the efficiency compared with the existing methods. Specially, we run zeroth-order perturbed accelerated gradient descent (Algorithm 1) and zeroth-order perturbed accelerated gradient descent with accelerated negative curvature finding (Algorithm 3) against the perturbed approximate gradient descent (PAGD) and the random search power iteration (RSPI) method. All experiments are performed on a computer with a six-core Intel Core i5-10500 CPU.

4.1 Cubic Regularization Problem

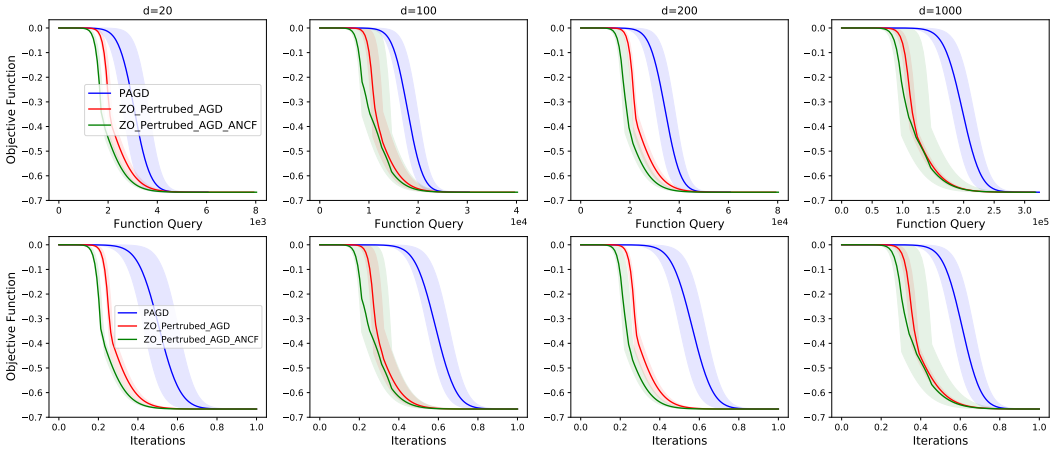


Figure 1: Performance of different algorithms to minimize the cubic regularization problem with growing dimensions. Confidence intervals show mini-max intervals over ten runs

We first consider the cubic regularization problem (Liu et al., 2018), which is defined as:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^\top\mathbf{A}\mathbf{x} + \frac{1}{6}\|\mathbf{x}\|^3. \tag{8}$$

Above, $\mathbf{A}$ is a randomly generated diagonal matrix with only one diagonal entry is -1 and the rest diagonal entries are uniformly distributed between $[1, 2]$. So that with increase of the dimension, the negative curvature directions that can escape from the saddle point will be more difficult to explore. In this experiment, we set $\epsilon = 10^{-2}$. To test the ability of different algorithms to escape from saddle points, we initialize all algorithms at a strict saddle point

$$\mathbf{x}_0 = (0, \ldots, 0)^\top$$

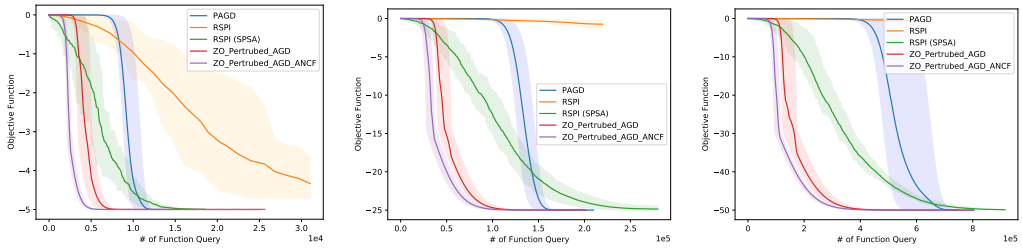
with $\lambda_{\min}(\nabla^2 f(\mathbf{x}_0)) = -1$.

In this experiment, we run Algorithm 1, 3, PAGD on the above cubic regularization problem from a strict saddle point. For Algorithm 1 and 3, the parameter settings basically follow Eq.5 and Eq.7. Specifically, we choose $\epsilon = 0.001$ and the perturbation radius r and r' are set to 0.001. The Lipschitz constants ℓ and ρ are selected based on a coarse grid search. Since all algorithms have certain randomness, we repeatedly run each algorithm multiple times and report the averaged function value versus the averaged number of function queries and the number of iterations in Figure 2.

The results in Fig. 1 illustrate that Algorithm 1, 3 can escape saddle points using less iterations than PAGD and converge faster than PAGD. On the other hand, in all dimensions, the number of

iterations for escaping saddle points are almost the same. This verifies the result in Lemma 6 and 7 that the number of iterations of Algorithm 1, 3 are only log dependent on the dimension d.

4.2 QUARTIC FUNCTION



(a) d=20 (b) d=100 (c) d=200

Figure 2: Performance of different algorithms to minimize the quartic function with growing dimensions. Confidence intervals show mini-max intervals over ten runs.

Then we consider the following quartic function (Lucchi et al., 2021),

$$f(x_1, x_2, \ldots, x_d, y) = \frac{1}{4}\sum_{i=1}^{d} x_i^4 - y\sum_{i=1}^{d} x_i + \frac{d}{2}y^2 \tag{9}$$

which has a strict saddle point at

$$\mathbf{x}_0 = (0, \ldots, 0)^\mathsf{T}$$

and two global minima at $(1, \ldots, 1)^\mathsf{T}$ and $(-1, \ldots, -1)^\mathsf{T}$.

In this experiment, we run Algorithm 1, 3, perturbed approximate gradient descent (PAGD) and Random Search Power Iteration (RSPI) on the above quartic function staring from its saddle point. Especially, we also run an acceleration version of RSPI, which replaces the finite difference gradient estimator in RSPI by the SPSA estimator (Spall et al., 1992). The parameter settings of PAGD are taken from Vlatakis-Gkaragkounis et al. (2019) and the parameters of RSPI are taken from the appendix of Lucchi et al. (2021). For Algorithm 1 and 3, the parameter settings basically follow Eq.5 and Eq.7. Specifically, we choose $\epsilon = 10^{-4}$ and the perturbation radius r and r' are set to 0.01. The Lipschitz constants ℓ and ρ are selected based on a coarse grid search. Since all algorithms have certain randomness, we repeatedly run each algorithm multiple times and report the averaged function value versus the averaged number of function queries in Figure 2.

The results in Fig.2 illustrate that both Algorithms 1 and 3 can efficiently escape saddle points and converge quickly to the global minimum. Note that, for all dimensions, Algorithms 1 and 3 escape saddle points with fewer function queries than PAGD. This verifies the theoretical result that algorithms 1 and 3 take $\tilde{\Theta}(\sqrt{\kappa})$ iterations for escaping saddle points when the initial point is a saddle point, while PAGD takes $\tilde{\Theta}(\kappa)$ iterations. For high dimensional problems, the computational cost of RSPI for escaping saddle points is expensive. In contrast, RSPI with SPSA estimator is much more efficient.

5 CONCLUSION

In this paper, we study the complexity of two zeroth-order AGD based algorithms for escaping saddle points and converging to SOSPs. The first method is a zeroth-order version of the perturbed AGD which uses the central finite difference version of the coordinate-wise gradient estimator. The second method extracts accelerated negative curvature findings by using the finite difference of two coordinate-wise gradient estimators. Both methods improve the function query complexity of prior zeroth-order methods for converging to SOSPs.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. Neon2: Finding local minima via first-order oracles. *Advances in Neural Information Processing Systems*, 31, 2018.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 154–169, 2018.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. "convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. In *International conference on machine learning*, pp. 654–663. PMLR, 2017.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1):71–120, 2020.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points ii: first-order methods. *Mathematical Programming*, 185(1):315–355, 2021.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui 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*, pp. 15–26, 2017.

Krzysztof Choromanski, Mark Rowland, Vikas Sindhwani, Richard Turner, and Adrian Weller. Structured evolution with compact architectures for scalable policy optimization. In *International Conference on Machine Learning*, pp. 970–978. PMLR, 2018.

Prateek Jain, Chi Jin, Sham Kakade, and Praneeth Netrapalli. Global convergence of non-convex gradient descent for computing matrix squareroot. In *Artificial Intelligence and Statistics*, pp. 479–488. PMLR, 2017.

Kaiyi Ji, Zhe Wang, Yi Zhou, and Yingbin Liang. Improved zeroth-order variance reduced algorithms and analysis for nonconvex optimization. In *International conference on machine learning*, pp. 3100–3109. PMLR, 2019.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pp. 1724–1732. PMLR, 2017.

Chi Jin, Lydia T Liu, Rong Ge, and Michael I Jordan. On the local minima of the empirical risk. *Advances in Neural Information Processing Systems*, 31, 2018a.

Chi Jin, Praneeth Netrapalli, and Michael I Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Conference On Learning Theory*, pp. 1042–1085. PMLR, 2018b.

Gangshan Jing, He Bai, Jemin George, Aranya Chakrabortty, and Piyush K Sharma. Asynchronous distributed reinforcement learning for lqr control via zeroth-order block coordinate descent. *arXiv preprint arXiv:2107.12416*, 2021.

Huan Li and Zhouchen Lin. Restarted nonconvex accelerated gradient descent: No more polylogarithmic factor in the $o(\epsilon^{-7/4})$ complexity. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12901–12916. PMLR, 17–23 Jul 2022.

Mingrui Liu, Zhe Li, Xiaoyu Wang, Jinfeng Yi, and Tianbao Yang. Adaptive negative curvature descent with applications in non-convex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Aurelien Lucchi, Antonio Orvieto, and Adamos Solomou. On the second-order convergence properties of random search methods. *Advances in Neural Information Processing Systems*, 34, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 506–519, 2017.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

James C Spall et al. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992.

Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. *Foundations of Computational Mathematics*, 18(5):1131–1198, 2018.

Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:742–749, 07 2019.

Emmanouil-Vasileios Vlatakis-Gkaragkounis, Lampros Flokas, and Georgios Piliouras. Efficiently avoiding saddle points with zero order methods: No gradients required. *Advances in Neural Information Processing Systems*, 32, 2019.

Haishan Ye, Zhichao Huang, Cong Fang, Chris Junchi Li, and Tong Zhang. Hessian-aware zeroth-order optimization for black-box adversarial attack. *arXiv preprint arXiv:1812.11377*, 2018.

Chenyi Zhang and Tongyang Li. Escape saddle points by a simple gradient-descent based algorithm. *Advances in Neural Information Processing Systems*, 34:8545–8556, 2021.