HOUDINI: Escaping from Moderately Constrained Saddles

Dmitrii Avdiukhin[1] and **Grigory Yaroslavtsev**[2]
[1]Indiana University, Bloomington, IN
[2]George Mason University, Fairfax, VA
davdyukh@iu.edu, grigory@grigory.us

Abstract

We give polynomial time algorithms for escaping from high-dimensional saddle points under a moderate number of constraints. Given gradient access to a smooth function $f\colon \mathbb{R}^d \to \mathbb{R}$ we show that (noisy) gradient descent methods can escape from saddle points under a logarithmic number of inequality constraints. While analogous results exist for unconstrained and equality-constrained problems, we make progress on the major open question of convergence to second-order stationary points in the case of inequality constraints, without reliance on NP-oracles or altering the definitions to only account for certain constraints. Our results hold for both regular and stochastic gradient descent.

1 Introduction

Achieving convergence of gradient descent to an (approximate) local minimum is a central question in non-convex optimization for machine learning. In recent years, breakthrough progress starting with the work of [Ge *et al.*, 2015] has led to a flurry of results in this area (see e.g. [Jin *et al.*, 2017; Du *et al.*, 2017; Mokhtari *et al.*, 2018; Jin *et al.*, 2018; Carmon *et al.*, 2017; Carmon and Duchi, 2018; Staib *et al.*, 2019; Carmon and Duchi, 2020; Jin *et al.*, 2021]), culminating in almost optimal bounds [Zhang and Li, 2021]. However, despite this success a key open question of [Ge *et al.*, 2015] still remains unanswered – can gradient methods efficiently escape from saddle points in *constrained* non-convex optimization? In fact, even basic linear inequality constraints still remain an obstacle: "Dealing with inequality constraints is left as future work" [Ge *et al.*, 2015][1]. This is due to the NP-hardness of the related copositivity problem [Murty and Kabadi, 1987], which corresponds to the case when the number of constraints is linear in the dimension. In this paper, we make progress on this open question in the case when the number of constraints depends moderately on the dimension.

Consider a feasible set defined by k linear inequality constraints: $S = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{A}\mathbf{x} \le \mathbf{b}\}$, where $\mathbf{A} \in \mathbb{R}^{k\times d}$ and $\mathbf{b} \in \mathbb{R}^k$. Let $\mathcal{B}_d(\mathbf{x}, r)$ be a d-dimensional closed ball of radius r centred at $\mathbf{x}$. We write $\mathcal{B}(\mathbf{x}, r)$ when the dimension is clear from the context and drop $\mathbf{x}$ when $\mathbf{x} = \mathbf{0}$. Our goal is to find $\min_{x\in S} f(x)$ for the objective function $f\colon \mathbb{R}^d \to \mathbb{R}$ over S. We first introduce the standard smoothness assumption.

Assumption 1 (Smoothness)**.** *The objective function f satisfies the following properties:*

1. *(First order) f has an L-Lipschitz gradient (f is L-smooth): $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.*
2. *(Second order) f has a ρ-Lipschitz Hessian: $\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\| \le \rho\|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.*

Definition 1 (Local minimum)**.** *For $S \subseteq \mathbb{R}^d$, let $f : \mathbb{R}^d \to \mathbb{R}$. A point $\mathbf{x}^\star$ is a local minimum of f in S if and only if there exists $r > 0$ such that $f(\mathbf{x}) \ge f(\mathbf{x}^\star)$ for all $\mathbf{x} \in S \cap \mathcal{B}(\mathbf{x}^\star, r)$.*

Since finding a local minimum is NP-hard even in the unconstrained case (see e.g. [Anandkumar and Ge, 2016] and the references within) the notion of a local minimum is typically relaxed as follows.

Definition 2 (Approximate local minimum)**.** *For $S \subseteq \mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}$ a point $\mathbf{x}^\star$ is a (δ, r)-approximate local minimum if $f(\mathbf{x}) \ge f(\mathbf{x}^\star) - \delta$ for all $\mathbf{x} \in S \cap \mathcal{B}(\mathbf{x}^\star, r)$.*

For smooth functions, one can define stationary points in terms of the gradient and the eigenvalues instead:

Definition 3 ([Nesterov and Polyak, 2006; Jin *et al.*, 2021])**.** *A point $\mathbf{x}$ is an ε-second-order stationary point (ε-SOSP) if $\|\nabla f(\mathbf{x})\| < \varepsilon$ and $\lambda_{\min}(\nabla^2 f(\mathbf{x})) > -\sqrt{\rho\varepsilon}$, where $\lambda_{\min}$ denotes the smallest eigenvalue.*

When applying this definition to the constrained case, eigenvectors and eigenvalues are not well-defined since there might be no eigenvectors inside the feasible set, while an escaping direction might exist. Moreover, for $f(\mathbf{x}) = -\frac{1}{2}\|\mathbf{x}\|^2$ and any compact feasible set, the Hessian is $-I$ at any point with $\lambda_{\min}(-I) = -1$. Hence an ε-SOSP doesn't exist according to the Definition 3, even though a local minimum exists. In fact, Definition 3 arises from the Taylor expansion, which justifies the choice of $\sqrt{\rho\epsilon}$ as the bound on the smallest eigenvalue. If the function has a ρ-Lipschitz Hessian:

$$\left| f(\mathbf{x}+\mathbf{h}) - f(\mathbf{x}) - \mathbf{h}^\top \nabla f(\mathbf{x}) - \frac{1}{2}\mathbf{h}^\top \nabla^2 f(\mathbf{x})\mathbf{h} \right| \le \frac{\rho}{6}\|\mathbf{h}\|^3$$

To guarantee that the discrepancy between the function and its quadratic approximation is small relative to δ (from Definition 2), a natural choice of r is $\sqrt[3]{\delta/\rho}$, which bounds the

[1] Using Lagrangian multipliers, equality constraints can be seen as reducing the dimension of the otherwise unconstrained problem.

discrepancy with $\Theta(\delta)$. Therefore, based on the quadratic approximation, one can distinguish a (δ, r)-approximate local minimum from not a $(c\delta, r)$-approximate local minimum for $c < 1$. By setting this r and selecting $\varepsilon = \sqrt[3]{\delta^2 \rho}$, we have $\sqrt{\rho\varepsilon} = \sqrt[3]{\delta\rho^2}$ and for any $\mathbf{h} \in \mathcal{B}(\mathbf{x}, r)$ (see the full version):

$$f(\mathbf{x}+\mathbf{h}) - f(\mathbf{x}) \geq \mathbf{h}^\top \nabla f(\mathbf{x}) + \frac{1}{2}\mathbf{h}^\top \nabla^2 f(\mathbf{x})\mathbf{h} - \frac{\rho}{6}\|\mathbf{h}\|^3 \geq -2\delta$$

Using the ball radius discussed above we arrive at the following version of Definition 2:

Definition 4 (Approximate SOSP). *For $S \subseteq \mathbb{R}^d$, let $f : \mathbb{R}^d \to \mathbb{R}$ be a twice-differentiable function with a ρ-Lipschitz Hessian. A point $\mathbf{x}^\star$ is a δ-second-order stationary point (δ-SOSP) if for $r = \sqrt[3]{\delta/\rho}$:*

$$\inf_{\mathbf{x} \in S \cap \mathcal{B}(\mathbf{x}^\star, r)} f(\mathbf{x}) \geq f(\mathbf{x}^\star) - \delta$$

Note that, while Definition 4 doesn't seemingly use second-order information, our choice of the ball radius guarantees that the function is close to its quadratic approximation. In particular, we can determine whether f is a $c\delta$-SOSP for a constant c by only using second-order information.

1.1 Our Contribution

Our results hold for stochastic gradient descent (SGD):

Assumption 2. *Access to a* stochastic gradient oracle $\mathbf{g}(\mathbf{x})$*:*

1. *(Unbiased expectation)* $\mathbb{E}[\mathbf{g}(\mathbf{x})] = \nabla f(\mathbf{x})$.
2. *(Variance)* $\mathbb{E}[\|\mathbf{g}(\mathbf{x}) - \nabla f(\mathbf{x})\|^2] \leq \sigma^2$.

Our main result is the following theorem which quantifies the complexity of finding an approximate SOSP under a moderate number of linear inequality constraints, showing that this problem is solvable in polynomial time for $k = O(\log d)$. We refer to a function as (L, ρ)-smooth if it satisfies Assumption 1 and simply *second-order smooth* if both smoothness parameters are constant.

Theorem 5. *Let S be a set defined by an intersection of k linear inequality constraints. Let f be a second-order smooth bounded function. Given access to a stochastic gradient oracle satisfying Assumption 2, there exists an algorithm which for any $\delta > 0$ finds a δ-SOSP in $\tilde{O}(\frac{1}{\delta}(d^3 2^k + \frac{d^2\sigma^2}{\delta^{4/3}}))$ time using $\tilde{O}(\frac{d}{\delta}(1 + \frac{d\sigma^2}{\delta^{4/3}}))$ stochastic gradient oracle calls. In the deterministic gradient case ($\sigma = 0$), the time complexity is $\tilde{O}(\frac{d^3 2^k}{\delta})$ and the number of gradient oracle calls is $\tilde{O}(\frac{d}{\delta})$.*

The exponential dependence of time complexity on k in our results (not required in the oracle calls) is most likely unavoidable due to the following hardness result, which implies that when $k = d$ then the complexity of this problem can't be polynomial in d under the standard hardness assumptions.

Remark 6 (Matrix copositivity [Murty and Kabadi, 1987]). *For a quadratic function $f(\mathbf{x}) = \mathbf{x}^T \mathbf{M} \mathbf{x}$ subject to constraints $x_i \geq 0$ for all i, it is NP-hard to decide whether there exists a solution with $f(\mathbf{x}) > 0$.*

Related results in convex optimization are covered in [Boyd and Vandenberghe, 2004; Bubeck and others, 2015]. Among related results in non-convex optimization, here we only focus on the algorithms using only gradient information.

1.2 Related Work

Unconstrained optimization. Recall that an *ϵ-first-order stationary point* (ϵ-FOSP) is defined so that $\|\nabla f(\mathbf{x})\| \leq \epsilon$. Analyses of convergence to an ε-FOSP are a cornerstone of non-convex optimization (see e.g. classic texts [Bertsekas, 1997; Nocedal and Wright, 1999]). Quantitative analysis of convergence to an ϵ-SOSP (Definition 3) started with the breakthrough work by [Ge *et al.*, 2015] further refined in [Jin *et al.*, 2017; Carmon and Duchi, 2018; Jin *et al.*, 2018; Carmon and Duchi, 2020; Jin *et al.*, 2021] and most recently in [Zhang and Li, 2021], who show an almost optimal bound. Due to the prevalence of SGD in deep learning, stochastic methods have attracted the most attention (see [Allen-Zhu, 2018; Allen-Zhu and Li, 2018; Fang *et al.*, 2018; Tripuraneni *et al.*, 2018; Xu *et al.*, 2018; Zhou and Gu, 2020; Zhou *et al.*, 2020] for the case of Lipschitz gradients and [Ge *et al.*, 2015; Daneshmand *et al.*, 2018] for non-Lipschitz gradients). For an in-depth summary of the previous work on unconstrained non-convex optimization we refer the reader to [Jin *et al.*, 2021].

Constrained optimization. The case of equality constraints is typically reducible to the unconstrained case by using Lagrangian multipliers (see e.g. [Ge *et al.*, 2015]). However, the general constrained case is substantially more challenging since even the definitions of stationarity require a substantial revision. For first-order convergence a rich literature exists, covering projected gradient, Frank-Wolfe, cubic regularization, etc (see e.g. [Mokhtari *et al.*, 2018] and the references within). For second-order convergence, the landscape of existing work is substantially sparser due to NP-hardness (Remark 6, [Murty and Kabadi, 1987]). A large body of work focuses on achieving convergence using various forms of NP-oracles (see e.g. [Bian *et al.*, 2015a; Cartis *et al.*, 2018; Mokhtari *et al.*, 2018; Haeser *et al.*, 2019; Nouiehed and Razaviyayn, 2020]), while another approach is to define stationarity in terms of tight constraints only [Avdiukhin *et al.*, 2019; Lu *et al.*, 2020].

Relationship with other definitions of SOSP. As discussed in Remark 6, second-order constrained optimization is NP-hard due to the hardness of the matrix copositivity problem. Definitions of constrained SOSP in the previous work fall into two categories: 1) definitions of scaled stationary points, 1) definitions that only consider active constraints ("active constraints only" definitions), 2) definitions that preserve the NP-hardness of the problem and rely on NP-oracles to achieve polynomial-time convergence:

1. (Scaled) For the constraints $\mathbf{x} \geq 0$, [Bian *et al.*, 2015b; O'Neill and Wright, 2020] consider the definition of scaled SOSP. The idea is to scale i-th coordinate by x_i: i.e. instead of bounding $\nabla f(\mathbf{x})$ it bounds $X\nabla f(\mathbf{x})$, where $X = \operatorname{diag}(x_1, \ldots, x_d)$, and instead of eigenvalues $\nabla^2 f(\mathbf{x})$ it considers eigenvalues of $X\nabla^2 f(\mathbf{x})X$. For this definition, $\mathbf{x} \geq 0$ restricts possible applications.

2. ("Active constraints only") In [Avdiukhin *et al.*, 2019; Lu *et al.*, 2020] definitions analogous to Definition 3, and the second-order conditions are given with respect to the set of *active* (i.e. tight for the current iterate) con-

straints. This bypasses the NP-hardness since the point at which the hardness of the copositivity problem applies now becomes a stationary point by definition.

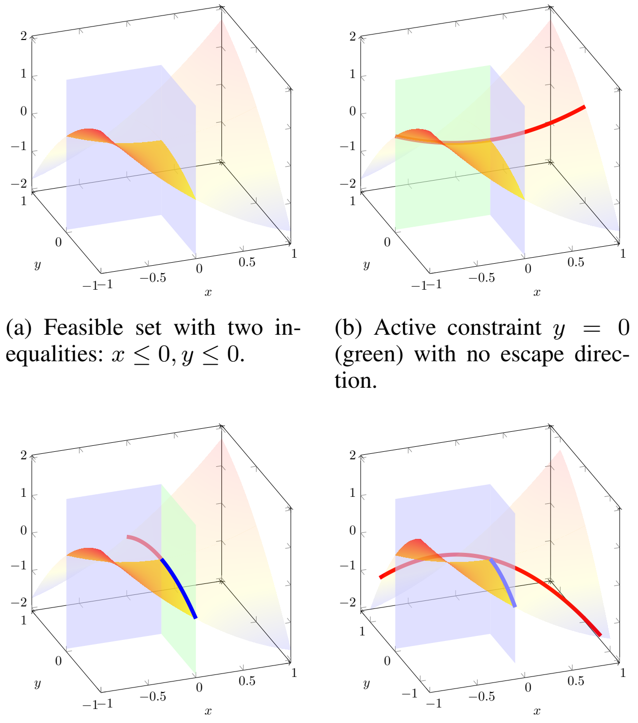
3. (NP-hard) In the results relying on NP-oracles (e.g. [Bian *et al.*, 2015a; Mokhtari *et al.*, 2018; Haeser *et al.*, 2019; Nouiehed and Razaviyayn, 2020]) the complexity is shifted on solving black-box quadratic optimization problems of a certain type. A key advantage of these types of approaches is that they can handle an arbitrary number of constraints and hence promising in certain machine learning applications.

What is currently lacking in the state of the art is a quantitative analysis of the complexity of convergence to a second-order stationary point, which shows full dependence on both the dimension and accuracy while defining stationarity with respect to the full set of constraints, instead of just active constraints only[2]. Our goal in Theorem 5 is to address this gap and give such an analysis.

1.3 Technical Overview

We address the NP-hardness of the copositivity problem by focusing on the case of a moderate number of constraints and arguing that it can be addressed using gradient-based methods. In order to streamline the presentation we first focus on the key challenge of escaping from a saddle point in a corner defined by the constraints when the underlying function is simply quadratic (Section 2). This is already enough to capture some of the key contributions, while more technical details and the full algorithm are given in Section 3.

Quadratic corner saddle point (Section 2). In this simplified scenario, the NP-hardness comes from the fact that the point we aim to find can lie in the intersection of an arbitrary subset of constraints. By doing an exhaustive search over this set of constraints (Algorithm 1) and enforcing them throughout the search process we are able to reduce to a setting similar to the equality-constrained case (Algorithm 2). We show different subsets of constraints enforced by the algorithm in an example in Figure 1. The key challenge is making this argument formal and arguing that this process converges to a constrained approximate SOSP as in Definition 4. This relies on performing a robust analysis of the properties of the points which lead to the hardness of copositivity. In particular, we show that after we guess the set of constraints correctly, the problem reduces to finding the smallest eigenvector (Lemma 9). Exact error analysis of the eigenvector process (Lemma 10, Lemma 11) is then required to complete the proof of the main theorem (Theorem 8).



(a) Feasible set with two inequalities: $x \leq 0, y \leq 0$.

(b) Active constraint $y = 0$ (green) with no escape direction.

(c) Active constraint $x = 0$ (green) with an escape direction $(0, -1)$. As shown in Lemma 10, out of two escape directions $(0, -1)$ and $(0, 1)$ in this constraint, at least one $(0, -1)$ (blue) lies in the feasible set, and is found by the algorithm.

(d) When no constraints are active, the algorithm finds red directions (negative eigenvectors) outside the feasible set. Note that escape directions with no active constraints exist (blue). Lemma 10 guarantees that we find them in some constrained space (Figure 1c).

Figure 1: Function $f(x, y) = (\frac{\sqrt{3}}{2}x + \frac{1}{2}y)^2 - (-\frac{1}{2}x + \frac{\sqrt{3}}{2}y)^2$ (composition of $x^2 - y^2$ and rotation by ${}^{\pi}\!/_{6}$).

General case (Section 3). In the algorithm for the general case, we address the three assumptions made in the quadratic corner saddle point case, while also handling the stochasticity in the gradient. The latter part is standard and is handled via variance reduction (see the full version). The full algorithm iterates the escape subroutine (Algorithm 3) until an escaping point is found. The escape subroutine first approximates the Hessian matrix using the gradient oracle and then performs an exhaustive search over the set of active constraints at the escaping point in a way similar to the quadratic corner case. After the correct subset of constraints is fixed the current iterate needs to be projected on this set of constraints, which also necessitates a recomputation of various related parameters. When this is done the problem is solved by a subroutine Algorithm 4, analogous to Algorithm 2 from the quadratic

[2] While unrefereed manuscripts ([Hsia and Sheu, 2018] and [Nouiehed *et al.*, 2020], relying on [Hsia and Sheu, 2018] as a subroutine) do contain related results, our work differs in a number of important aspects. [Hsia and Sheu, 2018] and [Nouiehed *et al.*, 2020] require access to the exact gradient and Hessian. We only assume access to the stochastic gradient oracle. Furthermore, [Hsia and Sheu, 2018] assumes access to matrix diagonalization. However, the diagonalization can only be found approximately, which compromises the stability of this approach, especially with respect to the linear term transformations. Our approach handles this issue by appropriate perturbation and using known results for matrix diagonalization. Finally, compared with [Hsia and Sheu, 2018], our main contribution is focused on solving a substantially different problem. While [Hsia and Sheu, 2018] find a global minimum of a quadratic problem, we find an approximate local minimum of an arbitrary smooth non-convex function.

corner case.

However, since the function is no longer quadratic and the gradient can be large, several modifications are required. The algorithm tries to find an escaping direction within a ball of radius r, within which the function is well approximated by a quadratic by the smoothness assumption (as discussed above). First, the algorithm tries to escape using the gradient term. If that doesn't work then we consider two cases: 1) the solution is inside the ball of radius $r = \sqrt[3]{\delta/\rho}$ (from Definition 4), 2) the solution is on the boundary of this ball. In the first case, the solution is a critical point and hence can be found using the Newton step. In the second case, we diagonalize the Hessian using an orthogonal transformation. This gives a quadratic function with a linear term whose critical points on the boundary of the ball can be found explicitly (up to the required precision) as roots of the corresponding polynomial. We note that the diagonalization performed in this case is the most computationally expensive step in the algorithm, resulting in polynomial dependence on the dimension.

Notation. For a set S let $\operatorname{Int} S$ be its interior and ∂S be its boundary. For $\mathbf{x} \in \mathbb{R}^d$ and $S \subseteq \mathbb{R}^d$, $\operatorname{Proj}_S(\mathbf{x}) = \operatorname{argmin}_{\mathbf{y}\in S} \|\mathbf{x} - \mathbf{y}\|$ is the projection of $\mathbf{x}$ on S. For a square matrix $\mathbf{M}$ with eigenvalues $\lambda_1 \le \ldots \le \lambda_d$, we denote $\lambda_{\min}(\mathbf{M}) = \lambda_1$ and $\lambda_{|max|}(\mathbf{M}) = \max(|\lambda_1|, |\lambda_d|)$. For $S = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{A}\mathbf{x} \le \mathbf{b}\}$ and $\mathbf{x} \in S$, we say that i-th constraint is active at $\mathbf{x}$ if $\mathbf{A}_i^\top \mathbf{x} = b_i$, where $\mathbf{A}_i$ is the i-th row of $\mathbf{A}$. $\tilde{O}$ notation hides polylogarithmic dependence on all parameters, including error probability.

2 Quadratic Corner Saddle Point Case

We introduce the key ideas of the analysis in a simplified setting when: 1) the function f is quadratic, 2) the gradient is small, 3) the current iterate is located in a corner of the constraint space. Intuitively, this represents the key challenge of the constrained saddle escape problem since its NP-hardness comes from the hardness of the matrix copositivity problem in Remark 6 (i.e. the function is exactly quadratic and has no gradient at the current iterate which lies in the intersection of all constraints). We refer to this setting as the Quadratic Corner Saddle Point problem defined formally below. By shifting the coordinate system, w.l.o.g. we can assume that the saddle point is $\mathbf{0}$ and $f(\mathbf{0}) = 0$[3]. If $\mathbf{0}$ is a (δ, r)-QCSP (as defined below), the objective can be decreased by δ within a ball of radius r.

Definition 7 (Quadratic Corner Saddle Point)**.** *Let $S = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \le \mathbf{0}\}$. For $\delta, r > 0$ and function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top\mathbf{M}\mathbf{x}$, we say that a point $\mathbf{0}$ is a:*

- *(δ, r)-Quadratic Corner Saddle Point ((δ, r)-QCSP) if $\min_{\mathbf{x}\in\mathcal{B}(r)\cap S} f(\mathbf{x}) < -\delta$.*
- *(δ, r)-boundary QCSP if $\min_{\mathbf{x}\in\mathcal{B}(r)\cap\partial S} f(\mathbf{x}) < -\delta$.*

In this section, we show how to escape from a (δ, r)-QCSP (see the full version for the full proof):

Theorem 8 (Quadratic Corner Saddle Point Escape)**.** *Let $\delta, r > 0$. Let $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top\mathbf{M}\mathbf{x}$ with $\lambda_{|max|}(\mathbf{M}) \le L$ and let $S = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \le \mathbf{0}\}$ be defined by k linear inequalities. If $\mathbf{0}$ is a (δ, r)-QCSP, then Algorithm 1 with probability at least $1 - \xi$ finds a point $\mathbf{x} \in S \cap \mathcal{B}(r)$ with $f(\mathbf{x}) < -\Omega(\delta)$ using $O\left(\frac{Lr^2 k 2^k}{\delta}\log\frac{1}{\xi}\right)$ deterministic gradient oracle calls.*

Algorithm 1: HOUDINIESCAPECORNER: Escaping from a corner for a quadratic function

```
input: Starting point $\mathbf{x}$, feasible set
        $S = \{\mathbf{y} \mid \mathbf{A}(\mathbf{y} - \mathbf{x}) \le \mathbf{0} \in \mathbb{R}^k\}$
parameters: $\delta$ and $r$ from definition of $(\delta, r)$-QCSP
for $\mathcal{I} \in 2^{[k]}$ – every subset of constraints do
    $\mathcal{A} \leftarrow \{\mathbf{y} \mid \mathbf{A}_i^\top(\mathbf{y} - \mathbf{x}) = 0$ for $i \in \mathcal{I}\}$, where $\mathbf{A}_i$
      is the $i$-th row of $\mathbf{A}$        // Optimize in $\mathcal{A}$
    $\mathbf{y} \leftarrow$ FINDINSIDECORNER$(\mathbf{x}, \mathcal{A})$
    if $\mathbf{y} \in S$ and $f(\mathbf{y}) < f(\mathbf{x}) - \frac{\delta}{2}$ then
        return $\mathbf{y}$
return $\perp$
```

Algorithm 2: FINDINSIDECORNER$(\mathbf{x}, \mathcal{A})$

```
input: Corner $\mathbf{x}$, affine subspace $\mathcal{A}$ with $\mathbf{x} \in \mathcal{A}$
parameters: $\delta$ and $r$ from Definition 7, step size
    $\eta = \frac{1}{L}$, number of iterations $T = \tilde{O}(\frac{Lr^2}{\delta})$
Sample $\xi \sim \mathcal{N}(\mathbf{0}, I)$, $\mathbf{x}_0 \leftarrow \operatorname{Proj}_{\mathcal{A}}(\mathbf{x} + \xi)$
for $t = 0, \ldots, T - 1$ do
    // Power method step
    $\mathbf{x}_{t+1} \leftarrow \operatorname{Proj}_{\mathcal{A}}(\mathbf{x}_t - \eta(\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x})))$
$\mathbf{e} \leftarrow r\frac{\mathbf{x}_T - \mathbf{x}}{\|\mathbf{x}_T - \mathbf{x}\|}$
return $\mathbf{x} + \mathbf{e}$
```

For the rest of the section, we assume that $\mathbf{0}$ is a (δ, r)-QCSP, i.e. $\min_{\mathbf{x}\in S\cap\mathcal{B}(r)} f(\mathbf{x}) < -\delta$. We consider two cases depending on whether $\mathbf{0}$ is a (δ, r)-boundary QCSP.

Case 1: $\mathbf{0}$ is a (δ, r)-boundary QCSP. For a subset of inequality constraints $\mathcal{I} \subseteq [k]$ we define the subspace where these constraints are active: $\mathcal{A}_{\mathcal{I}} = \{\mathbf{x} \mid \mathbf{A}_i^\top \mathbf{x} = 0$ for all $i \in \mathcal{I}\}$. Let $\mathcal{I}$ be a maximal[4] subset of constraints such that $\min_{\mathbf{x}\in\mathcal{A}_{\mathcal{I}}\cap\mathcal{B}(r)} f(\mathbf{x}) < -\delta$. If $\mathbf{P}$ is a projection operator on $\mathcal{A}_{\mathcal{I}}$, it suffices to optimize $g(\mathbf{x}) := f(\mathbf{P}\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top(\mathbf{P}\mathbf{M}\mathbf{P})\mathbf{x}$. Therefore, we reduced the original problem to minimizing a different quadratic form in the same feasible set. For any $i \in \mathcal{I}$, $\mathbf{A}_i^\top\mathbf{P}\mathbf{x} \le 0$ holds trivially, since $\mathbf{A}_i^\top\mathbf{y} = 0$ for any $\mathbf{y} \in \mathcal{A}$, and hence constraints from $\mathcal{I}$ can be ignored. If a constraint not from $\mathcal{I}$ is active in $\mathbf{P}\mathbf{x}$, then $f(\mathbf{P}\mathbf{x}) \ge -\delta$, since $\mathcal{I}$ is a maximal subset of constraints with $\min_{\mathbf{x}\in\mathcal{A}_{\mathcal{I}}\cap\mathcal{B}(r)} < -\delta$. Therefore, this reduces Case 1 to the next case.

Case 2: $\mathbf{0}$ is not a (δ, r)-boundary QCSP. In this case, we show that any $\mathbf{x} \in \mathcal{B}(r)$ with $f(\mathbf{x}) < -\delta$ must lie in S, and for $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top\mathbf{M}\mathbf{x}$ it suffices to find the eigenvector corresponding to the smallest eigenvalue of $\mathbf{M}$. We first show that there exists an eigenvector improving the objective.

[3] Algorithms 1 and 2 don't require saddle point $\mathbf{x}$ to be $\mathbf{0}$. All the statements are trivially adapted for the case when $\mathbf{x}$ is not $\mathbf{0}$

[4] As we don't know $\mathcal{I}$, Algorithm 1 tries all subsets of constraints.

Lemma 9. *Let $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{M}\mathbf{x}$ and $S = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{0}\}$. If $\mathbf{0}$ is not a (δ, r)-boundary QCSP for $\delta, r > 0$, then the following statements are equivalent:*

1. *$\mathbf{0}$ is (δ, r)-QCSP, i.e. $\min_{\mathbf{x}\in S\cap\mathcal{B}(r)} f(\mathbf{x}) < -\delta$.*
2. *There exists an eigenvector $\mathbf{e}$ of $\mathbf{M}$ such that $\mathbf{e} \in \operatorname{Int} S \cap \partial\mathcal{B}(r)$ and $f(\mathbf{e}) < -\delta$.*

Finding an exact eigenvector might be impossible. Hence, we show that finding $\mathbf{x} \in \mathcal{B}(r)$ with $f(\mathbf{x}) < -\delta$ suffices, since either $\mathbf{x}$ or $-\mathbf{x}$ are in S.

Lemma 10. *Let $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{M}\mathbf{x}$ and $S = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{0}\}$. For $\delta, r > 0$ and $\hat{\mathbf{x}} \in \partial\mathcal{B}(r)$, if $f(\hat{\mathbf{x}}) < -\delta$ and the following conditions hold, then either $\hat{\mathbf{x}} \in S$ or $-\hat{\mathbf{x}} \in S$:*

1. *$\min_{\mathbf{x}\in S\cap\mathcal{B}(r)} f(\mathbf{x}) < -\delta$, i.e. $\mathbf{0}$ is a (δ, r)-QCSP,*
2. *$\min_{\mathbf{x}\in \partial S\cap\mathcal{B}(r)} f(\mathbf{x}) \geq -\delta$, i.e $\mathbf{0}$ is not a (δ, r)-boundary QCSP,*

To show Lemma 10, we use the fact that, by Lemma 9, there exists an eigenvector $\mathbf{e}$ with $\mathbf{e} \in \operatorname{Int} S \cap \partial\mathcal{B}(r)$ and $f(\mathbf{e}) < -\delta$. For the sake of contradiction, if both $-\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}$ are not in S, at least one of them has a non-negative inner product with $\mathbf{e}$. W.l.o.g. we assume $\hat{\mathbf{x}}^\top \mathbf{e} \geq 0$, and we consider an arc on $\partial\mathcal{B}(r)$ connecting $\hat{\mathbf{x}}$ and $\mathbf{e}$. Since $\mathbf{e} \in S$ and $\hat{\mathbf{x}} \notin \mathbf{e}$, the arc intersects ∂S at some point. We show that any point $\mathbf{x}$ on the arc has $f(\mathbf{x}) < -\delta$, and hence this also holds for the point on the boundary, contradicting the assumption that $\mathbf{0}$ is not a (δ, r)-boundary QCSP, finishing the proof.

The main idea behind the algorithm is that Algorithm 2 emulates the power method on matrix $I - \frac{\mathbf{M}}{L}$. Hence, for any ε it allows us to find a vector $\mathbf{x}$ such that

$$\frac{\left|\mathbf{x}^\top (I - \frac{\mathbf{M}}{L})\mathbf{x}\right|}{\|\mathbf{x}\|^2} \geq (1-\varepsilon)\lambda_{|max|}\left(I - \frac{\mathbf{M}}{L}\right).$$

Since $\lambda_{|max|}(\mathbf{M}) \leq L$, all eigenvalues of $I - \frac{\mathbf{M}}{L}$ are positive, and hence the power method approximates the eigenvector corresponding to the largest eigenvalue of $I - \frac{\mathbf{M}}{L}$, and hence to the smallest eigenvalue of $\mathbf{M}$.

We know that $f(\mathbf{x}) < -\delta$, and we aim to find $\mathbf{x} \in \mathcal{B}(r)$ with $f(\mathbf{x}) < -(1-\varepsilon)\delta$ for a constant ε. Since there exists an eigenvector $\mathbf{e} \in \partial\mathcal{B}(r)$ of $\mathbf{M}$ with $\frac{1}{2}\mathbf{e}^\top \mathbf{M}\mathbf{e} < -\delta$, we have $\lambda_{\min}(\mathbf{M}) < \frac{-2\delta}{\|\mathbf{e}\|^2} = \frac{-2\delta}{r^2}$, and hence the largest eigenvalue of $I - \frac{\mathbf{M}}{L}$ is at least $1 - \frac{\lambda_{\min}(\mathbf{M})}{L} \geq 1 + \frac{2\delta}{Lr^2}$. Finding $\mathbf{x}$ with $\frac{1}{2}\mathbf{x}^\top \mathbf{M}\mathbf{x} < -(1-\varepsilon)\delta$ is equivalent to finding $\mathbf{x}$ with $\mathbf{x}^\top(I - \frac{\mathbf{M}}{L})\mathbf{x} \geq (1 + \frac{2(1-\varepsilon)\delta}{Lr^2})\|\mathbf{x}\|^2$, and the power method achieves this in $O(\log d + \frac{Lr^2}{\varepsilon\delta})$ iterations (see proof of Lemma 11 in the full version).

Lemma 11. *Let $\delta, r > 0$, $\mathbf{x} \in \mathbb{R}^d$ and $\varepsilon \in (0, 1)$. Let $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{M}\mathbf{x}$ with $\lambda_{|max|}(\mathbf{M}) \leq L$. Let $\mathcal{A}$ be a linear subspace of $\mathbb{R}^d$. If $\min_{\mathbf{x}\in\mathcal{A}\cap\mathcal{B}(r)} f(\mathbf{x}) < -\delta$, then Algorithm 2 finds $\mathbf{x} \in \mathcal{A} \cap \partial\mathcal{B}(r)$ with $f(\mathbf{x}) \leq -(1-\varepsilon)\delta$ after $T = \tilde{O}\left(\frac{Lr^2}{\varepsilon\delta}\right)$ iterations w.h.p.*

Finally, we now prove Theorem 8. First, by exhaustive search, we guess a maximal subset of active constraints $\mathcal{I}$ such that subspace $\mathcal{A}_\mathcal{I}$ formed by these linear constraints has $\mathbf{x} \in \mathcal{B}(r) \cap S$ with $f(\mathbf{x}) < \delta$. Using Algorithm 2, we find $\mathbf{y} \in \mathcal{B}(r) \cap \mathcal{A}_\mathcal{I}$ with $f(\mathbf{y}) < -(1-\varepsilon)\delta$. Then $\mathbf{y} \in S$ by Lemma 11, since $\mathcal{I}$ is a maximal subset of constraints with an escape direction.

Algorithm 3: HOUDINIESCAPE$(\mathbf{x}, S, \delta)$: Escaping from a saddle point

```
input : Saddle point $\mathbf{x}$, $\delta$ from definition of $\delta$-SOSP,
        feasible set $S = \{\mathbf{y} \mid \mathbf{A}\mathbf{y} \leq \mathbf{b} \in \mathbb{R}^k\}$
output: either reports that $\mathbf{x}$ is a $\delta$-SOSP or finds
        $\mathbf{u} \in S \cap \mathcal{B}(\mathbf{x}, r)$ with $f(\mathbf{u}) < f(\mathbf{x}) - \Omega(\delta)$
Construct $f'(\mathbf{x}+\mathbf{h})$ – quadratic approximation of
  $f(\mathbf{x}+\mathbf{h})$ – using stochastic gradient oracle calls
for $\mathcal{I}$ – every subset of constraints do
    $\mathcal{A} \leftarrow \{\mathbf{y} \mid \mathbf{A}_i^\top \mathbf{x} = b_i,\ i \in \mathcal{I}\}$, where $\mathbf{A}_i$ is the
      $i$-th row of $\mathbf{A}$            // Optimize in $\mathcal{A}$
    Let $\mathbf{p}$ be the projection of $\mathbf{x}$ on $\mathcal{A}$
    Let $\mathbf{O} \in \mathbb{R}^{d\times\dim\mathcal{A}}$ be an orthonormal basis of $\mathcal{A}$
    Define $g(\mathbf{y}) := f'(\mathbf{p} + \mathbf{O}\mathbf{y})$
    Algorithm 4 tries to find an escape direction for $g$
    if Algorithm 4 finds direction $\mathbf{y}$ then
        return $\mathbf{p} + \mathbf{O}\mathbf{y}$
Didn't find escape direction: report that $\mathbf{x}$ is a
  $\delta$-SOSP
```

Algorithm 4:
FINDINSIDE$(\mathbf{x}, \delta, (\mathbf{M}_\perp, \mathbf{v}_\perp), (r_\perp, S_\perp))$

```
input : $\delta$ from definition of $\delta$-SOSP,
        $g(\mathbf{y}) = \frac{1}{2}\mathbf{y}^\top \mathbf{M}_\perp \mathbf{y} + \mathbf{y}^\top \mathbf{v}_\perp$ – objective in
        $\mathbb{R}^{\dim\mathcal{A}}$,
        $S_\perp \cap \mathcal{B}(r_\perp)$ – feasible set in $\mathbb{R}^{\dim\mathcal{A}}$
output: Escaping direction, if exists

If any of the following candidates lies in the feasible
  set and decreases the objective by $\Omega(\delta)$, return it:
Case 1. Large gradient: $\operatorname{argmin}_{\mathbf{y}\in S_\perp\cap\mathcal{B}(r_\perp)} \mathbf{y}^\top \mathbf{v}_\perp$
Case 2. Solution in the interior: $\mathbf{y} \leftarrow -\mathbf{M}_\perp^{-1}\mathbf{v}_\perp$
Case 3. Solution on the boundary:
    Find orthogonal $\mathbf{Q}$ and diagonal
      $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_{\dim\mathcal{A}})$ such that
      $\mathbf{M}_\perp \approx \mathbf{Q}^\top \mathbf{\Lambda}\mathbf{Q}$
    Let $\tilde{\mathbf{v}} \leftarrow \mathbf{Q}\mathbf{v}$
    We consider points with coordinates $y_i \leftarrow \frac{\tilde{v}_i}{\mu_j - \lambda_i}$
      for some $\mu$
    Find the values of $\mu$ for which the points have norm
      $r_\perp$
    The candidates are the points corresponding to
      these values of $\mu$
If no candidate satisfies the condition, return $\perp$
```

3 Main Result

Our main result is the following theorem that shows how to find a δ-SOSP.

Theorem 12. *Let $S = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ be a set defined by an intersection of k linear inequality constraints. Let f satisfy Assumptions 1 and 2 and let $\min_{\mathbf{x}\in S} f(\mathbf{x}) = f^\star$. Then there exists an algorithm which for $\delta > 0$ finds a δ-SOSP in*

$\tilde{O}(\frac{f(\mathbf{x}_0)-f^\star}{\delta} d^3(2^k+\frac{\sigma^2}{\delta^{4/3}}))$ *time using* $\tilde{O}(\frac{f(\mathbf{x}_0)-f^\star}{\delta}(d+\frac{d^3\sigma^2}{\delta^{4/3}}))$ *stochastic gradient oracle calls.*

Our approach is outlined in Algorithms 3 and 4[5]:

- If $\mathbf{x}$ is not a δ-SOSP, Algorithm 3 finds an escape direction, i.e. a point $\mathbf{y} \in S \cap \mathcal{B}(\mathbf{x}, r)$ which significantly decreases the objective value: $f(\mathbf{y}) < f(\mathbf{x}) - \Omega(\delta)$. Therefore, if $\mathbf{x}_0$ is the initial point, our algorithm requires $O(\frac{f(\mathbf{x}_0)-f^\star}{\delta})$ calls of Algorithm 3.
- Algorithm 3 enumerates all possible sets of active constraints

Recall that we aim to find a δ-SOSP, i.e. $\mathbf{x} \in S$ such that $f(\mathbf{y}) \geq f(\mathbf{x}) - \delta, \forall \mathbf{y} \in S \cap \mathcal{B}(\mathbf{x}, r)$, where $r = \sqrt[3]{\delta/\rho}$.

Similarly to Section 2, we use a quadratic approximation and consider two cases depending on whether the minimizer $\mathbf{y}$ lies in the interior or on the boundary of S. The second case can be reduced to the first in a way similar to Section 2. To find the minimizer in the interior, we consider the following cases in Algorithm 3:

Case 1. When the gradient is large, we ignore the quadratic term, and optimize the function based on the gradient. This covers the situation when the objective is sensitive to the change in the argument.

Case 2. When the optimum lies in $\operatorname{Int} \mathcal{B}(\mathbf{x}, r)$, none of the constraints are active, and hence we can find the unique unconstrained critical point directly.

Case 3. When the optimum lies in $\partial \mathcal{B}(\mathbf{x}, r)$, the only active constraint is $c(\mathbf{y}) = \|\mathbf{y} - \mathbf{x}\|^2 - r^2 = 0$. By KKT conditions, for $\mathbf{y}$ to be the minimizer, there must exist μ such that $\nabla f(\mathbf{y}) = \mu \nabla c(\mathbf{y})$. We show that for each μ, there exists a unique $\mathbf{y}(\mu)$ satisfying the condition above, and only for $O(d)$ values of μ we have $\mathbf{y}(\mu) \in \partial \mathcal{B}(\mathbf{x}, r)$, resulting in $O(d)$ candidate solutions.

We now outline the proof of Theorem 12 (full proof is in the full version). As shown in Section 3.1, we can consider a quadratic approximation of the objective. By guessing which constraints are active at the minimizer $\mathbf{x}^\star$ and enforcing these constraints, we restrict the function to some affine subspace $\mathcal{A}$. By parameterizing $\mathcal{A}$, we eliminate enforced constraints, and, since the rest of the constraints are not active at $\mathbf{x}^\star$, we need to optimize a quadratic function in the intersection of a ball and linear inequality constraints (see the full version).

3.1 Algorithm 3: Quadratic Approximation

The goal is to build a quadratic approximation of the objective with some additional properties (see below). To simplify the presentation, w.l.o.g. (by shifting the coordinate system) we assume that the current saddle point is $\mathbf{0}$ and consider the quadratic approximation of the function:

$$f(\mathbf{x}) \approx f(\mathbf{0}) + \mathbf{x}^\top \nabla f(\mathbf{0}) + \frac{1}{2}\mathbf{x}^\top \nabla^2 f(\mathbf{0})\mathbf{x}.$$

Since f is ρ-Lipschitz and $r = \sqrt[3]{\delta/\rho}$, in $\mathcal{B}(r)$ the quadratic approximation deviates from f by at most $\frac{\delta}{6}$ (see derivation before Definition 4). For a small value ξ (to be specified later), we instead analyze a noisy function

$$f'(\mathbf{x}) = f(\mathbf{0}) + \mathbf{x}^\top \mathbf{v} + \frac{1}{2}\mathbf{x}^\top \mathbf{M}\mathbf{x},$$

where:

1. $\mathbf{v}$ is a perturbed approximation of $\nabla f(\mathbf{0})$. The perturbation guarantees that w.h.p. all coordinates of $\mathbf{v}$ are sufficiently separated from 0 and linear systems of the form $(\mathbf{M} - \mu I)\mathbf{x} = \mathbf{v}$ with $\operatorname{rank}(\mathbf{M} - \mu I) < d$ don't have any solutions, simplifying the analysis. To approximate the gradient, we use algorithm VRSG (see the full version), which w.h.p. estimates the gradient with precision $\tilde{\sigma}$ using $\tilde{O}(\frac{\sigma^2}{\tilde{\sigma}^2})$ stochastic gradient oracle calls.
2. $\mathbf{M}$ is a perturbed approximation of $\nabla^2 f(\mathbf{0})$. The perturbation guarantees that $\mathbf{M}$ is non-degenerate with probability 1. Since the i-th column of $\nabla^2 f(\mathbf{0})$ is by definition $\lim\limits_{\tau \to 0} \frac{\nabla f(\tau \mathbf{e}_i) - \nabla f(\mathbf{0})}{\tau}$, using a sufficiently small τ and approximating $\nabla f(\tau \mathbf{e}_i)$ and $\nabla f(\mathbf{0})$ using VRSG, we find good approximation of the Hessian.

Combining this with the derivation before Definition 4, we show the following:

Lemma 13. *Let f satisfy Assumptions 1 and 2. Let $f'(\mathbf{x}) = f(\mathbf{0}) + \mathbf{x}^\top \mathbf{v} + \frac{1}{2}\mathbf{x}^\top \mathbf{M}\mathbf{x}$, where $\mathbf{v}$ and $\mathbf{M}$ are as in Algorithm 3. For $\delta > 0$, $r = \sqrt[3]{\delta/\rho}$ we have $\|f'(\mathbf{x}) - f(\mathbf{x})\| < \frac{\delta}{2}$ for all $\mathbf{x} \in \mathcal{B}(r)$ w.h.p.*

Reducing Case $\mathbf{x}^\star \in \partial S$ to Case $\mathbf{x}^\star \in \operatorname{Int} S$. Similarly to Section 2, we reduce the case $\mathbf{x}^\star \in \partial S$ to the case $\mathbf{x}^\star \in \operatorname{Int} S$. If $\mathbf{x}^\star \in \partial S$, then there exist a non-empty set $\mathcal{I}$ of constraints active at $\mathbf{x}^\star$. Consider the iteration of Algorithm 3 where constraints from $\mathcal{I}$ are active. These active constraints define an affine subspace $\mathcal{A}$, which e parameterize: if $\mathbf{p} = \operatorname{Proj}_{\mathcal{A}}(\mathbf{x})$ and $\mathbf{O} \in \mathbb{R}^{d \times \dim \mathcal{A}}$ is an orthonormal basis of $\mathcal{A}$, then any point in $\mathcal{A}$ can be represented as $\mathbf{p} + \mathbf{O}\mathbf{y}$ for $\mathbf{y} \in \mathbb{R}^{\dim \mathcal{A}}$. Defining $g(\mathbf{y}) = f'(\mathbf{p} + \mathbf{O}\mathbf{y})$, minimizing f' in $\mathcal{A} \cap S \cap \mathcal{B}(r)$ is equivalent to minimizing g in $S_\perp \cap \mathcal{B}(r_\perp)$, where:

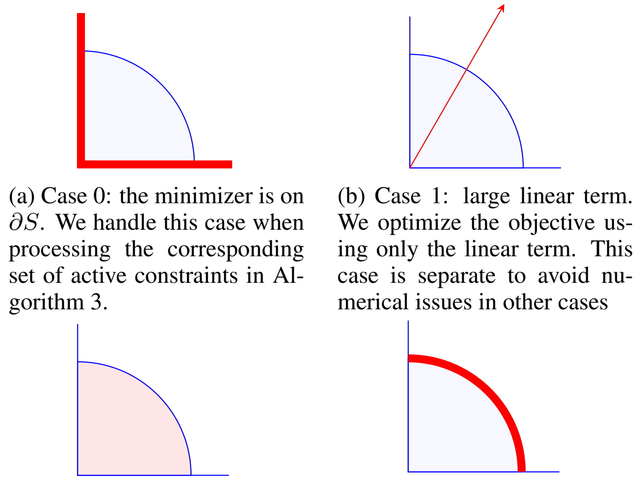
1. $S_\perp$ is a set of points $\mathbf{y} \in \mathbb{R}^{\dim \mathcal{A}}$ such that $\mathbf{p} + \mathbf{O}\mathbf{y} \in S$, namely $S_\perp = \{\mathbf{y} \mid \mathbf{A}_i(\mathbf{p} + \mathbf{O}\mathbf{y}) \leq b_i,\ i \notin \mathcal{I}\}$. Hence, $S_\perp$ is defined by linear inequalities, similarly to S.
2. $r_\perp$ is a radius such that condition $\mathbf{y} \in \mathcal{B}(r_\perp)$ is equivalent to $\mathbf{p} + \mathbf{O}\mathbf{y} \in \mathcal{B}(r)$. Since $\mathbf{p}$ is the projection of $\mathbf{0}$ on $\mathcal{A}$, we have $\mathbf{O}^\top \mathbf{p} = \mathbf{0}$, and hence $\|\mathbf{p} + \mathbf{O}\mathbf{y}\|^2 = \|\mathbf{p}\|^2 + \|\mathbf{O}\mathbf{y}\|^2$. Since $\mathbf{O}$ is an orthonormal basis of $\mathcal{A}$, $\|\mathbf{O}\mathbf{y}\| = \|\mathbf{y}\|$, and hence $r_\perp = \sqrt{r^2 - \|\mathbf{p}\|^2}$.

For $\mathbf{y}^\star$ such that $\mathbf{x}^\star = \mathbf{p} + \mathbf{O}\mathbf{y}^\star$, no constraints from $S_\perp$ are active, and hence $\mathbf{x}^\star \in \operatorname{Int} S_\perp$.

3.2 Algorithm 4: Escaping When $\mathbf{y}^\star \in \operatorname{Int} S_\perp$

In this section, we the find minimizer $\mathbf{y}^\star$ of function $g(\mathbf{y}) = \frac{1}{2}\mathbf{y}^\top \mathbf{M}_\perp \mathbf{y} + \mathbf{y}^\top \mathbf{v}_\perp + C$ in $S_\perp \cap \mathcal{B}(r_\perp)$, while assuming that $\mathbf{y}^\star \in \operatorname{Int} S_\perp$. Since the solutions we find can be approximate, we have to guarantee that the objective is not too sensitive to the change of its argument. It suffices to consider the case

[5] Algorithm 3 and 4 are simplified versions of rigorous algorithms in the full version



(a) Case 0: the minimizer is on ∂S. We handle this case when processing the corresponding set of active constraints in Algorithm 3.

(b) Case 1: large linear term. We optimize the objective using only the linear term. This case is separate to avoid numerical issues in other cases

(c) Case 2: the minimizer is in the interior. We can find the critical point analytically, by solving a linear equation.

(d) Case 3: the minimizer is on the boundary. Then, there are at most $O(d)$ candidates with norm $r_\perp$

Figure 2: Cases of Algorithm 4

when $\|\mathbf{v}_\perp\|$ is bounded, since for any $\mathbf{y} \in \mathcal{B}(r_\perp)$ and perturbation $\mathbf{h}$ there exists $\tau \in [0,1]$ such that:

$$\begin{aligned} |g(\mathbf{y}) - g(\mathbf{y}+\mathbf{h})| &= |(\nabla g(\mathbf{y}+\tau\mathbf{h}))^\top \mathbf{h}| \\ &\le (\|\nabla g(\mathbf{0})\| + L\|\mathbf{y}+\tau\mathbf{h}\|)\|\mathbf{h}_\perp\| \\ &\le (\|\mathbf{v}_\perp\| + L(r_\perp + \|\mathbf{h}\|))\|\mathbf{h}\|, \end{aligned}$$

where we used that the objective is L-smooth and hence $\|\nabla g(\mathbf{y}+\tau\mathbf{h})\| \le \|\nabla g(\mathbf{0})\| + L\|\mathbf{y}+\tau\mathbf{h}\|$. We consider the situation when $\|\mathbf{v}_\perp\|$ is large as a separate case. Otherwise, for $\mathbf{y}^\star$, there are only two options: either $\mathbf{y}^\star \in \operatorname{Int} \mathcal{B}(r_\perp)$ or $\mathbf{y}^\star \in \partial\mathcal{B}(r_\perp)$. Algorithm 4 handles these cases, as well as the case when $\|\mathbf{v}\|$ is large, separately.

Case 1: $\|\mathbf{v}_\perp\|$ is large. If $\|\mathbf{v}_\perp\|$ is large and we can find $\mathbf{y}$ with small $\mathbf{y}^\top\mathbf{v}_\perp$, the linear term alone suffices to improve the objective. We show that, if such $\mathbf{y}$ doesn't exist, then $g(\mathbf{y}^\star)$ requires $\mathbf{y}^\star \in \partial S_\perp$, which contradicts that $\mathbf{y}^\star \in \operatorname{Int} S_\perp$. Below we assume that $\|\mathbf{v}_\perp\|$ is bounded.

Case 2: $\mathbf{y} \in \operatorname{Int} \mathcal{B}(r_\perp)$. In this case, $\mathbf{y}^\star$ is an unconstrained critical point of g, and hence it must satisfy $\nabla g(\mathbf{y}) = \mathbf{0}$, implying $\mathbf{M}_\perp\mathbf{y} + \mathbf{v}_\perp = \mathbf{0}$ which gives the unique solution $\mathbf{y} = -\mathbf{M}_\perp^{-1}\mathbf{v}_\perp$. since $\mathbf{M}$ is a perturbed matrix, so is $\mathbf{M}_\perp$, and hence $\mathbf{M}_\perp$ is non-degenerate with probability 1. It remains to verify that $\mathbf{y} \in \mathcal{B}(r_\perp) \cap S_\perp$ and $\mathbf{y}$ decreases the objective by $\Omega(\delta)$.

Case 3: $\mathbf{y} \in \partial\mathcal{B}(r_\perp)$. Since the only active constraint at $\mathbf{y}^\star$ is $c(\mathbf{y}) = \frac{1}{2}(\|\mathbf{y}\|^2 - r_\perp^2) = 0$, by the KKT conditions, any critical point must satisfy $\nabla g(\mathbf{y}) = \mu\nabla c(\mathbf{y})$ for some $\mu \in \mathbb{R}$, which is equivalent to $\mathbf{M}_\perp\mathbf{y} + \mathbf{v}_\perp = \mu\mathbf{y}$. Hence, for any fixed μ, $\mathbf{y}$ must be a solution of linear system $(\mathbf{M}_\perp - \mu I)\mathbf{y} = -\mathbf{v}_\perp$. When $\mathbf{M}_\perp - \mu I$ is degenerate (i.e. μ is an eigenvalue of $\mathbf{M}_\perp$), due to perturbation of $\mathbf{v}_\perp$, the system doesn't have any solution with probability 1. It leaves us with the case when $\mathbf{M}_\perp - \mu I$ is non-degenerate, when there exists a unique solution $\mathbf{y}(\mu) := -(\mathbf{M}_\perp - \mu I)^{-1}\mathbf{v}_\perp$.

Diagonalization. Dependence of $(\mathbf{M}_\perp - \mu I)^{-1}$ on μ is non-trivial, but for a diagonal $\mathbf{M}_\perp$, the inverse can be found explicitly. Hence, we perform diagonalization of $\mathbf{M}_\perp$: we find orthogonal $\mathbf{Q}$ and diagonal $\mathbf{\Lambda}$ such that $\|\mathbf{M}_\perp - \mathbf{Q}^\top\mathbf{\Lambda}\mathbf{Q}\| < \varepsilon$ in time $O(d^3 \log {}^1\!/\!\varepsilon)$ [Pan and Chen, 1999]. Setting $\varepsilon = O({}^\delta\!/\!r_\perp^2)$, we guarantee that the function changes by at most $O(\delta)$ in $\mathcal{B}(r_\perp)$. The function $\mathbf{y}(\mu) = -(\mathbf{Q}^\top\mathbf{\Lambda}\mathbf{Q} - \mu I)^{-1}\mathbf{v}_\perp$ can be written as $\mathbf{Q}\mathbf{y}(\mu) = -(\mathbf{\Lambda} - \mu I)\mathbf{Q}\mathbf{v}_\perp$, and hence we work with rotated vectors $\tilde{\mathbf{y}}(\mu) := \mathbf{Q}\mathbf{y}(\mu)$ and $\tilde{\mathbf{v}} := \mathbf{Q}\mathbf{v}_\perp$.

Finding candidate μ. Since $\tilde{\mathbf{y}}(\mu) = -(\mathbf{M}_\perp - \mu I)^{-1}\tilde{\mathbf{v}}$, for the i-th coordinate of $\tilde{\mathbf{y}}(\mu)$ we have $\tilde{y}_i(\mu) = \frac{\tilde{v}_i}{\mu - \lambda_i}$. Since we are only interested in $\mathbf{y}(\mu) \in \partial\mathcal{B}(r_\perp)$ and $\mathbf{Q}$ is an orthogonal matrix, we must have:

$$\|r_\perp\|^2 = \|\mathbf{y}(\mu)\|^2 = \|\tilde{\mathbf{y}}(\mu)\|^2 = \sum_{i=1}^{d} \tilde{y}_i(\mu)^2 = \sum_i \frac{\tilde{v}_i^2}{(\mu - \lambda_i)^2}$$

After multiplying the equation by $\prod_i(\mu - \lambda_i)^2$, we get an equation of the form $p(\mu) = 0$, where p is a polynomial of degree $2d$. We find roots $\mu_1, \ldots, \mu_{2d}$ of the polynomial in time $O(d^2 \log d \cdot \log\log {}^1\!/\!\varepsilon)$ [Pan, 1987], where ε is the required root precision. For each i, we compute $\mathbf{y}(\mu_i)$ and verify whether it lies in $\mathcal{B}(r_\perp) \cap S_\perp$ and improves the objective by $-\Omega(\delta)$.

Precision. We find the roots of the polynomial approximately, and when μ is close to λ_i for some i, even a small perturbation of μ can strongly affect $y_i(\mu) = \frac{\tilde{v}_i}{\mu - \lambda_i}$. We solve this as follows: since $\|\tilde{\mathbf{y}}(\mu)\| = r_\perp$, for all i we have $|\tilde{y}_i(\mu)| \le r_\perp$, implying $|\mu - \lambda_i| \ge \frac{|\tilde{v}_i|}{r_\perp}$. Therefore, μ must be sufficiently far from any λ_i, where the lower bound on the distance depends on $r_\perp \le \sqrt[3]{{}^\delta\!/\!\rho}$ and on $|\tilde{v}_i|$. Noise added to $\mathbf{v}$ is preserved in $\tilde{\mathbf{v}}$, and each coordinate is sufficiently separated from 0 w.h.p. This reasoning is formalized in the full version.

4 Conclusion

In this paper, we have shown that it's possible to escape from a constrained second-order stationary point with the logarithmic number of constraints within polynomial time and using only a polynomial number of stochastic gradient oracle calls. We provide experimental results in the full version.

An open question is to determine the conditions that on one hand guarantee escaping from a saddle point in polynomial time even for the linear number of constraints, and on the other hand hold in practice. One such condition can be strict complementarity.

Another open question is handling non-linear constraints. We believe that it can be straightforwardly achieved using techniques from [Ge *et al.*, 2015] by using assumptions on curvature and linear independence of the constraints. Finally, an interesting question would be a simpler algorithm for the general case, e.g. an algorithm resembling the approach from Section 2.

References

[Allen-Zhu and Li, 2018] Zeyuan Allen-Zhu and Yuanzhi Li. NEON2: finding local minima via first-order oracles.

In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3720–3730, 2018.

[Allen-Zhu, 2018] Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2680–2691, 2018.

[Anandkumar and Ge, 2016] Animashree Anandkumar and Rong Ge. Efficient approaches for escaping higher order saddle points in non-convex optimization. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 81–102. JMLR.org, 2016.

[Avdiukhin *et al.*, 2019] Dmitrii Avdiukhin, Chi Jin, and Grigory Yaroslavtsev. Escaping saddle points with inequality constraints via noisy sticky projected gradient descent. In *Optimization for Machine Learning Workshop*, 2019.

[Bertsekas, 1997] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

[Bian *et al.*, 2015a] Wei Bian, Xiaojun Chen, and Yinyu Ye. Complexity analysis of interior point algorithms for non-lipschitz and nonconvex minimization. *Mathematical Programming*, 149(1):301–327, 2015.

[Bian *et al.*, 2015b] Wei Bian, Xiaojun Chen, and Yinyu Ye. Complexity analysis of interior point algorithms for non-lipschitz and nonconvex minimization. *Mathematical Programming*, 149(1):301–327, 2015.

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Bubeck and others, 2015] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[Carmon and Duchi, 2018] Yair Carmon and John C. Duchi. Analysis of krylov subspace solutions of regularized nonconvex quadratic problems. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10728–10738, 2018.

[Carmon and Duchi, 2020] Yair Carmon and John C. Duchi. First-order methods for nonconvex quadratic minimization. *SIAM Rev.*, 62(2):395–436, 2020.

[Carmon *et al.*, 2017] 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 Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 654–663. PMLR, 2017.

[Cartis *et al.*, 2018] Coralia Cartis, Nick IM Gould, and Philippe L Toint. Second-order optimality and beyond: Characterization and evaluation complexity in convexly constrained nonlinear optimization. *Foundations of Computational Mathematics*, 18(5):1073–1107, 2018.

[Daneshmand *et al.*, 2018] Hadi Daneshmand, Jonas Moritz Kohler, Aurélien Lucchi, and Thomas Hofmann. Escaping saddles with stochastic gradients. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1163–1172. PMLR, 2018.

[Du *et al.*, 2017] Simon S. Du, Chi Jin, Jason D. Lee, Michael I. Jordan, Aarti Singh, and Barnabás Póczos. Gradient descent can take exponential time to escape saddle points. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1067–1077, 2017.

[Fang *et al.*, 2018] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. SPIDER: near-optimal non-convex optimization via stochastic path-integrated differential estimator. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 687–697, 2018.

[Ge *et al.*, 2015] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points - online stochastic gradient for tensor decomposition. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 797–842. JMLR.org, 2015.

[Haeser *et al.*, 2019] Gabriel Haeser, Hongcheng Liu, and Yinyu Ye. Optimality condition and complexity analysis for linearly-constrained optimization without differentiability on the boundary. *Mathematical Programming*, 178(1):263–299, 2019.

[Hsia and Sheu, 2018] Yong Hsia and Ruey-Lin Sheu. Trust region subproblem with a fixed number of additional linear inequality constraints has polynomial complexity. *arXiv preprint arXiv:1312.1398*, 2018.

[Jin *et al.*, 2017] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1724–1732. PMLR, 2017.

[Jin *et al.*, 2018] Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1042–1085. PMLR, 06–09 Jul 2018.

[Jin *et al.*, 2021] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M. Kakade, and Michael I. Jordan. On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *J. ACM*, 68(2):11:1–11:29, 2021.

[Lu *et al.*, 2020] Songtao Lu, Meisam Razaviyayn, Bo Yang, Kejun Huang, and Mingyi Hong. Finding second-order stationary points efficiently in smooth nonconvex linearly constrained optimization problems. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[Mokhtari *et al.*, 2018] Aryan Mokhtari, Asuman Ozdaglar, and Ali Jadbabaie. Escaping saddle points in constrained optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[Murty and Kabadi, 1987] Katta G. Murty and Santosh N. Kabadi. Some np-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117–129, 1987.

[Nesterov and Polyak, 2006] Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

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

[Nouiehed and Razaviyayn, 2020] Maher Nouiehed and Meisam Razaviyayn. A trust region method for finding second-order stationarity in linearly constrained nonconvex optimization. *SIAM J. Optim.*, 30(3):2501–2529, 2020.

[Nouiehed *et al.*, 2020] Maher Nouiehed, Jason D Lee, and Meisam Razaviyayn. Convergence to second-order stationarity for constrained non-convex optimization. *arXiv preprint arXiv:1810.02024*, 2020.

[O'Neill and Wright, 2020] Michael O'Neill and Stephen J Wright. A log-barrier Newton-CG method for bound constrained optimization with complexity guarantees. *IMA Journal of Numerical Analysis*, 41(1):84–121, 04 2020.

[Pan and Chen, 1999] Victor Y. Pan and Zhao Q. Chen. The complexity of the matrix eigenproblem. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, STOC '99, page 507–516, New York, NY, USA, 1999. Association for Computing Machinery.

[Pan, 1987] V. Pan. Sequential and parallel complexity of approximate evaluation of polynomial zeros. *Computers & Mathematics with Applications*, 14(8):591–622, 1987.

[Staib *et al.*, 2019] Matthew Staib, Sashank Reddi, Satyen Kale, Sanjiv Kumar, and Suvrit Sra. Escaping saddle points with adaptive gradient methods. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5956–5965. PMLR, 09–15 Jun 2019.

[Tripuraneni *et al.*, 2018] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I. Jordan. Stochastic cubic regularization for fast nonconvex optimization. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2904–2913, 2018.

[Xu *et al.*, 2018] Yi Xu, Rong Jin, and Tianbao Yang. First-order stochastic algorithms for escaping from saddle points in almost linear time. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5535–5545, 2018.

[Zhang and Li, 2021] Chenyi Zhang and Tongyang Li. Escape saddle points by a simple gradient-descent based algorithm. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8545–8556. Curran Associates, Inc., 2021.

[Zhou and Gu, 2020] Dongruo Zhou and Quanquan Gu. Stochastic recursive variance-reduced cubic regularization methods. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 3980–3990. PMLR, 2020.

[Zhou *et al.*, 2020] Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduction for nonconvex optimization. *J. Mach. Learn. Res.*, 21:103:1–103:63, 2020.