On Nonconvex Optimization for Machine Learning: Gradients, Stochasticity, and Saddle Points

CHI JIN, University of California, Berkeley
PRANEETH NETRAPALLI, Microsoft Research, India
RONG GE, Duke University
SHAM M. KAKADE, University of Washington, Seattle
MICHAEL I. JORDAN, University of California, Berkeley

Gradient descent (GD) and stochastic gradient descent (SGD) are the workhorses of large-scale machine learning. While classical theory focused on analyzing the performance of these methods in *convex* optimization problems, the most notable successes in machine learning have involved *nonconvex* optimization, and a gap has arisen between theory and practice. Indeed, traditional analyses of GD and SGD show that both algorithms converge to stationary points efficiently. But these analyses do not take into account the possibility of converging to saddle points. More recent theory has shown that GD and SGD can avoid saddle points, but the dependence on dimension in these analyses is polynomial. For modern machine learning, where the dimension can be in the millions, such dependence would be catastrophic. We analyze perturbed versions of GD and SGD and show that they are truly efficient—their dimension dependence is only polylogarithmic. Indeed, these algorithms converge to second-order stationary points in essentially the same time as they take to converge to classical first-order stationary points.

CCS Concepts: • **Theory of computation → Nonconvex optimization**; **Machine learning theory**;

Additional Key Words and Phrases: Saddle points, (stochastic) gradient descent, perturbations, efficiency

ACM Reference format:
Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M. Kakade, and Michael I. Jordan. 2021. On Nonconvex Optimization for Machine Learning: Gradients, Stochasticity, and Saddle Points. *J. ACM* 68, 2, Article 11 (February 2021), 29 pages.
https://doi.org/10.1145/3418526

A preliminary version of this article, with a subset of the results that are presented here, was presented at ICML 2017 and appeared in the proceedings as Jin et al. [27].
We thank Yair Carmon, Tongyang Li and Quanquan Gu for valuable discussions. This work was supported in part by the Mathematical Data Science program of the Office of Naval Research under grant number N00014-18-1-2764. Rong Ge also acknowledges the funding support from NSF CCF-1704656, NSF CCF-1845171 (CAREER), Sloan Fellowship and Google Faculty Research Award.
Authors' addresses: C. Jin and M. I. Jordan, University of California, 2570 Hearst Ave, Berkeley, California, USA; emails: {chijin, jordan}@cs.berkeley.edu; P. Netrapalli, Microsoft Research, No. 9, Lavelle Road, Bengaluru, Karnataka, India; email: praneeth@microsoft.com; R. Ge, Duke University, 308 Research Dr, Durham, North Carolina, USA; email: rongge@cs.duke.edu; S. M. Kakade, University of Washington, 185 E Stevens Way NE, Seattle, Washington, USA; email: sham@cs.washington.edu.

0004-5411/2021/02-ART11 $15.00
https://doi.org/10.1145/3418526

1 INTRODUCTION

One of the principal discoveries in machine learning in recent years is an empirical one—that simple algorithms often suffice to solve difficult real-world learning problems. Machine learning algorithms generally arise via formulations as optimization problems, and, despite a massive classical toolbox of sophisticated optimization algorithms and a major modern effort to further develop that toolbox, the simplest algorithms—gradient descent, which dates to the 1840s [15] and stochastic gradient descent, which dates to the 1950s [44]—reign supreme in machine learning.

This empirical discovery is appealing in many ways. First, at the scale of modern machine learning applications—often involving many millions of data points and millions of parameters—complex algorithms are generally infeasible, so that the only hope is that simple algorithms might be not only feasible but successful. Second, simple algorithms are easier to implement, debug, and maintain. Third, as the field of machine learning transforms into a real-world engineering discipline, it will be necessary to develop solid theoretical foundations for entire systems that employ machine learning algorithms at their core, and such an effort seems less daunting if the basic ingredients are simple.

These developments were presaged and supported by optimization researchers such as Nemirovskii, Nesterov, and Polyak who, from the 1960s until the present day, have pursued an in-depth study of first-order, gradient-based algorithms, developing novel algorithms and accompanying theory [35, 37, 41]. Their results have included lower bounds and algorithms that achieve those lower bounds. This line of work has made clear that even simple algorithms require delicate theoretical treatment when they are studied in large-scale settings. Thus, much of the focus has been on the setting of convex optimization where many of the complexities have been stripped away. This has allowed the development of an elegant theory, and has provided a solid jumping-off point for further analysis that has brought additional computational constraints into play—including distributed platforms, fault tolerance, communication bottlenecks, and asynchronous computation [42, 46, 52].

The most notable machine-learning success stories, however, have generally involved nonconvex optimization formulations, and a gap has arisen between theory and practice. Attempts to fill this gap include Bhojanapalli et al. [8] for matrix sensing, Jain et al. [26] and Ge et al. [24] for matrix completion, Netrapalli et al. [40] and Ge et al. [23] for robust principal component analysis, Bandeira et al. [7] and Mei et al. [34] for synchronization and MaxCut, Boumal et al. [9] for smooth semidefinite programs, and Choromanska et al. [16] in the setting of learning multi-layer neural networks. But there remains a need to develop a general theory that relates the convergence of machine learning algorithms to geometry and dynamics.

In the optimization literature much of the focus of theoretical work on the performance of algorithms has been on the relationship between the number of iterations of the algorithm and a suitable notion of accuracy. Dimension is often neglected in such analyses, in part because in the convex setting the numbers of iterations to find a near-optimal solution for even the simplest algorithms, including gradient descent, are provably independent of dimension. In developing algorithmic theory for nonconvex optimization formulations of machine learning problems, however, it is critically important to study iteration complexity as a function of of dimension, which can be in the millions. Moreover, we cannot resort to asymptotics—we are interested in problems at all scales.

In the current article, we have three goals. The first is to show that significant progress has been made in recent years in the theoretical analysis of algorithms for nonconvex machine learning. The second is to extend that line of analysis to handle both stochastic and non-stochastic algorithms in a single framework. In both cases, we upper bound the iteration complexity as a function of both accuracy and dimension. The third is to exhibit a simple proof that exposes the core of the phenomenon that determines the dimension dependence.

Nonconvex optimization problems are intractable in general. Progress has been in machine learning by noting that in many problems the principal difficulty is not local minima, either because there are no spurious local minima (we review a list of such problems in Section 3) or because empirical work has shown that the local minima that are found by local gradient-based algorithms tend to be effective in terms of the ultimate goal of machine learning, which is performance on a test set. The problem then becomes one of avoiding saddle points, which are ubiquitous in machine learning architectures. Saddle points slow down gradient-based algorithms and in millions of dimensions they are potentially a major bottleneck for such algorithms. The theoretical problem becomes that of characterizing the iteration complexity of avoiding saddle points, as a function of target accuracy and dimension.

We briefly mention some of the most relevant theoretical context for this problem here, providing a more thorough review of related work in Section 1.1 and in the Appendix. Lee et al. [30, 31] showed that gradient descent, under random initialization or with perturbations, asymptotically avoids saddle points with probability one. Ge et al. [22] provided a more quantitative (nonasymptotic) characterization of gradient descent augmented with a suitable perturbation, showing that its convergence rate in the presence of saddle points is upper bounded by an expression of the form $\text{poly}(d, \epsilon^{-1})$, where d is the dimension and ϵ is the accuracy. While these convergence results are inspiring, they are significantly worse than the convergence of gradient descent in the convex setting, or its convergence in the nonconvex setting where convergence to a saddle point is not excluded—in both cases the rate is independent of d—and they do not seem to accord with the empirical success of gradient-based methods in high-dimensional problems. Thus we ask whether these results, which are upper bounds, can be improved.

The current article provides a positive answer to this question. We show that suitably-perturbed versions of gradient descent and stochastic gradient descent escape saddle points in a number of iterations that is only polylogarithmic in dimension. More technically, defining a notion of *ϵ-second-order stationarity* (see Section 2), which rules out saddle points, to be contrasted with classical *ϵ-first-order stationarity*, which simply means near vanishing of the gradient, and which therefore does not rule out saddle points, we show that:

- Perturbed gradient descent (PGD) finds ϵ-second-order stationary points in $\tilde{O}(\epsilon^{-2})$ iterations, where $\tilde{O}(\cdot)$ hides only absolute constants and polylogarithmic factors. Compared to the $O(\epsilon^{-2})$ iterations required by gradient descent (GD) to find first-order stationary points [37], this involves only additional *polylogarithmic* factors in d.
- In the stochastic setting where stochastic gradients are Lipschitz, perturbed stochastic gradient descent (PSGD) finds ϵ-second-order stationary points in $\tilde{O}(\epsilon^{-4})$ iterations. Compared to the $O(\epsilon^{-4})$ iterations required by stochastic gradient descent (SGD) to find first-order stationary points [25], this again incurs overhead that is only *polylogarithmic* in d.
- When stochastic gradients are not Lipschitz, PSGD finds ϵ-second-order stationary points in $\tilde{O}(d\epsilon^{-4})$ iterations—this involves an additional *linear* factor in d.

1.1 Related Work

In this section, we discuss related work on convergence guarantees for finding second-order stationary points. Some key comparisons are summarized in Table 1, and an augmented table is provided in Appendix A.

Non-stochastic settings. Classical approaches to finding second-order stationary points assume access to second-order information, in particular the Hessian matrix of second derivatives. Examples of such approaches include the cubic regularization method [39] and trust-region

Table 1. A High Level Summary of the Results of This Paper and Their Comparison to Prior State of the Art for GD and SGD Algorithms

Setting	Algorithm	Iterations	Guarantees
Non-stochastic	GD [38]	$O(\epsilon^{-2})$	first-order stationary point
	PGD	$\tilde{O}(\epsilon^{-2})$	second-order stationary point
Stochastic	SGD [25]	$O(\epsilon^{-4})$	first-order stationary point
	PSGD (*with* Assumption C)	$\tilde{O}(\epsilon^{-4})$	second-order stationary point
	PSGD (*no* Assumption C)	$\tilde{O}(d\epsilon^{-4})$	second-order stationary point

This table only highlights the dependences on d and ϵ. See Section 1 for a description of these results. See Section 1.1 and Appendix A for a more detailed comparison with other related works.

methods [17], both of which require $O(\epsilon^{-1.5})$ queries of gradients and Hessians. This favorable convergence rate is, however, obtained at a high cost per iteration, owing to the fact that Hessian matrices scale quadratically with respect to dimension. In practice researchers have turned to first-order methods, which only utilize gradients and are therefore are substantially cheaper per iteration.

Before turning to pure first-order algorithms, we mention a line of research that is based on the assumption of a Hessian-vector product oracle [1, 12]. For architectures such as deep neural networks, Hessian-vector products can be computed efficiently via automatic differentiation, and it is thus possible to obtain much of the effect of second-order methods with a complexity closer to that of first-order methods. Indeed, the algorithms have convergence rates of $\tilde{O}(\epsilon^{-1.75})$ gradient queries [1, 12]. Their implementation, however, involved nested loops, and accordingly a concern with the setting of hyperparameters. Thus, despite the favorable convergence rate, these algorithms have not yet found their way into practical implementations.

Given the preference among practitioners for simple, single-loop algorithms, and the striking empirical successes obtained with such algorithms, it is important to pin down the theoretical properties of such algorithms. In such analyses, the results for second-order and Hessian-vector algorithms serve as baselines. Of key interest is the convergence rate not merely as a function of the accuracy ϵ but also as a function of the dimension d. Indeed, second-order algorithms can use the structure of the Hessian to readily avoid unhelpful directions even in high-dimensional spaces. Without the Hessian there is a concern that algorithms may scale poorly as a function of dimension.

Ge et al. [22] and Levy [33] studied simple variants of gradient descent and found that while it has favorable scaling in terms of ϵ, it requires poly(d) gradient queries to find second-order stationary points. These results are only upper bounds, however. The practical success of gradient descent suggests that the poly(d) scaling may be overly pessimistic. Indeed, in an early version of the results presented here, Jin et al. [27] showed that a simple perturbed version of gradient descent finds second-order stationary points in $\tilde{O}(\epsilon^{-2})$ gradient queries, paying only a logarithmic overhead compared to the rate associated with finding first-order stationary points. This result is summarized in the first two lines of Table 1. In followup work, Jin et al. [29] show that a perturbed version of celebrated Nesterov's accelerated gradient descent [36] enjoys a faster convergence rate of $\tilde{O}(\epsilon^{-1.75})$, again with logarithmic dimension dependence.

Stochastic setting with Lipschitz gradient. We now turn to the setting in which the learning algorithm only has access to stochastic gradients and where the stochastic is extrinsic; i.e., not under the control of the algorithm. Most existing work assumes that the stochastic gradients are Lipschitz (or equivalently that the underlying functions are gradient-Lipschitz, see Assumption C). Under this assumption, and an additional *Hessian-vector product* oracle, Allen-Zhu [2], Tripuraneni et al.

[49], Zhou et al. [54] designed algorithms that have an iteration complexity of $\tilde{O}(\epsilon^{-3.5})$. Xu et al. [50] and Allen-Zhu and Li [3] obtain similar results without the requirement of a Hessian-vector product oracle. The sharpest rates in this category have been obtained by Fang et al. [20] and Zhou and Gu [53], who show that the iteration complexity can be further reduced to $\tilde{O}(\epsilon^{-3})$. Again, however, this line of works consists of double-loop algorithms, and it remains unclear whether they will have an impact on practice.

Among single-loop algorithms that are simple variants of SGD, Ge et al. [22] showed that a particular variant has an iteration complexity for finding second-order stationary points that is upper bounded by $d^4\text{poly}(\epsilon^{-1})$. Daneshmand et al. [18] presented an alternative variant of SGD and showed that if the variance of the stochastic gradient along the escaping direction of saddle points is at least γ for all saddle points, then the algorithm finds second-order stationary points in $\tilde{O}(\gamma^{-4}\epsilon^{-5})$ iterations. In general, however, γ scales as $1/d$, which implies a complexity of $\tilde{O}(d^4\epsilon^{-5})$.

In the current article, we demonstrate that a simple perturbed version of SGD achieves a convergence rate of $\tilde{O}(\epsilon^{-4})$, which matches the speed of SGD to find a first-order stationary point up to polylogarithmic factors in dimension. Concurrent to our work, Fang et al. [21] analyzed SGD with averaging over last few iterates, and obtained a faster convergence rate of $\tilde{O}(\epsilon^{-3.5})$.

General stochastic setting. There is significantly less work in the general setting in which the stochastic gradients are no longer guaranteed to be Lipschitz. In fact, only the results of Ge et al. [22] and Daneshmand et al. [18] apply here, and both of them require at least $\Omega(d^4)$ gradient queries to find second-order stationary points. The current article brings this dependence down to linear dimension dependence. See the last three lines in Table 1 for a summary of the results in the stochastic case.

Other settings. Finally, there are also several recent results in the setting in which objective functions can be written as a finite sum of individual functions. We refer readers to Allen-Zhu and Li [3], Reddi et al. [43], and Lei et al. [32] and the references therein for further reading.

1.2 Organization

In Section 2, we review some algorithmic and mathematical preliminaries. Section 3 presents several examples of nonconvex problems in machine learning, demonstrating how second-order stationarity can ensure approximate global optimality. In Section 4, we present the algorithms that we analyze and present our main theoretical results for perturbed GD and SGD. In Section 5, we present the proof for the non-stochastic case (perturbed GD), which illustrates some of our key ideas. The proof for the stochastic setting is presented in the Appendix. We conclude in Section 6.

2 BACKGROUND

In this section, we introduce our notation and present definitions and assumptions. We also overview existing results in nonconvex optimization in both the deterministic and stochastic settings.

2.1 Notation

We use bold uppercase letters $\mathbf{A}, \mathbf{B}$ to denote matrices and bold lowercase letters $\mathbf{x}, \mathbf{y}$ to denote vectors. For vectors we use $\|\cdot\|$ to denote the ℓ_2-norm, and for matrices we use $\|\cdot\|$ and $\|\cdot\|_{\text{F}}$ to denote spectral (or operator) norm and Frobenius norm, respectively. We use $\lambda_{\min}(\cdot)$ to denote the smallest eigenvalue of a matrix. For a function $f : \mathbb{R}^d \to \mathbb{R}$, we use ∇f and $\nabla^2 f$ to denote the gradient and Hessian and $f^\star$ to denote the global minimum of function f. We use the notation $O(\cdot), \Theta(\cdot), \Omega(\cdot)$ to hide only absolute constants that do not depend on any problem parameter, and the notation

$\tilde{O}(\cdot), \tilde{\Theta}(\cdot), \tilde{\Omega}(\cdot)$ to hide absolute constants and factors that are only polylogarithmically dependent on all problem parameters.

2.2 Nonconvex Optimization and Gradient Descent

In this article, we are interested in solving general unconstrained optimization problems of the form:

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

where f is a smooth function that can be nonconvex. In particular, we assume that f has Lipschitz gradients and Lipschitz Hessians, which ensures that the gradient and Hessian cannot change too rapidly.

Definition 2.1. A differentiable function f is **ℓ-gradient Lipschitz** if

$$\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \le \ell \|\mathbf{x}_1 - \mathbf{x}_2\| \quad \forall\, \mathbf{x}_1, \mathbf{x}_2.$$

Definition 2.2. A twice-differentiable function f is **ρ-Hessian Lipschitz** if

$$\|\nabla^2 f(\mathbf{x}_1) - \nabla^2 f(\mathbf{x}_2)\| \le \rho \|\mathbf{x}_1 - \mathbf{x}_2\| \quad \forall\, \mathbf{x}_1, \mathbf{x}_2.$$

Assumption A. *The function f is ℓ-gradient Lipschitz and ρ-Hessian Lipschitz.*

Our point of departure is the classical Gradient Descent (GD) algorithm, whose update takes following form:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t), \tag{1}$$

where $\eta > 0$ is a step size or learning rate. Since the problem of finding a global optimum for general nonconvex functions is NP-hard, the classical literature in optimization has resorted to a local surrogate—first-order stationarity.

Definition 2.3. For a differentiable function f, $\mathbf{x}$ is a **first-order stationary point** if $\nabla f(\mathbf{x}) = \mathbf{0}$.

Definition 2.4. For a differentiable function f, $\mathbf{x}$ is an **ϵ-first-order stationary point** if $\|\nabla f(\mathbf{x})\| \le \epsilon$.

It is of major importance that gradient descent converges to a first-order stationary point in a number of iterations that is independent of dimension. This fact, referred to as "dimension-free convergence" in the optimization literature, is captured in the following classical theorem.

Theorem 2.5 ([37]). *For any $\epsilon > 0$, assume the function $f(\cdot)$ is ℓ-gradient Lipschitz, and set the step size as $\eta = 1/\ell$. Then, the gradient descent algorithm in Equation (1) will visit an ϵ-stationary point at least once in the following number of iterations:*

$$\frac{\ell(f(\mathbf{x}_0) - f^\star)}{\epsilon^2}.$$

Note that in this formulation, the last iterate is not guaranteed to be a stationary point. However, it is not hard to figure out which iterate is the stationary point by calculating the norm of the gradient at every iteration.

A first-order stationary point can be a local minimum, a local maximum or even a saddle point:

Definition 2.6. For a differentiable function f, a stationary point $\mathbf{x}$ is a

- **local minimum**, if there exists $\delta > 0$ such that $f(\mathbf{x}) \le f(\mathbf{y})$ for any $\mathbf{y}$ with $\|\mathbf{y} - \mathbf{x}\| \le \delta$.
- **local maximum**, if there exists $\delta > 0$ such that $f(\mathbf{x}) \ge f(\mathbf{y})$ for any $\mathbf{y}$ with $\|\mathbf{y} - \mathbf{x}\| \le \delta$.
- **saddle point**, otherwise.

For minimization problems, both saddle points and local maxima are clearly undesirable. Our focus will be "saddle points," although our results also apply directly to local maxima as well. Unfortunately, distinguishing saddle points from local minima for smooth functions is still NP-hard in general [38]. To avoid these hardness results, we focus on a subclass of saddle points.

Definition 2.7. For a twice-differentiable function f, $\mathbf{x}$ is a **strict saddle point** if $\mathbf{x}$ is a stationary point and $\lambda_{\min}(\nabla^2 f(\mathbf{x})) < 0$.

A generic saddle point must satisfy that $\lambda_{\min}(\nabla^2 f(\mathbf{x})) \leq 0$. Being "strict" simply rules out the case where $\lambda_{\min}(\nabla^2 f(\mathbf{x})) = 0$. We reformulate our goal as that of finding stationary points that are not strict saddle points.

Definition 2.8. For twice-differentiable function $f(\cdot)$, $\mathbf{x}$ is a **second-order stationary point** if

$$\nabla f(\mathbf{x}) = \mathbf{0}, \quad \text{and} \quad \nabla^2 f(\mathbf{x}) \succeq \mathbf{0}.$$

Definition 2.9. For a ρ-Hessian Lipschitz function $f(\cdot)$, $\mathbf{x}$ is an **ϵ-second-order stationary point** if:

$$\|\nabla f(\mathbf{x})\| \leq \epsilon \quad \text{and} \quad \nabla^2 f(\mathbf{x}) \succeq -\sqrt{\rho\epsilon}\cdot \mathbf{I}.$$

Our definition again makes use of an ϵ-ball around the stationary point so that we can discuss rates, and the condition on the Hessian in Definition 2.4 uses the Hessian Lipschitz parameter ρ to retain a single accuracy parameter and to match the units of the gradient and Hessian, following the convention of Nesterov and Polyak [39].

Although second-order stationarity is only a necessary condition for being a local minimum, a line of recent work in the machine learning literature shows that for many popular models in machine learning, all ϵ-second-order stationary points are approximate global minima. Thus, for these models finding second-order stationary points is sufficient for solving those problems. See Section 3 for references and discussion of these results.

2.3 Stochastic Approximation

We turn to the stochastic approximation setting, where we cannot access the exact gradient $\nabla f(\cdot)$ directly. Instead for any point $\mathbf{x}$, a gradient query will return a stochastic gradient $\mathbf{g}(\mathbf{x};\theta)$, where θ is a random variable drawn from a distribution $\mathcal{D}$. The key property that we assume for stochastic gradients is that they are unbiased: $\nabla f(\mathbf{x}) = \mathbb{E}_{\theta\sim\mathcal{D}}[\mathbf{g}(\mathbf{x};\theta)]$. That is, the average value of the stochastic gradient equals the true gradient. In short, the update of SGD is

$$\text{Sample } \theta_t \sim \mathcal{D}, \qquad \mathbf{x}_{t+1} = \mathbf{x}_t - \eta\nabla\mathbf{g}(\mathbf{x}_t;\theta_t). \tag{2}$$

Other than being an unbiased estimator of true gradient, another standard assumption on the stochastic gradients is that their variance is bounded by some number σ^2:

$$\mathbb{E}_{\theta\sim\mathcal{D}}\left[\|\mathbf{g}(\mathbf{x},\theta) - \nabla f(\mathbf{x})\|^2\right] \leq \sigma^2.$$

When we are interested in high-probability bounds, we make the following stronger assumption on the tail of the distribution.

Assumption B. *For any $\mathbf{x}\in\mathbb{R}^d$, the stochastic gradient $\mathbf{g}(\mathbf{x};\theta)$ with $\theta\sim\mathcal{D}$ satisfies:*

$$\mathbb{E}\mathbf{g}(\mathbf{x};\theta) = \nabla f(\mathbf{x}), \qquad \mathbb{P}\left(\|\mathbf{g}(\mathbf{x};\theta) - \nabla f(\mathbf{x})\| \geq t\right) \leq 2\exp(-t^2/(2\sigma^2)), \qquad \forall t\in\mathbb{R}.$$

We note this assumption is more general than the standard notion of a sub-Gaussian random vector, which assumes $\mathbb{E}\exp(\langle \mathbf{v}, \mathbf{X} - \mathbb{E}\mathbf{X}\rangle) \leq \exp(\sigma^2\|\mathbf{v}\|^2/d)$ for any $\mathbf{v}\in\mathbb{R}^d$. The latter assumption requires the distribution to be "isotropic" while our assumption does not. By Lemma C.2, we

know that both bounded random vectors and standard sub-Gaussian random vector are special cases of our more general setting.

Prior work shows that stochastic gradient descent converges to first-order stationary points in a number of iterations that is independent of dimension.

Theorem 2.10 ([25]). *For any $\epsilon, \delta > 0$, assume that the function f is ℓ-gradient Lipschitz, that the stochastic gradient $\mathbf{g}$ satisfies Assumption B, and let the step size scale as $\eta = \tilde{\Theta}(\ell^{-1}(1+\sigma^2/\epsilon^2)^{-1})$. Then, with probability at least $1-\delta$, stochastic gradient descent will visit an ϵ-stationary point at least once in the following number of iterations:*

$$\tilde{O}\left(\frac{\ell(f(\mathbf{x}_0)-f^\star)}{\epsilon^2}\left(1+\frac{\sigma^2}{\epsilon^2}\right)\right).$$

3 ON THE SUFFICIENCY OF SECOND-ORDER STATIONARITY

In this section, we show that for a wide class of nonconvex problems in machine learning and signal processing, all second-order stationary points are global minima. Thus, for this class of problems, finding second-order stationary points efficiently is equivalent to solving the problem. We focus on the underlying global geometry that these problems have in common.

Problems for which second-order stationary points are global minima include tensor decomposition [22], dictionary learning [47], phase retrieval [48], synchronization and MaxCut [7, 34], smooth semidefinite programs [9], and many problems related to low-rank matrix factorization, such as matrix sensing [8], matrix completion [24], and robust principale component analysis [23]. In particular, these papers show that by adding appropriate regularization terms, and under mild conditions, there are two key geometric properties satisfied by the corresponding objective functions: (a) All local minima are global minima. There might be multiple local minima due to permutation, but they are all equally good. (b) All saddle points have at least one direction with strictly negative curvature, thus are strict saddle points. we summarize the consequences of these properties in the following proposition, for which we omit the proof as it follows essentially by definition.

Proposition 3.1. *If a function f satisfies (a) all local minima are global minima; (b) all saddle points (including local maxima) are strict saddle points, then all second-order stationary points are global minima.*

This implies that the core problem for these nonconvex machine-learning applications is to find second-order stationary points efficiently. If we can prove that some simple variants of GD and SGD converges to second-order stationary points efficiently, then we immediately establish global convergence results for these algorithms for all of the above applications (i.e. convergence from arbitrary initialization).

Before we turn to such convergence results, consider, as an illustrative example of this class of nonconvex problems, the problem of finding the leading eigenvector of a positive semidefinite matrix $\mathbf{M} \in \mathbb{R}^{d\times d}$. We define the following objective:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\mathbf{x}^\top - \mathbf{M}\|_{\mathrm{F}}^2. \tag{3}$$

Denote the eigenvalues and eigenvectors of $\mathbf{M}$ as $(\lambda_i, \mathbf{v}_i)$ for $i = 1, \ldots, d$, and assume there is a gap between the first and second eigenvalues: $\lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_d \geq 0$. In this case, the global optimal solutions are $\mathbf{x} = \pm\sqrt{\lambda_1}\mathbf{v}_1$ giving the top eigenvector direction.

The objective function (3) is nonconvex as a function of $\mathbf{x}$. To optimize this objective via gradient-descent methods, we need to analyze the global landscape of the objective function. Its

ALGORITHM 1: Perturbed Gradient Descent

Input: $\mathbf{x}_0$, step size η, perturbation radius r.
  for $t = 0, 1, \ldots,$ **do**
    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta(\nabla f(\mathbf{x}_t) + \xi_t), \qquad \xi_t \sim \mathcal{N}(\mathbf{0}, (r^2/d)\mathbf{I})$

gradient and Hessian are of the form:

$$\nabla f(\mathbf{x}) = (\mathbf{x}\mathbf{x}^\top - \mathbf{M})\mathbf{x}$$
$$\nabla^2 f(\mathbf{x}) = \|\mathbf{x}\|^2\mathbf{I} + 2\mathbf{x}\mathbf{x}^\top - \mathbf{M}.$$

Therefore, all stationary points satisfy the equation $\mathbf{M}\mathbf{x} = \|\mathbf{x}\|^2\mathbf{x}$. Thus they are $\mathbf{0}$ and $\pm\sqrt{\lambda_i}\mathbf{v}_i$ for $i = 1, \ldots, d$. We already know that $\pm\sqrt{\lambda_1}\mathbf{v}_1$ are global minima, thus they are also local minima and are equivalent up to a sign difference. For the remaining stationary points $\mathbf{x}^\dagger$, we note that their Hessian always has strict negative curvature along the $\mathbf{v}_1$ direction: $\mathbf{v}_1^\top \nabla^2 f(\mathbf{x}^\dagger)\mathbf{v}_1 \le \lambda_2 - \lambda_1 < 0$. Thus these points are strict saddle points. Having established the preconditions for Proposition 3.1, we are able to conclude:

Corollary 3.2. *Assume that* $\mathbf{M}$ *is a positive semidefinite matrix whose top two eigenvalues are* $\lambda_1 > \lambda_2 \ge 0$. *For the problem of minimizing the objective Equation (3), all second-order stationary points are global optima.*

Further analysis can be carried out to establish the ϵ-robust version of Corollary 3.2. Informally, it can be shown that under technical conditions, for polynomially small ϵ, all ϵ-second-order stationary point are close to global optima. We refer the readers to Ge et al. [23] for the formal statement.

4 MAIN RESULTS

In this section, we present our main results on the efficiency of GD and SGD in the nonconvex setting. We first study the case where the exact gradients are accessible, where we focus on an algorithm that we refer to as PGD. We then turn to the stochastic setting, and present the results for Perturbed SGD and its mini-batch version.

4.1 Non-stochastic Setting

We begin by considering the case in which exact gradients are available, such that GD can be implemented. For convex problems, GD is efficient, but, as can be seen in Equation (1), GD makes a non-zero step only when the gradient is non-zero, and thus in the nonconvex setting it will be stuck at saddle points if initialized there. We thus consider a simple variant of GD that adds randomness to the iterates at each step (Algorithm 1). The question is whether such a simple procedure can be efficient, particularly in terms of its dimension dependence.

At each iteration, Algorithm 1 is almost the same as gradient descent, except it adds a small isotropic random Gaussian perturbation to the gradient. The perturbation ξ_t is sampled from a zero-mean Gaussian with covariance $(r^2/d)\mathbf{I}$ so that $\mathbb{E}\|\xi_t\|^2 = r^2$. We note that Algorithm 1 is different from that studied in Jin et al. [27], where perturbation was added only when certain conditions hold.

We now show that if we pick $r = \tilde{\Theta}(\epsilon)$ in Algorithm 1, PGD will find an ϵ-second-order stationary point in a number of iterations that has only a polylogarithmic dependence on dimension.

Theorem 4.1. *Let the function* $f(\cdot)$ *satisfy Assumption A. Then, for any* $\epsilon, \delta > 0$, *the PGD algorithm (Algorithm 1), with parameters* $\eta = \tilde{\Theta}(1/\ell)$ *and* $r = \tilde{\Theta}(\epsilon)$, *will visit an* ϵ*-second-order stationary*

ALGORITHM 2: Perturbed Stochastic Gradient Descent (PSGD)

Input: $\mathbf{x}_0$, step size η, perturbation radius r.
for $t = 0, 1, \ldots,$ **do**
sample $\theta_t \sim \mathcal{D}$
$\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta(\mathbf{g}(\mathbf{x}_t; \theta_t) + \xi_t), \qquad \xi_t \sim \mathcal{N}(\mathbf{0}, (r^2/d)\mathbf{I})$

point at least once in the following number of iterations, with probability at least $1 - \delta$*:*

$$\tilde{O}\left(\frac{\ell(f(\mathbf{x}_0) - f^\star)}{\epsilon^2}\right),$$

where $\tilde{O}$ *and* $\tilde{\Theta}$ *hide polylogarithmic factors in* $d, \ell, \rho, 1/\epsilon, 1/\delta$*, and* $\Delta_f := f(\mathbf{x}_0) - f^\star$*.*

Remark 4.2. If we wish to output an ϵ-second-order stationary point with constant probability, then it suffices to run PGD for the same number of iterations as in Theorem 4.1 and output an iterate uniformly at random (see Section B.6 for the formal proof). We note that the ϵ-second-order stationary point that is output in this way may not be the point with the smallest function value among all iterates.

Remark 4.3. We have chosen the distribution of the perturbations to be Gaussian in Algorithm 1 for simplicity. This choice is not necessary. The key properties needed for the perturbation distributions are (a) that the tail of the distribution is sufficiently light such that an appropriate concentration inequality holds, and (b) the variance in every direction is bounded below.

Comparing Theorem 4.1 to the classical result in Theorem 2.5, our result shows that PGD finds second-order stationary points in almost the same time as GD finds first-order stationary points, up to only logarithmic factors. Therefore, strict saddle points are computationally benign for first-order gradient methods.

Comparing to Theorem 2.5, we see that Theorem 4.1 makes an additional assumption on the Lipschitz smoothness of the Hessian. Without such an assumption, a γ-strict saddle point (per Definition 2.7, with $\lambda_{\min}(\nabla^2 f(\mathbf{x})) < -\gamma$) can be arbitrarily close to a second-order stationary point. Therefore, the additional assumption is essential in separating strict saddle points from second-order stationary points.

4.2 Stochastic Setting

In the stochastic approximation setting, exact gradients $\nabla f(\cdot)$ are no longer available, and the algorithms only have access to unbiased stochastic gradients: $\mathbf{g}(\cdot; \theta)$ such that $\nabla f(\mathbf{x}) = \mathbb{E}_{\theta \sim \mathcal{D}}[\mathbf{g}(\mathbf{x}; \theta)]$.

In machine learning, the stochastic gradient $\mathbf{g}$ is often obtained as an exact gradient of a smooth function: $\mathbf{g}(\cdot; \theta) = \nabla f(\cdot; \theta)$. We formalize this assumption.

Assumption C. *For any* $\theta \in$ *supp*$(\mathcal{D})$, $\mathbf{g}(\cdot; \theta)$ *is* $\tilde{\ell}$*-Lipschitz:*

$$\|\mathbf{g}(\mathbf{x}_1; \theta) - \mathbf{g}(\mathbf{x}_2; \theta)\| \le \tilde{\ell}\|\mathbf{x}_1 - \mathbf{x}_2\| \quad \forall\, \mathbf{x}_1, \mathbf{x}_2.$$

In the special case where $\mathbf{g}(\cdot; \theta) = \nabla f(\cdot; \theta)$ for some twice-differentiable function $f(\cdot; \theta)$, Assumption C ensures that the spectral norm of the Hessian of function $f(\cdot; \theta)$ is bounded by $\tilde{\ell}$ for all θ. Therefore, the stochastic Hessian also enjoys good concentration properties, which allows algorithms to find points with a second-order characterization. In contrast, when Assumption C no longer holds, the problem of finding second-order stationary points becomes more of a challenge due to the lack of concentration of the stochastic Hessian. In the current article, we treat both cases, by allowing $\tilde{\ell} = +\infty$ to encode the assertion that Assumption C does not hold.

ALGORITHM 3: Mini-batch Perturbed Stochastic Gradient Descent (Mini-batch PSGD)

Input: $\mathbf{x}_0$, step size η, perturbation radius r.
  for $t = 0, 1, \ldots,$ **do**
    sample $\{\theta_t^{(1)}, \cdots \theta_t^{(m)}\} \sim \mathcal{D}$
    $\mathbf{g}_t(\mathbf{x}_t) \leftarrow \sum_{i=1}^m \mathbf{g}(\mathbf{x}_t; \theta_t^{(i)})/m$
    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta(\mathbf{g}_t(\mathbf{x}_t) + \xi_t), \qquad \xi_t \sim \mathcal{N}(\mathbf{0}, (r^2/d)\mathbf{I})$

We are now ready to present a guarantee on the efficiency of PSGD (Algorithm 2) for finding a second-order stationary point. We make the following choices of parameters for Algorithm 2:

$$\eta = \tilde{\Theta}\left(\frac{1}{\ell \cdot \mathfrak{R}}\right), \quad r = \tilde{\Theta}(\epsilon\sqrt{\mathfrak{R}}), \quad \text{where} \quad \mathfrak{R} = 1 + \min\left\{\frac{\sigma^2}{\epsilon^2} + \frac{\tilde{\ell}^2}{\ell\sqrt{\rho\epsilon}}, \frac{\sigma^2 d}{\epsilon^2}\right\}, \tag{4}$$

where $\ell, \rho, \sigma, \tilde{\ell}$ are the parameters defined in Assumption A, B, C.

Theorem 4.4. *Let the function f satisfy Assumption A and assume that the stochastic gradient $\mathbf{g}$ satisfies Assumption B (or Assumption C optionally). For any $\epsilon, \delta > 0$, the PSGD algorithm (Algorithm 2), with parameter (η, r) chosen as in Equation (4), will visit an ϵ-second-order stationary point at least once in the following number of iterations, with probability at least $1 - \delta$:*

$$\tilde{O}\left(\frac{\ell(f(\mathbf{x}_0) - f^\star)}{\epsilon^2} \cdot \mathfrak{R}\right).$$

We note that Remark 4.2 and Remark 4.3 apply directly to Theorem 4.4.

Theorem 4.4 provides a result for both the scenario in which Assumption C holds and when it does not. In the former case, taking ϵ such that $\sigma^2/\epsilon^2 \geq \tilde{\ell}^2/(\ell\sqrt{\rho\epsilon})$, we have that $\mathfrak{R} \approx 1 + \sigma^2/\epsilon^2$. Our results then show that perturbed SGD finds a second-order stationary points in $\tilde{O}(\epsilon^{-4})$ iterations, which matches Theorem 2.10 up to logarithmic factors.

In the general case where Assumption C does not hold ($\tilde{\ell} = \infty$), we have $\mathfrak{R} = 1 + \sigma^2 d/\epsilon^2$, and Theorem 4.4 guarantees that PSGD finds an ϵ-second-order stationary point in $\tilde{O}(d\epsilon^{-4})$ iterations. Comparing to Theorem 2.10, we see that the algorithm pays an additional cost that is linear in dimension d.

Finally, Theorem 4.4 can be easily extended to the mini-batch setting, where multiple samples are used per iteration to compute the stochastic gradient. This reduces the variance of the stochastictiy in each iteration while the gradients for multiple samples can be efficiently computed in parallel. We choose parameters in this setting as follows:

$$\eta = \tilde{\Theta}\left(\frac{1}{\ell \cdot \mathfrak{M}}\right), \quad r = \tilde{\Theta}(\epsilon\sqrt{\mathfrak{M}}), \quad \text{where} \quad \mathfrak{M} = 1 + \frac{1}{m}\min\left\{\frac{\sigma^2}{\epsilon^2} + \frac{\tilde{\ell}^2}{\ell\sqrt{\rho\epsilon}}, \frac{\sigma^2 d}{\epsilon^2}\right\}. \tag{5}$$

Corollary 4.5 (Mini-batch version). *Let the function f satisfy Assumption A and assume that the stochastic gradient $\mathbf{g}$ satisfies Assumption B (or C optionally). Then, for any $\epsilon, \delta, m > 0$, the mini-batch PSGD algorithm (Algorithm 3), with parameters (η, r) chosen as in Equation (5), will visit an ϵ-second-order stationary point at least once in the following number of iterations, with probability at least $1 - \delta$:*

$$\tilde{O}\left(\frac{\ell(f(\mathbf{x}_0) - f^\star)}{\epsilon^2} \cdot \mathfrak{M}\right).$$

Corollary 4.5 says that if the mini-batch size m is not too large—$m \leq \mathfrak{R}$, where $\mathfrak{R}$ is defined in Equation (4)—then mini-batch PSGD will reduce the number of iterations linearly, while not increasing the total number of stochastic gradient queries.

ALGORITHM 4: Perturbed Gradient Descent (Variant)

Input: $\mathbf{x}_0$, step size η, perturbation radius r, time interval $\mathscr{T}$, tolerance ϵ.

$t_{\text{perturb}} = 0$
for $t = 0, 1, \ldots, T$ **do**
  if $\|\nabla f(\mathbf{x}_t)\| \le \epsilon$ and $t - t_{\text{perturb}} > \mathscr{T}$ **then**
    $\mathbf{x}_t \leftarrow \mathbf{x}_t - \eta\xi_t$, $(\xi_t \sim \text{Uniform}(B_0(r)))$; $t_{\text{perturb}} \leftarrow t$
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta\nabla f(\mathbf{x}_t)$

5 PROOFS

We provide full proofs of our main results, Theorem 4.4 and Corollary 4.5, in Appendix B. (Theorem 4.1 follows directly from the proof of Theorem 4.4 by setting $\sigma = 0$). These proofs require novel concentration inequalities and other tools from stochastic analysis to handle the relatively complex way in which stochasticity interacts with geometry in the neighborhood of saddle points. In the current section, we circumvent some of these complexities by presenting a conceptually straightforward proof for an algorithm that is a variant of PGD. This algorithm, summarized in Algorithm 4, removes some of the stochasticity of PGD by restricting the way in which perturbation noise is added. The algorithm is more complex than PGD and is less preferred in practice especially due to the requirement of an additional hyperparameter $\mathscr{T}$. However, the proof of Algorithm 4 is streamlined, allowing the core concepts underlying the full proof to be conveyed more simply.

The following theorem is the specialization of Theorem 4.1 to the setting of Algorithm 4.

Theorem 5.1. *Let f satisfy Assumption A and define $\Delta_f := f(\mathbf{x}_0) - f^\star$. For any $\epsilon, \delta > 0$, the PGD (Variant) algorithm (Algorithm 4), with parameters $\eta, r, \mathscr{T}$ chosen as in Equation (6) and with $\iota = c \cdot \log(d\ell\Delta_f/(\rho\epsilon\delta))$, has the property that at least one half of its iterations of will be ϵ-second order stationary points, after the following number of iterations, and with probability at least $1 - \delta$:*

$$\tilde{O}\left(\frac{\ell\Delta_f}{\epsilon^2}\right).$$

Here c is an absolute constant.

We proceed to the proof of the theorem. We first specify the choice of hyperparameters η, r, and $\mathscr{T}$ and two quantities $\mathscr{F}$ and $\mathscr{S}$ that are frequently used:

$$\eta = \frac{1}{\ell}, \quad r = \frac{\epsilon}{400\iota^3}, \quad \mathscr{T} = \frac{\ell}{\sqrt{\rho\epsilon}} \cdot \iota, \quad \mathscr{F} = \frac{1}{50\iota^3}\sqrt{\frac{\epsilon^3}{\rho}}, \quad \mathscr{S} = \frac{1}{4\iota}\sqrt{\frac{\epsilon}{\rho}}. \tag{6}$$

Our high-level proof strategy is a proof by contradiction: When the current iterate is not an ϵ-second-order stationary point, it must either have a large gradient or have a strictly negative Hessian, and we prove that in either case, PGD must yield a significant decrease in function value in a controlled number of iterations. Also, since the function value cannot decrease more than $f(\mathbf{x}_0) - f^\star$, we know that the total number of iterates that are not ϵ-second order stationary points cannot be very large.

First, we bound the rate of decrease when the gradient is large.

Lemma 5.2 (Descent Lemma). *If $f(\cdot)$ satisfies Assumption A and $\eta \le 1/\ell$, then the gradient descent sequence $\{\mathbf{x}_t\}$ satisfies:*

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \le -\eta\|\nabla f(\mathbf{x}_t)\|^2/2.$$

Proof. Due to the ℓ-gradient Lipschitz assumption, we have:

$$f(\mathbf{x}_{t+1}) \le f(\mathbf{x}_t) + \langle \nabla f(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle + \frac{\ell}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2$$
$$= f(\mathbf{x}_t) - \eta\|\nabla f(\mathbf{x}_t)\|^2 + \frac{\eta^2\ell}{2}\|\nabla f(\mathbf{x}_t)\|^2 \le f(\mathbf{x}_t) - \frac{\eta}{2}\|\nabla f(\mathbf{x}_t)\|^2. \qquad \square$$

We now show that if the starting point has a strictly negative eigenvalue of the Hessian, then adding a perturbation and following by gradient descent will yield a significant decrease in function value in $\mathscr{T}$ iterations.

Lemma 5.3 (Escaping Saddle Points). *Assume that $f(\cdot)$ satisfies Assumption A, $\tilde{\mathbf{x}}$ satisfies $\|\nabla f(\tilde{\mathbf{x}})\| \le \epsilon$, and $\lambda_{\min}(\nabla^2 f(\tilde{\mathbf{x}})) \le -\sqrt{\rho\epsilon}$. Let $\mathbf{x}_0 = \tilde{\mathbf{x}} + \eta\xi$ ($\xi \sim$ Uniform$(B_0(r))$). Run gradient descent starting from $\mathbf{x}_0$. This yields:*

$$\mathbb{P}(f(\mathbf{x}_{\mathscr{T}}) - f(\tilde{\mathbf{x}}) \le -\mathscr{F}/2) \ge 1 - \frac{\ell\sqrt{d}}{\sqrt{\rho\epsilon}} \cdot \iota^2 2^{8-\iota},$$

where $\mathbf{x}_{\mathscr{T}}$ is the $\mathscr{T}$th gradient descent iterate starting from $\mathbf{x}_0$.

To prove this, we need to prove two lemmas, and the major simplification over Jin et al. [27] comes from the following lemma that says that if the function value does not decrease too much over t iterations, then all iterates $\{\mathbf{x}_\tau\}_{\tau=0}^t$ will remain in a small neighborhood of $\mathbf{x}_0$.

Lemma 5.4 (Improve or Localize). *Under the setting of Lemma 5.2, for any $t \ge \tau > 0$:*

$$\|\mathbf{x}_\tau - \mathbf{x}_0\| \le \sqrt{2\eta t(f(\mathbf{x}_0) - f(\mathbf{x}_t))}.$$

Proof. Given the gradient update, $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\nabla f(\mathbf{x}_t)$, we have that for any $\tau \le t$:

$$\|\mathbf{x}_\tau - \mathbf{x}_0\| \le \sum_{\tau=1}^{t} \|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\| \overset{(1)}{\le} \left[t\sum_{\tau=1}^{t}\|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\|^2\right]^{\frac{1}{2}}$$
$$= \left[\eta^2 t\sum_{\tau=1}^{t}\|\nabla f(\mathbf{x}_{\tau-1})\|^2\right]^{\frac{1}{2}} \overset{(2)}{\le} \sqrt{2\eta t(f(\mathbf{x}_0) - f(\mathbf{x}_t))},$$

where step (1) uses Cauchy-Schwarz inequality and step (2) is due to Lemma 5.2. $\square$

Second, we show that the region in which GD will get stuck for at least $\mathscr{T}$ iterations if initialized there (which we refer to as the "stuck region") is thin. We show this by tracking any pair of points that differ only in an escaping direction and are at least ω far apart. We show that at least one sequence of the two GD sequences initialized at these points is guaranteed to escape the saddle point with high probability, so that the width of the stuck region along an escaping direction is at most ω.

Lemma 5.5 (Coupling Sequence). *Suppose $f(\cdot)$ satisfies Assumption A and $\tilde{\mathbf{x}}$ satisfies $\lambda_{\min}(\nabla^2 f(\tilde{\mathbf{x}})) \le -\sqrt{\rho\epsilon}$. Let $\{\mathbf{x}_t\}$, $\{\mathbf{x}'_t\}$ be two gradient descent sequences that satisfy: (1) $\max\{\|\mathbf{x}_0 - \tilde{\mathbf{x}}\|, \|\mathbf{x}'_0 - \tilde{\mathbf{x}}\|\} \le \eta r$; and (2) $\mathbf{x}_0 - \mathbf{x}'_0 = \eta r_0 \mathbf{e}_1$, where $\mathbf{e}_1$ is the minimum eigenvector of $\nabla^2 f(\tilde{\mathbf{x}})$ and $r_0 > \omega := 2^{2-\iota}\ell\mathscr{S}$. Then:*

$$\min\{f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0), f(\mathbf{x}'_{\mathscr{T}}) - f(\mathbf{x}'_0)\} \le -\mathscr{F}.$$

Proof. Assume the contrary, that is, $\min\{f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0), f(\mathbf{x}'_{\mathscr{T}}) - f(\mathbf{x}'_0)\} > -\mathscr{F}$. Lemma 5.4 implies localization of both sequences around $\tilde{\mathbf{x}}$; that is, for any $t \le \mathscr{T}$:

$$\begin{aligned}\max\{\|\mathbf{x}_t - \tilde{\mathbf{x}}\|, \|\mathbf{x}'_t - \tilde{\mathbf{x}}\|\} &\le \max\{\|\mathbf{x}_t - \mathbf{x}_0\|, \|\mathbf{x}'_t - \mathbf{x}'_0\|\} + \max\{\|\mathbf{x}_0 - \tilde{\mathbf{x}}\|, \|\mathbf{x}'_0 - \tilde{\mathbf{x}}\|\} \\ &\le \sqrt{2\eta\mathscr{T}\mathscr{F}} + \eta r \le \mathscr{S}, \end{aligned} \tag{7}$$

where the last step is due to our choice of $\eta, r, \mathscr{T}, \mathscr{F}, \mathscr{S}$, as in Equation (6), and $\ell/\sqrt{\rho\epsilon} \ge 1$.[1] However, we can write the update equation for the difference $\hat{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}'_t$ as:

$$\begin{aligned}\hat{\mathbf{x}}_{t+1} &= \hat{\mathbf{x}}_t - \eta[\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}'_t)] = (\mathbf{I} - \eta\mathcal{H})\hat{\mathbf{x}}_t - \eta\Delta_t\hat{\mathbf{x}}_t \\ &= \underbrace{(\mathbf{I} - \eta\mathcal{H})^{t+1}\hat{\mathbf{x}}_0}_{\mathbf{p}(t+1)} - \underbrace{\eta\sum_{\tau=0}^{t}(\mathbf{I} - \eta\mathcal{H})^{t-\tau}\Delta_\tau\hat{\mathbf{x}}_\tau}_{\mathbf{q}(t+1)},\end{aligned}$$

where $\mathcal{H} = \nabla^2 f(\tilde{\mathbf{x}})$ and $\Delta_t = \int_0^1 [\nabla^2 f(\mathbf{x}'_t + \theta(\mathbf{x}_t - \mathbf{x}'_t)) - \mathcal{H}]\mathrm{d}\theta$. We note that $\mathbf{p}(t)$ arises from the initial difference $\hat{x}_0$, and $\mathbf{q}(t)$ is an error term that arises from the fact that the function f is not quadratic. We now use induction to show that the error term $\mathbf{q}(t)$ is always small compared to the leading term $\mathbf{p}(t)$. That is, we wish to show:

$$\|\mathbf{q}(t)\| \le \|\mathbf{p}(t)\|/2, \qquad t \in [\mathscr{T}].$$

The claim is true for the base case $t = 0$ as $\|\mathbf{q}(0)\| = 0 \le \|\hat{\mathbf{x}}_0\|/2 = \|\mathbf{p}(0)\|/2$. Now suppose the induction claim is true up to t. Denote $\lambda_{\min}(\nabla^2 f(\mathbf{x}_0)) = -\gamma$. Note that $\hat{\mathbf{x}}_0$ lies in the direction of the minimum eigenvector of $\nabla^2 f(\mathbf{x}_0)$. Thus for any $\tau \le t$, we have:

$$\|\hat{\mathbf{x}}_\tau\| \le \|\mathbf{p}(\tau)\| + \|\mathbf{q}(\tau)\| \le 2\|\mathbf{p}(\tau)\| = 2\|(\mathbf{I} - \eta\mathcal{H})^\tau\hat{\mathbf{x}}_0\| = 2(1+\eta\gamma)^\tau\eta r_0.$$

By the Hessian Lipschitz property, we have $\|\Delta_t\| \le \rho\max\{\|\mathbf{x}_t - \tilde{\mathbf{x}}\|, \|\mathbf{x}'_t - \tilde{\mathbf{x}}\|\} \le \rho\mathscr{S}$, therefore:

$$\begin{aligned}\|\mathbf{q}(t+1)\| &= \|\eta\sum_{\tau=0}^{t}(\mathbf{I} - \eta\mathcal{H})^{t-\tau}\Delta_\tau\hat{\mathbf{x}}_\tau\| \le \eta\rho\mathscr{S}\sum_{\tau=0}^{t}\|(\mathbf{I} - \eta\mathcal{H})^{t-\tau}\|\|\hat{\mathbf{x}}_\tau\| \\ &\le 2\eta\rho\mathscr{S}\sum_{\tau=0}^{t}(1+\eta\gamma)^t\eta r_0 \le 2\eta\rho\mathscr{S}\mathscr{T}(1+\eta\gamma)^t\eta r_0 \le 2\eta\rho\mathscr{S}\mathscr{T}\|\mathbf{p}(t+1)\|,\end{aligned}$$

where the second-to-last inequality uses $t + 1 \le \mathscr{T}$. By our choice of the hyperparameter in Equation (6), we have $2\eta\rho\mathscr{S}\mathscr{T} \le 1/2$, which finishes the inductive proof.

Finally, the induction claim implies:

$$\begin{aligned}\max\{\|\mathbf{x}_{\mathscr{T}} - \mathbf{x}_0\|, \|\mathbf{x}'_{\mathscr{T}} - \mathbf{x}_0\|\} &\ge \frac{1}{2}\|\hat{\mathbf{x}}(\mathscr{T})\| \ge \frac{1}{2}[\|\mathbf{p}(\mathscr{T})\| - \|\mathbf{q}(\mathscr{T})\|] \ge \frac{1}{4}[\|\mathbf{p}(\mathscr{T})\| \\ &= \frac{(1+\eta\gamma)^{\mathscr{T}}\eta r_0}{4} \overset{(1)}{\ge} 2^{t-2}\eta r_0 > \mathscr{S},\end{aligned}$$

where step (1) uses the fact $(1+x)^{1/x} \ge 2$ for any $x \in (0, 1]$. This contradicts the localization property of Equation (7), which finishes the proof. □

Equipped with Lemma 5.4 and Lemma 5.5, we are ready to prove Lemma 5.3.

[1] We note that when $\ell/\sqrt{\rho\epsilon} < 1$, ϵ-second-order stationary points are equivalent to ϵ-first-order stationary points due to the function f being ℓ-gradient Lipschitz. In this case, the problem of finding ϵ-second-order stationary points becomes straightforward.

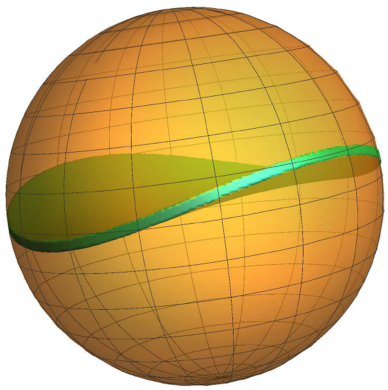

Fig. 1. Pertubation ball in 3D and "thin pancake" shape stuck region.

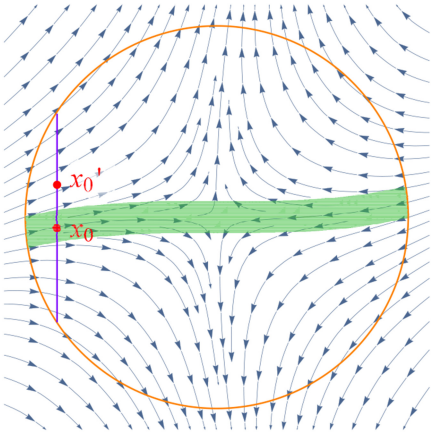

Fig. 2. Pertubation ball in 2D and "narrow band" stuck region under gradient flow.

Proof of Lemma 5.3. Recall $\mathbf{x}_0 \sim \text{Uniform}(B_{\tilde{\mathbf{x}}}(\eta r))$. We refer to $B_{\tilde{\mathbf{x}}}(\eta r)$ as the *perturbation ball* and define the *stuck region* within the perturbation ball to be the set of points starting from which GD requires more than $\mathscr{T}$ steps to escape:

$$\mathcal{X}_{\text{stuck}} := \{\mathbf{x} \in B_{\tilde{\mathbf{x}}}(\eta r) | \{\mathbf{x}_t\} \text{ is GD sequence with } \mathbf{x}_0 = \mathbf{x}, \text{and } f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0) > -\mathscr{F}\}.$$

See Figure 1 and Figure 2 for illustrations. Although the shape of the stuck region can be very complicated, according to Lemma 5.5 we know that the width of $\mathcal{X}_{\text{stuck}}$ along the $\mathbf{e}_1$ direction is at most $\eta\omega$. That is, $\text{Vol}(\mathcal{X}_{\text{stuck}}) \le \text{Vol}(\mathbb{B}_0^{d-1}(\eta r))\eta\omega$. Therefore:

$$\mathbb{P}(\mathbf{x}_0 \in \mathcal{X}_{\text{stuck}}) = \frac{\text{Vol}(\mathcal{X}_{\text{stuck}})}{\text{Vol}(\mathbb{B}_{\tilde{\mathbf{x}}}^d(\eta r))} \le \frac{\eta\omega \times \text{Vol}(\mathbb{B}_0^{d-1}(\eta r))}{\text{Vol}(\mathbb{B}_0^d(\eta r))} = \frac{\omega}{r\sqrt{\pi}} \frac{\Gamma(\frac{d}{2}+1)}{\Gamma(\frac{d}{2}+\frac{1}{2})} \le \frac{\omega}{r} \cdot \sqrt{\frac{d}{\pi}} \le \frac{\ell\sqrt{d}}{\sqrt{\rho\epsilon}} \cdot \iota^2 2^{8-\iota}.$$

On the event $\{\mathbf{x}_0 \notin \mathcal{X}_{\text{stuck}}\}$, due to our parameter choice in Equation (6), we have:

$$f(\mathbf{x}_{\mathscr{T}}) - f(\tilde{\mathbf{x}}) = [f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0)] + [f(\mathbf{x}_0) - f(\tilde{\mathbf{x}})] \le -\mathscr{F} + \epsilon\eta r + \frac{\ell\eta^2 r^2}{2} \le -\mathscr{F}/2.$$

This finishes the proof. □

With Lemma 5.2 and Lemma 5.3 in hand, it is not hard to establish Theorem 5.1.

Proof of Theorem 5.1. First, we set the total number of iterations T to be

$$T = 8\max\left\{\frac{(f(x_0) - f^\star)\mathscr{T}}{\mathscr{F}}, \frac{(f(x_0) - f^\star)}{\eta\epsilon^2}\right\} = O\left(\frac{\ell(f(x_0) - f^\star)}{\epsilon^2} \cdot \iota^4\right).$$

Next, we choose $\iota = c \cdot \log(\frac{d\ell\Delta_f}{\rho\epsilon\delta})$, with a large-enough absolute constant c such that:

$$(T\ell\sqrt{d}/\sqrt{\rho\epsilon}) \cdot \iota^2 2^{8-\iota} \le \delta.$$

We then argue that, with probability $1-\delta$, Algorithm 4 will add a perturbation at most $T/(4\mathscr{T})$ times. This is because if otherwise, then we can appeal to Lemma 5.3 every time we add a perturbation and conclude:

$$f(\mathbf{x}_T) \le f(\mathbf{x}_0) - T\mathscr{F}/(4\mathscr{T}) < f^\star,$$

which cannot happen. Finally, excluding those iterations that are within $\mathscr{T}$ steps after adding perturbations, we still have $3T/4$ steps left. They are either large gradient steps, $\|\nabla f(\mathbf{x}_t)\| \geq \epsilon$, or ϵ-second-order stationary points. Within them, we know that the number of large gradient steps cannot be more than $T/4$. This is true because if otherwise, by Lemma 5.2:

$$f(\mathbf{x}_T) \leq f(\mathbf{x}_0) - T\eta\epsilon^2/4 < f^\star,$$

which again cannot happen. Therefore, we conclude that at least $T/2$ of the iterates must be ϵ-second-order stationary points. □

6 CONCLUSIONS

We have shown that simple perturbed versions of GD and SGD escape saddle points and find second-order stationary points in essentially the same time that classical GD and SGD take to find first-order stationary points. The overheads are only polylogarithmic factors in dimensionality in both the non-stochastic setting and the stochastic setting with Lipschitz stochastic gradient. In the general stochastic setting, the overhead is a linear factor in dimension.

Combined with previous literature that shows that all second-order stationary points are global optima for a broad class of nonconvex optimization problems in machine learning and signal processing, our results directly provide efficient guarantees for solving those nonconvex problem via simple local search approaches. We now discuss several possible future directions, and further connections to other fields.

Optimal rates for finding second-order stationary points. Carmon et al. [13] have presented lower bounds that imply that GD achieves the optimal rate for finding stationary points for gradient Lipschitz functions. In our setting, we additionally assume that the Hessian is Lipschitz. This implies that GD is no longer necessarily an optimal algorithm. While our results show that variants of GD are efficient in this setting, one would additionally like to know whether they are close to optimality.

Focusing on first-order algorithms for functions with both Lipschitz gradient and Lipschitz Hessian, the best known gradient query complexity for finding second-order stationary points is $\tilde{O}(\epsilon^{-1.75})$ [1, 12, 29], while the best existing lower bound is $\Omega(\epsilon^{-12/7})$ by Carmon et al. [14]. Note that this lower bound is restricted to deterministic algorithms, and thus does not apply to most existing algorithms for escaping saddle points as they are all randomized algorithms. For the stochastic setting with Lipschitz stochastic gradients, the best-known query complexity is $\tilde{O}(\epsilon^{-3})$ [20, 53], while a matching lower bound $\Omega(\epsilon^{-3})$ has been obtained in recent work [5, 6][2]. See Appendix A for further discussion of the literature.

Escaping high-order saddle points. The current article focuses on escaping strict saddle points and finding second-order stationary points. More generally, one can define nth-order stationary points as points that satisfies the KKT necessary conditions for being local minima up to nth-order derivatives. It becomes more challenging to find nth-order stationary points as n increases, since it is necessary to escape higher-order saddle points. In terms of worst-case efficiency, Nesterov [38] rules out the possibility of efficient algorithms for finding nth-order stationary points for all $n \geq 4$, showing that the problem is NP-hard. Anandkumar and Ge [4] present a third-order algorithm that finds third-order stationary points in polynomial time. It remains open whether simple variants of GD can also find third-order stationary points efficiently. It is unlikely that the overhead will

[2]Despite the original hard instance in Arjevani et al. [6] does not satify the Hessian Lipschitz condition, it can be modified to satisfy such condition using the rescaling technique in Arjevani et al. [5], while giving the same lower bound $\Omega(\epsilon^{-3})$.

still be small or only logarithmic in this case, but it is not clear what to expect for the overhead. A related question is to identify applications where third-order stationarity is needed, in addition to second-order stationarity, to achieve global optimality.

Connection to gradient Langevin dynamics. The Bayesian counterpart of SGD is the Langevin Monte Carlo (LMC) algorithm [45], which performs the following iterative update:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta(\nabla f(\mathbf{x}_t) + \sqrt{2/(\eta\beta)}\mathbf{w}_t), \quad \text{where} \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Here β is known as the inverse temperature. When the step size η goes to zero, the distribution of the LMC iterates is known to converge to the stationary distribution $\mu(\mathbf{x}) \propto e^{-\beta f(\mathbf{x})}$ [45].

While the LMC algorithm is superficially very close to stochastic gradient descent, the goals for the two algorithms are quite different.

- **Convergence:** While the focus of the optimization literature is to find stationary points, the goal of the LMC algorithm is to converge to a stationary distribution (i.e., to mix rapidly).
- **Scaling of Noise:** The scaling of the stochasticity in SGD—and in particular the size of the perturbation that we consider in the current article—is much smaller than the scaling considered in the LMC literature. Running our algorithm is equivalent to running LMC with temperature $\beta^{-1} \propto d^{-1}$. In this low-temperature or small-noise regime, the algorithm can no longer mix efficiently for smooth nonconvex functions, as it requires $\Omega(e^d)$ steps in the worst case [10]. However, with this small amount of noise, the algorithm can still perform local search efficiently, and can find a second-order stationary point in a small number of iterations, as shown in Theorem 4.1.

Recent work of Zhang et al. [51] studied the time that LMC takes to hit a second-order stationary point as a criterion for convergence, instead of the traditional mixing time to a stationary distribution. In this analysis, the runtime is no longer exponential, but it is still polynomially dependent on dimension d with large degree.

On the necessity of adding perturbations. We have shown that adding perturbations to the iterations of GD or SGD allows these algorithms to escape saddle points efficiently. As an alternative, one can also simply run GD with random initialization, and try to escape saddle points using only the randomness due to the initialization. Although this alternative algorithm exhibits asymptotic convergence [31], it does not yield efficient convergence in general. Du et al. [19] shows that even with fairly natural random initialization schemes and non-pathological functions, randomly initialized GD can be significantly slowed by saddle points, taking exponential time to escape them.

APPENDICES

A TABLES OF RELATED WORK

In Table 2 and Table 3, we present a detailed comparison of our results with other related work in both non-stochastic and stochastic settings. See Section 1.1 for the full text descriptions. We note that our algorithms are simple variants of standard GD and SGD, which are the simplest among all the algorithms listed in the table.

Table 2. A Summary of Related Work on First-order Algorithms to Find Second-order Stationary Points in *Non-stochastic* Settings

Algorithm	Iterations	Simplicity
Noisy GD [22]	$d^4\text{poly}(\epsilon^{-1})$	single-loop
Normalized GD [33]	$\tilde{O}(d^3 \cdot \text{poly}(\epsilon^{-1}))$	
PGD (conference version of this work) [27]	$\tilde{O}(\epsilon^{-2})$	
†Perturbed AGD [29]	$\tilde{O}(\epsilon^{-1.75})$	
FastCubic [1]	$\tilde{O}(\epsilon^{-1.75})$	double-loop
Carmon et al. [12]	$\tilde{O}(\epsilon^{-1.75})$	
Carmon and Duchi [11]	$\tilde{O}(\epsilon^{-2})$	

This table only highlights the dependencies on d and ϵ.† denotes followup work on the conference version of this article [27].

Table 3. A Summary of Related Work on First-order Algorithms to Find Second-order Stationary Points in the *Stochastic* Setting

Algorithm	Iterations (*with* Assumption C)	Iterations (*no* Assumption C)	Simplicity
Noisy GD [22]	$d^4\text{poly}(\epsilon^{-1})$	$d^4\text{poly}(\epsilon^{-1})$	single-loop
CNC-SGD [18]	$\tilde{O}(d^4\epsilon^{-5})$	$\tilde{O}(d^4\epsilon^{-5})$	
PSGD (this work)	$\tilde{O}(\epsilon^{-4})$	$\tilde{O}(d\epsilon^{-4})$	
*SGD with averaging [21]	$\tilde{O}(\epsilon^{-3.5})$	×	
Natasha 2 [2]	$\tilde{O}(\epsilon^{-3.5})$	×	double-loop
Stochastic Cubic [49]	$\tilde{O}(\epsilon^{-3.5})$	×	
SPIDER [20]	$\tilde{O}(\epsilon^{-3})$	×	
SRVRC [53]	$\tilde{O}(\epsilon^{-3})$	×	

This table only highlights the dependencies on d and ϵ. * denotes independent work.

B PROOFS FOR STOCHASTIC SETTING

In this section, we provide proofs for our main results—Theorem 4.4 and Corollary 4.5. Theorem 4.1 can be proved as a special case of Theorem 4.4 by taking $\sigma = 0$.

B.1 Notation

Recall the update equation of Algorithm 2 is $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta(\mathbf{g}(\mathbf{x}_t;\theta_t) + \xi_t)$, where $\xi_t \sim \mathcal{N}(\mathbf{0}, (r^2/d)\mathbf{I})$. Throughout this section, we denote $\zeta_t := \mathbf{g}(\mathbf{x}_t;\theta_t) - \nabla f(\mathbf{x}_t)$, as the noise part within the stochastic gradient. For simplicity, we also denote $\tilde{\zeta}_t := \zeta_t + \xi_t$, which is the summation of noise from the stochastic gradient and the injected perturbation, and $\tilde{\sigma}^2 := \sigma^2 + r^2$. Then the update equation can be rewritten as $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta(\nabla f(\mathbf{x}_t) + \tilde{\zeta}_t)$. We also denote $\mathcal{F}_t = \sigma(\zeta_0, \xi_0, \ldots, \zeta_t, \xi_t)$ as the corresponding filtration up to time step t. We choose parameters in Algorithm 2 as follows:

$$\eta = \frac{1}{\iota^9 \cdot \ell\mathfrak{R}}, \qquad r = \iota \cdot \epsilon\sqrt{\mathfrak{R}}, \qquad \mathscr{T} := \frac{\iota}{\eta\sqrt{\rho\epsilon}}, \qquad \mathscr{F} := \frac{1}{\iota^5}\sqrt{\frac{\epsilon^3}{\rho}}, \qquad \mathscr{S} := \frac{2}{\iota^2}\sqrt{\frac{\epsilon}{\rho}}, \tag{8}$$

where $\mathfrak{R}$ and the log factor ι are defined as:

$$\mathfrak{R} = 1 + \min\left\{\frac{\sigma^2}{\epsilon^2} + \frac{\tilde{\ell}^2}{\ell\sqrt{\rho\epsilon}}, \frac{\sigma^2 d}{\epsilon^2}\right\}, \qquad \iota = \mu \cdot \log\left(\frac{d\ell\Delta_f\mathfrak{R}}{\rho\epsilon\delta}\right).$$

Here, μ is a sufficiently large absolute constant to be determined later. Note also that throughout this section c denotes an absolute constant that does not depend on the choice of μ. Its value may change from line to line.

B.2 Descent Lemma

We first prove that the change in the function value can be always decomposed into the decrease due to the magnitudes of gradients and the possible increase due to randomness in both the stochastic gradients and the perturbations.

Lemma B.1 (Descent Lemma). *Under Assumption A, B, there exists an absolute constant c such that, for any fixed $t, t_0, \iota > 0$, if $\eta \le 1/\ell$, then with at least $1 - 4e^{-\iota}$ probability, the sequence PSGD(η, r) (Algorithm 2) satisfies: (denote $\tilde{\sigma}^2 = \sigma^2 + r^2$)*

$$f(\mathbf{x}_{t_0+t}) - f(\mathbf{x}_{t_0}) \le -\frac{\eta}{8}\sum_{i=0}^{t-1}\|\nabla f(\mathbf{x}_{t_0+i})\|^2 + c\cdot\eta\tilde{\sigma}^2(\eta\ell t + \iota).$$

Proof. Since Algorithm 2 is Markovian, the operations in each iteration do not depend on time step t. Thus, it suffices to prove Lemma B.1 for special case $t_0 = 0$. Recall the update equation:

$$\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta(\nabla f(\mathbf{x}_t) + \tilde{\zeta}_t),$$

where $\tilde{\zeta}_t = \zeta_t + \xi_t$. By assumption, we know $\zeta_t|\mathcal{F}_{t-1}$ is zero-mean nSG(σ) (see Definition C.1). Also $\xi_t|\mathcal{F}_{t-1}$ comes from $\mathcal{N}(\mathbf{0}, (r^2/d)\mathbf{I})$, and thus by Lemma C.2 it is zero-mean nSG$(c\cdot r)$ for some absolute constant c. By a Taylor expansion and the assumptions of an ℓ-gradient Lipschitz and $\eta \le 1/\ell$, we have:

$$\begin{aligned}
f(\mathbf{x}_{t+1}) &\le f(\mathbf{x}_t) + \langle\nabla f(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t\rangle + \frac{\ell}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2\\
&\le f(\mathbf{x}_t) - \eta\langle\nabla f(\mathbf{x}_t), \nabla f(\mathbf{x}_t) + \tilde{\zeta}_t\rangle + \frac{\eta^2\ell}{2}\left[\frac{3}{2}\|\nabla f(\mathbf{x}_t)\|^2 + 3\|\tilde{\zeta}_t\|^2\right]\\
&\le f(\mathbf{x}_t) - \frac{\eta}{4}\|\nabla f(\mathbf{x}_t)\|^2 - \eta\langle\nabla f(\mathbf{x}_t), \tilde{\zeta}_t\rangle + \frac{3}{2}\eta^2\ell\|\tilde{\zeta}_t\|^2.
\end{aligned}$$

Summing over this inequality, we have the following:

$$f(\mathbf{x}_t) - f(\mathbf{x}_0) \le -\frac{\eta}{4}\sum_{i=0}^{t-1}\|\nabla f(\mathbf{x}_i)\|^2 - \eta\sum_{i=0}^{t-1}\langle\nabla f(\mathbf{x}_i), \tilde{\zeta}_i\rangle + \frac{3}{2}\eta^2\ell\sum_{i=0}^{t-1}\|\tilde{\zeta}_i\|^2. \tag{9}$$

For the second term on the right-hand side, applying Lemma C.8, there exists an absolute constant c, such that, with probability $1 - 2e^{-\iota}$:

$$-\eta\sum_{i=0}^{t-1}\langle\nabla f(\mathbf{x}_i), \tilde{\zeta}_i\rangle \le \frac{\eta}{8}\sum_{i=0}^{t-1}\|\nabla f(\mathbf{x}_i)\|^2 + c\eta\tilde{\sigma}^2\iota.$$

For the third term on the right-hand side of Equation (9), applying Lemma C.7, with probability $1 - 2e^{-\iota}$:

$$\frac{3}{2}\eta^2\ell\sum_{i=0}^{t-1}\|\tilde{\zeta}_i\|^2 \le 3\eta^2\ell\sum_{i=0}^{t-1}(\|\zeta_i\|^2 + \|\xi_i\|^2) \le c\eta^2\ell\tilde{\sigma}^2(t+\iota).$$

Substituting these inequalities into Equation (9), and noting the fact $\eta \le 1/\ell$, we have with probability $1 - 4e^{-\iota}$:

$$f(\mathbf{x}_t) - f(\mathbf{x}_0) \le -\frac{\eta}{8}\sum_{i=0}^{t-1}\|\nabla f(\mathbf{x}_i)\|^2 + c\eta\tilde{\sigma}^2(\eta\ell t + \iota).$$

This finishes the proof. □

The descent lemma allows us to show the following "Improve or Localize" phenomenon for perturbed SGD. That is, with high probability over a small number of iterations, either the function value decreases significantly, or the iterates stay within a small local region.

Lemma B.2 (Improve or Localize). *Under the same setting as in Lemma B.1, with probability at least* $1-8dt\cdot e^{-\iota}$, *the sequence PSGD*(η, r) *(Algorithm 2) satisfies:*

$$\forall \tau \le t : \|\mathbf{x}_{t_0+\tau} - \mathbf{x}_{t_0}\|^2 \le c\eta t \cdot [f(\mathbf{x}_{t_0}) - f(\mathbf{x}_{t_0+\tau}) + \eta\tilde{\sigma}^2(\eta\ell t + \iota)].$$

Proof. By a similar argument as in the proof of Lemma B.1, it suffices to prove Lemma B.2 in the special case $t_0 = 0$. According to Lemma B.1, with probability $1-4e^{-\iota}$, for some absolute constant c:

$$\sum_{i=0}^{t-1} \|\nabla f(\mathbf{x}_i)\|^2 \le \frac{8}{\eta}[f(\mathbf{x}_0) - f(\mathbf{x}_t)] + c\tilde{\sigma}^2(\eta\ell t + \iota).$$

Therefore, for any fixed $\tau \le t$, with probability $1-8d\cdot e^{-\iota}$,

$$\begin{aligned}
\|\mathbf{x}_\tau - \mathbf{x}_0\|^2 &= \eta^2 \left\|\sum_{i=0}^{\tau-1}(\nabla f(\mathbf{x}_i) + \tilde{\zeta}_i)\right\|^2 \le 2\eta^2 \left[\left\|\sum_{i=0}^{\tau-1}\nabla f(\mathbf{x}_i)\right\|^2 + \left\|\sum_{i=0}^{\tau-1}\tilde{\zeta}_i\right\|^2\right] \\
&\overset{(1)}{\le} 2\eta^2 t\sum_{i=0}^{\tau-1}\|\nabla f(\mathbf{x}_i)\|^2 + c\eta^2\tilde{\sigma}^2 t\iota \le 2\eta^2 t\sum_{i=0}^{t-1}\|\nabla f(\mathbf{x}_i)\|^2 + c\eta^2\tilde{\sigma}^2 t\iota \\
&\le c\eta t[f(\mathbf{x}_0) - f(\mathbf{x}_t) + \eta\tilde{\sigma}^2(\eta\ell t + \iota)].
\end{aligned}$$

where in step (1) we use the Cauchy-Schwarz inequality and Lemma C.5. Finally, applying a union bound for all $\tau \le t$, we finish the proof. □

B.3 Escaping Saddle Points

Lemma B.1 shows that large gradients contribute to the fast decrease of the function value. In this subsection, we will show that starting in the vicinity of strict saddle points will also enable PSGD to decrease the function value rapidly. Concretely, this entire subsection will be devoted to proving the following lemma:

Lemma B.3 (Escaping Saddle Point). *Given Assumption A, B, there exists an absolute constant* $c_{\max}$ *such that, for any fixed* $t_0 > 0, \iota > c_{\max}\log(\ell\sqrt{d/(\rho\epsilon)})$, *if* $\eta, r, \mathscr{F}, \mathscr{T}$ *are chosen as in Equation (8), and* $\mathbf{x}_{t_0}$ *satisfies* $\|\nabla f(\mathbf{x}_{t_0})\| \le \epsilon$ *and* $\lambda_{\min}(\nabla^2 f(\mathbf{x}_{t_0})) \le -\sqrt{\rho\epsilon}$, *then the sequence PSGD*(η, r) *(Algorithm 2) satisfies:*

$$\begin{aligned}
&\mathbb{P}(f(\mathbf{x}_{t_0+\mathscr{T}}) - f(\mathbf{x}_{t_0}) \le 0.1\mathscr{F}) \ge 1 - 4e^{-\iota} \qquad \text{and} \\
&\mathbb{P}(f(\mathbf{x}_{t_0+\mathscr{T}}) - f(\mathbf{x}_{t_0}) \le -\mathscr{F}) \ge 1/3 - 5d\mathscr{T}^2\cdot\log(\mathscr{S}\sqrt{d}/(\eta r))e^{-\iota}.
\end{aligned}$$

Since Algorithm 2 is Markovian, the operations in each iterations do not depend on time step t. Thus, it suffices to prove Lemma B.3 for special case $t_0 = 0$. To prove this lemma, we first need to introduce the concept of a coupling sequence.

Notation. Throughout this subsection, we let $\mathcal{H} := \nabla^2 f(\mathbf{x}_0)$, $\mathbf{e}_1$ be the minimum eigendirection of $\mathcal{H}$, and $\gamma := \lambda_{\min}(\mathcal{H})$. We also let $\mathcal{P}_{-1}$ be the projection onto the subspace complement of $\mathbf{e}_1$.

Definition B.4 (Coupling Sequence). Consider sequences $\{\mathbf{x}_i\}$ and $\{\mathbf{x}'_i\}$ that are obtained as separate runs of the PSGD (algorithm 2), both starting from $\mathbf{x}_0$. They are coupled if both sequences share the same randomness $\mathcal{P}_{-1}\xi_\tau$ and θ_τ, while in $\mathbf{e}_1$ direction we have $\mathbf{e}_1^\top\xi_\tau = -\mathbf{e}_1^\top\xi'_\tau$.

The first thing we can show is that if the function values of both sequences do not exhibit a sufficient decrease, then both sequences are localized in a small ball around $\mathbf{x}_0$ within $\mathscr{T}$ iterations.

Lemma B.5 (Localization). *Under the notation of Lemma B.6, we have:*

$$\mathbb{P}(\min\{f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0), f(\mathbf{x}'_{\mathscr{T}}) - f(\mathbf{x}_0)\} \le -\mathscr{F}, \text{ or } \forall t \le \mathscr{T} : \max\{\|\mathbf{x}_t - \mathbf{x}_0\|^2, \|\mathbf{x}'_t - \mathbf{x}_0\|^2\} \le \mathscr{S}^2) \ge 1 - 16d\mathscr{T} \cdot e^{-\iota}.$$

Proof. This lemma follows from applying Lemma B.2 on both sequences and using a union bound. □

The overall proof strategy for Lemma B.3 is to show that localization happens with a very small probability, thus at least one of the sequence must have sufficient descent. To prove this, we study the dynamics of the difference of the coupling sequence.

Lemma B.6 (Dynamics of the Coupling Sequence Difference). *Consider coupling sequences $\{\mathbf{x}_i\}$ and $\{\mathbf{x}'_i\}$ as in Definition B.4 and let $\hat{\mathbf{x}}_t := \mathbf{x}_i - \mathbf{x}'_i$. Then $\hat{\mathbf{x}}_t = -\mathbf{q}_h(t) - \mathbf{q}_{sg}(t) - \mathbf{q}_p(t)$, where:*

$$\mathbf{q}_h(t) := \eta \sum_{\tau=0}^{t-1} (\mathbf{I} - \eta\mathcal{H})^{t-1-\tau} \Delta_\tau \hat{\mathbf{x}}_\tau, \mathbf{q}_{sg}(t) := \eta \sum_{\tau=0}^{t-1} (\mathbf{I} - \eta\mathcal{H})^{t-1-\tau} \hat{\zeta}_\tau, \mathbf{q}_p(t) := \eta \sum_{\tau=0}^{t-1} (\mathbf{I} - \eta\mathcal{H})^{t-1-\tau} \hat{\xi}_\tau.$$

Here $\Delta_t := \int_0^1 \nabla^2 f(\psi \mathbf{x}_t + (1-\psi)\mathbf{x}'_t) \mathrm{d}\psi - \mathcal{H}$, and $\hat{\zeta}_\tau := \zeta_\tau - \zeta'_\tau$, $\hat{\xi}_\tau := \xi_\tau - \xi'_\tau$.

Proof. Recall $\zeta_i = \mathbf{g}(\mathbf{x}_i; \theta_i) - \nabla f(\mathbf{x}_i)$, thus, we have the update formula:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta(\mathbf{g}(\mathbf{x}_t; \theta_t) + \xi_t) = \mathbf{x}_t - \eta(\nabla f(\mathbf{x}_t) + \zeta_t + \xi_t).$$

Taking the difference between $\{\mathbf{x}_i\}$ and $\{\mathbf{x}'_i\}$:

$$\begin{aligned}
\hat{\mathbf{x}}_{t+1} &= \mathbf{x}_{t+1} - \mathbf{x}'_{t+1} = \hat{\mathbf{x}}_t - \eta(\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}'_t) + \zeta_t - \zeta'_t + (\xi_t - \xi'_t)) \\
&= \hat{\mathbf{x}}_t - \eta[(\mathcal{H} + \Delta_t)\hat{\mathbf{x}}_t + \hat{\zeta}_t + \hat{\xi}_t] = (\mathbf{I} - \eta\mathcal{H})\hat{\mathbf{x}}_t - \eta[\Delta_t \hat{\mathbf{x}}_t + \hat{\zeta}_t + \mathbf{e}_1 \mathbf{e}_1^\top \hat{\xi}_t] \\
&= -\eta \sum_{\tau=0} t(I - \eta\mathcal{H})^{t-\tau}(\Delta_\tau \hat{\mathbf{x}}_\tau + \hat{\zeta}_\tau + \hat{\xi}_\tau)),
\end{aligned}$$

where $\Delta_t := \int_0^1 \nabla^2 f(\psi \mathbf{x}_t + (1-\psi)\mathbf{x}'_t) \mathrm{d}\psi - \mathcal{H}$. This finishes the proof. □

At a high level, we will show that with constant probability, $\mathbf{q}_p(t)$ is the dominating term that controls the behavior of the dynamics, and $\mathbf{q}_h(t)$ and $\mathbf{q}_{sg}(t)$ will stay small compared to $\mathbf{q}_p(t)$. To show this, we prove the following three lemmas.

Lemma B.7. *Denote $\alpha(t) := [\sum_{\tau=0}^{t-1} (1+\eta\gamma)^{2(t-1-\tau)}]^{\frac{1}{2}}$ and $\beta(t) := (1+\eta\gamma)^t / \sqrt{2\eta\gamma}$. If $\eta\gamma \in [0,1]$, then (1) $\alpha(t) \le \beta(t)$ for any $t \in \mathbb{N}$; and (2) $\alpha(t) \ge \beta(t)/\sqrt{3}$ for $t \ge \ln(2)/(\eta\gamma)$.*

Proof. By summing a geometric sequence:

$$\alpha^2(t) := \sum_{\tau=0}^{t-1} (1+\eta\gamma)^{2(t-1-\tau)} = \frac{(1+\eta\gamma)^{2t} - 1}{2\eta\gamma + (\eta\gamma)^2}.$$

Thus, the claim that $\alpha(t) \le \beta(t)$ for any $t \in \mathbb{N}$ follows immediately. However, note that for $t \ge \ln(2)/(\eta\gamma)$, we have $(1+\eta\gamma)^{2t} \ge 2^{2\ln 2} \ge 2$, where the second claim follows by a short calculation. □

Lemma B.8. *Under the notation of Lemma B.6 and Lemma B.7, letting $-\gamma := \lambda_{\min}(\mathcal{H})$, we have $\forall t > 0$:*

$$\mathbb{P}(\|\mathbf{q}_p(t)\| \le \frac{c\beta(t)\eta r}{\sqrt{d}} \cdot \sqrt{\iota}) \ge 1 - 2e^{-\iota}$$
$$\mathbb{P}(\|\mathbf{q}_p(\mathscr{T})\| \ge \frac{\beta(\mathscr{T})\eta r}{10\sqrt{d}}) \ge \frac{2}{3}.$$

Proof. Note that $\hat{\xi}_\tau$ is one-dimensional Gaussian random variable with standard deviation $2r/\sqrt{d}$ along the $\mathbf{e}_1$ direction. As an immediate result, $\eta\sum_{\tau=0}^{t}(I-\eta\mathcal{H})^{t-\tau}\hat{\xi}_\tau$ is also a one-dimensional Gaussian distribution, since the summation of Gaussians is again Gaussian. Finally note that $\mathbf{e}_1$ is an eigendirection of $\mathcal{H}$ with corresponding eigenvalue $-\gamma$, and by Lemma B.7 we have $\alpha(t) \le \beta(t)$. Then, the first inequality immediately follows from the standard concentration of Gaussian measure, and the second inequality follows from the fact if $Z \sim \mathcal{N}(0,\sigma^2)$, then $\mathbb{P}(|Z| \le \lambda\sigma) \le 2\lambda/\sqrt{2\pi} \le \lambda$. □

Lemma B.9. *There exists an absolute constant $c_{\max}$ such that, for any $\iota \ge c_{\max}$, under the notation of Lemma B.6 and Lemma B.7, and letting $-\gamma := \lambda_{\min}(\mathcal{H})$, we have:*

$$\mathbb{P}(\min\{f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0), f(\mathbf{x}'_{\mathscr{T}}) - f(\mathbf{x}_0)\} \le -\mathscr{F}, \text{ or}$$
$$\forall t \le \mathscr{T} : \|\mathbf{q}_h(t) + \mathbf{q}_{sg}(t)\| \le \frac{\beta(t)\eta r}{20\sqrt{d}}) \ge 1 - 10d\mathscr{T}^2 \cdot \log\left(\frac{\mathscr{S}\sqrt{d}}{\eta r}\right)e^{-\iota}.$$

Proof. For simplicity, we denote $\mathfrak{E}$ as the event $\{\forall\tau \le t : \max\{\|\mathbf{x}_\tau - \mathbf{x}_0\|^2, \|\mathbf{x}'_\tau - \mathbf{x}_0\|^2\} \le \mathscr{S}^2\}$. We use induction to prove following claim for any $t \in [0, \mathscr{T}]$:

$$\mathbb{P}(\mathfrak{E} \Rightarrow \forall\tau \le t : \|\mathbf{q}_h(\tau) + \mathbf{q}_{sg}(\tau)\| \le \frac{\beta(\tau)\eta r}{20\sqrt{d}}) \ge 1 - 10d\mathscr{T}t \cdot \log\left(\frac{\mathscr{S}\sqrt{d}}{\eta r}\right)e^{-\iota}.$$

Then Lemma B.9 follows directly from combining Lemma B.5 and the induction claim.

Clearly for the base case $t = 0$, the claim holds trivially, as $\mathbf{q}_{sg}(0) = \mathbf{q}_h(0) = \mathbf{0}$. Suppose the claim holds for t, then by Lemma B.8, with probability at least $1 - 2\mathscr{T}e^{-\iota}$, we have for any $\tau \le t$:

$$\|\hat{\mathbf{x}}_\tau\| \le \eta\|\mathbf{q}_h(\tau) + \mathbf{q}_{sg}(\tau)\| + \eta\|\mathbf{q}_p(\tau)\| \le \frac{c\beta(\tau)\eta r}{\sqrt{d}} \cdot \sqrt{\iota}.$$

Then, under the condition $\max\{\|\mathbf{x}_\tau - \mathbf{x}_0\|^2, \|\mathbf{x}'_\tau - \mathbf{x}_0\|^2\} \le \mathscr{S}^2$, by the Hessian Lipschitz property, we have $\|\Delta_\tau\| = \|\int_0^1 \nabla^2 f(\psi\mathbf{x}_\tau + (1-\psi)\mathbf{x}'_\tau)\mathrm{d}\psi - \mathcal{H}\| \le \rho\max\{\|\mathbf{x}_\tau - \mathbf{x}_0\|, \|\mathbf{x}'_\tau - \mathbf{x}_0\|\} \le \rho\mathscr{S}$. This gives bounds on $\mathbf{q}_h(t+1)$ terms as:

$$\|\mathbf{q}_h(t+1)\| \le \eta\sum_{\tau=0}^{t}(1+\eta\gamma)^{t-\tau}\rho\mathscr{S}\|\hat{\mathbf{x}}_\tau\| \le \eta\rho\mathscr{S}\mathscr{T}\frac{c\beta(t)\eta r}{\sqrt{d}} \le \frac{\beta(t)\eta r}{40\sqrt{d}},$$

where the last step is due to $\eta\rho\mathscr{S}\mathscr{T} = 1/\iota$ by Equation (8). By picking ι larger than the absolute constant $40c$, we have $c\eta\rho\mathscr{S}\mathscr{T} \le 1/40$.

Recall also that $\hat{\zeta}_\tau|\mathcal{F}_{\tau-1}$ is the summation of a nSG(σ) random vector and a nSG($c\cdot r$) random vector. By Lemma C.5, we know that with probability at least $1 - 4de^{-\iota}$:

$$\|\mathbf{q}_{sg}(t+1)\| \le c\beta(t+1)\eta\sigma\sqrt{\iota}$$

However, when assumption C is avaliable, we also have $\hat{\zeta}_\tau|\mathcal{F}_{\tau-1} \sim \mathrm{nSG}(\tilde{\ell}\|\hat{\mathbf{x}}_\tau\|)$, by applying Lemma C.6 with $B = \alpha^2(t)\cdot\eta^2\tilde{\ell}^2\mathscr{S}^2; b = \alpha^2(t)\cdot\eta^2\tilde{\ell}^2\cdot\eta^2r^2/d$, we know with probability at least

$1 - 4d \cdot \log(\mathscr{S}\sqrt{d}/(\eta r)) \cdot e^{-\iota}$:

$$\|\mathbf{q}_{sg}(t+1)\| \le c\eta\tilde{\ell}\sqrt{\sum_{\tau=0}^{t}(1+\eta\gamma)^{2(t-\tau)} \cdot \max\{\|\hat{\mathbf{x}}_\tau\|^2, \frac{\eta^2 r^2}{d}\}\iota} \le \eta\tilde{\ell}\sqrt{\mathscr{T}} \cdot \frac{c\beta(t)\eta r}{\sqrt{d}} \cdot \sqrt{\iota}. \tag{10}$$

Finally, combining both cases, and by our choice of step size η, r as in Equation (8) with ι large enough:

$$\|\mathbf{q}_{sg}(t+1)\| \le c\frac{\beta(t)\eta r}{\sqrt{d}} \cdot \min\left\{\eta\tilde{\ell}\sqrt{\mathscr{T}\iota}, \frac{\sigma\sqrt{d\iota}}{r}\right\} \le \frac{\beta(t)r}{40\sqrt{d}},$$

the induction follows by the triangle inequality and a union bound. □

We are ready to prove Lemma B.3, which is the focus of this subsection.

Proof of Lemma B.3. We first prove the first claim $\mathbb{P}(f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0) \le 0.1\mathscr{F}) \ge 1 - 4e^{-\iota}$. Because of our choice of step size and Lemma B.1, we have with probability $1 - 4e^{-\iota}$:

$$f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0) \le c\eta\tilde{\sigma}^2(\eta\ell\mathscr{T} + \iota) \le 0.1\mathscr{F},$$

where the last step is because our choice of parameters in Equation (8) implies $c\eta\tilde{\sigma}^2(\eta\ell\mathscr{T} + \iota) \le 2c\mathscr{F}/\iota$ and we pick ι to be larger than an absolute constant $20c$.

For the second claim, $\mathbb{P}(f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0) \le -\mathscr{F}) \ge 1/3 - 5d\mathscr{T}^2 \cdot \log(\mathscr{S}\sqrt{d}/(\eta r))e^{-\iota}$, we consider coupling sequences $\{\mathbf{x}_i\}$ and $\{\mathbf{x}'_i\}$ as defined in Definition B.4. Given Lemma B.8 and Lemma B.9, we know that with probability at least $2/3 - 10d\mathscr{T}^2 \cdot \log(\mathscr{S}\sqrt{d}/(\eta r))e^{-\iota}$ if $\min\{f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0), f(\mathbf{x}'_{\mathscr{T}}) - f(\mathbf{x}_0)\} > -\mathscr{F}$—i.e., both sequences are stuck around the saddle point—we must have:

$$\|\mathbf{q}_p(\mathscr{T})\| \ge \frac{\beta(\mathscr{T})\eta r}{10\sqrt{d}}, \quad \|\mathbf{q}_h(\mathscr{T}) + \mathbf{q}_{sg}(\mathscr{T})\| \le \frac{\beta(\mathscr{T})\eta r}{20\sqrt{d}}.$$

By Lemma B.6, when $\iota \ge c \cdot \log(\ell\sqrt{d/(\rho\epsilon)})$ for a large absolute constant c, we have:

$$\begin{aligned}\max\{\|\mathbf{x}_{\mathscr{T}} - \mathbf{x}_0\|, \|\mathbf{x}'_{\mathscr{T}} - \mathbf{x}_0\|\} \ge& \frac{1}{2}\|\hat{\mathbf{x}}(\mathscr{T})\| \ge \frac{1}{2}[\|\mathbf{q}_p(\mathscr{T})\| - \|\mathbf{q}_h(\mathscr{T}) + \mathbf{q}_{sg}(\mathscr{T})\|] \\ \ge& \frac{\beta(\mathscr{T})\eta r}{40\sqrt{d}} = \frac{(1+\eta\gamma)^{\mathscr{T}}\eta r}{40\sqrt{2\eta\gamma d}} \le \frac{2^\iota \eta r}{80\sqrt{\eta\ell d}}, > \mathscr{S},\end{aligned}$$

which contradicts with Lemma B.5. Therefore, we can conclude that $\mathbb{P}(\min\{f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0), f(\mathbf{x}'_{\mathscr{T}}) - f(\mathbf{x}_0)\} \le -\mathscr{F}) \ge 2/3 - 10d\mathscr{T}^2 \cdot \log(\mathscr{S}\sqrt{d}/(\eta r))e^{-\iota}$. We also know that the marginal distribution of $\mathbf{x}_{\mathscr{T}}$ and $\mathbf{x}'_{\mathscr{T}}$ is the same, thus they have same probability to escape the saddle point. That is,

$$\begin{aligned}\mathbb{P}(f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0) \le -\mathscr{F}) \ge& \frac{1}{2}\mathbb{P}(\min\{f(\mathbf{x}_{\mathscr{T}}) - f(\mathbf{x}_0), f(\mathbf{x}'_{\mathscr{T}}) - f(\mathbf{x}_0)\} \le -\mathscr{F}) \\ \ge& 1/3 - 5d\mathscr{T}^2 \cdot \log(\mathscr{S}\sqrt{d}/(\eta r))e^{-\iota}.\end{aligned}$$

This finishes the proof. □

B.4 Proof of Theorem 4.4

Lemma B.1 and Lemma B.3 describe the speed of decrease in the function values when either large gradients or strictly negative curvatures are present. Combining them gives the proof for our main theorem.

Proof of Theorem 4.4. First, we set the total number of iterations T to be

$$T = 100 \max\left\{\frac{(f(x_0)-f^\star)\mathscr{T}}{\mathscr{F}}, \frac{(f(x_0)-f^\star)}{\eta\epsilon^2}\right\} = O\left(\frac{\ell(f(x_0)-f^\star)}{\epsilon^2}\cdot\mathfrak{R}\cdot\iota^9\right).$$

We will show that the following **two claims** hold simultaneously with probability $1-\delta$:

(1) At most $T/4$ iterates have large gradient; i.e., $\|\nabla f(\mathbf{x}_t)\| \geq \epsilon$;
(2) At most $T/4$ iterates are close to saddle points; i.e., $\|\nabla f(\mathbf{x}_t)\| \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(\mathbf{x}_t)) \leq -\sqrt{\rho\epsilon}$.

Therefore, at least $T/2$ iterates are ϵ-second-order stationary point. We prove the two claims separately.

Claim 1. *Suppose that within T steps, we have more than $T/4$ iterates for which gradient is large (i.e., $\|\nabla f(\mathbf{x}_t)\| \geq \epsilon$). Recall that by Lemma B.1 we have with probability $1-4e^{-\iota}$:*

$$f(\mathbf{x}_T) - f(\mathbf{x}_0) \leq -\frac{\eta}{8}\sum_{i=0}^{T-1}\|\nabla f(\mathbf{x}_i)\|^2 + c\eta\tilde{\sigma}^2(\eta\ell T+\iota) \leq -\eta\left[\frac{T\epsilon^2}{32} - \tilde{\sigma}^2(\eta\ell T+\iota)\right].$$

Note that by our choice of η, r, T and picking ι larger than some absolute constant, we have $T\epsilon^2/32 - \tilde{\sigma}^2(\eta\ell T+\iota) \geq T\epsilon^2/64$ and thus $f(\mathbf{x}_T) \leq f(x_0) - T\eta\epsilon^2/64 < f^\star$, which is not possible.

Claim 2. *We first define the stopping times that allow us to invoke Lemma B.3:*

$$z_1 = \inf\{\tau | \|\nabla f(\mathbf{x}_\tau)\| \leq \epsilon \text{ and } \lambda_{\min}(f(\mathbf{x}_\tau)) \leq -\sqrt{\rho\epsilon}\}$$
$$z_i = \inf\{\tau | \tau > z_{i-1} + \mathscr{T} \text{ and } \|\nabla f(\mathbf{x}_\tau)\| \leq \epsilon \text{ and } \lambda_{\min}(f(\mathbf{x}_\tau)) \leq -\sqrt{\rho\epsilon}\}, \qquad \forall i > 1.$$

Clearly, z_i is a stopping time, and it is the ith time in the sequence along which we can apply Lemma B.3. We also let M be the random variable $M = \max\{i | z_i + \mathscr{T} \leq T\}$. We can decompose the decrease $f(\mathbf{x}_T) - f(\mathbf{x}_0)$ as follows:

$$f(\mathbf{x}_T) - f(\mathbf{x}_0) = \underbrace{\sum_{i=1}^{M}[f(\mathbf{x}_{z_i+\mathscr{T}}) - f(\mathbf{x}_{z_i})]}_{T_1}$$
$$+ \underbrace{[f(\mathbf{x}_T) - f(\mathbf{x}_{z_M})] + [f(\mathbf{x}_{z_1}) - f(\mathbf{x}_0)] + \sum_{i=1}^{M-1}[f(\mathbf{x}_{z_{i+1}}) - f(\mathbf{x}_{z_i+\mathscr{T}})]}_{T_2}.$$

For the first term T_1, by Lemma B.3 and a supermartingale concentration inequality, for each fixed $m \leq T$:

$$\mathbb{P}\left(\sum_{i=1}^{m}[f(\mathbf{x}_{z_i+\mathscr{T}}) - f(\mathbf{x}_{z_i})] \leq -(0.9m - c\sqrt{m\cdot\iota})\mathscr{F}\right) \geq 1 - 5d\mathscr{T}^2 T\cdot\log(\mathscr{S}\sqrt{d}/(\eta r))e^{-\iota}.$$

Since the random variable $M \leq T/\mathscr{T} \leq T$, by a union bound, we know that with probability $1-5d\mathscr{T}^2T^2\cdot\log(\mathscr{S}\sqrt{d}/(\eta r))e^{-\iota}$:

$$T_1 \leq -(0.9M - c\sqrt{M\cdot\iota})\mathscr{F}.$$

For the second term, by a union bound and Lemma B.1 for all $0 \leq t_1, t_2 \leq T$, with probability $1-4T^2e^{-\iota}$:

$$T_2 \leq c\cdot\eta\tilde{\sigma}^2(\eta\ell T + 2M\iota)$$

Therefore, if within T steps we have more than $T/4$ saddle points, then $M \geq T/4\mathscr{T}$, and with probaility $1 - 10d\mathscr{T}^2T^2 \cdot \log(\mathscr{S}\sqrt{d}/(\eta r))e^{-\iota}$:

$$f(\mathbf{x}_T) - f(\mathbf{x}_0) \leq -(0.9M - c\sqrt{M \cdot \iota})\mathscr{F} + c \cdot \eta\tilde{\sigma}^2(\eta\ell T + 2M\iota) \leq -0.4M\mathscr{F} \leq -0.4T\mathscr{F}/\mathscr{T}.$$

This will gives $f(\mathbf{x}_T) \leq f(x_0) - 0.4T\mathscr{F}/\mathscr{T} < f^\star$, which is not possible.

Finally, it is not hard to verify, by choosing $\iota = c \cdot \log(\frac{d\ell\Delta_f\mathfrak{R}}{\rho\epsilon\delta})$ for a large enough value of the absolute constant c, we can make both claims hold with probability $1 - \delta$. □

B.5 Proof of Corollary 4.5

Our proofs for PSGD easily generalize to the mini-batch setting.

Proof of Corollary 4.5. The proof is essentially the same as the proof of Theorem 4.4. The only difference is that, up to a log factor, mini-batch PSGD reduces the variance σ^2 and $\tilde{\ell}^2\|\hat{\mathbf{x}}_\tau\|^2$ in Equation (10) by a factor of m, where m is the size of the mini-batch. □

B.6 Proof of Remark 4.2

The proof of Theorem 4.4 can be easily modified to prove Remark 4.2.

Proof of Remark 4.2. In the proof of Theorem 4.4, we have shown that if we set the total number of iterations T to be:

$$T = O\left(\frac{\ell(f(x_0) - f^\star)}{\epsilon^2} \cdot \mathfrak{R} \cdot \iota^9\right),$$

where $\iota = c \cdot \log(\frac{d\ell\Delta_f\mathfrak{R}}{\rho\epsilon\delta})$ for a large enough value of the absolute constant c, with at least $1 - \delta$ probability, at least $T/2$ iterates are ϵ-second-order stationary point. Therefore, for $\delta \leq 1/6$, if we output an iterate uniformly at random, with at least $1/3$ probability, then we will output an ϵ-second-order stationary point. □

C CONCENTRATION INEQUALITIES

In this section, we present the concentration inequalities required for this article. Please refer to the technical note [28] for the proofs of Lemmas C.2, C.3, C.5, and C.6.

Recall the definition of a norm-subGaussian random vector.

Definition C.1. A random vector $\mathbf{X} \in \mathbb{R}^d$ is *norm-subGaussian* (or nSG(σ)), if there exists σ so that:

$$\mathbb{P}\left(\|\mathbf{X} - \mathbb{E}\mathbf{X}\| \geq t\right) \leq 2e^{-\frac{t^2}{2\sigma^2}}, \qquad \forall t \in \mathbb{R}.$$

Note that a bounded random vector and a subGaussian random vector are two special cases of a norm-subGaussian random vector.

Lemma C.2. *There exists an absolute constant c so that following random vectors are nSG($c \cdot \sigma$).*

(1) A bounded random vector $\mathbf{X} \in \mathbb{R}^d$ such that $\|\mathbf{X}\| \leq \sigma$.
(2) A random vector $\mathbf{X} \in \mathbb{R}^d$, where $\mathbf{X} = \xi\mathbf{e}_1$ and the random variable $\xi \in \mathbb{R}$ is σ-subGaussian.
(3) A random vector $\mathbf{X} \in \mathbb{R}^d$ that is $(\sigma/\sqrt{d})$-subGaussian.

Second, we have that if $\mathbf{X}$ is norm-subGaussian, then its norm square is subExponential, and its component along a single direction is subGaussian.

Lemma C.3. *There is an absolute constant c so that if the random vector $\mathbf{X} \in R^d$ is zero-mean nSG(σ), then $\|\mathbf{X}\|^2$ is $c \cdot \sigma^2$-subExponential, and for any fixed unit vector $\mathbf{v} \in \mathbb{S}^{d-1}$, $\langle \mathbf{v}, \mathbf{X}\rangle$ is $c \cdot \sigma$-subGaussian.*

For concentration, we are interested in the properties of norm-subGaussian martingale difference sequences. Concretely, they are sequences satisfying the following conditions.

Condition C.4. *Consider random vectors $\mathbf{X}_1, \ldots, \mathbf{X}_n \in \mathbb{R}^d$, and corresponding filtrations $\mathcal{F}_i = \sigma(\mathbf{X}_1, \ldots, \mathbf{X}_i)$ for $i \in [n]$, such that $\mathbf{X}_i | \mathcal{F}_{i-1}$ is zero-mean $nSG(\sigma_i)$ with $\sigma_i \in \mathcal{F}_{i-1}$. That is:*

$$\mathbb{E}[\mathbf{X}_i | \mathcal{F}_{i-1}] = 0, \quad \mathbb{P}\left(\|\mathbf{X}_i\| \geq t | \mathcal{F}_{i-1}\right) \leq 2e^{-\frac{t^2}{2\sigma_i^2}}, \qquad \forall t \in \mathbb{R}, \forall i \in [n].$$

Similarly to subGaussian random variables, we can also prove a Hoeffding-type inequality for norm-subGaussian random vectors that is tight up to a $\log(d)$ factor.

Lemma C.5 (Hoeffding-type inequality for norm-subGaussian). *Given $\mathbf{X}_1, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ that satisfy condition C.4, with fixed $\{\sigma_i\}$, then for any $\iota > 0$, there exists an absolute constant c such that, with probability at least $1 - 2d \cdot e^{-\iota}$:*

$$\left\| \sum_{i=1}^n \mathbf{X}_i \right\| \leq c \cdot \sqrt{\sum_{i=1}^n \sigma_i^2 \cdot \iota}.$$

When $\{\sigma_i\}$ is also random, we have the following.

Lemma C.6. *Given $\mathbf{X}_1, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ that satisfy condition C.4, then for any $\iota > 0$, and $B > b > 0$, there exists an absolute constant c such that, with probability at least $1 - 2d \log(B/b) \cdot e^{-\iota}$:*

$$\sum_{i=1}^n \sigma_i^2 \geq B \quad \text{or} \quad \left\| \sum_{i=1}^n \mathbf{X}_i \right\| \leq c \cdot \sqrt{\max\left\{ \sum_{i=1}^n \sigma_i^2, b \right\} \cdot \iota}.$$

Finally, we can also provide concentration inequalities for the sum of norm squares of norm-subGaussian random vectors, and for the sum of inner products of norm-subGaussian random vectors with another set of random vectors.

Lemma C.7. *Given $\mathbf{X}_1, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ that satisfy Condition C.4 with fixed $\sigma_1 = \cdots = \sigma_n = \sigma$, then there exists an absolute constant c such that, for any $\iota > 0$, with probability at least $1 - e^{-\iota}$:*

$$\sum_{i=1}^n \|\mathbf{X}_i\|^2 \leq c \cdot \sigma^2 (n + \iota).$$

Proof. Note there exists an absolute constant c such that $\mathbb{E}[\|\mathbf{X}_i\|^2 | \mathcal{F}_{i-1}] \leq c \cdot \sigma^2$, and $\|\mathbf{X}_i\|^2 | \mathcal{F}_{i-1}$ is $c \cdot \sigma^2$-subExponential. This lemma directly follows from standard Bernstein concentration inequalities for subExponential random variables. □

Lemma C.8. *Given $\mathbf{X}_1, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ that satisfy Condition C.4 and random vectors $\{\mathbf{u}_i\}$ that satisfy $\mathbf{u}_i \in \mathcal{F}_{i-1}$ for all $i \in [n]$, then for any $\iota > 0$, $\lambda > 0$, there exists absolute constant c such that, with probability at least $1 - e^{-\iota}$:*

$$\sum_i \langle \mathbf{u}_i, \mathbf{X}_i \rangle \leq c \cdot \lambda \sum_i \|\mathbf{u}_i\|^2 \sigma_i^2 + \frac{1}{\lambda} \cdot \iota.$$

Proof. For any $i \in [n]$ and fixed $\lambda > 0$, since $\mathbf{u}_i \in \mathcal{F}_{i-1}$, according to Lemma C.3 there exists a constant c such that $\langle \mathbf{u}_i, \mathbf{X}_i \rangle | \mathcal{F}_{i-1}$ is $c \cdot \|\mathbf{u}_i\| \sigma_i$-subGaussian. Thus:

$$\mathbb{E}[e^{\lambda \langle \mathbf{u}_i, \mathbf{X}_i \rangle} | \mathcal{F}_{i-1}] \leq e^{c \cdot \lambda^2 \|\mathbf{u}_i\|^2 \sigma_i^2}.$$

Therefore, consider the following quantity:

$$\begin{aligned}\mathbb{E}e^{\sum_{i=1}^{t}(\lambda\langle \mathbf{u}_i,\mathbf{X}_i\rangle - c\cdot\lambda^2\|\mathbf{u}_i\|^2\sigma_i^2)} &= \mathbb{E}\left[e^{\sum_{i=1}^{t-1}\lambda\langle \mathbf{u}_i,\mathbf{X}_i\rangle - c\cdot\sum_{i=1}^{t}\lambda^2\|\mathbf{u}_i\|^2\sigma_i^2}\cdot\mathbb{E}\left(e^{\lambda\langle \mathbf{u}_t,\mathbf{X}_t\rangle}|\mathcal{F}_{t-1}\right)\right]\\ &\le \mathbb{E}\left[e^{\sum_{i=1}^{t-1}\lambda\langle \mathbf{u}_i,\mathbf{X}_i\rangle - c\cdot\sum_{i=1}^{t}\lambda^2\|\mathbf{u}_i\|^2\sigma_i^2}\cdot e^{c\cdot\lambda^2\|\mathbf{u}_t\|^2\sigma_t^2}\right]\\ &= \mathbb{E}e^{\sum_{i=1}^{t-1}(\lambda\langle \mathbf{u}_i,\mathbf{X}_i\rangle - c\cdot\lambda^2\|\mathbf{u}_i\|^2\sigma_i^2)} \le 1.\end{aligned}$$

By Markov's inequality, for any $t > 0$:

$$\begin{aligned}\mathbb{P}\left(\sum_{i=1}^{t}(\lambda\langle \mathbf{u}_i,\mathbf{X}_i\rangle - c\cdot\lambda^2\|\mathbf{u}_i\|^2\sigma_i^2) \ge t\right) &\le \mathbb{P}\left(e^{\sum_{i=1}^{t}(\lambda\langle \mathbf{u}_i,\mathbf{X}_i\rangle - c\cdot\lambda^2\|\mathbf{u}_i\|^2\sigma_i^2)} \ge e^t\right)\\ &\le e^{-t}\mathbb{E}e^{\sum_{i=1}^{t}(\lambda\langle \mathbf{u}_i,\mathbf{X}_i\rangle - c\cdot\lambda^2\|\mathbf{u}_i\|^2\sigma_i^2)} \le e^{-t}.\end{aligned}$$

This finishes the proof. □

REFERENCES

[1] Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. 2017. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 1195–1199.

[2] Zeyuan Allen-Zhu. 2018. Natasha 2: Faster non-convex optimization than SGD. In *Advances in Neural Information Processing Systems*. 2680–2691.

[3] Zeyuan Allen-Zhu and Yuanzhi Li. 2018. Neon2: Finding local minima via first-order oracles. In *Advances in Neural Information Processing Systems*. 3716–3726.

[4] Animashree Anandkumar and Rong Ge. 2016. Efficient approaches for escaping higher-order saddle points in non-convex optimization. In *Conference on Learning Theory*. 81–102.

[5] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Ayush Sekhari, and Karthik Sridharan. 2020. Second-order information in non-convex stochastic optimization: Power and limitations. In *Conference on Learning Theory*.

[6] Yossi Arjevani, Yair Carmon, John C. Duchi, Dylan J. Foster, Nathan Srebro, and Blake Woodworth. 2019. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365* (2019).

[7] Afonso S. Bandeira, Nicolas Boumal, and Vladislav Voroninski. 2016. On the low-rank approach for semidefinite programs arising in synchronization and community detection. In *Conference on Learning Theory*. 361–382.

[8] Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. 2016. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems*. 3873–3881.

[9] Nicolas Boumal, Vlad Voroninski, and Afonso Bandeira. 2016. The non-convex Burer-Monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*. 2757–2765.

[10] Anton Bovier, Michael Eckhoff, Véronique Gayrard, and Markus Klein. 2004. Metastability in reversible diffusion processes I: Sharp asymptotics for capacities and exit times. *Journal of the European Mathematical Society* 6, 4 (2004), 399–424.

[11] Yair Carmon and John Duchi. 2019. Gradient descent finds the cubic-regularized nonconvex Newton step. *SIAM Journal on Optimization* 29, 3 (2019), 2146–2178.

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

[13] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. 2019. Lower bounds for finding stationary points I. *Mathematical Programming* (2019).

[14] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. 2019. Lower bounds for finding stationary points II: First-order methods. *Mathematical Programming* (2019).

[15] Louis Augustin Cauchy. 1847. Méthode générale pour la résolution des systémes d'équations simultanees. *C. R. Acad. Sci. Paris* (1847).

[16] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. 2015. The loss surface of multilayer networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[17] Frank E Curtis, Daniel P Robinson, and Mohammadreza Samadi. 2014. A trust region algorithm with a worst-case iteration complexity of $O(\epsilon^{-3/2})$ for nonconvex optimization. *Mathematical Programming* (2014), 1–32.

[18] Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. 2018. Escaping saddles with stochastic gradients. In *International Conference on Machine Learning*.

[19] Simon S Du, Chi Jin, Jason D Lee, Michael I Jordan, Aarti Singh, and Barnabas Poczos. 2017. Gradient descent can take exponential time to escape saddle points. In *Advances in Neural Information Processing Systems*. 1067–1077.

[20] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. 2018. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*. 687–697.
[21] Cong Fang, Zhouchen Lin, and Tong Zhang. 2019. Sharp analysis for nonconvex SGD escaping from saddle points. *arXiv preprint arXiv:1902.00247* (2019).
[22] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. 2015. Escaping from saddle points—Online stochastic gradient for tensor decomposition. In *Conference on Computational Learning Theory (COLT)*.
[23] Rong Ge, Chi Jin, and Yi Zheng. 2017. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning (ICML)*.
[24] Rong Ge, Jason D. Lee, and Tengyu Ma. 2016. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*. 2973–2981.
[25] Saeed Ghadimi and Guanghui Lan. 2013. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization* 23, 4 (2013), 2341–2368.
[26] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. 2013. Low-rank matrix completion using alternating minimization. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*. 665–674.
[27] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. 2017. How to escape saddle points efficiently. In *International Conference on Machine Learning (ICML)*.
[28] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M. Kakade, and Michael I. Jordan. 2019. A short note on concentration inequalities for random vectors with subGaussian norm. *arXiv preprint arXiv:1902.03736* (2019).
[29] Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. 2018. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Conference of Learning Theory (COLT)*.
[30] Jason D. Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. 2019. First-order methods almost always avoid strict saddle points. *Mathematical Programming* 176, 1–2 (2019), 311–337.
[31] Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. 2016. Gradient descent only converges to minimizers. In *Conference on Learning Theory*. 1246–1257.
[32] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I. Jordan. 2018. Finding approximate local minima faster than gradient descent. In *Advances in Neural Information Processing (NIPS) 31*. Curran Associates, Red Hook, NY.
[33] Kfir Y. Levy. 2016. The power of normalization: Faster evasion of saddle points. *arXiv preprint arXiv:1611.04831* (2016).
[34] Song Mei, Theodor Misiakiewicz, Andrea Montanari, and Roberto I. Oliveira. 2017. Solving SDPs for synchronization and MaxCut problems via the Grothendieck inequality. In *Conference on Learning Theory (COLT)*. 1476–1515.
[35] Arkadii Nemirovskii and David Borisovich Yudin. 1983. *Problem Complexity and Method Efficiency in Pptimization*. Wiley.
[36] Yurii Nesterov. 1983. A method of solving a convex programming problem with convergence rate O (1/k2). *Soviet Mathematics Doklady* 27 (1983), 372–376.
[37] Yurii Nesterov. 1998. *Introductory Lectures on Convex Programming Volume I: Basic Course*. Springer.
[38] Yurii Nesterov. 2000. Squared functional systems and optimization problems. In *High Performance Optimization*. Springer, 405–440.
[39] Yurii Nesterov and Boris T. Polyak. 2006. Cubic regularization of Newton method and its global performance. *Mathematical Programming* 108, 1 (2006), 177–205.
[40] Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. 2014. Non-convex robust PCA. In *Advances in Neural Information Processing Systems*. 1107–1115.
[41] Boris T. Polyak. 1963. Gradient methods for the minimisation of functionals. *U. S. S. R. Comput. Math. and Math. Phys.* 3, 4 (1963), 864–878.
[42] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*. 693–701.
[43] Sashank Reddi, Manzil Zaheer, Suvrit Sra, Barnabas Poczos, Francis Bach, Ruslan Salakhutdinov, and Alex Smola. 2018. A generic approach for escaping saddle points. In *International Conference on Artificial Intelligence and Statistics*. 1233–1242.
[44] Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *Annals of Mathematical Statistics* (1951), 400–407.
[45] Gareth O. Roberts, Richard L. Tweedie, et al. 1996. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli* 2, 4 (1996), 341–363.
[46] Virginia Smith, Simone Forte, Chenxin Ma, Martin Takac, Michael I. Jordan, and Martin Jaggi. 2018. CoCoA: A general framework for communication-efficient distributed optimization. *Journal of Machine Learning Research* 18 (2018), 1–49.
[47] Ju Sun, Qing Qu, and John Wright. 2016. Complete dictionary recovery over the sphere I: Overview and the geometric picture. *IEEE Transactions on Information Theory* (2016).

[48] Ju Sun, Qing Qu, and John Wright. 2016. A geometric analysis of phase retrieval. In *IEEE International Symposium on Information Theory (ISIT'16)*. IEEE, 2379–2383.
[49] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I Jordan. 2018. Stochastic cubic regularization for fast nonconvex optimization. In *Advances in Neural Information Processing Systems*. 2904–2913.
[50] Yi Xu, Jing Rong, and Tianbao Yang. 2018. First-order stochastic algorithms for escaping from saddle points in almost linear time. In *Advances in Neural Information Processing Systems*. 5535–5545.
[51] Yuchen Zhang, Percy Liang, and Moses Charikar. 2018. A hitting time analysis of stochastic gradient Langevin dynamics. In *Conference of Learning Theory (COLT)*.
[52] Yuchen Zhang, Martin J. Wainwright, and John C. Duchi. 2012. Communication-efficient algorithms for statistical optimization. In *Advances in Neural Information Processing Systems*. 1502–1510.
[53] Dongruo Zhou and Quanquan Gu. 2020. Stochastic recursive variance-reduced cubic regularization methods. In *International Conference on Artificial Intelligence and Statistics*. 3980–3990.
[54] Dongruo Zhou, Pan Xu, and Quanquan Gu. 2018. Finding local minima via stochastic nested variance reduction. *arXiv preprint arXiv:1806.08782* (2018).

Received September 2019; revised July 2020; accepted August 2020