OPT2021: 13th Annual Workshop on Optimization for Machine Learning

Escaping Local Minima With Stochastic Noise

Harshvardhan HHARSHVARDHAN@UCSD.EDU
UCSD

Sebastian U. Stich STICH@CISPA.DE
CISPA

Abstract

Non-convex optimization problems are ubiquitous in machine learning, especially in Deep Learning. While such complex problems can often be successfully optimized in practice by using stochastic gradient descent (SGD), theoretical analysis cannot adequately explain this success. In particular, the standard analyses don't show global convergence of SGD on non-convex functions, and instead show convergence to stationary points (which can also be local minima or saddle points). In this work, we identify a broad class of nonconvex functions for which we can show that perturbed SGD (gradient descent perturbed by stochastic noise—covering SGD as a special case) converges to a global minimum, in contrast to gradient descent without noise that can get stuck in local minima. In particular, for non-convex functions that are relative close to a convex-like (strongly convex or PŁ) function we prove that SGD converges linearly to a global optimum.

1. Introduction

Non-convex optimization problems are ubiquitous in deep learning and computer vision [10]. The training of a neural network amounts to minimizing a non-convex loss function $f\colon \mathbb{R}^d \to \mathbb{R}$,

$$f^\star = \min_{\mathbf{x}\in\mathbb{R}^d} \left[f(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D}}\, f(\mathbf{x},\boldsymbol{\xi}) \right], \tag{1}$$

where stochastic gradients $\nabla f(\mathbf{x},\boldsymbol{\xi})$ can be evaluated on samples $\boldsymbol{\xi} \sim \mathcal{D}$ of the data distribution (this formulation covers both the online setting or training on a finite set of samples). Stochastic gradient descent methods, like SGD [67] or ADAM [41], are core components for training neural networks. In addition to their simplicity, and almost universal applicability, the solutions obtained by stochastic methods often generalize remarkably well [see e.g. 40].

In the field of convex optimization, the convergence of SGD is very well understood [10, 25]. The convergence proofs all exploit the property that stochastic gradients are approximations of the true gradient, $\mathbb{E}_{\boldsymbol{\xi}}\, \nabla f(\mathbf{x},\boldsymbol{\xi}) = \nabla f(\mathbf{x})$ and that SGD updates can be viewed as gradient descent steps perturbed by noise. From this point of view, stochastic noise is an undesirable influence which is not conducive to optimization and must therefore be tamed: For instance by averaging techniques [7, 69], decreasing stepsizes [43] or variance reduction [38, 50, 71, 77].

In stark contrast, stochastic noise has been observed to have beneficial effects in non-convex optimization: For instance, it has been proven that stochastic noise can allow SGD to escape saddle points [15, 24, 34], and under certain conditions noise allows SGD to escape local minima [30, 42]. While many important insights have been developed in such past works, none of these prove global convergence results on non-convex functions. This is because finding a global solution on smooth optimization problems is NP hard in general [59]. We can break this complexity barrier by

considering a new function class that can more closely model the difficulty of non-convex problems encountered in practice.

In this work, we characterize a new class of non-convex functions for which stochastic gradient methods can provably escape local minima. In particular, we characterize non-convex functions on which stochastic methods converge *linearly* to a global solution (in contrast, only sublinear convergence rates to local minima are known on general non-convex functions, 23, 47).[1] The main difference to prior work is that we assume that the objective function f has a hidden structure, namely that f is the composition of two components $g, h\colon \mathbb{R}^d \to \mathbb{R}$:

$$f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x})\,. \tag{2}$$

We study algorithms can only query $\nabla f(\mathbf{x}, \boldsymbol{\xi})$ (such as SGD) and *do not have* access to g or h separately. This structural assumption allows us to derive much finer convergence guarantees than are possible with the standard black box model. For instance under the assumption that g satisfies the Polyak-Łojasiewicz (PŁ) condition (we consider other cases too) and that the perturbations induced by h are not too strong relative to g, we show that the SGD trajectory follows the gradient flow of g and converges linearly to a neighborhood of the global solution.

Such hidden structures appear in many common ML optimization problems. As an example, consider the training of a classifier in the presence of random label noise. A common solution approach is to modify the surrogate loss function to attain unbiased estimators—however this new optimization target might not be convex, even when starting from a convex loss function (such as least square regression). Natarajan et al. [53, Theorem 6] prove that this non-convex optimization target f is uniformly close to a convex function g, i.e. h is bounded. The function classes we consider contains this class of problems, yet we also cover more general cases where h is not uniformly bounded.

Contributions. Our contributions can be summarized as:

- We introduce a new class of structured non-convex functions. By studying convergence on this function class, we can circumvent the lower complexity bounds that constrain the SGD analyses on general non-convex smooth functions [6] and we are able to derive improved complexity estimates for *perturbed SGD* methods—a class of algorithms that perturb iterates by stochastic perturbations and that contains SGD as a special case.
- We characterize settings where perturbed SGD methods converge linearly to a neighborhood of the global solution, while traditional analyses can only show sublinear convergence to local minma or stationary points (which can be arbitrary far from the global minima).
- Utilizing the insights developed in [42], we are able to link our convergence results to the behavior of SGD.

2. Perturbed SGD

Our main goal is to study the convergence of SGD on problem (1). The SGD algorithm is defined as

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)\,, \tag{SGD}$$

for a constant stepsize γ and a uniform stochastic sample $\boldsymbol{\xi}_t \sim \mathcal{D}$. This update can equivalently be written as

1. Concretely, $\mathcal{O}(1/\epsilon^{3.5})$ complexity to find an $\sqrt{\epsilon}$-approximate local minima with $\|\nabla f(\mathbf{x})\| \leq \epsilon$ [23, 47].

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

by defining $\mathbf{w}_t := \nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)$. Let $\mathbf{w}_t \sim \mathcal{W}(\mathbf{x}_t)$, where $\mathcal{W}(\mathbf{x}_t)$ denotes the distribution of $\mathbf{w}_t$, which can depend on the iterate $\mathbf{x}_t$.

Standard approach. Standard analyses of SGD on non-convex L-smooth functions typically derive an upper bound on the *expected* one step progress [e.g. Thm. 4.8 in 10]. This gives

$$\mathbb{E}\, f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \gamma \left\|\nabla f(\mathbf{x}_t)\right\|^2 + (\gamma^2 L/2)\mathbb{V}\mathrm{ar}(\mathbf{w}_t) \,.$$

However, following this methodology, stochastic updates can always only guarantee a *smaller* expected one step progress than the gradient method, as the variance is always positive.

Our approach. To circumvent the aforementioned limitation, we adopt two key changes. First, by utilizing the structure (2) we study the one step progress on g and secondly, we formulate the algorithm slightly differently. Concretely, we study **perturbed SGD** (Algorithm 1) that we formally define as

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t) \,, \tag{perturbed SGD}$$

for a random perturbation $\mathbf{u}_t \sim \mathcal{U}(\mathbf{x}_t)$. For this method, the expected one step progress can be estimated as,

$$\mathbb{E}\, g(\mathbf{x}_{t+1}) \leq g(\mathbf{x}_t) - \gamma \underbrace{\nabla g(\mathbf{x}_t)^\intercal \, \mathbb{E}_{\mathbf{u}_t, \boldsymbol{\xi}_t}[\nabla f(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t)]}_{①} + \underbrace{\frac{\gamma^2 L}{2} \mathbb{V}\mathrm{ar}_{\mathbf{u}_t, \boldsymbol{\xi}_t}(\nabla f(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t))}_{②} \tag{3}$$

The above formulation allows us to obtain larger progress than standard analysis, by the virtue of considering g and by using an appropriate smoothing distribution $\mathcal{U}$. To establish convergence, we will impose appropriate conditions on terms ① and ② in (3), which forms the basis for our Assumptions in Section 3.2.

It is easy to see that perturbed SGD comprises SGD, for instance when $\mathbf{u}_t \equiv 0$ a.s. However, there are more possibilities to trade-off the randomness in $\boldsymbol{\xi}_t$ and $\mathbf{u}_t$. In Appendix 5 below we derive more general connections between perturbed SGD and vanilla SGD.

To summarize, we introduce perturbed SGD with the purpose to study the impact of smoothing $\mathbf{u} \sim \mathcal{U}$ and stochastic gradient noise $\boldsymbol{\xi} \sim \mathcal{D}$ separately. Perturbed SGD is illustrated in Algorithm 1 and implements a stochastic smoothing oracle by only accessing stochastic gradients of f. For simplicity, we assume constant step length γ.

Algorithm 1 Perturbed SGD

Require: $\gamma, f(\mathbf{x}), T, \mathcal{U}(\mathbf{x}), \mathbf{x}_0$
  for $t = 0$ to $T-1$ **do**
    sample $\mathbf{u}_t \sim \mathcal{U}(\mathbf{x}_t)$ ▷ smoothing distribution
    sample $\boldsymbol{\xi}_t \sim \mathcal{D}$ ▷ (mini-batch) data sample
    $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t)$ ▷ SGD update (or ADAM/momentum)
  end for

3. Setting and Assumptions

We will now introduce the main assumption on the objective function f with structure (2) and give an illustrative example.

3.1. Smoothing

To formalize the notion of perturbations (i.e. the $\mathbf{u}_t$'s in Algorithm 1), we utilize the framework of smoothing [21]. Convolution-based smoothing of a function $f\colon \mathbb{R}^d \to \mathbb{R}$ is defined as[2]

$$f_{\mathcal{U}}(\mathbf{x}) := E_{\mathbf{u}\sim\mathcal{U}} f(\mathbf{x}-\mathbf{u})\,, \qquad \forall \mathbf{x}\in\mathbb{R}^d\,, \tag{4}$$

for a probability distribution $\mathcal{U}$ (sometimes we will allow $\mathcal{U}(\mathbf{x})$ to depend on $\mathbf{x}$).

Smoothing is a linear operator $(g+h)_{\mathcal{U}} = g_{\mathcal{U}} + h_{\mathcal{U}}$ and when f is convex, then $f_{\mathcal{U}}$ is convex as well.

The smoothing (4) cannot be computed exactly without having access to f, but one can resort to a stochastic approximation in practice. For a given f, we can query stochastic gradients of $\nabla f_{\mathcal{U}}$ by sampling $\mathbf{u}\sim\mathcal{U}$ and evaluating $\nabla f(\mathbf{x}-\mathbf{u})$. Many works that analyze smoothing need to formulate concrete assumptions on the smoothing distribution $\mathcal{U}$, for instance that variance $\mathbb{E}_{\mathbf{u}\sim\mathcal{U}(\mathbf{x})}\left\|\mathbf{u}\right\|^2 \leq \zeta^2$ is bounded by a parameter $\zeta^2 > 0$. This is for instance satisfied for for smoothing distributions with bounded support [see 21] or subgaussian noise, in particular for the normalized Gaussian kernel $\mathbf{u}\sim\mathcal{N}(\mathbf{0},\zeta^2/d\,\mathbf{I}_d)$. In our case, we do not need to formulate such an assumption on $\mathcal{U}$ directly, instead we formulate a new assumption that jointly governs and smoothing and stochastic noise in the next section.

3.2. Main Assumptions

As mentioned earlier, these assumptions seek to improve the one step progress for perturbed SGD (Algorithm 1) by exploiting the key terms of $\mathbf{u}$, and in (3)—in Assumptions 1 and 2 respectively.

We now list the main assumptions for the paper.

Assumption 1 (Stochastic noise) *The stochastic noise is unbiased, $\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D}} f(\mathbf{x},\boldsymbol{\xi}) = f(\mathbf{x})$, the smoothing distribution is zero-mean and $\mathbb{E}_{\mathbf{u}\sim\mathcal{U}(\mathbf{x})}[\mathbf{u}] = 0$, there exists parameters $\sigma'^2 \geq 0$, $M' \geq 0$, such that after smoothing with $\mathcal{U}(\mathbf{x})$, $\forall \mathbf{x}\in\mathbb{R}^d$:*

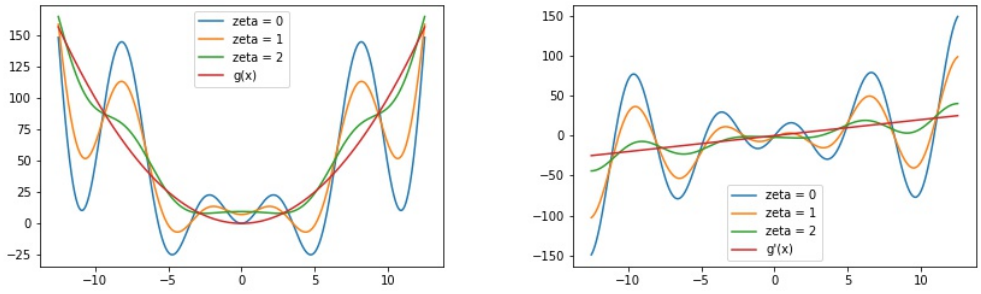
$$\mathbb{E}_{\mathbf{u},\boldsymbol{\xi}} \left\|\nabla f(\mathbf{x}-\mathbf{u},\boldsymbol{\xi}) - \nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2 \leq \sigma'^2 + M' \left\|\nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2 . \tag{5}$$

Note that $\mathbb{E}_{\mathbf{u},\boldsymbol{\xi}} \nabla f(\mathbf{x}-\mathbf{u},\boldsymbol{\xi}) = \nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})$. Therefore (5) allows to bound the variance term ② in (3). This extends the standard noise assumption in SGD settings [10, 69] which are of the form $\sigma^2 + M\left\|\nabla f(\mathbf{x})\right\|^2$ (we recover this assumption when $\mathbf{u}\equiv 0$, a.s.). While in non-convex settings this prior assumption is could be restrictive (as $\left\|\nabla f(\mathbf{x})\right\|^2$ is small for stationary points, enforcing large σ'), in contrast, $\|\nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x}))\|^2$ will still be positive large at saddles or sharp local minima, and thus in general σ' in (5) can be chosen much smaller.

Additionally, if the stochastic and smoothing noise are bounded and independent, we can recover the above assumption. We discuss this in detail in Appendix C.

We shift our attention on how to control the term ① in (3). Through the next assumption, we neatly tie this to the structure of the objective function in (2).

2. If $\mathcal{U}$ is symmetric, this is equivalent to the more standard definition $E_{\mathcal{U}}[f(\mathbf{x}+\mathbf{u})]$.



(a) $f_{\mathcal{U}}(x)$ and $g(x) = x^2$ (b) $\nabla f_{\mathcal{U}}(x)$ and $\nabla g_{\mathcal{U}}(x)$

Figure 1: Illustration of the effect of smoothing $f(x) = x^2 + 10x\sin(x)$ (blue) with the Gaussian kernel $\mathcal{N}(0, \zeta^2)$ for different $\zeta \in \{0, 1, 2\}$. $f_{\mathcal{U}}(\mathbf{x})$ does not become convex even for arbitrarily large $\zeta^2 > 0$.

Assumption 2 (Structural properties of g and h) *The objective function $f\colon \mathbb{R}^d \to \mathbb{R}$ can be written in the form (2), with g being L_g-smooth, and there exists parameters $0 \leq m < 1$ and $\Delta \geq 0$, such that, $\forall \mathbf{x} \in \mathbb{R}^d$:*

$$\left\|\nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x}) - \nabla g(\mathbf{x})\right\|^2 \leq \Delta + m\left\|\nabla g(\mathbf{x})\right\|^2 . \tag{6}$$

While this function does not explicitly clarify the role of h, to illustrate we can split the term on LHS as $\nabla h_{\mathcal{U}(\mathbf{x})}(\mathbf{x}) + (\nabla g_{\mathcal{U}(\mathbf{x})}(\mathbf{x}) - \nabla g(\mathbf{x}))$. The difference term $(g_{\mathcal{U}(\mathbf{x})}(\mathbf{x}) - \nabla g(\mathbf{x}))$ can be bounded if $\mathcal{U}(\mathbf{x})$ has bounded variance and g is smooth. The purpose of this assumption then becomes controlling $\nabla h_{\mathcal{U}(\mathbf{x})}(\mathbf{x})$, which essentially is the non-convex perturbation in f. Note that this assumption allows possibly unbounded h, however after smoothing, $\nabla h_{\mathcal{U}(\mathbf{x})}(\mathbf{x})$ must be dominated by $\nabla g(\mathbf{x})$. This assumption is an extension of biased gradient oracles of Ajalloeian and Stich [2].

3.3. Illustrative Example

We now provide an illustrative example which satisfies our assumptions while displaying a high degree of non-convexity. Consider the following 1-dimensional function,

$$f(x) = x^2 + ax\sin(bx)\,, \tag{7}$$

for parameters $a, b > 0$. We can chose $g(x) = x^2$ as the convex part, while $h(x) = ax\sin(bx)$ denotes the possibly unbounded non-convex perturbation. For $ab \geq 2$, this function can have infinitely many local minima, arbitrarily far away from its global minima.

Even after smoothing with a Gaussian distribution $\mathcal{N}(0, \zeta^2)$, the non-convex perturbations do not disappear, and it cannot be convex for any ζ (for more details see Appendix F.3). However, these perturbations become smaller with respect to g for larger ζ, as shown in Fig. 1. This (provably) allows the function to satisfy Assumption 2 for some m and Δ.

4. Convergence Analysis

The convergence analysis in this sections partially follows the biased gradient framework [2]. We provide convergence results when g is PŁ, and defer the proofs and additional extensions to Appendix D.

4.1. Convergence under PŁ Conditions

Theorem 1 *Let f satisfy Assumptions 1 and 2, and assume g to be μ_g-PŁ. Then there exists a stepsize γ such that for any $\epsilon > 0$,*

$$T = \tilde{\mathcal{O}}\left((M'+1)\log\frac{1}{\epsilon} + \frac{\sigma'^2}{\epsilon(1-m)\mu + \Delta} \right) \frac{\kappa}{1-m}$$

iterations are sufficient to obtain $\mathcal{G}_T = \mathcal{O}(\epsilon + \frac{\Delta}{\mu(1-m)})$, where $\kappa := \frac{L_g}{\mu}$, $\mathcal{G}_t = \mathbb{E}[g(x_t)] - \min_{\mathbf{x}\in\mathbb{R}^d} g(\mathbf{x})$ and $\tilde{\mathcal{O}}$ hides only log terms.

If $\sigma'^2 = 0$ then this theorem shows linear convergence in $\mathcal{O}\big(\frac{\kappa}{1-m}\log\frac{1}{\epsilon}\big)$ steps to a neighborhood of the global solution. When σ'^2 is large, the rate is dominated by the second term, $\mathcal{O}\big(\frac{\sigma'^2}{\epsilon(1-m)^2}\big)$. This matches the $\mathcal{O}\big(\frac{\sigma^2}{\epsilon}\big)$ convergence rate of vanilla SGD on PŁ functions. However, note that in our case f does not need to be PŁ to enjoy these convergence guarantees.

4.2. Insights

In this section we have derived convergence results under our novel structural assumption (2) for Perturbed SGD (Alg. 1). Our results depict the impact of the smoothing $\mathcal{U}$ and the stochastic noise $\mathcal{D}$, and when $\mathcal{U} \equiv 0$ a.s. (no smoothing), we recover the known convergence results for SGD.

All convergence results depend on the joint effect of smoothing and stochastic noise, $\sigma'^2 = \sigma^2 + L^2\zeta^2$ (see Remark 4). This means, that any smoothing with $\zeta^2 \leq \frac{1}{L^2}\sigma^2$ does *not worsen* the convergence estimates one would get by analyzing vanilla SGD alone. Moreover, smoothing allows convergence to the minima of g, and to avoid local minima of f at a linear rate. Note that this is much faster and simpler than existing methods [35, 78] which can only converge to approximate local minima. In particular, smoothing f with the scaled gradient noise $\frac{1}{L}\mathcal{D}$ we get for free a method that enjoys much more favorable convergence guarantees than SGD [24]. But is it even necessary to implement Pertubed SGD, or does vanilla SGD suffice? We argue in the next section that this might indeed be the case.

5. Connection to SGD

We now explain how the analysis from the previous section is connected to the standard SGD algorithm. (that does not implement the smoothing perturbation $\mathbf{u} \sim \mathcal{U}(\mathbf{x})$ explicitly).

This follows directly from insights in [42]. Let $\mathbf{x}_t$ be the SGD iterates as defined in (SGD), with noise $\mathbf{w}_t \sim \mathcal{W}(\mathbf{x}_t)$, where $\mathcal{W}(\mathbf{x}_t)$ is the gradient noise distribution. Kleinberg et al. [42] propose to study the alternate sequence $\mathbf{y}_t$ defined as

$$\mathbf{y}_{t+1} = \mathbf{x}_t - \gamma\nabla f(\mathbf{x}_t)\,.$$

Let $\mathbf{z}_t$ define the iterates of Algorithm 1 as defined in (perturbed SGD), with only smoothing, $\mathbf{u}_t$, and no gradient noise, ξ_t. Let $\mathbf{u}_t \sim \mathcal{U}(\mathbf{x}_t)$, where $\mathcal{U}(\mathbf{x}_t)$ is the smoothing distribution.

Lemma 2 (Equality in Expectation, [adopted from 42]) *For $\mathbf{x}_t, \mathbf{y}_t$ and $\mathbf{z}_t$ defined as above, if $\mathbf{z}_0 = \mathbf{y}_0$ and $\mathcal{U}(\mathbf{x}_t) = \gamma\mathcal{W}(\mathbf{x}_t)$ for all $t \geq 0$, then*

$$\mathbb{E}[\mathbf{z}_t] = \mathbb{E}[\mathbf{y}_t]\,.$$

We refer the reader to [42] for the proof.

This Lemma establishes the intuition, that SGD is performing approximately gradient descent on a smooth version of f. Note that we establish only a weak equivalence in expectation. However, even this weak equivalence is sufficient to use our main results from Theorem 10 for SGD analysis. We refer the reader to Appendix E for additional analysis and experiments.

6. Numerical Illustrations

In this section we provide numerical illustrations to demonstrate that Perturbed SGD is able to escape local minima in contrast to gradient descent (GD). We compare the performance of our Algorithm 1 with GD on our toy example $f(x) = x^2 + 10x\sin(x)$ with $\mathcal{U} = \mathcal{N}(0, \zeta^2)$ smoothing. The results (averaged over 1000 independent runs) are illustrate in Figure 2. For this function there are two global minima located near ± 4.7. We observe that while GD gets stuck at poor local minima most of the time, our algorithm is able to escape these local minima. Further, increasing smoothing by increasing ζ helps in escaping local minima, and allows convergence to the minima of $g(x) = x^2$, which is close to the global minima of f.

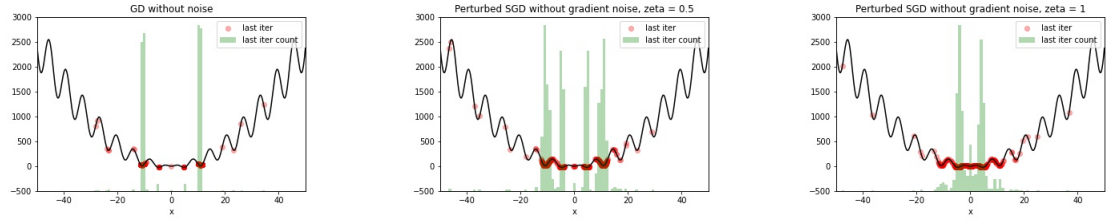


Figure 2: Distribution of the last iterates from GD and Algorithm 1 on the toy example (7). We run Perturbed SGD and SGD for 1000 random initializations between $[-400, 400]$ and for $T = 100$ iterations. Here $\sigma = 0, M = 0$ and $\zeta \in \{0.5, 1\}$ with Gaussian smoothing. We select the step size form a grid search over a grid of 4 step sizes from $[10^{-5}, 1]$, which are exponentially separated. For better visualization, we plot the locations and histogram of the last iterate from these runs, restricted to the interval $[-50, 50]$.

7. Discussion and Outlook

There is a growing discrepancy between the theoretical complexity results for SGD and its much better good empirical performance, which is often observed in practice. This is because the theoretical modeling of the functional class—typically smooth non-convex losses—does not reflect well the practical challenges. To break this complexity barrier, we propose a new class of functions that allow us to justify why stochastic methods (SGD or Perturbed SGD) can provably avoid local minima and can converge (at a linear rate) to a global optimal solution. However, it remains an interesting open question to prove that our structural assumption holds for real DL tasks.

We believe that it possible to develop more advanced versions of Perturbed SGD, such as counterparts of momentum SGD, ADAM, or variance reduced methods that are specifically designed for (hidden) composite functions. Another direction could aim at proving convergence results for SGD on targets with hidden structure in a more direct way, without the detour via Perturbed SGD. Research in this direction may for example shed new light on why variance reduced methods struggle on non-convex tasks [16] and can lead to more efficient training methods for neural networks in general.

References

[1] Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 1195–1199, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450345286. doi: 10.1145/3055399.3055464. URL https://doi.org/10.1145/3055399.3055464.

[2] Ahmad Ajalloeian and Sebastian U. Stich. Analysis of SGD with Biased Gradient Estimators. *arXiv:2008.00051 [cs, math, stat]*, July 2020. URL http://arxiv.org/abs/2008.00051. arXiv: 2008.00051.

[3] Zeyuan Allen-Zhu. How to make the gradients small stochastically: Even faster convex and nonconvex sgd. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/996a7fa078cc36c46d02f9af3bef918b-Paper.pdf.

[4] Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than sgd. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/79a49b3e3762632813f9e35f4ba53d6c-Paper.pdf.

[5] Zeyuan Allen-Zhu and Yuanzhi Li. Neon2: Finding local minima via first-order oracles. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/d4b2aeb2453bdadaa45cbe9882ffefcf-Paper.pdf.

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

[7] Francis R. Bach and Eric Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *NIPS - Advances in Neural Information Processing Systems 24*, pages 451–459. Curran Associates, Inc., 2011. URL http://papers.nips.cc/paper/4316-non-asymptotic-analysis-of-stochastic-approximation-algorithms-for-machine-learning.pdf.

[8] Alexandre Belloni, Tengyuan Liang, Hariharan Narayanan, and Alexander Rakhlin. Escaping the local minima via simulated annealing: Optimization of approximately convex functions. In *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 240–265, Paris, France, 03–06 Jul 2015. PMLR. URL https://proceedings.mlr.press/v40/Belloni15.html.

[9] Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT press Cambridge, 1987.

[10] L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018. URL https://doi.org/10.1137/16M1080173.

[11] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for non-convex optimization. *ArXiv*, abs/1611.00756, 2016.

[12] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. "Convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 654–663. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/carmon17a.html.

[13] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-SGD: Biasing gradient descent into wide valleys. In *International Conference on Learning Representations*, 2017.

[14] Li Chen, Shuisheng Zhou, and Zhuan Zhang. Stochastic Variance Reduction Gradient for a Non-convex Problem Using Graduated Optimization. July 2017. URL https://arxiv.org/abs/1707.02727v1.

[15] Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. Escaping saddles with stochastic gradients. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1155–1164. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/daneshmand18a.html.

[16] Aaron Defazio and Leon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/84d2004bf28a2095230e8e14993d398d-Paper.pdf.

[17] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*, 2017.

[18] Alejandro Domínguez. A history of the convolution operation. *IEEE Pulse*, 2015.

[19] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[20] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *Conference on Learning Theory*, 2010.

[21] John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. Randomized Smoothing for Stochastic Optimization. *arXiv:1103.4296 [math, stat]*, April 2012. URL http://arxiv.org/abs/1103.4296. arXiv: 1103.4296.

[22] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/1543843a4723ed2ab08e18053ae6dc5b-Paper.pdf.

[23] Cong Fang, Zhouchen Lin, and Tong Zhang. Sharp analysis for nonconvex sgd escaping from saddle points. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1192–1234. PMLR, 25–28 Jun 2019. URL https://proceedings.mlr.press/v99/fang19a.html.

[24] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points — online stochastic gradient for tensor decomposition. In *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 797–842. PMLR, 2015. URL http://proceedings.mlr.press/v40/Ge15.html.

[25] Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. doi: 10.1137/120880811. URL https://doi.org/10.1137/120880811.

[26] Saeed Ghadimi and Guanghui Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Math. Program.*, 156(1):59–99, March 2016. ISSN 1436-4646. doi: 10.1007/s10107-015-0871-8. URL https://doi.org/10.1007/s10107-015-0871-8.

[27] Nikolaus Hansen and Andreas Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 06 2001. ISSN 1063-6560. doi: 10.1162/106365601750190398. URL https://doi.org/10.1162/106365601750190398.

[28] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1225–1234. PMLR, 2016. URL http://proceedings.mlr.press/v48/hardt16.html.

[29] Kosuke Haruki, Taiji Suzuki, Yohei Hamakawa, Takeshi Toda, Ryuji Sakai, Masahiro Ozawa, and Mitsuhiro Kimura. Gradient noise convolution (GNC): Smoothing loss function for distributed large-batch SGD. *arXiv preprint arXiv:1906.10822*, 2019.

[30] Elad Hazan, Kfir Yehuda Levy, and Shai Shalev-Shwartz. On graduated optimization for stochastic non-convex problems. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1833–1841. PMLR, 2016. URL http://proceedings.mlr.press/v48/hazanb16.html.

[31] Oliver Hinder, Aaron Sidford, and Nimit Sohoni. Near-optimal methods for minimizing star-convex functions and beyond. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 1894–1938. PMLR, 09–12 Jul 2020. URL http://proceedings.mlr.press/v125/hinder20a.html.

[32] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 01 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.1.1. URL https://doi.org/10.1162/neco.1997.9.1.1.

[33] Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

[34] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1724–1732. PMLR, 06–11 Aug 2017. URL http://proceedings.mlr.press/v70/jin17a.html.

[35] Chi Jin, Lydia T. Liu, Rong Ge, and Michael I Jordan. On the local minima of the empirical risk. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/da4902cb0bc38210839714ebdcf0efc3-Paper.pdf.

[36] Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1042–1085. PMLR, 06–09 Jul 2018. URL https://proceedings.mlr.press/v75/jin18a.html.

[37] Jikai Jin. On The Convergence of First Order Methods for Quasar-Convex Optimization. *arXiv:2010.04937 [cs, math, stat]*, October 2020. URL http://arxiv.org/abs/2010.04937. arXiv: 2010.04937.

[38] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, volume 26. Curran

Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/ac1dd209cbcc5e5d1c6e28598e8cbbe8-Paper.pdf.

[39] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak-\L{}ojasiewicz Condition. *arXiv:1608.04636 [cs, math, stat]*, September 2016. URL http://arxiv.org/abs/1608.04636. arXiv: 1608.04636.

[40] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017.

[41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[42] Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does SGD escape local minima? In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2698–2707. PMLR, 2018. URL http://proceedings.mlr.press/v80/kleinberg18a.html.

[43] Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an O(1/t) convergence rate for the projected stochastic subgradient method. *arXiv:1212.2002 [cs, math, stat]*, December 2012. URL http://arxiv.org/abs/1212.2002. arXiv: 1212.2002.

[44] Jean-Baptiste le Rond d'Alembert. *Recherches sur différents points importants du systæme du monde*. 1754.

[45] Jasper C. H. Lee and Paul Valiant. Optimizing Star-Convex Functions. *arXiv:1511.04466 [cs]*, May 2016. URL http://arxiv.org/abs/1511.04466. arXiv: 1511.04466.

[46] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/81ca0262c82e712e50c580c032d99b60-Paper.pdf.

[47] Zhize Li. SSRGD: Simple stochastic recursive gradient descent for escaping saddle points. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[48] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. Extrapolation for large-batch training in deep learning. *arXiv preprint arXiv:2006.05720*, 2020.

[49] S Łojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. In *Les Èquations aux Dérivées Partielles*, pages 87–89, Paris, 1963. Èditions du Centre National de la Recherche Scientifique.

[50] Mehrdad Mahdavi, Lijun Zhang, and Rong Jin. Mixed optimization for smooth functions. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/f73b76ce8949fe29bf2a537cfa420e8f-Paper.pdf.

[51] J. Matyas. Random optimization. *Automation and Remote Control*, 26:246–253, 1965.

[52] Wenlong Mou, Liwei Wang, Xiyu Zhai, and Kai Zheng. Generalization bounds of sgld for non-convex learning: Two theoretical viewpoints. In *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 605–638. PMLR, 06–09 Jul 2018. URL http://proceedings.mlr.press/v75/mou18a.html.

[53] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[54] I. Necoara, Yu Nesterov, and F. Glineur. Linear convergence of first order methods for non-strongly convex optimization. *arXiv:1504.06298 [math]*, August 2016. URL http://arxiv.org/abs/1504.06298. arXiv: 1504.06298.

[55] Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

[56] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.

[57] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140 (1), 2013.

[58] Yu. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, May 2005. ISSN 1436-4646. doi: 10.1007/s10107-004-0552-5. URL https://doi.org/10.1007/s10107-004-0552-5.

[59] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, 1 edition, 2014.

[60] Yurii Nesterov and B.T. Polyak. Cubic regularization of Newton method and its global performance. *Math. Program.*, 108(1):177–205, August 2006. ISSN 1436-4646. doi: 10.1007/s10107-006-0706-8. URL https://doi.org/10.1007/s10107-006-0706-8.

[61] Yurii Nesterov and Vladimir Spokoiny. Random Gradient-Free Minimization of Convex Functions. *Found Comput Math*, 17(2):527–566, April 2017. ISSN 1615-3383. doi: 10.1007/s10208-015-9296-2. URL https://doi.org/10.1007/s10208-015-9296-2.

[62] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[63] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*, 2017.

[64] B. T. Polyak. Gradient methods for minimizing functionals. *Zh. Vychisl. Mat. Mat. Fiz.*, pages 643–653, 1963.

[65] L. A. Rastrigin. The convergence of the random search method in theextremal control of a many-parameter system. *Automation and Remote Control*, 24:1337–1342, 1963.

[66] Sashank Reddi, Manzil Zaheer, Suvrit Sra, Barnabas Poczos, Francis Bach, Ruslan Salakhutdinov, and Alex Smola. A generic approach for escaping saddle points. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1233–1242. PMLR, 09–11 Apr 2018. URL https://proceedings.mlr.press/v84/reddi18a.html.

[67] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407, September 1951.

[68] M. Schumer and K. Steiglitz. Adaptive step size random search. *IEEE Transactions on Automatic Control*, 13(3):270–276, 1968.

[69] Sebastian U. Stich. Unified Optimal Analysis of the (Stochastic) Gradient Method. *arXiv:1907.04232 [cs, math, stat]*, December 2019. URL http://arxiv.org/abs/1907.04232. arXiv: 1907.04232.

[70] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I Jordan. Stochastic cubic regularization for fast nonconvex optimization. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/db1915052d15f7815c8b88e879465a1e-Paper.pdf.

[71] Chong Wang, Xi Chen, Alexander J Smola, and Eric P Xing. Variance reduction for stochastic gradient optimization. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/9766527f2b5d3e95d4a733fcfb77bd7e-Paper.pdf.

[72] Wei Wen, Yandan Wang, Feng Yan, Cong Xu, Chunpeng Wu, Yiran Chen, and Hai Li. Smoothout: Smoothing out sharp minima to improve generalization in deep learning. *arXiv preprint arXiv:1805.07898*, 2018.

[73] Peng Xu, Fred Roosta, and Michael W. Mahoney. Newton-type methods for non-convex optimization under inexact Hessian information. *Math. Program.*, 184(1):35–70, November 2020. ISSN 1436-4646. doi: 10.1007/s10107-019-01405-z. URL https://doi.org/10.1007/s10107-019-01405-z.

[74] Yi Xu, Rong Jin, and Tianbao Yang. First-order stochastic algorithms for escaping from saddle points in almost linear time. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/217e342fc01668b10cb1188d40d3370e-Paper.pdf.

[75] Yi Xu, Rong Jin, and Tianbao Yang. Neon+: Accelerated gradient methods for extracting negative curvature for non-convex optimization, 2018.

[76] Yaodong Yu, Pan Xu, and Quanquan Gu. Third-order smoothness helps: Faster stochastic optimization algorithms for finding local minima. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/fea9c11c4ad9a395a636ed944a28b51a-Paper.pdf.

[77] Lijun Zhang, Mehrdad Mahdavi, and Rong Jin. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/37f0e884fbad9667e38940169d0a3c95-Paper.pdf.

[78] Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient langevin dynamics. In *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1980–2022. PMLR, 07–10 Jul 2017. URL https://proceedings.mlr.press/v65/zhang17b.html.

[79] Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic Variance-Reduced Cubic Regularization Methods. *Journal of Machine Learning Research*, 20(134):1–47, 2019. URL http://jmlr.org/papers/v20/19-055.html.

[80] Yi Zhou, Junjie Yang, Huishuai Zhang, Yingbin Liang, and Vahid Tarokh. SGD Converges to Global Minimum in Deep Learning via Star-convex Path. *arXiv:1901.00451 [cs, stat]*, January 2019. URL http://arxiv.org/abs/1901.00451. arXiv: 1901.00451.

Contents of the Appendix

The code for all the experiments and plots in this paper has been uploaded to the following repository.

https://github.com/harshv834/smooth-sgd-code

Appendix A. Additional Discussion of Related Work

In this section, we provide a more comprehensive overview of the literature for our problem. We have two major directions of research with which we can compare our work. The major direction deals with convergence to stationary points of the a general non-convex function. We compare with existing works in this direction in Table 1. The second direction deals with approximately convex functions, where the convexity condition is gradually relaxed. We also cover existing literature on smoothing and optimization of composite functions. **Benefits of Injecting Noise:** It has been observed that the noise in the gradient can help SGD to escape saddle points [24] or achieve better generalization [28, 52]. This is often explained by arguing that SGD finds 'flat' minima with favorable generalization properties [32, 33, 40], though also 'sharp' minima can generalize well [17]. These advantageous properties of SGD decrease as the batch size is increased [40] or with variance reduction techniques [16]. Several authors proposed to artificially inject noise into the SGD process for improved generalization [13, 55, 63], in particular in the context of large batch training [29, 48, 72].

Approximate Minima in Non-Convex Functions: Despite their NP-hardness, several works have studied non-convex optimization problems. Standard analysis for smooth functions can guarantee convergence to a first order stationary point ($\|\nabla f(\mathbf{x})\| \leq \epsilon$) only [25, 26] at rate $\mathcal{O}(\epsilon^{-2})$. Recently, there has been much interest in second-order stationary points, where ϵ-SOSP is defined as $\|\nabla f(\mathbf{x})\| \leq \epsilon, \lambda_{\min}(\nabla^2 f(\mathbf{x})) \geq -\sqrt{\epsilon}$ [5, 24, 75]. If all saddle points are strict, then all ϵ-SOSP are approximate local minima [34]. Thus, convergence to ϵ-SOSP allows us to escape all saddle points. While SGD guarantees $\mathcal{O}(\epsilon^{-4})$ convergence to ϵ-SOSP, utilizing acceleration and second-order approximations improves it to $\mathcal{O}(\epsilon^{-3.5})$ [1, 11, 12, 35, 36]. Other methods, with same or slightly better rates, utilize efficient subroutines [3, 4], negative curvature of the loss [22, 74, 76], adaptive regularization [60, 70, 73] and variance reduction [46, 66, 79]. For Table 1, we will use the following definitions:

- First-order stationary point: A point $\mathbf{x} \in \mathbb{R}^d$ is called a ϵ-first-order stationary point if $\|\nabla f(\mathbf{x})\| \leq \epsilon$.
- Second-order stationary point: A point $\mathbf{x} \in \mathbb{R}^d$ is called a $(\epsilon, \sqrt{\rho\epsilon}))$-second-order stationary point if $\|\nabla f(\mathbf{x})\| \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(\mathbf{x})) \geq -\sqrt{\rho\epsilon}$, where ρ is the Lipschitz constant of the Hessian. Under certain conditions on the saddle points of the function [34], these points can be regarded as approximate local minima, and convergence to them implies escaping all saddle points.

Additionally, all methods converging to first-order stationary points require gradients to be Lipschitz, while for all methods converging to second-order stationary points, the function, its gradient and its Hessian should all be Lipschitz.

In contrast our methods, require only stochastic gradients, smoothness of g and Assumption 2, to allow linear convergence to a neighborhood of global minima. Moreover, the algorithms achieving best rates need more complicated algorithms than our Perturbed SGD.

Smoothing: Injecting artificial noise is classically also known as *smoothing* or *convolution* [18, 44] and has found countless applications in various domains and communities. In the context of optimization, smoothing has been used at least since the 1960s in [51, 65, 68]. While most proofs apply to the convex setting only [56, 61], smoothing is more prominently used in heuristic search procedures for non-convex problems [9, 27]. One of the outstanding features of the smoothing

Table 1: Comparison to related works on non-convex optimization

Output	Assumptions	Oracle	Method	Rate
First-order stationary point	Gradient Lipschitz	Gradient	[26]	$\mathcal{O}(\epsilon^{-2})$
			[12]	$\tilde{\mathcal{O}}(\epsilon^{-1.75})$
Second-order stationary point	Function, Gradient and Hessian Lipschitz	Hessian	[60]	$\mathcal{O}(\epsilon^{-1.5})$
		Hessian-vector product	[11]	$\tilde{\mathcal{O}}(\epsilon^{-2})$
			[1]	$\tilde{\mathcal{O}}(\epsilon^{-1.75})$
		Gradient	[34]	$\tilde{\mathcal{O}}(\epsilon^{-2})$
			[36]	$\tilde{\mathcal{O}}(\epsilon^{-1.75})$
		Stochastic Gradient	[78]	$poly(\epsilon^{-1})$
			[24]	$\tilde{\mathcal{O}}(\epsilon^{-4})$
			[23]	$\tilde{\mathcal{O}}(\epsilon^{-3.5})$
			[70]	
			[4]	
			[66]	
			[46]	$\tilde{\mathcal{O}}(\epsilon^{-3.33})$
			[5]	$\tilde{\mathcal{O}}(\epsilon^{-3.25})$
			[22]	$\tilde{\mathcal{O}}(\epsilon^{-3})$
Global Minima	Assumption 2, $\nabla g(\mathbf{x})$ Lipschitz g convex		**This paper**	$\mathcal{O}(\log(\epsilon^{-1}) + \epsilon^{-1})$

technique is that it allows to reduce the optimization complexity of non-smooth optimization problems [21, 58].

Compositional structure: Often in machine learning settings, an inherent structure $f = g + h$ is explicitly known, for instance when one term denotes a regularizer. In this case, optimization methods can be designed that exploit favorable properties of the regularizer (such as strong convexity) [20, 57]. However, this is different from our approach, as these algorithms need to have explicit knowlege of the regulariser. We, instead, use the structure (2) only as an analysis tool [opposed to e.g. 14], while the algorithm has only access to stochastic gradients of f.

Approximately convex functions: Another approach for analysis of non-convex functions investigates weaker forms of convexity. The most common formulations include PŁ functions [39, 49, 64], where all minima are global minima, star-convex functions [45, 80], which are convex about the minima and approximately convex functions, which differ from convex functions by a bounded constant [8, 35, 78]. These functions are analyzed using standard techniques used for convex function. also extend this notion. Necoara et al. [54] provide a survey of when this analysis can lead to linear convergence. The class of non-convex functions that we consider subsume most mild cases of non-convexity like PŁ, star-convexity or approximate convexity and can be extended to stronger ones like quasar-convexity [31, 37].

The key idea for these function classes is to relax the convexity condition to include non-convex functions. We will show that our settings allow us to encompass most function classes which are

approximately convex. First of all, our function class includes convex and PŁ functions [39], by setting $h(\mathbf{x}) = 0$. Another class of functions which are discussed extensively are those with bounded h. Belloni et al. [8] consider a class of functions where $h(\mathbf{x})$ is bounded. A similar notion is covered by Zhang et al. [78], where SGLD is used for derivative free optimization of functions, where the stochastic function oracle is bounded from the function value. This can also be encompassed by our function class as we only need the stochastic oracle to be unbiased. Another simpler function class covered by our settings, is that of bounded gradients for h. We discuss this in detail in Appendix D.5.2. There are other forms of non-convexity, like star-convexity [37], where the function is convex about only the global minima, and its generalization to quasar-convexity [31]. While our settings do not cover these classes, we can extend our analysis to include them, by considering that function g is quasar or star convex instead of strongly-convex, ensuring that all results still hold.

Non-convex smoothing: A theoretical connection between stochastic optimization and smoothing as been established in [42]. They study smoothing with distributions with bounded support (while we do not make this restriction) and prove convergence under the assumption the smooth $f_{\mathcal{U}}$ is star convex [31]. In [30] a graduated smoothing technique was analyzed under the assumption the smoothed function is strongly convex on a sufficiently large neighborhood of the optimal solution. Further, smoothing has been used in the context of derivative free optimization or in Langevin dynamics in non-convex regimes, most notably in [8, 35, 78], however these works do not show global linear convergence in stronger paradigms of non-convexity.

Apart from these works, we do a rigorous comparison to [30] and [42] in Appendix F, since they are closely related to our settings.

Appendix B. Notation

For the reader's convenience, we summarize here a few standard definitions [59]. We say that a function $f\colon \mathbb{R}^d \to \mathbb{R}$ is L-smooth if its gradient is L-Lipschitz continuous:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\| \,, \qquad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d. \tag{8}$$

A function $f\colon \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex for $\mu \geq 0$, if

$$\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq \mu \|\mathbf{x} - \mathbf{y}\|^2 \,, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$$

Sometimes relaxations of this condition are considered. A function $f\colon \mathbb{R}^d \to \mathbb{R}$ satisfies the Polyak-Łojasiewicz (μ-PŁ) condition with respect to $\mathbf{x}^\star$ if

$$2\mu(f(\mathbf{x}) - f(\mathbf{x}^\star)) \leq \|\nabla f(\mathbf{x})\|^2 \,, \qquad \forall \mathbf{x} \in \mathbb{R}^d \,. \tag{9}$$

If follows that $\mathbf{x}^\star = \arg\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$ is unique. We provide additional useful standard consequences of these inequalities in Appendix C.

Appendix C. Additional Technical Tools

We list here a few useful properties, sometimes used in the proofs. Further, we also provide missing proofs and additional analysis for Remark 4 and Lemma 2 in Section 5.

C.1. On Smooth and Convex Functions

We first provide additional definitions and formulations for smooth functions, which we will use later.

A function is μ-star-convex with respect to $\mathbf{x}^\star$ if

$$\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star \rangle \geq \mu \left\| \mathbf{x} - \mathbf{x}^\star \right\|^2 , \qquad \forall \mathbf{x} \in \mathbb{R}^d . \tag{10}$$

Strongly convex functions are both PŁ and star convex.

The smoothness assumption (8) is often equivalently written as

$$|f(\mathbf{y}) - f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle| \leq \frac{L}{2} \left\| \mathbf{y} - \mathbf{x} \right\|^2 , \qquad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d \tag{11}$$

Remark 3 *Note that if a function f is $L-$ smooth and has a minimizer $\mathbf{x}^\star \in \arg\min_{\mathbf{x}} f(\mathbf{x})$, then it holds satisfies*

$$\left\| \nabla f(\mathbf{x}) \right\|^2 \leq 2L(f(\mathbf{x}) - r(\mathbf{x}^\star)) \qquad \forall \mathbf{x} \in \mathbb{R}^d . \tag{12}$$

Proof Let $\mathbf{y} = \mathbf{x} - \frac{1}{L}\nabla f(\mathbf{x})$, then, substituting these $\mathbf{x}$ and $\mathbf{y}$ in above definition –

$$\left\| \nabla r(\mathbf{x}) \right\|^2 \leq 2L(r(\mathbf{x}) - r(\mathbf{x} - \frac{1}{L}\nabla f(\mathbf{x}))) .$$

Since $r(\mathbf{x} - \frac{1}{L}\nabla f(\mathbf{x})) \geq r(\mathbf{x}^\star)$, we can substitute this in the upper bound. ■

Strong convexity is often written as

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2} \left\| \mathbf{y} - \mathbf{x} \right\|^2 \qquad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d . \tag{13}$$

C.2. Discussion on Assumption 1

We obtain a provide additional details on obtaining Assumption 1 as a decomposition of the smoothing and stochastic noise.

Remark 4 *If the smoothing distribution, $\mathcal{U}(\mathbf{x})$ has variance bounded by $\zeta^2 + Z \left\| \nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x}) \right\|^2$, and the variance of stochastic gradients have variance bounded as $\sigma^2 + M \left\| \nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x}) \right\|^2$, for some $\sigma^2, \zeta^2, M, Z \geq 0$, then under independence of $\mathcal{U}$ and $\mathcal{D}$ and L-smoothness of f, we can chose the terms in Assumption 1 as $\sigma'^2 := \sigma^2 + 2(L\zeta)^2$ and $M' := M + 2(LZ)^2$.*

The above remark allows us to separate the contributions of smoothing noise and stochastic noise. Further, setting the terms of smoothing (ζ, Z) to 0, we recover the standard assumptions for SGD with unbounded variance.

To prove Remark 4, we first restate a more general version of the assumptions on the smoothing distribution $\mathcal{U}(\mathbf{x}_t)$ and noise distribution $\mathcal{D}$ (in the main text we assumed $Z = 0$ for simplicity).

Assumption 3 (Smoothing noise) *For given $f \colon \mathbb{R}^n \to \mathbb{R}$, the smoothing distribution $\mathcal{U}(\mathbf{x})$ is zero-mean ($\mathbb{E}_{\mathbf{u} \sim \mathcal{U}(\mathbf{x})} \mathbf{u} = \mathbf{0}$), can possibly depend on $\mathbf{x} \in \mathbb{R}^d$ and there exists constants ($\zeta^2 \geq 0, Z^2 \geq 0$) such that the variance can be bounded as*

$$\mathbb{E}_{\mathbf{u} \sim \mathcal{U}(\mathbf{x})} \left\| \mathbf{u} \right\|^2 \leq \zeta^2 + Z^2 \left\| \nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x}) \right\|^2 , \qquad \forall \mathbf{x} \in \mathbb{R}^d . \tag{14}$$

This Assumption is modeled similar to our Assumption 1. Further, setting $Z = 0$, we obtain a bound on the variance of the smoothing distribution, which is valid for subgaussian variables [21].

We can use the above assumption to obtain bounds on variance of the perturbed gradient.

Lemma 5 (Stochastic Approximation) *If f is L-smooth and Assumption 3, the variance is bounded as*

$$\mathbb{E}_{\mathbf{u}\sim\mathcal{U}(\mathbf{x})} \left\|\nabla f(\mathbf{x}-\mathbf{u}) - \nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2 \leq 2L^2\zeta^2 + 2L^2Z^2 \left\|\nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2, \qquad \forall \mathbf{x}\in\mathbb{R}^d. \quad (15)$$

Proof By Jensen's inequality and smoothness

$$\begin{aligned}\mathbb{E}_{\mathbf{u}} \left\|\nabla f(\mathbf{x}-\mathbf{u}) - \nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2 &= \mathbb{E}_{\mathbf{u}} \left\|\nabla f(\mathbf{x}-\mathbf{u}) - \mathbb{E}_{\mathbf{v}\sim\mathcal{U}} \nabla f_{(\mathbf{x}-\mathbf{v})}(\mathbf{x})\right\|^2 \\ &\leq \mathbb{E}_{\mathbf{u},\mathbf{v}} \left\|\nabla f(\mathbf{x}-\mathbf{u}) - \nabla f(\mathbf{x}-\mathbf{v})\right\|^2 \\ &\leq L^2\, \mathbb{E}_{\mathbf{u},\mathbf{v}} \left\|\mathbf{u}-\mathbf{v}\right\|^2 \leq 2L^2\zeta^2 + 2L^2Z^2 \left\|\nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2.\end{aligned}$$

■

Now, that we have defined all the terms for the smoothing distribution in Remark 4, we introduce a common assumption for the stochastic noise.

Assumption 4 *For given $f\colon \mathbb{R}^n \to \mathbb{R}$, the perturbed stochastic gradient can be expressed as*

$$\nabla f(\mathbf{x}-\mathbf{u}, \boldsymbol{\xi}) = \nabla f(\mathbf{x}-\mathbf{u}) + \mathbf{w} \quad (16)$$

where $\mathbf{w} \sim \mathcal{W}(\mathbf{x})$ and $\mathcal{W}(\mathbf{x})$ denotes the zero-mean noise distribution, and there exist constants ($\sigma^2 > 0, M > 0$), such its variance can be bounded as

$$\mathbb{E}_{\mathbf{w}\sim\mathcal{W}(\mathbf{x})} \left\|\mathbf{w}\right\|^2 \leq \sigma^2 + M \left\|\nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2, \qquad \forall \mathbf{x}\in\mathbb{R}^d. \quad (17)$$

Now, we are ready to present the complete the proof for Remark 4. We first present its extended version as a Lemma below and then prove it.

Lemma 6 (Extension of Remark 4) *If f is L-smooth, Assumptions 3 and 4 are satisfied, and the noise ($\mathcal{W}(\mathbf{x})$) and smoothing distributions ($\mathcal{U}(\mathbf{x})$) are independent for $\mathbf{x}$, then,*

$$\mathbb{E}_{\mathbf{u},\boldsymbol{\xi}} \left\|\nabla f(\mathbf{x}-\mathbf{u}, \boldsymbol{\xi}) - \nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2 \leq (\sigma^2 + 2(L\zeta)^2) + (M + 2(LZ)^2) \left\|\nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2. \quad (18)$$

Note that this is identical to Assumption 1, with $\sigma'^2 = \sigma^2 + 2(L\zeta)^2$ and $M' = M + 2(LZ)^2$.

Proof Consider the term on the left hand side,

$$\begin{aligned}\mathbb{E}_{\mathbf{u},\boldsymbol{\xi}} \left\|\nabla f(\mathbf{x}-\mathbf{u}, \boldsymbol{\xi}) - \nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2 &= \mathbb{E}_{\mathbf{w},\boldsymbol{\xi}} \left\|\nabla f(\mathbf{x}-\mathbf{u}) + \mathbf{w} - \nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2 \\ &= \mathbb{E}_{\boldsymbol{\xi}} \left\|\nabla f(\mathbf{x}-\mathbf{u}) - \nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2 + \mathbb{E}_{\mathbf{w}} \left\|\mathbf{w}\right\|^2 \\ &\leq (\sigma^2 + 2(L\zeta)^2) + (M + 2(LZ)^2) \left\|\nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})\right\|^2.\end{aligned}$$

The first step is obtained by applying Assumption 4 to separate $\mathbf{w}$. We can then separate terms of $\mathbf{u}$ and $\mathbf{w}$ since their distributions are independent. Then, we use Lemma 5 and Assumption 4 to bound the two variance terms. ■

Appendix D. Deferred Proofs

In this section we give the proofs that are missing from the main text.

First, we state and prove an intermediate lemma for sufficient decrease which resembles (3). Using this Lemma, we can easily prove the corresponding theorems for gradient noise, PŁ and strongly-convex functions. Additionally, we restate the complete theorems for these cases which contain all the details about step sizes and exact convergence rate.

D.1. One Step Progress

Lemma 7 (One Step Progress) *Let f satisfy Assumptions 1 and 2 and, assume g to be L_g-smooth and $\mathbf{x}_t$ generated according to Algorithm 1. Then, for $\gamma \leq \frac{1}{L_g(M'+1)}$, it holds*

$$\frac{(1-m)}{2}\,\mathbb{E}[\|\nabla g(\mathbf{x}_t)\|^2] \leq \frac{\mathcal{G}_t - \mathcal{G}_{t+1}}{\gamma} + \frac{\Delta}{2} + \frac{\gamma L_g}{2}\sigma'^2\,,$$

where $\mathcal{G}_t$ and $g^\star$ are as defined before.

Proof Using L_g-smoothness of g, we can write

$$\begin{aligned}
g(\mathbf{x}_{t+1}) \leq& g(\mathbf{x}_t) + \langle \nabla g(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t\rangle + \frac{L_g}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 \\
\leq& g(\mathbf{x}_t) - \gamma\,\langle \nabla g(\mathbf{x}_t), \nabla g(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t) + \nabla h(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t)\rangle \\
&+ \frac{\gamma^2 L_g}{2}\|\nabla g(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t) + \nabla h(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t)\|^2\,.
\end{aligned}$$

Taking expectation wrt $\boldsymbol{\xi}_t$ and $\mathbf{u}_t$, and using the inequality $\mathbb{E}[\|X\|^2] = \mathbb{E}[\|X - \mathbb{E}[X]\|^2] + \|\mathbb{E}[X]\|^2$, and using the definition of smoothness we get

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}_t,\mathbf{u}_t}[g(\mathbf{x}_{t+1})] \leq& g(\mathbf{x}_t) - \gamma\,\left\langle \nabla g(\mathbf{x}_t), \mathbb{E}_{\boldsymbol{\xi}_t,\mathbf{u}_t}[\nabla g(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t) + \nabla h(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t)]\right\rangle \\
&+ \frac{\gamma^2 L_g}{2}\,\mathbb{E}_{\boldsymbol{\xi}_t,\mathbf{u}_t}\|\nabla g(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t) + \nabla h(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t)\|^2 \\
\leq& g(\mathbf{x}_t) - \gamma\,\left\langle \nabla g(\mathbf{x}_t), \nabla g_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) + \nabla h_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t)\right\rangle + \frac{\gamma^2 L_g}{2}\left\|\nabla f_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t)\right\|^2 \\
&+ \frac{\gamma^2 L_g}{2}\,\mathbb{E}_{\boldsymbol{\xi}_t,\mathbf{u}_t}[\left\|\nabla f(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t) - \nabla f_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t)\right\|^2]\,.
\end{aligned}$$

Using Assumption 1, with $\gamma \leq \frac{1}{L_g(M'+1)}$

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}_t,\mathbf{u}_t}[g(\mathbf{x}_{t+1})] \leq& g(\mathbf{x}_t) - \gamma\,\left\langle \nabla g(\mathbf{x}_t), \nabla g_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) + \nabla h_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t)\right\rangle \\
&+ \frac{\gamma^2 L_g(M'+1)}{2}\left\|\nabla f_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t)\right\|^2 + \frac{\gamma^2 L_g}{2}\sigma'^2\,. \qquad (19)\\
\mathbb{E}_{\boldsymbol{\xi}_t,\mathbf{u}_t}[g(\mathbf{x}_{t+1})] \leq& -\frac{\gamma}{2}\left(\|\nabla g(\mathbf{x}_t)\|^2 - \left\|\nabla h_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) + g_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) - g(\mathbf{x}_t)\right\|^2\right) \\
&+ g(\mathbf{x}_t) + \frac{\gamma^2 L_g}{2}\sigma'^2\,.
\end{aligned}$$

Now, using Assumption 2.

$$\mathbb{E}_{\boldsymbol{\xi}_t,\mathbf{u}_t}[g(\mathbf{x}_{t+1})] \leq g(\mathbf{x}_t) - \frac{\gamma(1-m)}{2}\left\|\nabla g(\mathbf{x}_t)\right\|^2 + \frac{\gamma\Delta}{2} + \frac{\gamma^2 L_g}{2}\sigma'^2\,.$$

Taking full expectation on both sides and subtracting $g^\star$ from both sides, we get the required result. ■

D.2. Gradient Norm Convergence

Theorem 8 *(Gradient Norm convergence) Under the assumptions in Lemma 7, for stepsize $\gamma \leq \frac{1}{L_g(M'+1)}$, after running the Algorithm 1 for T steps, it holds:*

$$\Phi_T \leq \left(\frac{2\mathcal{G}_0}{T\gamma(1-m)} + \frac{\gamma L_g\sigma'^2}{1-m}\right) + \frac{\Delta}{1-m}\,,$$

where $\Phi_T = \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla g(\mathbf{x}_t)\|^2]$.
Further, for $\epsilon > 0$ and $\gamma = \min\{\frac{1}{L_g(M'+1)}, \frac{\epsilon(1-m)+\Delta}{2L_g\sigma'^2}\}$, then

$$T = \mathcal{O}\left(\frac{M'+1}{\epsilon(1-m)+\Delta} + \frac{\sigma'^2}{\epsilon^2(1-m)^2+\Delta^2}\right)L_g\mathcal{G}_0$$

iterations are sufficient to obtain $\Phi_T = \mathcal{O}(\epsilon + \frac{\Delta}{1-m})$

Proof We can sum the terms of Lemma 7 for $t = 0$ to $T-1$, and divide both sides by T,to obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla g(\mathbf{x}_t)\|^2] \leq \frac{2(\mathcal{G}_0 - \mathcal{G}_T)}{T\gamma(1-m)} + \frac{\Delta}{(1-m)} + \frac{\gamma L_g}{(1-m)}\sigma'^2\,,$$

This proves the first part of the above Theorem. We can choose step sizes according to obtain rates in terms of ϵ. This can be found in [2, Lemma 3] and [2, Theorem 4] with different constants and notation. ■

D.3. Convergence for PŁ functions (Proof of Theorem 1)

We state the extended version of Theorem 1.

Theorem 9 *Under Assumptions of Lemma 7 and the additional assumption that g is μ-PŁ, it holds for any stepsize $\gamma \leq \frac{1}{L_g(M'+1)}$,*

$$\mathcal{G}_T \leq (1-\gamma\mu(1-m))^T\mathcal{G}_0 + \frac{1}{2}\Xi\,, \qquad \textit{where} \qquad \Xi = \frac{\Delta}{\mu(1-m)} + \frac{\gamma L_g\sigma'^2}{\mu(1-m)}\,.$$

Further, by choosing $\gamma = \min\{\frac{1}{L_g(M'+1)}, \frac{\epsilon(1-m)\mu+\Delta}{L_g\sigma'^2}\}$, for any $\epsilon > 0$,

$$T = \tilde{\mathcal{O}}\left((M'+1)\log\frac{1}{\epsilon} + \frac{\sigma'^2}{\epsilon(1-m)\mu+\Delta}\right)\frac{\kappa}{1-m}$$

iterations are sufficient to obtain $\mathcal{G}_T = \mathcal{O}(\epsilon + \frac{\Delta}{\mu(1-m)})$, where $\kappa := \frac{L_g}{\mu}$ and $\tilde{\mathcal{O}}$ hides only log terms.

Proof We use the PL condition in Lemma 7, to obtain

$$\mu\mathcal{G}_t \leq \frac{(\mathcal{G}_t - \mathcal{G}_{t+1})}{\gamma(1-m)} + \frac{\Delta}{2(1-m)} + \frac{\gamma L_g}{2(1-m)}\sigma'^2$$
$$\mathcal{G}_{t+1} \leq (1-\mu\gamma(1-m))\mathcal{G}_t + \frac{\Delta\gamma}{2} + \frac{\gamma^2 L_g}{2}\sigma'^2$$

Unfolding the above recursion from $t = 0$ to $t = T-1$, we get the first part of above Theorem. For the convergence rates in terms of ϵ, we can choose step size γ accordingly. This is similar to [2, Theorem 6] with different constants and notation. ■

D.4. Convergence under Strong Convexity

We now extend our results to the case when g is strongly convex. Note that while Theorem 1 still applies (all strongly convex functions are PŁ), applying this result in the for PŁ case admits a weaker convergence rate by a factor proportional to κ in contrast to the improved result in Theorem 10. This result is not covered in prior frameworks, as matching convergence rates were previously only derived for $m < 1/\kappa$ [2, Remark 7]. To achieve, this, we slightly refine our Assumption 2, ensuring we still are able to retain its expressivity.

Assumption 5 (Structural properties) *The objective function $f\colon \mathbb{R}^d \to \mathbb{R}$ can be written in the form (2) with g being L_g-smooth, and there exists parameters $\Delta \geq 0$, $0 \leq m < 1$ such that, $\forall \mathbf{x} \in \mathbb{R}^d$:*

$$\left|\left(\mathbf{r}(\mathbf{x})\right)_g\right|^2 \leq m\left\|\nabla g(\mathbf{x})\right\|^2, \qquad \left|\left(\mathbf{r}(\mathbf{x})\right)_{g_\perp}\right|^2 \leq \Delta,$$

where $\mathbf{r}(\mathbf{x}) = (\nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x}) - \nabla g(\mathbf{x}))$ and $(\mathbf{r}(\mathbf{x}))_g$ and $(\mathbf{r}(\mathbf{x}))_{g_\perp}$ denote the components of $\mathbf{r}(\mathbf{x})$, along the direction of $\nabla g(\mathbf{x})$ and perpendicular to it, respectively.

Our main idea is to split the bound in Assumption 2 to its respective components. Note that we can easily verify that this is stronger than Assumption 2 by computing $\|\mathbf{r}(\mathbf{x})\|^2$.

To ensure the same level of expressivity for both the structural assumptions, we can verify that they have similar worst-case scenarios for a biased oracle, that is, when $\mathbf{r}(\mathbf{x})$ points in the opposite direction of ∇g with squared norm $m\left\|\nabla g(\mathbf{x})\right\|^2$, ignoring the constant terms of Δ. Thus, our new assumption can still deal with worst-case oracles obeying Assumption 2 while still admitting a better analysis.

Theorem 10 *Under Assumptions 1 and 5, and if g is μ-strongly convex, running Algorithm 1 for T steps, with $\gamma \leq \frac{1-\sqrt{m}}{L_g(1+\sqrt{m})^2(M'+1)}$, there exist non-negative weights $\{w_t\}_{t=0}^T$, with $W_T = \sum_{t=0}^T w_t$, such that*

$$\frac{1}{W_T}\sum_{t=0}^{T} w_t\mathcal{G}_T + \frac{\mu}{2}d_{T+1} = \mathcal{O}\left(\frac{d_0}{\gamma(1-\sqrt{m})}\exp\left(-\frac{(1-\sqrt{m})\gamma\mu T}{2}\right) + \Xi\right)$$

where $\mathcal{G}_t$ is same as defined previously before, $d_t = \mathbb{E}[\left\|\mathbf{x}_t - \mathbf{x}_g^\star\right\|^2]$, $\mathbf{x}_g^\star = \arg\min_{\mathbf{x}\in\mathbb{R}^d} g(\mathbf{x})$, and

$$\Xi = \frac{\gamma(\sigma'^2 + \Delta(M'+1))}{(1-\sqrt{m})} + \frac{2\Delta}{\mu(1-\sqrt{m})^2}.$$

Further, choosing $\gamma = \min\left\{\frac{(1-\sqrt{m})}{L_g(M'+1)(1+\sqrt{m})^2}, \frac{\mu\epsilon(1-\sqrt{m})^2+4\Delta}{2(\sigma'^2+\Delta(M'+1))(1-\sqrt{m})\mu}\right\}$,

$$T = \tilde{\mathcal{O}}\left(\frac{2\kappa(M'+1)(1+\sqrt{m})^2}{(1-\sqrt{m})^2}\log\frac{1}{\epsilon} + \frac{4(\sigma'^2+\Delta(M'+1))}{\mu\epsilon(1-\sqrt{m})^2+4\Delta}\right)$$

iterations are sufficient to obtain $\frac{1}{W_T}\sum_{t=0}^{T} w_t\mathcal{G}_T = \mathcal{O}(\epsilon + \frac{4\Delta}{\mu(1-\sqrt{m})^2})$.

Comparing Theorems 1 and 10, we find that the κ dependence is no longer present in the noise term, while our proof holds for arbitrary $m < 1$. Thus, we have addressed both the problems which we mentioned at the start of this subsection. However, this does not come for free, as the convergence rate is inversely proportional to $(1-\sqrt{m})$, instead of $1-m$, in the PŁ case and $1-\sqrt{m} < 1-m$. Also, we have a larger noise term $(\sigma'^2 + \Delta(M'+1))$, than with PŁ, which also depends on Δ.

Proof Consider $\left\|\mathbf{x}_t - \mathbf{x}_g^\star\right\|^2$, and take expectations with respect to $\mathbf{u}_t, \boldsymbol{\xi}_t$, on both sides, further use $\mathbb{E}[\|X\|^2] = \mathbb{E}[\|X - \mathbb{E}[X]\|^2] + \|\mathbb{E}[X]\|^2$ and Assumption 1.

$$\begin{aligned}
\left\|\mathbf{x}_{t+1} - \mathbf{x}_g^\star\right\|^2 &= \left\|\mathbf{x}_t - \mathbf{x}_g^\star\right\|^2 - 2\gamma\left\langle\nabla f(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t), \mathbf{x}_t - \mathbf{x}_g^\star\right\rangle \\
&\quad + \gamma^2\left\|\nabla f(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t)\right\|^2 \\
\mathbb{E}_{\mathbf{u}_t,\boldsymbol{\xi}_t}[\left\|\mathbf{x}_{t+1} - \mathbf{x}_g^\star\right\|^2] &= \left\|\mathbf{x}_t - \mathbf{x}_g^\star\right\|^2 - 2\gamma\left\langle\nabla f_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_g^\star\right\rangle + \gamma^2\left\|\nabla f_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t)\right\|^2 \\
&\quad + \gamma^2\,\mathbb{E}_{\mathbf{u}_t,\boldsymbol{\xi}_t}[\left\|\nabla f(\mathbf{x}_t - \mathbf{u}_t, \boldsymbol{\xi}_t) - \nabla f_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t)\right\|^2] \\
&\leq \left\|\mathbf{x}_t - \mathbf{x}_g^\star\right\|^2 - 2\gamma\left\langle\nabla g(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_g^\star\right\rangle + \gamma^2\sigma'^2 \\
&\quad - 2\gamma\left\langle\nabla g_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) + \nabla h_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) - \nabla g(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_g^\star\right\rangle \\
&\quad + \gamma^2(M'+1)\left\|\nabla g_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) + \nabla h_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t)\right\|^2 . \qquad (20)
\end{aligned}$$

Let $\widehat{\nabla g(\mathbf{x}_t)}$ and $\widehat{\nabla g(\mathbf{x}_t)}_\perp$ be the units vector in direction of $\nabla g(\mathbf{x}_t)$ and perpendicular to it, respectively. For clarity of notations, let $(\nabla h_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) + \nabla g_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) - \nabla g(\mathbf{x}_t)) = \mathbf{r}(\mathbf{x}_t)$. First, we bound the component perpendicular to $\nabla g(\mathbf{x}_t)$, using Assumption 2

$$\begin{aligned}
(\mathbf{r}(\mathbf{x}_t))_{g_\perp}\left\langle\widehat{\nabla g(\mathbf{x}_t)}_\perp, \mathbf{x}_t - \mathbf{x}_g^\star\right\rangle &\leq \frac{\mu}{4}\left\|\mathbf{x}_t - \mathbf{x}_g^\star\right\|^2 + \frac{1}{\mu}\left|(\mathbf{r}(\mathbf{x}_t))_{g_\perp}\right|^2 \\
&\leq \frac{\mu(1-\sqrt{m})}{4}\left\|\mathbf{x}_t - \mathbf{x}_g^\star\right\|^2 + \frac{\Delta}{\mu(1-\sqrt{m})} . \qquad (21)
\end{aligned}$$

Now, consider the component along $\nabla g(\mathbf{x}_t)$ and strong convexity of g implies $\left\langle\nabla g(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_g^\star\right\rangle \geq 0$, and using Assumption 2

$$\begin{aligned}
(\mathbf{r}(\mathbf{x}_t))_g\left\langle\widehat{\nabla g(\mathbf{x}_t)}_g, \mathbf{x}_t - \mathbf{x}_g^\star\right\rangle &\geq -\frac{|(\mathbf{r}(\mathbf{x}_t))_g|}{\|\nabla g(\mathbf{x}_t)\|}\left\langle\nabla g(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_g^\star\right\rangle \\
&\geq -\sqrt{m}\left\langle\nabla g(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_g^\star\right\rangle . \qquad (22)
\end{aligned}$$

Additionally, consider $\left\|\nabla g_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) + \nabla h_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t)\right\|^2$ and use Assumption 5.

$$\begin{aligned}\left\|\nabla g_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) + \nabla h_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t)\right\|^2 &\leq \left\|\nabla g_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) + \nabla h_{\mathcal{U}(\mathbf{x}_t)}(\mathbf{x}_t) - \nabla g(\mathbf{x}_t) + \nabla g(\mathbf{x}_t)\right\|^2 \\ &\leq \|\mathbf{v} + \nabla g(\mathbf{x}_t)\|^2 \\ &\leq \left\|(\mathbf{v})_g \nabla \hat{g}(\mathbf{x}_t) + (\mathbf{v})_{g_\perp} \nabla \hat{g}(\mathbf{x}_t)_\perp + \nabla g(\mathbf{x}_t)\right\|^2 \\ &\leq |(\mathbf{v})_g + \|\nabla g(\mathbf{x}_t\|\|^2 + |(\mathbf{v})_{g_\perp}|^2 \\ &\leq (1+\sqrt{m})^2 \|\nabla g(\mathbf{x}_t)\|^2 + \Delta\,. \end{aligned} \tag{23}$$

Using Eqns. (21), (22) and (23) in Eq. (20), we get

$$\begin{aligned}\mathbb{E}_{\mathbf{u}_t,\boldsymbol{\xi}_t}[\left\|\mathbf{x}_{t+1} - \mathbf{x}_g^\star\right\|^2] \leq \left\|\mathbf{x}_t - \mathbf{x}_g^\star\right\|^2 (1 + \frac{\gamma\mu(1-\sqrt{m})}{2}) - 2\gamma(1-\sqrt{m})\left\langle \nabla g(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_g^\star\right\rangle \\ + \gamma^2(M'+1)(1+\sqrt{m})^2 \|\nabla g(\mathbf{x}_t)\|^2 + \gamma^2(\sigma'^2 + \Delta(M'+1)) \\ + \frac{2\gamma\Delta}{\mu(1-\sqrt{m})}\,.\end{aligned}$$

Now, using strong-convexity and smoothness of g, we get

$$\begin{aligned}\mathbb{E}_{\mathbf{u}_t,\boldsymbol{\xi}_t}[\left\|\mathbf{x}_{t+1} - \mathbf{x}_g^\star\right\|^2] \leq \left\|\mathbf{x}_t - \mathbf{x}_g^\star\right\|^2 \left(1 - \frac{\gamma\mu(1-\sqrt{m})}{2}\right) + \gamma^2(\sigma'^2 + \Delta(M'+1)) + \frac{2\gamma\Delta}{\mu(1-\sqrt{m})} \\ - 2\gamma(1-\sqrt{m})\left(1 - \frac{\gamma L_g(M'+1)(1+\sqrt{m})^2}{2(1-\sqrt{m})}\right)(g(\mathbf{x}_t) - g(\mathbf{x}_g^\star))\,.\end{aligned}$$

Now, taking $\gamma \leq \frac{(1-\sqrt{m})}{L_g(M'+1)(1+\sqrt{m})^2}$, taking complete expectations, and substituting $\mathcal{G}_t = \mathbb{E}[g(\mathbf{x}_t)] - g(\mathbf{x}_g^\star)$ and $d_t = \mathbb{E}[\left\|\mathbf{x}_t - \mathbf{x}_g^\star\right\|^2]$.

$$\begin{aligned}d_{t+1} \leq d_t\left(1 - \frac{\gamma\mu(1-\sqrt{m})}{2}\right) + \gamma^2(\sigma'^2 + \Delta(M'+1)) + \frac{2\gamma\Delta}{\mu(1-\sqrt{m})} \\ - \gamma(1-\sqrt{m})\mathcal{G}_t\,.\end{aligned}$$

We follow analysis in [69, Lemma 2] to multiply both sides by $w_t = \left(1 - \frac{\gamma\mu(1-\sqrt{m})}{2}\right)^{-(t+1)}$. If $\frac{\gamma\mu(1-\sqrt{m})}{2} < 1$, we sum over $t = 0$ to T and divide both sides by $W_T = \sum_{t=0}^T w_t$. We obtain the following results after performing these steps,

$$\frac{(1-\sqrt{m})}{W_T}\sum_{t=0}^T w_t\mathcal{G}_t + \frac{w_T d_{T+1}}{\gamma W_T} \leq \frac{d_t}{\gamma W_T} + \frac{2\Delta}{\mu(1-\sqrt{m})} + \gamma(\sigma'^2 + \Delta(M'+1))\,.$$

Since $W_T \leq \frac{w_T}{(\gamma\mu(1-\sqrt{m})/2)\gamma}$ and $W_T \geq w_T$, we obtain the first inequality

$$\begin{aligned}\frac{1}{W_T}\sum_{t=0}^T w_t\mathcal{G}_t + \frac{\mu}{2}\frac{d_{T+1}}{\gamma W_T} \leq &\frac{d_0}{\gamma(1-\sqrt{m})}\exp\left(-\frac{\mu\gamma(1-\sqrt{m})T}{2}\right) + \frac{2\gamma\Delta}{\mu(1-\sqrt{m})^2} \\ &+ \frac{\gamma(\sigma'^2 + \Delta(M'+1))}{(1-\sqrt{m})}\,.\end{aligned}$$

For the second part, first let $\alpha = \sigma'^2 + \Delta(M'+1)$ and $\beta = M'+1$ Then, we denote the RHS of the main convergence result in terms of γ and T.

$$\Theta(\gamma, T) = \frac{d_0}{\gamma(1-\sqrt{m})} \exp\left(-\frac{\mu\gamma(1-\sqrt{m})T}{2}\right) + \frac{\alpha}{(1-\sqrt{m}} + \frac{2\Delta}{\mu(1-\sqrt{m})^2}\,.$$

We show that our bound for $\Theta(\gamma, T) = \mathcal{O}\big(\epsilon + \frac{4\Delta}{\mu(1-\sqrt{m})^2}\big)$ is achieved by $\gamma = \min\{\gamma_1, \gamma_2\}$ and $T = \max\{T_1, T_2\}$

$$\begin{aligned}
\gamma_1 =& \frac{(1-\sqrt{m})}{L_g M'(1+\sqrt{m})^2}, \qquad \gamma_2 = \frac{\mu\epsilon(1-\sqrt{m})^2 + 4\Delta}{2\alpha(1-\sqrt{m})\mu} \\
T_1 =& \frac{2\beta L_g(1+\sqrt{m})^2}{\mu(1-\sqrt{m})^2} \log\left(\frac{2L_g\beta d_0(1+\sqrt{m})^2}{\epsilon(1-\sqrt{m})^2}\right), \\
T_2 =& \frac{4\beta}{\mu\epsilon(1-\sqrt{m})^2 + 4\Delta} \log\left(\frac{4d_0\alpha\mu}{(\mu\epsilon(1-\sqrt{m})^2 + 4\Delta)\epsilon}\right).
\end{aligned}$$

If $\gamma = \gamma_1$, then $\frac{\gamma\alpha}{(1-\sqrt{m})} \leq \frac{\epsilon}{2} + \frac{2\Delta}{\mu(1-\sqrt{m})^2}$. Then, we can choose $T \geq T_1$, so that $\Theta(\gamma, T) \leq \epsilon + \frac{4\Delta}{\mu(1-\sqrt{m})^2}$

Similarly, if $\gamma = \gamma_2$, then $\frac{\gamma\alpha}{(1-\sqrt{m})} \leq \frac{\epsilon}{2} + \frac{2\Delta}{\mu(1-\sqrt{m})^2}$. Then, we can choose $T \geq T_2$, so that $\Theta(\gamma, T) \leq \epsilon + \frac{4\Delta}{\mu(1-\sqrt{m})^2}$. ■

D.5. Additional Settings

In this subsection, we present alternative formulations to our Assumptions, namely, for bounded perturbed h and for exact smooth oracle $\nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})$, instead of the perturbed gradient.

D.5.1. CONVERGENCE FOR EXACT SMOOTH ORACLE $\nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x})$

While we have derived all results assuming we have access to $\nabla f(\mathbf{x}+\mathbf{u};\boldsymbol{\xi})$, our results can be extended to the case when we have access to $\nabla f_{\mathcal{U}(\mathbf{x})}(\mathbf{x};\boldsymbol{\xi})$. This extension is similar to extensions of SGD results to GD. This is mainly done by setting the variance of gradients to 0, by setting $\sigma^2 = M = 0$. Similarly, for our case setting $\zeta^2 = Z = 0$, yields converge rates with gradient oracle $\nabla f_{\mathcal{U}(\mathbf{x})}$. This does not mean that the smoothing distribution $\mathcal{U}(\mathbf{x})$ has 0 variance, just that the contribution to gradient noise due to smoothing is 0, again motivating the connection between smoothing and SGD.

D.5.2. PERTURBATION h WITH BOUNDED GRADIENTS

In this section, we explore a class of non-convex functions satisfying our formulation (2), but which are easy to solve. Consider as before that $g(\mathbf{x})$ and $h(\mathbf{x})$ denote the convex part and non-convex perturbation of $f(\mathbf{x})$, respectively. We now provide a few definitions which we will use later.

A point $\mathbf{x} \in \mathbb{R}^d$ is a stationary point of a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ if

$$\nabla f(\mathbf{x}) = 0\,.$$

Let $\mathcal{X}^\star$ denote the set of stationary points of f. Additionally, let $g^\star = \min_{\mathbf{x}\in\mathbb{R}^d} g(\mathbf{x})$ and $\mathbf{x}_g^\star = \arg\min_{\mathbf{x}\in\mathbb{R}^d} g(\mathbf{x})$

A function $h : \mathbb{R}^d \to \mathbb{R}$ has B_2-bounded gradients if

$$\|\nabla h(\mathbf{x})\|^2 \leq B_2 \qquad \forall \mathbf{x} \in \mathbb{R}^d \,. \tag{24}$$

A function $h : \mathbb{R}^d \to \mathbb{R}$ is B_1-bounded if

$$|h(\mathbf{x})| \leq B_1 \qquad \forall \mathbf{x} \in \mathbb{R}^d \,. \tag{25}$$

With these definitions, we provide the below lemma, which illustrates the impact of a simple (bounded and gradient bounded) h on the stationary points of f.

Lemma 11 *Let f satisfy structure (2) with convex part g and non-convex part h.*

- *If g is μ-PŁ and h is B_2-gradient bounded*

$$g^\star \leq g(\mathbf{x}) \leq g^\star + \frac{B_2}{2\mu} \,, \qquad \forall \mathbf{x} \in \mathcal{X}^\star \,.$$

- *If g is μ-strongly convex and h is B_2-gradient-bounded*

$$\left\|\mathbf{x} - \mathbf{x}_g^\star\right\|^2 \leq \frac{B_2}{\mu^2} \,, \qquad \forall \mathbf{x} \in \mathcal{X}^\star, \mathcal{X}_g = \{\mathbf{x}_g^\star\} \,.$$

- *If g is μ-PŁ and h is B_1-bounded and B_2-gradient bounded*

$$g^\star - B_1 \leq f(\mathbf{x}) \leq g^\star + B_1 + \frac{B_2}{2\mu} \,, \qquad \forall \mathbf{x} \in \mathcal{X}^\star \,,$$
$$|f(\mathbf{x}) - f(\mathbf{y})| \leq 2B_1 + \frac{B_2}{2\mu} \,, \qquad \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}^\star \,.$$

Proof Let $\mathbf{y}$ be a stationary point of f. Then,

$$\nabla g(\mathbf{y}) = -\nabla h(\mathbf{y}) \,.$$

For the first part, since g is and h is B_2-gradient bounded,

$$2\mu(g(\mathbf{y}) - g^\star) \leq \|\nabla g(\mathbf{y})\|^2 = \|\nabla h(\mathbf{y})\|^2 \leq B_2 \,.$$

For the second part, since g is μ-strongly convex with global minima $\mathbf{x}_g^\star$

$$g(\mathbf{y}) \geq g^\star + \frac{\mu}{2}\left\|\mathbf{y} - \mathbf{x}_g^\star\right\|^2 \,,$$

and the claim follows together with the first part of this lemma (all μ-strongly convex functions are also μ-PŁ).

For the third part, assuming h is B_1- bounded with the result from first part,

$$g^\star + h(\mathbf{y}) \leq g(\mathbf{y}) + h(\mathbf{y}) \leq g^\star + h(\mathbf{y}) + \frac{B_2}{2\mu} \,,$$
$$g^\star - B_1 \leq f(\mathbf{y}) \leq g^\star + B_1 + \frac{B_2}{2\mu} \,.$$

■

From the above lemma, we can see that if h is gradient bounded, all its stationary points are close to minima of g. Thus, even GD on such a function should always end up close to the global minima. Note that Assumption 2 is weaker than bounded gradients for h, as we allow h to have unbounded gradients and its stationary points are also not constrained to a neighborhood. This is demonstrated by our toy example $f(x) = x^2 + ax\sin(bx)$, which we describe in detail in the next section.

Appendix E. Additional Details about connection to SGD

In this section we provide additional details about the connection between Algorithm 1 and SGD. We provide a lemma to connect the results for strongy-convex case for Algorithm 1 to SGD iterates. Later, we discuss how to extend this to the finite-sum case and provide experiments to verify this assumption. Additionally, we verify these claims for Deep Learning examples in Appendix G.

Lemma 12 *Let $\mathbf{x}_t, \mathbf{y}_t$ and $\mathbf{z}_t$ be as defined above. Define $\bar{\mathbf{y}}_T := \frac{1}{W_T}\sum_{t=0}^{T} \mathbf{w}_t \mathbb{E}[\mathbf{y}_t]$, for $\{\mathbf{w}_t\}_{t=0}^{T}$ and W_T as defined in Theorem 10. If Lemma 2 holds, g is convex,*

$$g(\bar{\mathbf{y}}_T) - g(\mathbf{x}_g^\star) \leq \frac{1}{W_T}\sum_{t=0}^{T} \mathbf{w}_t \mathcal{G}_t$$

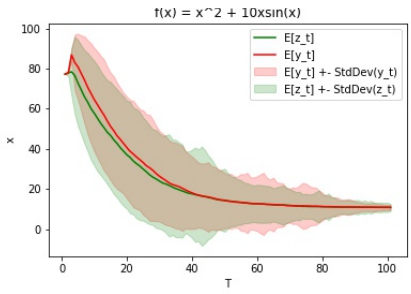
where $\mathbf{x}_g^\star$ and $\mathcal{G}_t$ are as defined in Theorem 10 .

Proof Consider the term $g(\bar{\mathbf{y}}_t) - g^\star$.

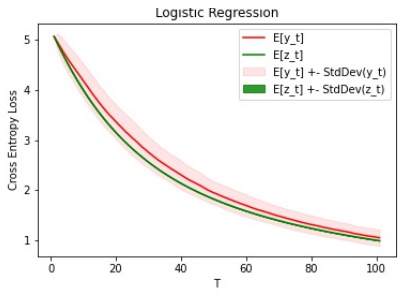
$$\begin{aligned}
g(\bar{\mathbf{y}}_t) - g^\star &\leq \frac{1}{W_T}\sum_{t=0}^{T} \mathbf{w}_t (g(\mathbb{E}[\mathbf{y}_t]) - g^\star) \\
&\leq \frac{1}{W_T}\sum_{t=0}^{T} \mathbf{w}_t (g(\mathbb{E}[\mathbf{z}_t]) - g^\star) \\
&\leq \frac{1}{W_T}\sum_{t=0}^{T} \mathbf{w}_t (\mathbb{E}[g(\mathbf{z}_t)] - g^\star) \\
&\leq \frac{1}{W_T}\sum_{t=0}^{T} \mathbf{w}_t \mathcal{G}_t \,.
\end{aligned}$$

For the first step, we use convexity of g with coefficients $\left\{\frac{\mathbf{w}_t}{W_T}\right\}_{t=0}^{T}$. The second step is obtained from equality in expectation. The third step is obtained from Jensen's inequality on convex g and the last term is the definition of $\mathcal{G}_t$. ■

This lemma allows us to utilize the results of Thm. 10 for SGD iterates defined by $\mathbf{y}_t$.



(*a*) Uniform noise setting



(*b*) Noise dependent on $\mathbf{x}$

Figure 3: Equivalent trajectories of SGD and Perturbed SGD. Mean trajectories of 1000 independent runs of SGD and Perturbed SGD with the same γ selected by grid search, as described in Fig. 2, and $\mathbf{z}_0 = \mathbf{y}_0$. Solid lines depict mean and shaded areas standard deviations.

E.1. Finite-Sum Setting

In this section, we explain the connection between SGD and our Algorithm 1 for a finite-sum objective. Consider the objective function, $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$, which is a sum of n terms. For SGD, at each step t,

$$\nabla f(\mathbf{x}_t, \boldsymbol{\xi}) = \nabla f_i(\mathbf{x}_t)$$

where i is sampled uniformly at random from $[n]$. Thus, the noise in each gradient step, $\mathbf{w}_t$, is,

$$\mathbf{w}_t = \nabla f_i(\mathbf{x}_t) - \nabla f(\mathbf{x}_t) = \nabla f_i(\mathbf{x}_t) - \tfrac{1}{n}\textstyle\sum_{j=1}^{n} \nabla f_j(\mathbf{x}_t)\,. \tag{26}$$

To find an equivalent smoothing distribution, we can set $\mathcal{U}(\mathbf{x}) = \gamma\mathcal{W}(\mathbf{x})$ as described above. However, the resulting distribution would require to compute $\mathbf{u}_t = \gamma\big(\nabla f_k(\mathbf{x}_t) - \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(\mathbf{x}_t)\big)$ for an uniformly at random sampled index k. This involves computation of a full batch gradient, rendering the resulting procedure very inefficient. To overcome this, we can define $\mathbf{u}_t$ in the following way:

$$\mathbf{u}_t = \gamma(\nabla f_k(\mathbf{x}_t) - \nabla f_j(\mathbf{x}_t))\,, \tag{27}$$

where k, j are sampled uniformly at random from $[n]$. This results in an efficient oracle with variance

$$\mathbb{E}_{\mathcal{U}(\mathbf{x}_t)}[\|\mathbf{u}_t\|^2] = 2\gamma^2\,\mathbb{E}_{\mathcal{W}(\mathbf{x}_t)}[\|\mathbf{w}_t\|^2]\,.$$

Note that this resembles the method implemented in [29] in a distributed setting.

E.2. Experimental Verification

We empirically demonstrate the connections between our algorithm and SGD in two settings, when noise is– a) independent of $\mathbf{x}$ and b) dependent on $\mathbf{x}$.

For our first setting (depicted in Figure **??**), we use our toy problem $f(x) = x^2 + 10x\sin(x)$. We fix the initial point for SGD as $x_0 = 100$ and $\zeta = 0.1$. We add a Gaussian noise sampled from $\mathcal{N}(0, \sigma^2)$ to the gradients, where $\sigma^2 = \gamma\zeta^2$.

For our second setting (depicted in Figure 3(*b*)subfigure), we consider a finite-sum objective. The objective function is of the form $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$, where n is the number of datapoints

and $f_i(\mathbf{x})$ is the cross-entropy loss for the n^{th} datapoint. For SGD, we sample 1 datapoint from the dataset at each step, while for the smoothing distribution, we use the formulation in (26), as described above.

The stochastic noise arises from sampling one datapoint in the finite sum with replacement, and is thus dependent on $\mathbf{x}$. We use logistic regression with cross entropy loss on the Digits dataset [19] from scikit-learn [62]. The dataset consists of 8×8 images of handwritten digits from `0` to `9`, from which we use only images of `0` and `1`. For SGD, $\mathbf{x}_0$ is sampled uniformly from $[-0.5, 0.5]^{64}$. We choose the same sampling for $\mathcal{U}(\mathbf{x})$, to obtain $\mathcal{U}(\mathbf{x}) = \gamma \mathcal{W}(\mathbf{x})$.

For both of these cases, the mean trajectories for $\mathbf{y}_t$ and $\mathbf{z}_t$ are very close, verifying our analysis. For the uniform noise setting, the variances of the trajectories are also very similar. However, the variance for our algorithm is much smaller than SGD for the logistic regression example. Note that we also illustrate this connection for deep learning examples in Appendix G.

Appendix F. Investigating Examples

In this section, we further investigate our toy example $f(x) = x^2 + ax\sin(bx)$ and utilize it to compare our settings to other applications of non-convex smoothing in [30, 42].

For any finite value of ζ, the function $f_{\mathcal{U}}$ is never convex. However, for every $\zeta > \frac{1}{b}\sqrt{(2\ln(ab) - \ln(4))}$, we can always find $m < 1, \Delta > 0$ which satisfies our Assumption 2.

F.1. Comparison to other Applications of Non-Convex Smoothing

In [30], the notion of graduated optimization is utilized, by successively smoothing with decreasing δ variance, to converge to global optima of a class of non-convex Lipschitz functions in a bounded domain $\mathcal{X}$ ((c, δ)-nice, [30, Definition 3.2]). Convergence of their method relies on the function becoming strongly-convex on $\mathcal{X}$ after $c\delta$-smoothing. For a fixed domain, we can set $\zeta = c\delta > \frac{1}{b}\sqrt{(2\ln(ab) - \ln(4))}$, with appropriate a, b such that our toy example is never strongly convex in a fixed interval inside $\mathcal{X}$, but satisfies our Assumption 2. Thus, their analysis fails on our example. Further, on a bounded domain, if a function is strongly-convex after smoothing, it satisfies our Assumption 2 for the same smoothing with $m = \Delta = 0$. Thus, all (c, δ)-nice functions also satisfy this assumption.

Our assumptions are weaker than those required in [42]. Notably, [42] consider only smoothing with bounded support, while we do not have this restriction. Moreover, they need to assume that for given $\mathcal{U}$, $f_{\mathcal{U}}$ is star convex. We see from Figure 1(*a*)subfigure that our toy function is not star convex for all ζ^2, while our Assumption 2 holds. This shows, that our setting allows more flexibility and trade-offs in the parameters.

Consider $f(x) = x^2 + ax\sin(bx)$ and $\mathcal{U} = \mathcal{N}(0, \zeta^2)$ as in the main text. For $g(x) = x^2$ and $h(x) = ax\sin(bx)$, we observe that

$$
\begin{aligned}
g_{\mathcal{U}}(x) = x^2 + \zeta^2, \quad & h_{\mathcal{U}}(x) = ae^{-(b^2\zeta^2)/2}(b\zeta^2\cos(bx) + x\sin(bx)) \\
\nabla g_{\mathcal{U}}(x) = 2x, \quad & \nabla h_{\mathcal{U}}(x) = abe^{-(b^2\zeta^2)/2}((1 - b\zeta^2)\sin(bx) + x\cos(bx)) \\
\|\nabla h_{\mathcal{U}}(x) + \nabla g_{\mathcal{U}}(x) - \nabla g(x)\|^2 &\leq a^2b^2e^{-b^2\zeta^2}(x^2 + (b\zeta^2 - 1)^2) \\
&\leq \frac{a^2b^2e^{-b^2\zeta^2}}{4}(\|\nabla g(x)\|^2 + 4(b\zeta^2 - 1)^2)
\end{aligned}
$$

To satisfy Assumption 2 we can choose $m = \frac{1}{4}a^2b^2e^{-b^2\zeta^2}$ or $\zeta = \frac{1}{b}\sqrt{2\ln(ab) - \ln(4m)}$ (note that $m < 1$) and $\Delta = a^2b^2e^{-b^2\zeta^2}(b\zeta^2 - 1)^2 = \frac{4m}{b^2}(2\ln(ab) - \ln(4m) - b)^2$.

F.2. Comparing to (c, δ)-Nice Functions [30]

We consider the toy example which is (c, δ)-nice, mentioned in [30], and show that this function can be optimized under our biased gradient assumptions as well. Consider $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d) \in \mathbb{R}^d$

$$f(\mathbf{x}) = 0.5\left\|\mathbf{x}\right\|^2 - \alpha e^{-\frac{x_1-1}{2\lambda^2}}$$

This function is $(\sqrt{d}, 0.5)$-nice for $\lambda \leq 0.1$ and $\alpha \in \left[0, \frac{1}{200}\right]$. Note that, if we consider $g(\mathbf{x}) = \mathbf{x}^2$ and $h(\mathbf{x}) = -\alpha e^{-\frac{x_1-1}{2\lambda^2}}$, after smoothing with $\mathcal{U} = \mathcal{N}(\mathbf{0}, \zeta^2 I_d)$, we obtain –

$$\left\|\nabla h_{\mathcal{U}}(\mathbf{x}) + \nabla g_{\mathcal{U}}(\mathbf{x}) - \nabla g(\mathbf{x})\right\|^2 \leq \frac{\alpha^2\zeta^4}{\lambda^2(\zeta^2+\lambda^2)^3}\left(\left\|\nabla g(\mathbf{x})\right\|^2 + 1\right).$$

Here, choosing $\zeta = k\lambda$, this function satisfies Assumption 2 with $\Delta = m = \frac{\alpha^2k^4}{(k^2+1)^3\lambda^4}$. For every valid α, λ, we can choose k such that $m < 1$.

F.3. Toy Example is not convex after smoothing

Consider the toy example again, $f(x) = x^2 + 10x\sin(x)$, with smoothing f with $\mathcal{N}(0, \zeta^2)$. We obtain:

$$f_{\mathcal{U}}(x) = x^2 + \zeta^2 + ae^{-(b\zeta)^2/2}\left(b\zeta^2\cos(bx) + x\sin(bx)\right). \tag{28}$$

According to our structure (2), we can pick $g(x) = x^2$ and $h(x) = ax\sin(bx)$. We observe that smoothing reduces the non-convexity in the function and it starts resembling its convex component g. This is better visualized in Figure 1, where we plot the function and its gradient for parameters $a = 10$ and $b = 1$ and $\zeta \in \{0, 1, 2\}$, where $\zeta = 0$ corresponds to no smoothing.

Further, if we take our toy example again, $f(x) = x^2 + 10x\sin(x)$, we can see that even after smoothing f with $\mathcal{N}(0, \zeta^2)$, $f_{\mathcal{U}}$ still has local minima and is not strongly-convex. To generate a concrete example, consider $\zeta = 2$, and denote the smoothed function with f_ζ which is plotted in Figure 1(*a*)subfigure, and for better visualization additionally in Figure 4. The smoothed function f_2 has two minima, close to $x \approx -2.56$ and $x \approx 2.56$ and an additional stationary point at $x = 0$. Therefore, the function f_2 is not strongly convex on a 3ζ-ball around its minima (as each such ball contains also $x = 0$ and the other minima). Therefore, the example function f_2 does not satisfy the local strong convexity condition that is required for (c, δ)-nice functions, but it satisfies our Assumption 2 (note that $\zeta > 2$ satisfies the sufficient condition derived in Section F.1 above).

F.4. Additional experiments on toy example

We perform additional experiments on our toy example for the same settings as Section 6. We implement Perturbed SGD with no gradient noise and different smoothing by controlling ζ and SGD, with a Gaussian gradient noise distribution, $\mathcal{W} = \mathcal{N}(0, \sigma^2)$.

From Figure 5, we can see that SGD and Perturbed SGD have similar behaviours when we start increasing the noise level, as the last iterates are able to escape local minimas. But, if we keep

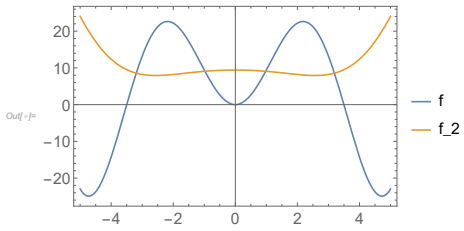
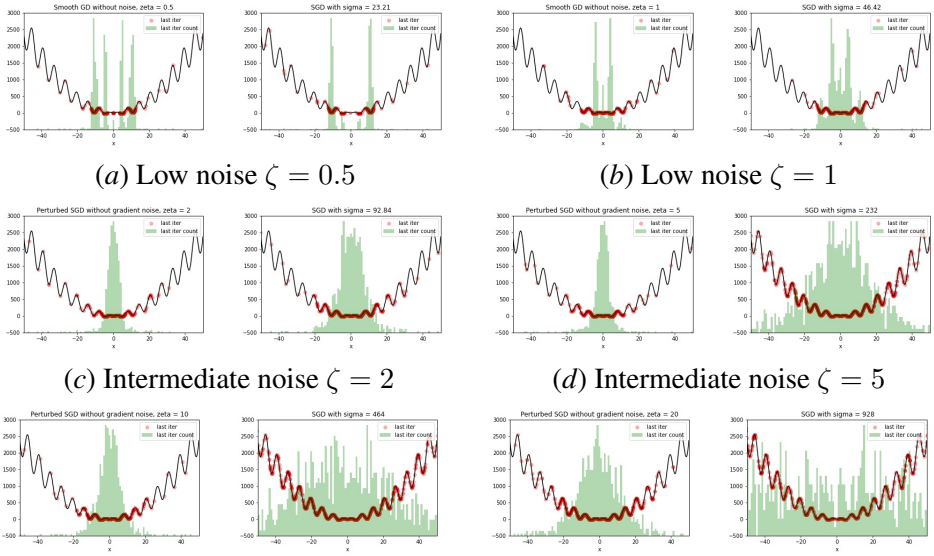


Figure 4: Function $f(x)$ and $f_\zeta(x)$ for $\zeta = 2$ (the same function as in Figure 1(a)subfigure, highlighting that f_2 is not strongly convex in a 3ζ-ball around its minima, as required for (c, ζ)-nice functions, but f_2 satisfies Assumption 2.



(a) Low noise $\zeta = 0.5$ (b) Low noise $\zeta = 1$

(c) Intermediate noise $\zeta = 2$ (d) Intermediate noise $\zeta = 5$

(e) High noise $\zeta = 10$ (f) High Noise $\zeta = 20$

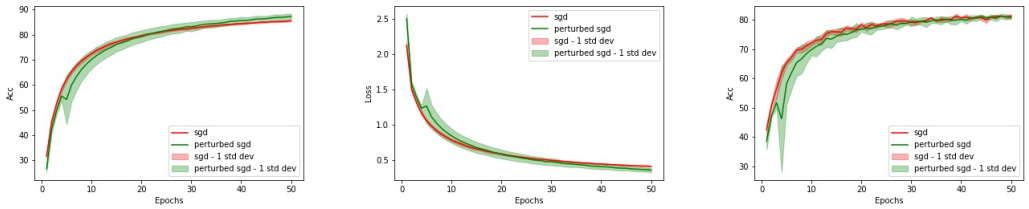
Figure 5: Comparison of Last iterate positions for SGD and Perturbed SGD without gradient noise for same noise levels. In each subfigure, ζ decides the noise level of both SGD and Perturbed SGD, as γ is constant.

increasing the noise level, SGD starts performing poorly and its last iterates get spread out evenly over the domain. In contrast, Perturbed SGD at the same noise level concentrates around the global minima, and only at the highest noise level of $\zeta = 20$, its last iterates start spreading out. Although SGD and Perturbed SGD are equal in expectation, there are key differences especially in high noise setting.

Appendix G. Deep Learning Examples

We further investigate the equivalence between SGD and Perturbed SGD for a standard deep learning problem–Resnet18 on CIFAR10 dataset . Note that in deep learning settings, our loss function is $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$, where $f_i(\mathbf{x})$ is the loss, in this case cross-entropy loss, for the i^{th} datapoint in the dataset for network with weights given by $\mathbf{x}$.

We compare our Algorithm 1 with mini-batch SGD with batch size 128. In Section E.1, we describe two possible implementations for the finite-sum setting– (26) and (27). Since we require the full-batch gradient in each step of (26), we cannot use this in deep learning settings with large



(*a*) Training Accuracy (*b*) Training Loss (*c*) Validation Accuracy

Figure 6: Equivalent Trajectories for SGD and finite-sum implementations of Perturbed SGD (Algorithm 1) according to (27). Mean trajectories after 5 independent runs of SGD and Perturbed SGD with same $\gamma = 0.1$, momentum = 0.9, weight decay = 10^{-4} for 50 epochs with same initialization and noise levels. Solid lines depict means and shaded areas standard deviations.

dataset sizes. In (27), we utilize only minibatch gradients, so we can apply it to deep learning problems. In our pytorch implementation, we break down Algorithm 1 into two steps–perturbation step which computes $\mathbf{u}_t$, and the gradient step which updates parameters with $\nabla f(\mathbf{x}_t + \mathbf{u}_t, \boldsymbol{\xi}_t)$.

To verify the equivalence of SGD and Perturbed SGD, we need to ensure the same noise levels and the number of steps for both algorithms. We briefly describe how this is achieved for finite-sum implementation of Perturbed SGD described in eqrefeq:structuredsmoothing.

For (27) , the perturbation step and the gradient step have 3 times the noise of SGD, as the perturbation step has 2 times the noise of SGD. To ensure the same noise levels, we set the batch size for both steps as $128 \times 3 = 384$. To ensure the same number of steps as SGD in one epoch, we repeat perturbation + gradient step 3 times in each epoch.

From Fig 6, we can see that the efficient finite-sum implementation of Perturbed SGD and SGD have very similar trajectories for training accuracy, training loss and validation accuracy. This verifies our claim of equivalence of SGD and Perturbed SGD on DL examples, with the same noise levels. Moreover, the variance is higher for Perturbed SGD than SGD, despite similar gradient noise level, providing further motivation for how adding perturbations to SGD can improve generalization and escape saddles [24].