How to Escape Sharp Minima with Random Perturbations

Kwangjun Ahn [1 2] **Ali Jadbabaie** [1] **Suvrit Sra** [1 3]

Abstract

Modern machine learning applications have witnessed the remarkable success of optimization algorithms that are designed to find flat minima. Motivated by this design choice, we undertake a formal study that (i) formulates the notion of flat minima, and (ii) studies the complexity of finding them. Specifically, we adopt the trace of the Hessian of the cost function as a measure of flatness, and use it to formally define the notion of approximate flat minima. Under this notion, we then analyze algorithms that find approximate flat minima efficiently. For general cost functions, we discuss a gradient-based algorithm that finds an approximate flat local minimum efficiently. The main component of the algorithm is to use gradients computed from randomly perturbed iterates to estimate a direction that leads to flatter minima. For the setting where the cost function is an empirical risk over training data, we present a faster algorithm that is inspired by a recently proposed practical algorithm called sharpness-aware minimization, supporting its success in practice.

1. Introduction

In modern machine learning applications, the training loss function $f : \mathbb{R}^d \to \mathbb{R}$ to be optimized often has a continuum of local/global minima, and the central question is which minima lead to good prediction performance. Among many different properties for minima, "flatness" of minima has been a promising candidate extensively studied in the literature (Hochreiter and Schmidhuber, 1997; Keskar et al., 2017; Dinh et al., 2017; Dziugaite and Roy, 2017; Neyshabur et al., 2017; Sagun et al., 2017; Yao et al., 2018; Chaudhari et al., 2019; He et al., 2019; Mulayoff and Michaeli, 2020; Tsuzuku et al., 2020; Xie et al., 2021).

Recently, there has been a resurgence of interest in flat minima due to various advances in both empirical and theoretical domains. Motivated by the extensive research on flat minima, this work undertakes a formal study that

(i) delineates a clear definition for flat minima, and
(ii) studies the upper complexity bounds of finding them.

We begin by emphasizing the significance of flat minima, based on recent advancements in the field.

1.1. Why Flat Minima?

Several recent optimization methods that are explicitly designed to find flat minima have achieved substantial empirical success (Chaudhari et al., 2019; Izmailov et al., 2018; Foret et al., 2021; Wu et al., 2020; Zheng et al., 2021; Norton and Royset, 2021; Kaddour et al., 2022). One notable example is sharpness-aware minimization (SAM) (Foret et al., 2021), which has shown significant improvements in prediction performance of deep neural network models for image classification problems (Foret et al., 2021) and language processing problems (Bahri et al., 2022). Furthermore, research by Liu et al. (2023) indicates that for language model pretraining, the flatness of minima serves as a more reliable predictor of model efficacy than the pretraining loss itself, particularly when the loss approaches its minimum values.

Complementing the empirical evidence, recent theoretical research underscores the importance of flat minima as a desirable attribute for optimization. Key insights include:

- *Provable generalization of flat minima.* For overparameterized models, research by Ding et al. (2022) demonstrates that flat minima correspond to the true solutions in low-rank matrix recovery tasks, such as matrix/bilinear sensing, robust PCA, matrix completion, and regression with a single hidden layer neural network, leading to better generalization. This is further extended by Gatmiry et al. (2023) to deep linear networks learned from linear measurements. In other words, in a range of nonconvex problems with multiple minima, flat minima yield superior predictive performance.
- *Benifits of flat minima in pretraining.* Along with the empirical validations, Liu et al. (2023) prove that in simplified masked language models, flat minima correlate with the most generalizable solutions.
- *Inductive bias of algorithms towards flat minima.* It

[1]MIT [2]Microsoft Research [3]TU Munich. Correspondence to: Kwangjun Ahn <kjahn@mit.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

has been proved that various practical optimization algorithms inherently favor flat minima. This includes stochastic gradient descent (SGD) (Blanc et al., 2020; Wang et al., 2022; Damian et al., 2021; Li et al., 2022; Liu et al., 2023), gradient descent (GD) with large learning rates (Arora et al., 2022; Damian et al., 2023; Ahn et al., 2023), sharpness-aware minimization (SAM) (Bartlett et al., 2022; Wen et al., 2022; Compagnoni et al., 2023; Dai et al., 2023), and a communication-efficient variant of SGD (Gu et al., 2023). The practical success of these algorithms indicates that flatter minima might be linked to better generalization properties.

Motivated by such recent advances, the main goal of this work is to initiate a formal study of the behavior of algorithms for finding flat minima, especially an understanding of their upper complexity bounds.

1.2. Overview of Our Main Results

In this work, we formulate a particular notion of flatness for minima and design efficient algorithms for finding them. We adopt the trace of Hessian of the loss $\mathrm{tr}(\nabla^2 f(\boldsymbol{x}))$ as a measure of "flatness," where lower values of the trace imply flatter regions within the loss landscape. The reasons governing this choice are many, especially its deep relevance across a rich variety of research, as summarized in Subsection 2.1. With this metric, we characterize a flat minimum as a local minimum where any local enhancement in flatness would result in an increased cost, effectively delineating regions where the model is both stable and efficient in terms of performance. More formally, we define the notion of (ϵ, ϵ')-flat minima in Definition 3. See Section 2 for precise details.

Given the notion of flat minima, the main goal of this work is to design algorithms that find an approximate flat minimum efficiently. At first glance, the goal of finding a flat minimum might seem computationally expensive because minimizing $\mathrm{tr}(\nabla^2 f)$ would require information about second or higher derivatives. Notably, this work demonstrates that one can reach a flat minimum using only first derivatives (gradients).

- In Section 3, we present a gradient-based algorithm called the randomly smoothed perturbation algorithm (Algorithm 1) which finds a $(\epsilon, \sqrt{\epsilon})$-flat minimum within $\mathcal{O}\left(\epsilon^{-3}\right)$ iterations for general costs without structure (Theorem 1). The main component of the algorithm is to use gradients computed from randomly perturbed iterates to estimate a direction that leads to flatter minima.
- In Section 4, we consider the setting where f is the training loss over a training data set and the initialization is near the set of global minima, motivated by overparametrized models in practice (Zhang et al., 2021). In such a setting, we present another gradient-based algorithm called the sharpness-aware perturbation algorithm (Algorithm 2), inspired by sharpness-aware minimization (SAM) (Foret et al., 2021). We show that this algorithm finds a $(\epsilon, \sqrt{\epsilon})$-flat minimum within $\mathcal{O}\left(d^{-1}\epsilon^{-2}(1 \vee \frac{1}{d^3\epsilon})\right)$ iterations (Theorem 2) – here d denotes the dimension of the domain. This demonstrates that a practical algorithm like SAM can find flat minima much faster than the randomly smoothed perturabtion algorithm in high dimensional settings.

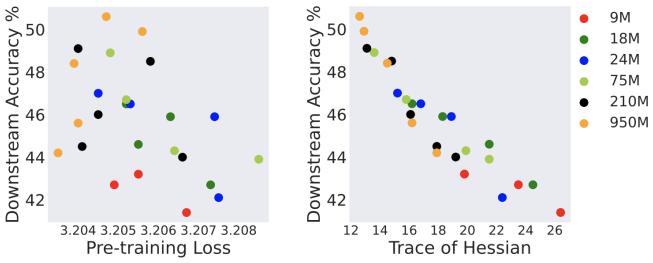

Figure 1. Figure from Liu et al. (2023). They pretrain language models for probabilistic context-free grammar with different optimization methods, and compare their downstream accuracy. As shown in the plot, the trace of Hessian is a better indicator of the performance than the pretraining loss itself.

See Table 1 for a high level summary of our results.

2. Formulating Flat Minima

In this section, we formally define the notion of flat minima.

2.1. Measure of Flatness

Within the literature reviewed in Subsection 1.1, a recurring metric for evaluating "flatness" in loss landscapes is the trace of the Hessian matrix of the loss function $\mathrm{tr}(\nabla^2 f(\boldsymbol{x}))$. This metric intuitively reflects the curvature of the loss landscape around minima, where the Hessian matrix $\nabla^2 f(\boldsymbol{x})$ is expected to be positive semi-definite. Consequently, lower values of the trace indicate regions where the loss landscape is flatter. For simplicity, we will refer to this metric as *the trace of Hessian*. Key insights from recent research include:

- *Overparameterized low-rank matrix recovery.* In this domain, the trace of Hessian has been identified as the *correct notion of flatness*. Ding et al. (2022) show that the most desirable minima, which correspond to ground truth solutions, are those with the lowest trace of Hessian values. This principle is also applicable to the analysis of deep linear networks, as highlighted by Gatmiry et al. (2023), where the same measure plays a pivotal role.
- *Language model pretraining.* The importance of the trace of Hessian extends to language model pretraining, as demonstrated by Liu et al. (2023). More specifically,

Setting	Iterations for $(\epsilon, \sqrt{\epsilon})$-flat minima (Definition 3)	Algorithm
General loss	$\mathcal{O}\left(\epsilon^{-3}\right)$ **gradient queries** (Theorem 1)	**Randomly Smoothed Perturbation** (Algorithm 1)
Training loss (Setting 1)	$\mathcal{O}\left(d^{-1}\epsilon^{-2}(1 \vee \frac{1}{d^3\epsilon})\right)$ **gradient queries** (Theorem 2)	**Sharpness-Aware Perturbation** (Algorithm 2)

Table 1. A high level summary of the main results with emphasis on the dependence on d and ϵ.

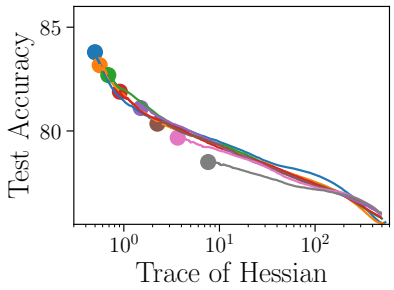

Figure 2. Figure from Damian et al. (2021). For training ResNet-18 on CIFAR 10, they measure the trace of Hessian across the iterates of SGD with label noise and have observed an inspiring relation between $\mathrm{tr}(\nabla^2 f(\boldsymbol{x}_t))$ and prediction performance.

Liu et al. (2023) conduct an insightful experiment (see Figure 1) that demonstrates the effectiveness of the trace of Hessian as a good measure of model performance. This observation is backed by their theoretical results for simple language models.

- *Model output stability.* Furthermore, the work (Ma and Ying, 2021) links the trace of Hessian to the stability of model outputs in deep neural networks relative to input data variations. This relationship underscores the significance of the trace of Hessian in improving model generalization and enhancing adversarial robustness.
- *Practical optimization algorithms.* Lastly, various practical optimization algorithms are shown to be inherently biased toward achieving lower values of the trace of Hessian. This includes SGD with label noise, as discussed in works by Blanc et al. (2020); Damian et al. (2021); Li et al. (2022), and without label noise for the language modeling pretraining (Liu et al., 2023). In particular, Damian et al. (2021) conduct an inspiring experiment showing a strong correlation between the trace of Hessian and the prediction performance of models (see Figure 2). Additionally, stochastic SAM is proven to prefer lower trace of Hessian values (Wen et al., 2022).

Remark 1 (Other notions of flatness?). *Perhaps, another popular notion of flat minima in the literature is the maximum eigenvalue of Hessian $\lambda_{\max}(\nabla^2 f(\boldsymbol{x}))$. However, recent empirical works (Kaur et al., 2023; Andriushchenko et al., 2023) have shown that the maximum eigenvalue of Hessian has limited correlation with the goodness of models (*e.g.*, generalization). On the other hand, as we detailed above, the trace of Hessian has been consistently brought up as a promising candidate, both* **theoretically and empirically**. *Hence, we adopt the trace of Hessian as the measure of flatness throughout.*

2.2. Formal Definition of Flat Minima

Motivated by the previous works discussed above, we adopt the trace of Hessian as the measure of flatness. Specifically, we consider the (normalized) trace of Hessian $\mathrm{tr}(\nabla^2 f(\boldsymbol{x}))/\mathrm{tr}(I_d) = \mathrm{tr}(\nabla^2 f(\boldsymbol{x}))/d$. Here we use the normalization to match the scale of flatness with the loss. For simplicity, we henceforth use the following notation:

$$\boxed{\overline{\mathsf{tr}}(x) := \frac{\mathrm{tr}(\nabla^2 f(\boldsymbol{x}))}{\mathrm{tr}(I_d)} = \frac{\mathrm{tr}(\nabla^2 f(\boldsymbol{x}))}{d}.}$$

The reason we consider the normalized trace is to match its scale with that of loss $f(\boldsymbol{x})$: the trace is in general the sum of d second derivatives, so it's scale is d times of that of $f(\boldsymbol{x})$. Also, the normalization can be potentially beneficial in practice where models have different sizes. Larger models would typically have a higher trace of Hessian due to having more parameters, and the normalization could put them on the same scale.

Given this choice, our notion of flat minima at a high level is *a local minimum (of f) for which one cannot locally decrease $\overline{\mathsf{tr}}$ without increasing the cost f*. In particular, this concept becomes nontrivial when the set of local minima is connected (or locally forms a manifold), which is indeed the case for the over-parametrized neural networks, as shown empirically in (Draxler et al., 2018; Garipov et al., 2018) and theoretically in (Cooper, 2021).

One straightforward way to define a (locally) flat minimum is the following: a local minimum which is also a local minimum of $\overline{\mathsf{tr}}$. However, this definition is not well-defined as the set of local minima of f can be disjoint from that of $\overline{\mathsf{tr}}$ as shown in the following example.

Example 1. *Consider a two-dimensional function f :*

$(x_1, x_2) \mapsto (x_1 x_2 - 1)^2$. *Then it holds that*

$$\nabla f(\boldsymbol{x}) = 2(x_1 x_2 - 1) \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} \quad \textit{and}$$

$$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} 2x_2^2 & 4x_1x_2 - 2 \\ 4x_1x_2 - 2 & 2x_1^2 \end{bmatrix}.$$

Hence, the set of minima is $\mathcal{X}^\star = \{(x_1, x_2) : x_1 x_2 = 1\}$ *and* $\overline{\mathsf{tr}}(\boldsymbol{x}) = (2x_1^2 + 2x_2^2)/2 = x_1^2 + x_2^2$. *The unique minimum of* $\overline{\mathsf{tr}}$ *is* $(0, 0)$ *which does not intersect with* $\mathcal{X}^\star$. *When restricted to* $\mathcal{X}^\star$, $\overline{\mathsf{tr}}$ *achieve its minimum at* $(1, 1)$ *and* $(-1, -1)$, *so those two points are flat minima.*

Hence, we consider the local optimality of $\overline{\mathsf{tr}}$ restricted to the set of local minima $\mathcal{X}^\star$. In practice, finding local minima with respect to $\overline{\mathsf{tr}}$ might be too stringent, so as an initial effort, we set our goal to find *a local minimum that is also a stationary point of* $\overline{\mathsf{tr}}$ *restricted to the set of local minima*. To formalize this, we introduce the limit map under the gradient flow, following (Li et al., 2022; Arora et al., 2022; Wen et al., 2022).

Definition 1 (Limit point under gradient flow)**.** *Given a point* $\boldsymbol{x}$, *let* $\Phi(\boldsymbol{x})$ *be the limiting point of the gradient flow on* f *starting at* $\boldsymbol{x}$. *More formally, letting* $\boldsymbol{x}(t)$ *be the iterate at time* t *of the gradient flow starting at* $\boldsymbol{x}$, *i.e.,* $\boldsymbol{x}(0) = \boldsymbol{x}$ *and* $\dot{\boldsymbol{x}}(t) = -\nabla f(\boldsymbol{x}(t))$, $\Phi(\boldsymbol{x})$ *is defined as* $\lim_{t\to\infty} \boldsymbol{x}(t)$.

The intuition behind such a definition is the following. Since we are focusing on the first-order optimization algorithms that has access to gradients of f, the natural notion of optimality is the local optimality. In other words, we want to ensure that at a flat minimum, locally deviating away from the minimum will either increase the loss or the trace of Hessian. This condition precisely corresponds to $[\nabla(\overline{tr} \circ \Phi)](x^\star) = 0$, since Φ maps each point x to its "closest" local minimum.

When $\boldsymbol{x}$ is near a set of local minima, $\Phi(\boldsymbol{x})$ is approximately equal to the projection onto the local minima set. Thus, the trace of Hessian $\overline{\mathsf{tr}}$ along the manifold can be captured by the functional $\overline{\mathsf{tr}}(\Phi(x))$. Therefore, we say a local minimum $\boldsymbol{x}^\star$ is a stationary point of $\overline{\mathsf{tr}}$ restricted to $\mathcal{X}^\star$ if

$$\nabla\left[\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}^\star))\right] = \partial\Phi(\boldsymbol{x}^\star)\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}^\star)) = \mathbf{0}\,.$$

In particular, if $\nabla\left[\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}^\star))\right] \neq \mathbf{0}$, moving along the direction of $-\nabla\left[\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}^\star))\right]$ will locally decrease the value $\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}))$ while staying within the set of minima, hence leading to a flatter minimum. Moreover, if $\boldsymbol{x}^\star$ is an isolated local minimum, then $\partial\Phi(\boldsymbol{x}^\star) = \mathbf{0}$, and hence $\nabla\left[\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}^\star)))\right] = \mathbf{0}$. This leads to the following definition.

Definition 2 (**Flat local minima**)**.** *We say a point* $\boldsymbol{x}$ *is a flat local minimum if it is a local minimum,* i.e., $\boldsymbol{x} \in \mathcal{X}^\star$, *and satisfies*

$$\nabla\left[\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}))\right] = \partial\Phi(\boldsymbol{x})\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x})) = \mathbf{0}\,. \tag{1}$$

Again, the intuition behind Definition 2 is that we want to ensure that at a flat minimum, locally deviating away from the minimum will either increase the loss or the trace of Hessian. This condition precisely corresponds to (1), since Φ maps each point x to its "closest" local minimum.

Having defined the notion of flat local minima, we define an approximate version of them such that we can discuss the iteration complexity of finding them.

Definition 3 (**(ϵ, ϵ')-flat local minima**)**.** *We say a point* $\boldsymbol{x}$ *is an* (ϵ, ϵ')*-flat local minimum if for* $\boldsymbol{x}^\star = \Phi(\boldsymbol{x})$, *it satisfies*

$$\|\boldsymbol{x} - \boldsymbol{x}^\star\| \le \epsilon \quad \textit{and} \quad \|[\nabla(\overline{\mathsf{tr}} \circ \Phi)](\boldsymbol{x}^\star)\| \le \epsilon'\,.$$

In other words, a $(0, 0)$*-flat local minimum is a flat local minimum.*

3. Randomly Smoothed Perturbation Escapes Sharp Minima

In this section, we present a gradient-based algorithm for finding an approximate flat minimum. We first discuss the setting for our analysis.

In order for our notion of flat minima (Definition 3) to be well defined, we assume that the loss function is four times continuously differentiable near the local minima set $\mathcal{X}^\star$. More formally, we make the following assumption about the loss function.

Assumption 1 (Loss near minima)**.** *There exists* $\zeta > 0$ *such that within* ζ*-neighborhood of the set of local minima* $\mathcal{X}^\star$, *the following properties hold:*

(a) f *is four-times continuously differentiable.*
(b) The limit map under gradient flow Φ *(Definition 1) is well-defined and is twice Lipschitz differentiable. Also,* $\Phi(\boldsymbol{x}) \in \mathcal{X}^\star$ *and the gradient flow starting at* $\boldsymbol{x}$ *is contained within the* ζ*-neighborhood of* $\mathcal{X}^\star$.
(c) The Polyak–Łojasiewicz (PL) inequality holds locally, i.e., $f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})) \le \frac{1}{2\alpha}\|\nabla f(\boldsymbol{x})\|^2$.

It fact, the last two conditions (b), (c) are consequences of f being four-times continuously differentiable (Arora et al., 2022, Appendix B). We include them for concreteness.

We also discuss a preliminary step for our analysis. Since the question of finding candidates for approximate local minima (or second order stationary points) is well-studied, thanks to the vast literature on the topic over the last decade (Ge et al., 2015; Agarwal et al., 2017; Carmon et al., 2018; Fang et al., 2018; Jin et al., 2021), we do not further explore it, but single out the question of seeking flatness by assuming that the initial iterate $\boldsymbol{x}_0$ is already close to the set of local minima $\mathcal{X}^\star$. For instance, assuming that the loss f satisfies strict saddle properties (Ge et al., 2015; Jin et al., 2017), one can find a point $\boldsymbol{x}_0$ that satisfies $\|\nabla f(\boldsymbol{x}_0)\| \le \mathcal{O}(\epsilon)$ within

$\tilde{\mathcal{O}}(\epsilon^{-2})$iterations. Now thanks to Assumption 1, since we assume f to be four-times continuously differentiable, it follows that $\|\boldsymbol{x}_0 - \Phi(\boldsymbol{x}_0)\| \leq \mathcal{O}(\|\nabla f(\boldsymbol{x}_0)\|) \leq \mathcal{O}(\epsilon)$. Hence, we will often start our analysis with the initialization that is sufficiently close to the set of local minima $\mathcal{X}^\star$.

We also define the following notation, which we will utilize throughout.

Definition 4 (Projecting-out operator)**.** *For two vectors $\boldsymbol{u}, \boldsymbol{v}$, $\mathrm{Proj}_{\boldsymbol{u}}^{\perp}\boldsymbol{v}$ is the "projecting-out" operator, i.e.,*

$$\mathrm{Proj}_{\boldsymbol{u}}^{\perp}\boldsymbol{v} := \boldsymbol{v} - \left\langle \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}, \boldsymbol{v} \right\rangle \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}.$$

3.1. Main Result

Under this setting, we present a gradient-based algorithm for finding approximate flat minima and its theoretical guarantees. Our proposed algorithm is called the randomly smoothed perturbation algorithm (Algorithm 1). The main component of the algorithm is the perturbed gradient step that is employed whenever the gradient norm is smaller than a tolerance ϵ_0:

$$\begin{aligned} \boldsymbol{x}_{t+1} &\leftarrow \boldsymbol{x}_t - \eta\left(\nabla f(\boldsymbol{x}_t) + \boldsymbol{v}_t\right), \quad \text{where} \\ \boldsymbol{v}_t &:= \mathrm{Proj}_{\nabla f(\boldsymbol{x}_t)}^{\perp} \nabla f(\boldsymbol{x}_t + \rho \boldsymbol{g}_t) \end{aligned} \tag{2}$$

Here $\boldsymbol{g}_t \sim \mathrm{Unif}(\mathbb{S}^{d-1})$ is a random unit vector. At a high level, (2) adds a perturbation direction $\boldsymbol{v}_t$ to the ordinary gradient step, where the perturbation direction $\boldsymbol{v}_t$ is computed using gradients at a randomly perturbed iterate $\boldsymbol{x}_t + \rho \boldsymbol{g}_t$ and then projecting out the gradient $\nabla f(\boldsymbol{x}_t)$. The gradient of a randomly perturbed iterate $\nabla f(\boldsymbol{x}_t + \rho \boldsymbol{g}_t)$ can be also interpreted as the (stochastic) gradient of widely known *randomized smoothing* of f (hence its name "randomly smoothed perturbation")—a widely known technique for nonsmooth optimization (Duchi et al., 2012). In some sense, this work discovers a new property of randomized smoothing for nonconvex optimization: *randomized smoothing seeks flat minima!* We now present the theoretical guarantee of Algorithm 1.

Theorem 1. *Let Assumption 1 hold and f have β-Lipschitz gradients. Let the target accuracy $\epsilon > 0$ be chosen sufficiently small, and $\delta \in (0, 1)$. Suppose that $\boldsymbol{x}_0$ is ζ-close to $\mathcal{X}^\star$. Then, the randomly smoothed perturbation algorithm (Algorithm 1) with parameters $\eta = \mathcal{O}(\delta\epsilon)$, $\eta' = 1/\beta$, $\rho = \mathcal{O}(\delta\sqrt{\epsilon})$, $\epsilon_0 = \mathcal{O}\left(\delta^{1.5}\epsilon\right)$ returns an $(\epsilon, \sqrt{\epsilon})$-flat minimum with probability at least $1 - \mathcal{O}(\delta)$ after $T = \mathcal{O}\left(\epsilon^{-3}\delta^{-4}\right)$ iterations.*

[1]The uniformly chosen iterate is for the sake of analysis, and it's a standard approach often used in convergence to stationary points analysis. See, *e.g.*, (Ghadimi and Lan, 2013; Reddi et al., 2016).

Algorithm 1 Randomly Smoothed Perturbation

Input: $\boldsymbol{x}_0$, learning rates η, η', perturbation radius ρ, tolerance ϵ_0, the number of steps T.
for $t = 0, 1, \ldots, T-1$ **do**
  if $\|\nabla f(\boldsymbol{x}_t)\| \leq \epsilon_0$ **then**
    $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t - \eta\left(\nabla f(\boldsymbol{x}_t) + \boldsymbol{v}_t\right)$,
      where $\boldsymbol{v}_t := \mathrm{Proj}_{\nabla f(\boldsymbol{x}_t)}^{\perp} \nabla f(\boldsymbol{x}_t + \rho \boldsymbol{g}_t)$
    and $\boldsymbol{g}_t \sim \mathrm{Unif}(\mathbb{S}^{d-1})$
  else
    $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t - \eta' \nabla f(\boldsymbol{x}_t)$
  end if
end for
return $\widehat{\boldsymbol{x}}$ uniformly at random[1] from $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$

Minimizing flatness only using gradients? At first glance, finding a flat minimum seems computationally expensive since minimizing $\mathrm{tr}(\nabla^2 f)$ would require information about second or higher derivatives. Thus, Theorem 1 may sound quite surprising to some readers since Algorithm 1 only uses gradients which only pertains to information about first derivatives.

However, it turns out using the gradients from the perturbed iterates $\boldsymbol{x}_t + \rho \boldsymbol{g}_t$ lets us get access to specific third derivatives of f in a parsimonious way. More precisely, as we shall see in our proof sketch, the crux of the perturbation step (2) is that the gradients of $\overline{\mathsf{tr}}$ can be estimated using gradients from perturbed iterates. In particular, we show that (see (5)) in expectation, it holds that

$$\mathbb{E}\boldsymbol{v}_t = \frac{1}{2}\rho^2 \mathrm{Proj}_{\nabla f(\boldsymbol{x}_t)}^{\perp} \nabla \overline{\mathsf{tr}}(\boldsymbol{x}_t) + \text{lower order terms.}$$

Using this property, one can prove that each step of the perturbed gradient step decrease the trace of Hessian along the local minima set; see Lemma 2. We remark that this general principle of estimating higher order derivatives from gradients in a parsimonious way is inspired by recent works on understanding dynamics of sharpness-aware minimization (Bartlett et al., 2022; Wen et al., 2022) and gradient descent at edge-of-stability (Arora et al., 2022; Damian et al., 2023).

3.2. Proof Sketch of Theorem 1

In this section, we provide a proof sketch of Theorem 1, The full proof can be found in Appendix B. We first sketch the overall structure of the proof and then detail each part:

1. We first show that iterates enters an $\mathcal{O}(\epsilon_0)$-neighborhood of the local minima set $\mathcal{X}^\star$ in a few steps, and the subsequent iterates remain there.
2. When the iterates is $\mathcal{O}(\epsilon_0)$-near $\mathcal{X}^\star$, we show that the perturbed gradient step in Algorithm 1 decreases the trace of Hessian $\overline{\mathsf{tr}}(\Phi)$ in expectation as long as

$\|\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\| \geq \sqrt{\epsilon}$.

3. We then combine the above two properties to show that Algorithm 1 finds a flat minimum.

Perturbation does not increase the cost too much. First, since $\boldsymbol{x}_0$ is ζ-close to $\mathcal{X}^\star$ where the loss function satisfies the Polyak–Łojasiewicz (PL) inequality, the standard linear convergence result of gradient descent guarantees that the iterate enters an $\mathcal{O}(\epsilon_0)$-neighborhood of $\mathcal{X}^\star$. We thus assume that $\boldsymbol{x}_0$ itself satisfies $\|\nabla f(\boldsymbol{x}_0)\| \leq \epsilon_0$ without loss of generality. We next show that the perturbation $\boldsymbol{v}_t$ we add at each step to the gradient only leads to a small increase in the cost. This claim follows from the following variant of well-known descent lemma.

Lemma 1. *For $\eta \leq 1/\beta$, consider a one-step of the perturbed gradient step of Algorithm 1: $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t - \eta(\nabla f(\boldsymbol{x}_t) + \boldsymbol{v}_t)$. Then we have*

$$f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t) - \frac{1}{2}\eta\|\nabla f(\boldsymbol{x}_t)\|^2 + \frac{\beta\eta^2}{2}\|\boldsymbol{v}_t\|^2 .$$

The proof of Lemma 1 uses the fact that $\boldsymbol{v}_t \perp \nabla f(\boldsymbol{x}_t)$. Now, with the β-Lipschitz gradient condition, one can show that $\|\boldsymbol{v}_t\| = \mathcal{O}(\rho)$. Hence, whenever the gradient becomes large as $\|\nabla f(\boldsymbol{x}_t)\| \gtrsim \eta\rho^2$, the perturbed update starts decreasing the loss again and brings the iterates back close to $\mathcal{X}^\star$. Using this property, one can show that the iterates $\boldsymbol{x}_t$ remain in an ϵ_0-neighborhood of $\mathcal{X}^\star$, i.e., $\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\| = \mathcal{O}(\|\nabla f(\boldsymbol{x}_t)\|) = \mathcal{O}(\epsilon_0)$. See Lemma 8 for precise details.

Perturbation step decreases $\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))$ in expectation. Now the main part of the analysis is to show that the perturbation updates lead to decrease in the trace Hessian along $\mathcal{X}^\star$, i.e., decrease in $\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))$, as show in the following result.

Lemma 2. *Let Assumption 1 hold. Let the target accuracy $\epsilon > 0$ be chosen sufficiently small, and $\delta \in (0,1)$. Consider the perturbed gradient step of Algorithm 1, i.e., $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t - \eta(\nabla f(\boldsymbol{x}_t) + \boldsymbol{v}_t)$ starting from $\boldsymbol{x}_t$ such that $\|\nabla f(\boldsymbol{x}_t)\| \leq \epsilon_0$ with parameters $\eta = \mathcal{O}(\delta\epsilon)$, $\rho = \mathcal{O}(\delta\sqrt{\epsilon})$ and $\epsilon_0 = \mathcal{O}(\delta^{1.5}\epsilon)$. Assume that $\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))$ has sufficiently large gradient*

$$\|\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\| \geq \sqrt{\epsilon} .$$

Then the trace of Hessian $\overline{\mathsf{tr}}(\Phi)$ decreases as

$$\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \leq -\Omega(\delta^3\epsilon^3) , \tag{3}$$

where the expectation is over the perturbation $\boldsymbol{g}_t \sim \mathrm{Uniform}(\mathbb{S}^{d-1})$ in Algorithm 1.

Proof sketch of Lemma 2: We begin with the Taylor expansion of the perturbed gradient:

$$\nabla f(\boldsymbol{x}_t + \rho\boldsymbol{g}_t) = \nabla f(\boldsymbol{x}_t) + \rho\nabla^2 f(\boldsymbol{x}_t)\boldsymbol{g}_t + \frac{1}{2}\rho^2\nabla^3 f(\boldsymbol{x}_t)[\boldsymbol{g}_t, \boldsymbol{g}_t] + \mathcal{O}(\rho^3) . \tag{4}$$

Now let us compute the expectation of the projected out version of the perturbed gradient, i.e., $\mathbb{E}\mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}\nabla f(\boldsymbol{x}_t + \rho\boldsymbol{g}_t)$. First, note that in (4), the projection operator removes $\nabla f(\boldsymbol{x}_t)$, and using the fact $\mathbb{E}[\boldsymbol{g}_t] = \mathbf{0}$, the second term $\rho\nabla^2 f(\boldsymbol{x}_t)\boldsymbol{g}_t$ also vanishes in expectation. Turning to the third term, an interesting thing happens. Since $\mathbb{E}[\boldsymbol{g}_t\boldsymbol{g}_t^\top] = \frac{1}{d}\mathbf{I}_d$, using the fact $\nabla^3 f(\boldsymbol{x}_t)[\boldsymbol{g}_t, \boldsymbol{g}_t] = \nabla(\nabla^2 f(\boldsymbol{x}_t)[\boldsymbol{g}_t, \boldsymbol{g}_t]) = \nabla\,\mathrm{tr}\left(\nabla^2 f(\boldsymbol{x}_t)\boldsymbol{g}_t\boldsymbol{g}_t^\top\right)$, it follows that

$$\mathbb{E}\boldsymbol{v}_t = \frac{1}{2}\rho^2\mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}\nabla\overline{\mathsf{tr}}(\boldsymbol{x}_t) + \mathcal{O}(\rho^3) , \tag{5}$$

Now, with the high-order smoothness properties of f, we obtain

$$\begin{aligned}
&\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \\
&\quad = \overline{\mathsf{tr}}(\Phi(\Phi(\boldsymbol{x}_{t+1}))) - \overline{\mathsf{tr}}(\Phi(\Phi(\boldsymbol{x}_t))) \\
&\quad \leq \langle\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)), \Phi(\boldsymbol{x}_{t+1}) - \Phi(\boldsymbol{x}_t)\rangle \\
&\qquad + \mathcal{O}\left(\|\Phi(\boldsymbol{x}_{t+1}) - \Phi(\boldsymbol{x}_t)\|^2\right) .
\end{aligned}$$

Using (5) and carefully bounding terms, one can prove the following upper bound on $\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))$: ($\boldsymbol{\nabla}_t \coloneqq \partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))$)

$$-\eta\rho^2\|\boldsymbol{\nabla}_t\|^2 + O\left(\eta\rho^2\epsilon_0\|\boldsymbol{\nabla}_t\| + \eta\rho^3\|\boldsymbol{\nabla}_t\| + \eta^2\rho^2\right) \tag{6}$$

The inequality (3) implies that as long as $\|\boldsymbol{\nabla}_t\| \geq \Omega(\max\{\epsilon_0, \rho, \sqrt{\eta}\})$, $\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))$ decreases in expectation by $\eta\rho^2\|\boldsymbol{\nabla}_t\|^2$. Due to our choices of ρ, η, ϵ_0, Lemma 2 follows. □

Using similar argument, one can show that the perturbation step does not increase the trace Hessian value too much even when $\|\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\| \leq \sqrt{\epsilon}$.

Lemma 3. *Under the same setting as Lemma 2, assume now that $\|\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\| \leq \sqrt{\epsilon}$. Then we have $\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \leq \mathcal{O}(\delta^4\epsilon^3)$.*

Putting things together. Using the results so far, we establish a high probability result by returning one of the iterates uniformly at random, following (Ghadimi and Lan, 2013; Reddi et al., 2016; Daneshmand et al., 2018). For $t = 1, 2, \ldots, T$,

let A_t be the event $\|\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\| \geq \sqrt{\epsilon}$,

and Let P_t denote the probability of event A_t. Then, the probability of returning a $(\epsilon, \sqrt{\epsilon})$-flat minimum is simply

equal to $\frac{1}{T}\sum_{t=1}^{T}(1-P_t)$. It turns out one can upper bound the sum of P_t's using Lemma 2; see Appendix B for details. In particular, choosing $T=\Omega\left(\epsilon^{-3}\delta^{-4}\right)$, we get

$$\frac{1}{T}\sum_{t=1}^{T}P_t \lesssim \frac{\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_0))}{T\delta^3\epsilon^3}+\delta=\mathcal{O}\left(\delta\right).$$

This concludes the proof of Theorem 1.

4. Faster Escape with Sharpness-Aware Perturbation

In this section, we present another gradient-based algorithm for finding an approximate flat minima for the case where the loss f is a training loss over a training data set. More formally, we consider the following setting for training loss, following the one in (Wen et al., 2022).

Setting 1 (Training loss over data)**.** *Let n be the number of training data, and for $i=1,\ldots,n$, let $p_i(\boldsymbol{x})$ be the model prediction output on the i-th data, and y_i be the i-th label. For a loss function ℓ, let f be defined as the following training loss*

$$f(\boldsymbol{x})=\frac{1}{n}\sum_{i=1}^{n}f_i(\boldsymbol{x})\coloneqq\frac{1}{n}\sum_{i=1}^{n}\ell(p_i(\boldsymbol{x}),y_i).$$

Here ℓ satifies $\arg\min_{z\in\mathbb{R}}\ell(z,y)=y\ \forall y$, and $\frac{\partial^2\ell(z,y)}{\partial^2 z}|_{z=y}>0$. Lastly, we consider $\mathcal{X}^\star$ to be the set of global minima, i.e., $\mathcal{X}^\star=\{\boldsymbol{x}\in\mathbb{R}^d:p_i(\boldsymbol{x})=y_i,\forall i=1,\ldots,n\}$. We assume that $\nabla p_i(\boldsymbol{x})\neq\mathbf{0}$ for $\boldsymbol{x}\in\mathcal{X}^\star$, $\forall i=1,\ldots,n$.

We note that the assumption that $\nabla p_i(\boldsymbol{x})\neq\mathbf{0}$ for $\boldsymbol{x}\in\mathcal{X}^\star$ is without loss of generality. More precisely, by Sard's Theorem, $\mathcal{X}^\star$ defined above is just equal to the set of global minima, except for a measure-zero set of labels.

4.1. Main Result

Under Setting 1, we present another gradient-based algorithm for finding approximate flat minima (Algorithm 2). The main component of our proposed algorithm is the perturbed gradient step

$$\boldsymbol{x}_{t+1}\leftarrow\boldsymbol{x}_t-\eta\left(\nabla f(\boldsymbol{x}_t)+\boldsymbol{v}_t\right),\quad\text{where}\tag{7}$$
$$\boldsymbol{v}_t\coloneqq\mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}\nabla f_i\left(\boldsymbol{x}_t+\rho\sigma_t\frac{\nabla f_i(\boldsymbol{x}_t)}{\|\nabla f_i(\boldsymbol{x}_t)\|}\right)$$

for random samples $i\sim[n]$ and $\sigma_t\sim\{\pm1\}$.

Remark 2. *Here, note that the direction $\nabla f_i(\boldsymbol{x}_t)/\|\nabla f_i(\boldsymbol{x}_t)\|$ could be ill-defined when the stochastic gradient exactly vanishes at $\boldsymbol{x}_t$. In that case, one can use $\nabla f_i(\boldsymbol{x}_t+\boldsymbol{\xi})/\|\nabla f_i(\boldsymbol{x}_t+\boldsymbol{\xi})\|$ where $\boldsymbol{\xi}$ is a random vector with a small norm, say ϵ^3. Hence, to avoid tedious technicality, we assume for the remaining of the paper that* (7) *is well-defined at each step.*

Algorithm 2 Sharpness-Aware Perturbation

Input: $\boldsymbol{x}_0$, learning rates η,η', perturbation radius ρ, tolerance ϵ_0, the number of steps T.
for $t=0,1,\ldots,T-1$ **do**
  if $\|\nabla f(\boldsymbol{x}_t)\|\le\epsilon_0$ **then**
    $\boldsymbol{x}_{t+1}\leftarrow\boldsymbol{x}_t-\eta\left(\nabla f(\boldsymbol{x}_t)+\boldsymbol{v}_t\right)$,
    where $\boldsymbol{v}_t\coloneqq\mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}\nabla f_i(\boldsymbol{x}_t+\rho\sigma_t\frac{\nabla f_i(\boldsymbol{x}_t)}{\|\nabla f_i(\boldsymbol{x}_t)\|})$
    for $i\sim\mathrm{Unif}([n])$ and $\sigma_t\sim\mathrm{Unif}(\{\pm1\})$.
  else
    $\boldsymbol{x}_{t+1}\leftarrow\boldsymbol{x}_t-\eta'\nabla f(\boldsymbol{x}_t)$
  end if
end for
return $\widehat{\boldsymbol{x}}$ uniformly at random from $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T\}$

Notice the distinction between (2) and (7). In particular, for the randomly smoothed perturbation algorithm, $\boldsymbol{v}_t$ is computed using the gradient at a randomly perturbed iterate. On the other hand, in the update (7), $\boldsymbol{v}_t$ is computed using the *stochastic* gradient at an iterate perturbed along the *stochastic gradient direction*. The idea of computing the (stochastic) gradient at an iterate perturbed along the (stochastic) gradient direction is inspired by sharpenss-aware minimization (SAM) of Foret et al. (2021), a practical optimization algorithm showing substantial success in practice. Hence, we call our algorithm the *sharpness-aware perturbation* algorithm.

As we shall see in detail in Theorem 2, the sharpness-aware perturbation step (7) leads to an improved guarantee for finding a flat minimum. The key idea—as we detail in Subsection 4.2—is that this perturbation leads to faster decrease in $\overline{\mathsf{tr}}$. In particular, Lemma 4 shows that each sharpness-aware perturbation decreases $\overline{\mathsf{tr}}$ by $\Omega(d\min\{1,\epsilon d^3\}\cdot\delta^3\epsilon^2)$, which is $d\epsilon^{-1}\min\{1,\epsilon d^3\}$ times larger than the decrease of $\Omega(\delta^3\epsilon^3)$ due to the randomly smoothed perturbation (shown in Lemma 2). We now present the theoretical guarantee of Algorithm 2.

Theorem 2. *Under Setting 1, let Assumption 1 hold and each f_i is four times coutinuously differentiable within the ζ-neighborhood of $\mathcal{X}^\star$ and have β-Lipschitz gradients. Let the target accuracy $\epsilon>0$ be chosen sufficiently small, and $\delta\in(0,1)$. Suppose that $\boldsymbol{x}_0$ is ζ-close to $\mathcal{X}^\star$. Then, for $\nu\coloneqq\min\{d,\epsilon^{-1/3}\}$, the sharpness-aware perturbation algorithm (Algorithm 2) with parameters $\eta=\mathcal{O}\left(\nu\delta\epsilon\right)$, $\eta'=1/\beta$, $\rho=\mathcal{O}\left(\nu\delta\sqrt{\epsilon}\right)$, $\epsilon_0=\mathcal{O}\left(\nu^{1.5}\delta^{1.5}\epsilon\right)$ returns an $(\mathcal{O}\left(\epsilon_0\right),\sqrt{\epsilon})$-flat minimum with probability at least $1-\mathcal{O}\left(\delta\right)$ after $T=\mathcal{O}\left(d^{-1}\epsilon^{-2}\cdot\max\left\{1,\frac{1}{d^3\epsilon}\right\}\cdot\delta^{-4}\right)$ iterations. From this $(\mathcal{O}\left(\epsilon_0\right),\sqrt{\epsilon})$-flat minimum, gradient descent with step size $\eta=\mathcal{O}\left(\epsilon\right)$ reaches a $(\epsilon,\sqrt{\epsilon})$-flat minimum within $\mathcal{O}\left(\epsilon^{-1}\log(1/\epsilon)\right)$ iterations.*

Curious role of stochastic gradients. Some readers might wonder the role of stochastic gradients in (7)—for

instance, what happens if we replace them by *full-batch* gradients ∇f? Empirically, it has been observed that for SAM's performance, it is important to use stochastic gradients over full-batch (Foret et al., 2021; Kaddour et al., 2022; Kaur et al., 2023). Our analysis (see the proof sketch of Lemma 4) provides a partial explanation for the success of using stochastic gradients, from the perspective of finding flat minima. In particular, we show that stochastic gradients are important for faster decrease in the trace of the Hessian.

4.2. Proof Sketch of Theorem 2

In this section, we sketch a proof of Theorem 2; for the full proof please see Appendix C. First, similarly to the proof of Theorem 1, one can show that once $\boldsymbol{x}_t$ enters an $\mathcal{O}(\epsilon_0)$-neighborhood of $\mathcal{X}^\star$, all subsequent iterates $\boldsymbol{x}_t$ remain in the neighborhood. Now we sketch the proof of decrease in the trace of the Hessian.

Sharpness-aware perturbation decreases $\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))$ faster. Similarly to Lemma 2, the main part is to show that the trace of Hessian decreases during each perturbed gradient step.

Lemma 4. *Let Assumption 1 hold. Let the target accuracy $\epsilon > 0$ be chosen sufficiently small, and $\delta \in (0,1)$. Consider the perturbed gradient step of Algorithm 2, i.e., $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t - \eta\left(\nabla f(\boldsymbol{x}_t) + \boldsymbol{v}_t\right)$ starting from $\boldsymbol{x}_t$ such that $\|\nabla f(\boldsymbol{x}_t)\| \leq \epsilon_0$ with parameters $\eta = \mathcal{O}(\nu\delta\epsilon)$, $\rho = (\nu\delta\sqrt{\epsilon})$, $\epsilon_0 = \mathcal{O}\left(\nu^{1.5}\delta^{1.5}\epsilon\right)$. Assume that $\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))$ has sufficiently large gradient*

$$\|\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\| \geq \sqrt{\epsilon}\,.$$

Then the trace of Hessian $\overline{\mathsf{tr}}(\Phi)$ decreases as

$$\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \leq -\Omega(d\nu^3\delta^3\epsilon^3)\,,$$

where the expectation is over the random samples $i \sim \mathrm{Unif}([n])$ and $\sigma_t \sim \mathrm{Unif}(\{\pm 1\})$ in Algorithm 2.

Proof sketch of Lemma 4: For notational simplicity, let $\boldsymbol{g}_{i,t} := \frac{\nabla f_i(\boldsymbol{x}_t)}{\|\nabla f_i(\boldsymbol{x}_t)\|}$. To illustrate the main idea effectively, we make the simplifying assumption that for $\boldsymbol{x}^\star \in \mathcal{X}^\star$, the gradient of model outputs $\{\nabla p_i(\boldsymbol{x})\}_{i=1}^n$ are orthogonal; our full proof in Appendix C does not require this assumption. To warm-up, let us first consider the case where we use the *full-batch gradient* instead of the stochastic gradient for the outer part of the perturbation, i.e., consider

$$\boldsymbol{v}_t' := \mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}\nabla f(\boldsymbol{x}_t + \rho\sigma_t\boldsymbol{g}_{i,t})$$

Because $\mathbb{E}[\sigma_t\boldsymbol{g}_t] = \mathbf{0}$ (since $\mathbb{E}[\sigma_t] = 0$), a similar calculation as the proof of Lemma 2, we arrive at

$$\mathbb{E}\boldsymbol{v}_t' = \frac{1}{2}\rho^2\mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}\nabla\,\mathrm{tr}\left(\nabla^2 f(\boldsymbol{x}_t)\boldsymbol{g}_{i,t}\boldsymbol{g}_{i,t}^\top\right) + \mathcal{O}\left(\rho^3\right)\,.$$

Now the key observation, inspired by Wen et al. (2022), is that at a minimum $\boldsymbol{x}^\star \in \mathcal{X}^\star$, the Hessian is given as

$$\nabla^2 f(\boldsymbol{x}^\star) = \frac{1}{n}\sum_{i=1}^n \ell''(p_i(\boldsymbol{x}^\star), y_i)\nabla p_i(\boldsymbol{x}^\star)\nabla p_i(\boldsymbol{x}^\star)^\top\,.$$

Hence, due to our simplifying assumption for this proof sketch, namely the orthogonality of the gradient of model outputs $\{\nabla p_i(\boldsymbol{x}^\star)\}_{i=1}^n$, it follows that $\boldsymbol{u}_i(\boldsymbol{x}^\star) := \nabla p_i(\boldsymbol{x}^\star)/\|\nabla p_i(\boldsymbol{x}^\star)\|$ is the eigenvector of the Hessian. Let $\lambda_i(\boldsymbol{x}^\star)$ be the corresponding eigenvalue. Furthermore, we have $\nabla f_i(\boldsymbol{x}_t) = \frac{\partial\ell(z,y_i)}{\partial z}|_{z=p_i(\boldsymbol{x}_t)}\nabla p_i(\boldsymbol{x}_t)$, which implies $\frac{\nabla f_i(\boldsymbol{x}_t)}{\|\nabla f_i(\boldsymbol{x}_t)\|} = \frac{\nabla p_i(\boldsymbol{x}_t)}{\|\nabla p_i(\boldsymbol{x}_t)\|} = \boldsymbol{u}_i(\boldsymbol{x}_t)$ as long as $\nabla f_i(\boldsymbol{x}_t) \neq 0$. Hence, as long as $\boldsymbol{x}_t$ stays near $\Phi(\boldsymbol{x}_t)$, it follows that

$$\mathbb{E}\,\mathrm{tr}\left(\nabla^2 f(\boldsymbol{x}_t)\boldsymbol{g}_{i,t}\boldsymbol{g}_{i,t}^\top\right) \approx \mathbb{E}\lambda_i(\Phi(\boldsymbol{x}_t)) = \frac{d}{n}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\,,$$

which notably gives us $\frac{d}{n}$ times larger gradient than the randomly smoothed perturbation (5). On the other hand, one can do even better by choosing the stochastic gradient for the outerpart of perturbation. Similar calculations to the above yield

$$\mathbb{E}\,\mathrm{tr}\left(\nabla^2 f_i(\boldsymbol{x}_t)\boldsymbol{g}_{i,t}\boldsymbol{g}_{i,t}^\top\right) \approx n\mathbb{E}\lambda_i(\Phi(\boldsymbol{x}_t)) = d\cdot\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\,,$$

which now leads to d times larger gradient than (5). This leads to the following inequality that is an improvement of (6): $\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) - d\cdot\eta\rho^2\|\boldsymbol{\nabla}_t\|^2 + \mathcal{O}\left(d\eta\rho^2\epsilon_0\|\boldsymbol{\nabla}_t\| + \eta\rho^3\|\boldsymbol{\nabla}_t\| + \eta^2\rho^2\right)$. This inequality implies that as long as $\|\boldsymbol{\nabla}_t\| \geq \Omega(\max\{\epsilon_0, \rho/d, \sqrt{\eta/d}\})$, $\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))$ decreases in expectation by $d\eta\rho^2\|\boldsymbol{\nabla}_t\|^2$. Due to our choices of ρ, η, ϵ_0, Lemma 4 follows. □

Using Lemma 4, and following the analysis presented in Subsection 3.2, it can be shown that Algorithm 2 returns a $(\epsilon_0, \sqrt{\epsilon})$-flat minimum $\widehat{\boldsymbol{x}}$ with probability at least $1 - \mathcal{O}(\delta)$ after $T = \mathcal{O}\left(d^{-1}\epsilon^{-3}\nu^{-3}\delta^{-4}\right)$ iterations. From this $(\epsilon_0, \sqrt{\epsilon})$-flat minimum $\widehat{\boldsymbol{x}}$, one can find a $(\epsilon, \sqrt{\epsilon})$-flat minimum in a few iterations.

5. Experiments

We run experiments based on training ResNet-18 on the CIFAR10 dataset to test the ability of proposed algorithms to escape sharp global minima. Following (Damian et al., 2021), the algorithms are initialized at a point corresponding to a sharp global minimizer that achieve poor test accuracy. Crucially, we choose this setting because (Damian et al., 2021, Figure 1) verify that test accuracy is inversely correlated with the trace of Hessian (see Figure 2). This bad global minimizer, due to (Liu et al., 2020), achieves 100% training accuracy, but only 48% test accuracy. We choose the constant learning rate of $\eta = 0.001$, which is small

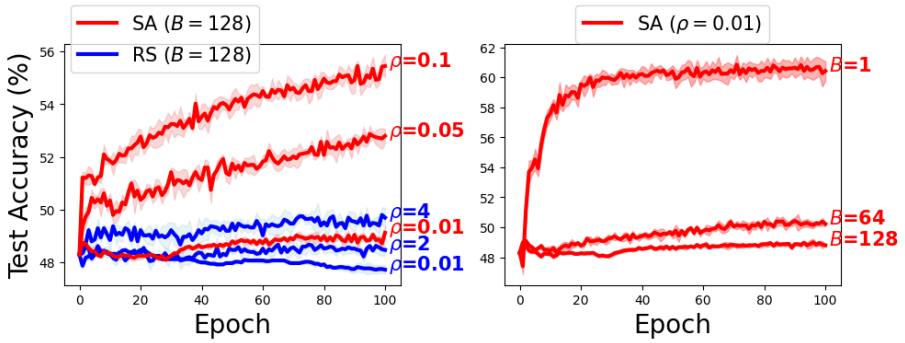

Figure 3. (Left) The comparison between Randomly Smoothed Perturbation ("RS") and Sharpness-Aware Perturbation ("SA"). (Right) Comparison of SA with different batch sizes. Here, we highlight that we do observe that **the trace of Hessian value monotonically decreases** along the algorithm iterates, similarly to (Damian et al., 2021) (see also Figure 2). We decide to present the test accuracy instead of the trace of Hessian, as it has more practical values.

enough such that SGD baseline without any perturbation does not escape.

We discuss the results one by one. First of all, we highlight that the training accuracy stays at 100% for all algorithms.

- **Comparison between two methods.** In the left plot of Figure 3, we compare the performance of Randomly Smoothed Perturbation ("RS") and Sharpness-Aware Perturbation ("SA"). We choose the batch size of 128 for both methods. Consistent with our theory, one can see that SA is more effective in escaping sharp minima even with a smaller perturbation radius ρ.
- **Different batch sizes.** Our theory suggests that batch size 1 should be effective in escaping sharp minima. We verify this in the right plot of Figure 3 by choosing the batch size to be $B = 1, 64, 128$. We do see that the case of $B = 1$ is quite effective in escaping sharp minima.

6. Related Work and Future Work

The last decade has seen a great success in theoretical studies on the question of finding (approximate) stationary points (Ghadimi and Lan, 2013; Ge et al., 2015; Agarwal et al., 2017; Daneshmand et al., 2018; Carmon et al., 2018; Fang et al., 2018; Allen-Zhu, 2018; Zhou and Gu, 2020; Jin et al., 2021). This work extends this line of research to a new notion of stationary point, namely an approximate flat minima. We believe that further studies on defining/refining practical notions of flat minima and designing efficient algorithms for them would lead to better understanding of practical nonconvex optimization for machine learning. In the same spirit, we believe that characterizing lower bounds would be of great importance, similar to the ones for the stationary points (Carmon et al., 2020; Drori and Shamir, 2020; Carmon et al., 2021; Arjevani et al., 2023).

Another important direction is to further investigate the effectiveness of the flatness. As we discussed in Remark 1, recent results have shown that other notions of flatness are not always a good indicator of model efficacy (Andriushchenko et al., 2023; Wen et al., 2023). It would be interesting to understand the precise role of flatness, given that we have a lot of evidence of its success. Moreover, studying other notions of flatness, such as the "effective size of basin" as considered in (Kleinberg et al., 2018; Feng et al., 2020), or the constrained settings (*e.g.*, (Feng et al., 2020)), and exploring the algorithmic questions there would be also interesting future directions.

Based on our analysis, we suspect that replacing the full-batch gradients with the stochastic gradients in our proposed algorithms also leads to an efficient algorithm, with a more careful stochastic analysis. Moreover, we suspect that our results have sub-optimal dependence on the error probability δ, and a more advanced analysis will likely leads to a better dependence (Jin et al., 2021). Lastly, based on our experiments, it seems that a smaller batch size has the same effect as using a larger perturbation radius ρ. Whether one can capture this effect theoretically would be also an intriguing direction. However, the main scope of this work is to initiate the study of complexity of finding flat minima, and we leave all of this to future works.

Acknowledgements

Kwangjun Ahn and Ali Jadbabaie were supported by the ONR grant (N00014-20-1-2394) and MIT-IBM Watson as well as a Vannevar Bush fellowship from Office of the Secretary of Defense. Kwangjun Ahn and Suvrit Sra acknowledge support from an NSF CAREER grant (1846088), and NSF CCF-2112665 (TILOS AI Research Institute). Suvrit Sra also thanks the Alexander von Humboldt Foundation for their generous support.

Kwangjun Ahn thanks Xiang Cheng, Yan Dai, Hadi Daneshmand, and Alex Gu for helpful discussions that led the au-

thor to initiate this work.

Impact Statement

This paper aims to advance our theoretical understanding of flat minima optimization. Our work is theoretical in nature, and we do not see any immediate potential societal consequences.

References

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*, pages 1195–1199, 2017.

Kwangjun Ahn, Sebastien Bubeck, Sinho Chewi, Yin Tat Lee, Felipe Suarez, and Yi Zhang. Learning threshold neurons via edge of stability. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=9cQ6kToLnJ.

Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than sgd. *Advances in neural information processing systems*, 31, 2018.

Maksym Andriushchenko, Francesco Croce, Maximilian Müller, Matthias Hein, and Nicolas Flammarion. A modern look at the relationship between sharpness and generalization. *arXiv preprint arXiv:2302.07011*, 2023.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 199(1-2):165–214, 2023.

Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on the edge of stability in deep learning. In *International Conference on Machine Learning*, pages 948–1024. PMLR, 2022.

Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7360–7371, 2022.

Peter L Bartlett, Philip M Long, and Olivier Bousquet. The dynamics of sharpness-aware minimization: Bouncing across ravines and drifting towards wide minima. *arXiv preprint arXiv:2210.01513*, 2022.

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In *Conference on learning theory*, pages 483–513. PMLR, 2020.

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

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1-2):71–120, 2020.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points ii: first-order methods. *Mathematical Programming*, 185 (1-2):315–355, 2021.

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. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12): 124018, 2019.

Enea Monzio Compagnoni, Luca Biggio, Antonio Orvieto, Frank Norbert Proske, Hans Kersting, and Aurelien Lucchi. An sde for modeling sam: Theory and insights. In *International Conference on Machine Learning*, pages 25209–25253. PMLR, 2023.

Yaim Cooper. Global minima of overparameterized neural networks. *SIAM Journal on Mathematics of Data Science*, 3(2):676–691, 2021.

Yan Dai, Kwangjun Ahn, and Suvrit Sra. The crucial role of normalization in sharpness-aware minimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=zq4vFneRiA.

Alex Damian, Tengyu Ma, and Jason D. Lee. Label noise SGD provably prefers flat global minimizers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=x2TMPhseWAW.

Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=nhKHA59gXz.

Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. Escaping saddles with stochastic gradients. In Jennifer Dy and Andreas Krause, editors, *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 https://proceedings.mlr.press/v80/daneshmand18a.html.

Lijun Ding, Dmitriy Drusvyatskiy, and Maryam Fazel. Flat minima generalize for low-rank matrix recovery. *arXiv preprint arXiv:2203.03756*, 2022.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR, 2017.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pages 1309–1318. PMLR, 2018.

Yoel Drori and Ohad Shamir. The complexity of finding stationary points with stochastic gradient descent. In *International Conference on Machine Learning*, pages 2658–2667. PMLR, 2020.

John C Duchi, Peter L Bartlett, and Martin J Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

Han Feng, Haixiang Zhang, and Javad Lavaei. A dynamical system perspective for escaping sharp local minima in equality constrained optimization problems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 4255–4261. IEEE, 2020.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *arXiv preprint arXiv:1802.10026*, 2018.

Khashayar Gatmiry, Zhiyuan Li, Tengyu Ma, Sashank J. Reddi, Stefanie Jegelka, and Ching-Yao Chuang. What is the inductive bias of flatness regularization? a study of deep matrix factorization models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=2hQ7MBQApp.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Xinran Gu, Kaifeng Lyu, Longbo Huang, and Sanjeev Arora. Why (and when) does local SGD generalize better than SGD? In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=svCcui6Drl.

Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. *arXiv preprint arXiv:1902.00744*, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

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

Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M Kakade, and Michael I Jordan. On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *Journal of the ACM (JACM)*, 68(2):1–29, 2021.

Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt J Kusner. When do flat minima optimizers work? *Advances in Neural Information Processing Systems*, 35:16577–16595, 2022.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*, pages 795–811. Springer, 2016.

Simran Kaur, Jeremy Cohen, and Zachary Chase Lipton. On the maximum hessian eigenvalue and generalization. In *Proceedings on*, pages 51–65. PMLR, 2023.

Nitish Shirish Keskar, Jorge Nocedal, Ping Tak Peter Tang, Dheevatsa Mudigere, and Mikhail Smelyanskiy. On large-batch training for deep learning: Generalization gap and

sharp minima. In *International Conference on Learning Representations*, 2017.

Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does sgd escape local minima? In *International conference on machine learning*, pages 2698–2707. PMLR, 2018.

Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What happens after SGD reaches zero loss? –a mathematical framework. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=siCt4xZn5Ve.

Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. Same pre-training loss, better downstream: Implicit bias matters for language models. In *International Conference on Machine Learning*, pages 22188–22214. PMLR, 2023.

Shengchao Liu, Dimitris Papailiopoulos, and Dimitris Achlioptas. Bad global minima exist and SGD can reach them. *Advances in Neural Information Processing Systems*, 33:8543–8552, 2020.

Chao Ma and Lexing Ying. On linear stability of SGD and input-smoothness of neural networks. *Advances in Neural Information Processing Systems*, 34:16805–16817, 2021.

Rotem Mulayoff and Tomer Michaeli. Unique properties of flat minima in deep networks. In *International conference on machine learning*, pages 7108–7118. PMLR, 2020.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

Matthew D Norton and Johannes O Royset. Diametrical risk minimization: Theory and computations. *Machine Learning*, pages 1–19, 2021.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323. PMLR, 2016.

Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis. In *International Conference on Machine Learning*, pages 9636–9647. PMLR, 2020.

Yuqing Wang, Minshuo Chen, Tuo Zhao, and Molei Tao. Large learning rate tames homogeneity: Convergence and balancing effect. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=3tbDrs77LJ5.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How does sharpness-aware minimization minimize sharpness? *arXiv preprint arXiv:2211.05729*, 2022.

Kaiyue Wen, Zhiyuan Li, and Tengyu Ma. Sharpness minimization algorithms do not only minimize sharpness to achieve better generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Dkmpa6wCIx.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33: 2958–2969, 2020.

Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=wXgk_iCiYGo.

Zhewei Yao, Amir Gholami, Qi Lei, Kurt Keutzer, and Michael W Mahoney. Hessian-based analysis of large batch training and robustness to adversaries. *Advances in Neural Information Processing Systems*, 31, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Yaowei Zheng, Richong Zhang, and Yongyi Mao. Regularizing neural networks via adversarial model perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8156–8165, 2021.

Dongruo Zhou and Quanquan Gu. Stochastic recursive variance-reduced cubic regularization methods. In *International Conference on Artificial Intelligence and Statistics*, pages 3980–3990. PMLR, 2020.

Appendix

A. Preliminaries

In this section, we present background information and useful lemmas for our analysis. We start with several notations and conventions for our analysis.

- We will highlight the dependence on the relevant quantities ϵ, δ, d and will often hide the dependence on other parameters in the notations $\mathcal{O}(\cdot), \Theta(\cdot), \Omega(\cdot)$.
- We will sometimes abuse our notation as follows: when the two vectors $\boldsymbol{u}, \boldsymbol{v}$ satisfy $\|\boldsymbol{u}-\boldsymbol{v}\| = \mathcal{O}(g(\epsilon,\delta,d))$ for some function g of ϵ, δ, d, then we will simply write

$$\boldsymbol{u} = \boldsymbol{v} + \mathcal{O}(g(\epsilon,\delta,d)) .$$

- For a ℓ-th order tensor $\mathcal{T} \in \mathbb{R}^{d_1\times\cdots\times d_\ell}$, the spectral norm is defined as

$$\|\mathcal{T}\|_2 \coloneqq \sup_{\boldsymbol{u}_i\in\mathbb{R}^{d_i}, \|\boldsymbol{u}_i\|=1} \mathcal{T}[\boldsymbol{u}_1,\ldots,\boldsymbol{u}_\ell] .$$

- For a tensor $\mathcal{T}(\boldsymbol{x})$ that depends on $\boldsymbol{x}$ (e.g., $\nabla^2 f(\boldsymbol{x}), \nabla^3 f(\boldsymbol{x}), \partial\Phi(\boldsymbol{x})$ etc), let $L_{\mathcal{T}}$ be the upper bound on the spectrum norm $\|\mathcal{T}(\boldsymbol{x})\|_2$ within the ζ-neighborhood of $\mathcal{X}^\star$ (ζ is defined in Assumption 1).

We also recall our main assumption (Assumption 1) for reader's convenience.

Assumption 1 (Loss near minima)**.** *There exists $\zeta > 0$ such that within ζ-neighborhood of the set of local minima $\mathcal{X}^\star$, the following properties hold:*

(a) f is four-times continuously differentiable.
(b) The limit map under gradient flow Φ (Definition 1) is well-defined and is twice Lipschitz differentiable. Also, $\Phi(\boldsymbol{x}) \in \mathcal{X}^\star$ and the gradient flow starting at $\boldsymbol{x}$ is contained within the ζ-neighborhood of $\mathcal{X}^\star$.
(c) The Polyak–Łojasiewicz (PL) inequality holds locally, i.e., $f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})) \le \frac{1}{2\alpha}\|\nabla f(\boldsymbol{x})\|^2$.

A.1. Auxiliary Lemmas

We first present the following geometric result that compares the cost, gradient norm, and the distance to $\Phi(\boldsymbol{x})$ near $\mathcal{X}^\star$.

Lemma 5. *Let Assumption 1 hold and f have β-Lipschitz gradients. If $\boldsymbol{x}$ is in the ζ-neighborhood of $\mathcal{X}^\star$, then it holds that*

- $\|\boldsymbol{x}-\Phi(\boldsymbol{x})\| \le \sqrt{\frac{2}{\alpha}}\sqrt{f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x}))}$ *and* $\sqrt{f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x}))} \le \frac{\beta}{\sqrt{2\alpha}}\|\boldsymbol{x}-\Phi(\boldsymbol{x})\|$.
- $\|\boldsymbol{x}-\Phi(\boldsymbol{x})\| \le \frac{1}{\alpha}\|\nabla f(\boldsymbol{x})\|$ *and* $\|\nabla f(\boldsymbol{x})\| \le \beta\|\boldsymbol{x}-\Phi(\boldsymbol{x})\|$.
- $\sqrt{f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x}))} \le \frac{1}{\sqrt{2\alpha}}\|\nabla f(\boldsymbol{x})\|$ *and* $\|\nabla f(\boldsymbol{x})\| \le \sqrt{\frac{2\beta^2}{\alpha}}\sqrt{f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x}))}$.

Proof. See Subsection D.2. □

We next present an important property of the limit point under the gradient flow, Φ.

Lemma 6. *For any $\boldsymbol{x}$ at which Φ is defined and differentiable, we have that $\partial\Phi(\boldsymbol{x})\nabla f(\boldsymbol{x}) = \boldsymbol{0}$.*

Proof. See (Wen et al., 2022, Lemma 3.2) or (Li et al., 2022, Lemma C.2). □

We next prove the following results about the distance in terms of Φ between two adjacent iterates.

Lemma 7. *Let Assumption 1 hold and f have β-Lipschitz gradients. For a vector $\boldsymbol{v}$ satisfying $\boldsymbol{v} \perp \nabla f(\boldsymbol{x})$ and $\|\boldsymbol{v}\| = \mathcal{O}(1)$, consider the update $\boldsymbol{x}^+ - \boldsymbol{x} = -\eta(\nabla f(\boldsymbol{x}) + \boldsymbol{v})$. Then, for suffciently small η, if $\boldsymbol{x}$ is in ζ-neighborhood of $\mathcal{X}^\star$, the following holds:*

- $\Phi(x^+) - \Phi(x) = -\eta\partial\Phi(x)\boldsymbol{v} + \mathcal{O}\left(L_{\partial\Phi}\eta^2(\|\nabla f(\boldsymbol{x})\|^2 + \|\boldsymbol{v}\|^2)\right)$.
- $\|\Phi(\boldsymbol{x}^+) - \Phi(\boldsymbol{x})\|^2 \le 4L_{\partial\Phi}^2\eta^2\|\boldsymbol{v}\|^2 + 3L_{\partial\Phi}^2\eta^4\|\nabla f(\boldsymbol{x})\|^4$.

- $|f(\Phi(\boldsymbol{x}^+)) - f(\Phi(\boldsymbol{x}))| = \mathcal{O}\left((L_{\partial^2\Phi}L_{\nabla f} + L_{\partial\Phi}L_{\nabla^2 f})L_{\partial\Phi}^2(\|\boldsymbol{v}\|^2 + \eta^2\|\nabla f(\boldsymbol{x})\|^4)\right)$.

Proof. See Subsection D.3. □

We next present the result about iterates staying near the local minima set $\mathcal{X}^\star$.

Lemma 8. *Let Assumption 1 hold and f have β-Lipschitz gradients. For a vector $\boldsymbol{v}$ satisfying $\boldsymbol{v} \perp \nabla f(\boldsymbol{x})$ and $\|\boldsymbol{v}\| = \mathcal{O}(1)$, consider the update $\boldsymbol{x}^+ - \boldsymbol{x} = -\eta \cdot (\nabla f(\boldsymbol{x}) + \boldsymbol{v})$. For sufficiently small $\eta > 0$, we have the following:*

$$\text{if } f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})) \le \frac{2\beta}{\alpha}\eta\|\boldsymbol{v}\|^2, \text{ then } f(\boldsymbol{x}^+) - f(\Phi(\boldsymbol{x}^+)) \le \frac{2\beta}{\alpha}\eta\|\boldsymbol{v}\|^2 \text{ as well.}$$

Proof. See Subsection D.4. □

B. Proof of Theorem 1

In this section, we present the proof of Theorem 1. The overall structure of the proof follows the proof sketch in Subsection 3.2. We consider the following choice of parameters for Algorithm 1:

$$\eta = \Theta\left(\frac{1}{L_0 L_{\partial\Phi}^2 \beta^2}\delta\epsilon\right), \quad \rho = \Theta\left(\frac{1}{L_0}\delta\sqrt{\epsilon}\right), \quad \epsilon_0 = \Theta\left(\frac{\beta^{3/2}}{\alpha L_0^{3/2} L_{\partial\Phi}}\delta^{1.5}\epsilon\right). \tag{8}$$

where $L_0 \coloneqq L_{\partial^2\Phi}L_{\nabla^3 f} + L_{\partial\Phi}L_{\nabla^4 f}$. Then, note that $\partial\Phi(\boldsymbol{x})\nabla\overline{\mathsf{tr}}(\boldsymbol{x})$ is L_0-Lipschitz in the ζ-neighborhood of $\mathcal{X}^\star$.

First, since $\boldsymbol{x}_0$ is ζ-close to $\mathcal{X}^\star$, the standard linear convergence result of gradient descent for the cost function satisfying the Polyak–Łojasiewicz (PL) inequality (Karimi et al., 2016) together with Lemma 5 imply that with the step size $\eta' = \frac{1}{\beta}$, within $T_0 = \mathcal{O}(\log(1/\epsilon_0))$ steps, one can reach the point $\boldsymbol{x}_{T_0}$ satisfying $\|\nabla f(\boldsymbol{x}_{T_0})\| \le \epsilon_0$. Thus, we henceforth assume that $\boldsymbol{x}_0$ itself satisfies $\|\nabla f(\boldsymbol{x}_0)\| \le \epsilon_0$ without loss of generality.

Next, we show that for $\boldsymbol{v}_t$ defined as in Algorithm 1, i.e., $\boldsymbol{v}_t \coloneqq \mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}\nabla f(\boldsymbol{x}_t + \rho\boldsymbol{g}_t)$, we have

$$\|\boldsymbol{v}_t\| \le \beta\rho \text{ at each step } t.$$

This holds because $\boldsymbol{v}_t = \mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}(\nabla f(\boldsymbol{x}_t + \rho\boldsymbol{g}_t) - \nabla f(\boldsymbol{x}_t))$, and the "projecting-out" operator $\mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}$ only decreases the norm of the vector: it follows that $\|\boldsymbol{v}_t\| \le \|\nabla f(\boldsymbol{x}_t + \rho\boldsymbol{g}_t) - \nabla f(\boldsymbol{x}_t)\| \le \beta\rho$, as desired.

Then, by Lemma 8, for sufficiently small ϵ, it holds that $f(\boldsymbol{x}_t) - f(\Phi(\boldsymbol{x}_t)) \le \frac{2\beta^3}{\alpha}\eta\rho^2$ during each step t. This implies together with Lemma 5 that $\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\|^2 \le \frac{4\beta^3}{\alpha^2}\eta\rho^2$ and $\|\nabla f(\boldsymbol{x}_t)\|^2 \le \frac{4\beta^5}{\alpha^2}\eta\rho^2$ during each step t. Thus, due to the choice (8), we conclude that

$$\|\nabla f(\boldsymbol{x}_t)\| \le \epsilon_0 \text{ and } \|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\| \le \mathcal{O}(\epsilon_0) \quad \text{hold during each step } t. \tag{9}$$

We now characterize the direction $\partial\Phi(\boldsymbol{x}_t)\boldsymbol{v}_t$.

Lemma 9. *Let Assumption 1 hold and consider the parameter choice (8). Then, for sufficiently small $\epsilon > 0$, under the condition (9), $\boldsymbol{v}_t$ defined in Algorithm 1 satisfies*

$$\mathbb{E}\partial\Phi(\boldsymbol{x}_t)\boldsymbol{v}_t = \frac{1}{2}\rho^2\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) + \mathcal{O}\left(L_0\rho^3\right).$$

Proof. See Subsection D.5. □

Using Lemma 9, we can prove the following formal statement of Lemma 2.

Lemma 10. *Let Assumption 1 hold and choose the parameters as per (8). Let $\epsilon > 0$ be chosen sufficiently small and $\delta \in (0, 1)$. Then, there exists an absolute constant $c > 0$ s.t. the following holds: if $\|\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\| \ge \sqrt{\epsilon}$, then*

$$\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \le -\frac{c}{L_0^3 L_{\partial\Phi}^2\beta^2}\delta^3\epsilon^3.$$

On the other hand, if $\|\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\| \le \sqrt{\epsilon}$*, then*

$$\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \le \frac{c}{L_0^3 L_{\partial\Phi}^2 \beta^2}\delta^4\epsilon^3 .$$

Proof. See Subsection D.6. □

Now the rest of the proof follows the probabilistic argument in the proof sketch (Subsection 3.2). For $t = 1, 2, \ldots, T$, let A_t be the event where $\|\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\| \ge \sqrt{\epsilon}$, and let R be a random variable equal to the ratio of desired flat minima visited among the iterates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$. Then,

$$R = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\left(A_t^c\right) ,$$

where $\mathbb{1}$ is the indicator function. Let P_t denote the probability of event A_t. Then, the probability of returning a $(\epsilon, \sqrt{\epsilon})$-flat minimum is simply equal to $\mathbb{E}R = \frac{1}{T}\sum_{t=1}^{T}(1 - P_t)$. Now the key idea is that although estimating individual P_t's might be difficult, one can upper bound the sum of P_t's using Lemma 10. More specifically, Lemma 10 implies that

$$\begin{aligned}\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) &\le -\frac{c}{L_0^3 L_{\partial\Phi}^2 \beta^2}\delta^3\epsilon^3 \cdot (P_t - \delta(1 - P_t)) \\ &= \frac{c}{L_0^3 L_{\partial\Phi}^2 \beta^2}\delta^3\epsilon^3 \cdot \{\delta - (1+\delta)P_t\} ,\end{aligned}$$

which after taking sum over $t = 0 \ldots, T-1$ and rearranging yields

$$\frac{1}{T}\sum_{t=1}^{T} P_t \le \frac{L_0^3 L_{\partial\Phi}^2 \beta^2}{cT\delta^3\epsilon^3} + \delta .$$

Hence choosing

$$T = \Theta\left(\frac{L_0^3 L_{\partial\Phi}^2 \beta^2}{\epsilon^3\delta^4}\right) ,$$

$\mathbb{E}R$ is lower bounded by $1 - \mathcal{O}(\delta)$, which concludes the proof of Theorem 1.

C. Proof of Theorem 2

In this section, we present the proof of Theorem 2. The overall structure of the proof is similar to that of Theorem 1in Appendix B. We consider the following choice of parameters for Algorithm 2: for $\nu := \min\{d, \epsilon^{-1/3}\}$,

$$\eta = \Theta\left(\frac{1}{L_0 L_{\partial\Phi}^2 \beta^2}\nu\delta\epsilon\right), \;\; \rho = \Theta\left(\frac{1}{L_0}\nu\delta\sqrt{\epsilon}\right), \;\; \epsilon_0 = \Theta\left(\frac{\beta^{3/2}}{\alpha L_0^{3/2} L_{\partial\Phi}}\nu^{3/2}\delta^{3/2}\epsilon\right). \tag{10}$$

where this time we define $L_0 := \max_{i=1,\ldots,n} L_{\partial^2\Phi} L_{\nabla^3 f_i} + L_{\partial\Phi} L_{\nabla^4 f_i}$. Then, again note that $\partial\Phi(\boldsymbol{x})\nabla\overline{\mathsf{tr}}(\boldsymbol{x})$ is L_0-Lipschitz in the ζ-neighborhood of $\mathcal{X}^\star$.

Again, similarly to the proof in Appendix B, within $T_0 = \mathcal{O}(\log(1/\epsilon_0))$ steps, one can reach $\boldsymbol{x}_{T_0}$ s.t. $\|\nabla f(\boldsymbol{x}_{T_0})\| \le \epsilon_0$, so we assume that $\boldsymbol{x}_0$ satisfies $\|\nabla f(\boldsymbol{x}_0)\| \le \epsilon_0$ without loss of generality.

We first show that for $\boldsymbol{v}_t$ defined as $\mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}\nabla f_i(\boldsymbol{x}_t + \rho\sigma_t\frac{\nabla f_i(\boldsymbol{x}_t)}{\|\nabla f_i(\boldsymbol{x}_t)\|})$, we have

$$\|\boldsymbol{v}_t\| \le \|\nabla f_i(\boldsymbol{x}_t)\| + \beta\rho \text{ at each step } t. \tag{11}$$

This holds since the β-Lipschitz gradient condition implies

$$\left\|\nabla f_i\left(\boldsymbol{x}_t + \rho\sigma_t\frac{\nabla f_i(\boldsymbol{x}_t)}{\|\nabla f_i(\boldsymbol{x}_t)\|}\right)\right\| \le \|\nabla f_i(\boldsymbol{x}_t)\| + \beta\left\|\rho\sigma_t\frac{\nabla f_i(\boldsymbol{x}_t)}{\|\nabla f_i(\boldsymbol{x}_t)\|}\right\| = \|\nabla f_i(\boldsymbol{x}_t)\| + \beta\rho ,$$

and the "projecting-out" operator $\mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}$ only decreases the norm of the vector. Hence, it follows that $\|\boldsymbol{v}_t\| \leq \|\nabla f_i(\boldsymbol{x}_t)\| + \beta\rho$.

Now we show by induction that $f(\boldsymbol{x}_t) - f(\Phi(\boldsymbol{x}_t)) \leq \frac{8\beta^3}{\alpha}\eta\rho^2$ holds during each step t. Suppose that it holds for $\boldsymbol{x}_t$ and consider $\boldsymbol{x}_{t+1}$. Then from Lemma 5, it holds that $\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\|^2 \leq \frac{16\beta^3}{\alpha^2}\eta\rho^2$, which implies that $\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\| = \mathcal{O}\left(\eta^{1/2}\rho\right) = o\left(\rho\right)$ as long as ϵ is sufficiently small. Thus, from (11), it follows that $\|\boldsymbol{v}_t\| \leq 2\beta\rho$, and hence, Lemma 8 implies that $f(\boldsymbol{x}_{t+1}) - f(\Phi(\boldsymbol{x}_{t+1})) \leq \frac{8\beta^3}{\alpha}\eta\rho^2$.

This conclusion together with Lemma 5 and the choice (8) imply the following conclusion:

$$\|\nabla f(\boldsymbol{x}_t)\| \leq \epsilon_0 \;\text{ and }\; \|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\| \leq \mathcal{O}\left(\epsilon_0\right) \text{ hold during each step } t. \tag{12}$$

We now characterize the direction $\partial\Phi(\boldsymbol{x}_t)\boldsymbol{v}_t$.

Lemma 11. *Let Assumption 1 hold and each f_i is four times coutinuously differentiable within the ζ-neighborhood of $\mathcal{X}^\star$ and have β-Lipschitz gradients. Consider the parameter choice (8). Then, for sufficiently small $\epsilon > 0$, under the condition (12), $\boldsymbol{v}_t$ defined in Algorithm 2 (assume that it is well-defined as per Remark 2) satisfies*

$$\mathbb{E}\partial\Phi(\boldsymbol{x}_t)\boldsymbol{v}_t = \frac{1}{2}d\rho^2\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) + \mathcal{O}\left(L_1L_2d\rho^2\epsilon_0\right) + \mathcal{O}\left(L_0\rho^3\right) ,$$

where $L_1 \coloneqq \max_{i=1,\ldots,n} L_{\partial(\nabla p_i/\|\nabla p_i\|)}$ and $L_2 \coloneqq L_{\partial\Phi}L_{\nabla^3 f_i}$.

Proof. See Subsection D.7. □

Notice an multiplicative factor of d appearing in the equation above, which shows an improvement over Lemma 9. Using Lemma 11, we can prove the following formal statement of Lemma 4.

Lemma 12. *Let Assumption 1 hold and choose the parameters as per (8). Let $\epsilon > 0$ be chosen sufficiently small and $\delta \in (0,1)$. Then, under the condition (12), there exists an absolute constant $c > 0$ s.t. the following holds: if $\|\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\| \geq \sqrt{\epsilon}$, then*

$$\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \leq -\frac{c}{L_0^3L_{\partial\Phi}^2\beta^2}d\nu^3\delta^3\epsilon^3 .$$

On the other hand, if $\|\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\| \leq \sqrt{\epsilon}$, then

$$\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \leq \frac{c}{L_0^3L_{\partial\Phi}^2\beta^2}d\nu^3\delta^4\epsilon^3 .$$

Proof. See Subsection D.8. □

Now the rest of the proof follows the probabilistic argument in Appendix B. For $t = 1, 2, \ldots, T$, let A_t be the event where $\|\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))\| \geq \sqrt{\epsilon}$, and let R be a random variable equal to the ratio of desired flat minima visited among the iterates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$. Let P_t denote the probability of event A_t. Then, the probability of returning a $(\epsilon, \sqrt{\epsilon})$-flat minimum is simply equal to $\mathbb{E}R = \frac{1}{T}\sum_{t=1}^{T}(1 - P_t)$. Similarly to Appendix B, using Lemma 12, we have

$$\begin{aligned}\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) &\leq -\frac{c}{L_0^3L_{\partial\Phi}^2\beta^2}d\nu^3\delta^3\epsilon^3 \cdot (P_t - \delta(1 - P_t)) \\ &= \frac{c}{L_0^3L_{\partial\Phi}^2\beta^2}d\nu^3\delta^3\epsilon^3 \cdot \{\delta - (1+\delta)P_t\} ,\end{aligned}$$

which after taking sum over $t = 0 \ldots, T-1$ and rearranging yields

$$\frac{1}{T}\sum_{t=1}^{T}P_t \leq \frac{L_0^3L_{\partial\Phi}^2\beta^2}{cTd\nu^3\delta^3\epsilon^3} + \delta \leq 2\delta .$$

Hence choosing

$$T = \Theta\left(\frac{L_0^3 L_{\partial\Phi}^2 \beta^2}{d\nu^3\epsilon^3\delta^4}\right),$$

$\mathbb{E}R$ is lower bounded by $1 - \mathcal{O}(\delta)$, which shows that Theorem 2. This shows that $\widehat{\boldsymbol{x}}$ is an $(\mathcal{O}(\epsilon_0), \sqrt{\epsilon})$-flat minimum with probability at least $1 - \mathcal{O}(\delta)$.

Now we prove the refinement part. Let $\widehat{\boldsymbol{x}}_0 := \widehat{\boldsymbol{x}}$. Since $\nu := \min\{d, \epsilon^{-1/3}\} \le \epsilon^{-1/3}$,

$$\epsilon_0 = \Theta\left(\frac{\beta^{3/2}}{\alpha L_0^{3/2} L_{\partial\Phi}}\nu^{3/2}\delta^{3/2}\epsilon\right) \le \mathcal{O}\left(\sqrt{\epsilon}\right)$$

Hence, from Lemma 5, it then follows that $\|\nabla f(\widehat{\boldsymbol{x}}_0)\| \le \mathcal{O}(\sqrt{\epsilon})$ and $f(\boldsymbol{x}_t) - f(\Phi(\boldsymbol{x}_t)) \le \mathcal{O}(\epsilon)$. Then, the linear convergence of GD under the PL inequality shows that GD with step size $\mathcal{O}(\epsilon)$ finds an point $\widehat{\boldsymbol{x}}_{T_0}$ s.t. $\|\widehat{\boldsymbol{x}}_{T_0} - \Phi(\widehat{\boldsymbol{x}}_{T_0})\| \le \epsilon/2$ in $T_0 = \mathcal{O}\left(\epsilon^{-1}\log(1/\epsilon)\right)$ steps. On the other hand, applying Lemma 7 with $\boldsymbol{v} = \mathbf{0}$, it holds that

$$\|\Phi(\widehat{\boldsymbol{x}}_{t+1}) - \Phi(\widehat{\boldsymbol{x}}_t)\| = \mathcal{O}\left(\eta^2\|\nabla f(\widehat{\boldsymbol{x}}_t)\|^2\right) = \mathcal{O}\left(\epsilon^3\right).$$

Therefore, it follows that

$$\|\Phi(\widehat{\boldsymbol{x}}_{T_0}) - \Phi(\widehat{\boldsymbol{x}}_0\|) = \mathcal{O}\left(\epsilon^3 \cdot \epsilon^{-1}\log(1/\epsilon)\right) = \mathcal{O}\left(\epsilon^{-2}\log(1/\epsilon)\right).$$

Thus, we conclude that $\widehat{\boldsymbol{x}}_{T_0}$ is a $(\epsilon, \sqrt{\epsilon})$-flat minimum. This concludes the proof of Theorem 2.

D. Proof of Auxiliary Lemmas

D.1. Proof of Lemma 1

Due to the β-gradient Lipschitz assumption, we have:

$$\begin{aligned} f(\boldsymbol{x}^+) &\le f(\boldsymbol{x}) + \langle\nabla f(\boldsymbol{x}), \boldsymbol{x}^+ - \boldsymbol{x}\rangle + \frac{\beta}{2}\left\|\boldsymbol{x}^+ - \boldsymbol{x}\right\|^2 \\ &= f(\boldsymbol{x}) - \eta\|\nabla f(\boldsymbol{x})\|^2 + \frac{\eta^2\beta}{2}\|\nabla f(\boldsymbol{x}) + \boldsymbol{v}\|^2 \\ &\le f(\boldsymbol{x}) - \frac{1}{2}\eta(2-\eta\beta)\frac{1}{2\beta}\|\nabla f(\boldsymbol{x})\|^2 + \frac{\beta\eta^2}{2}\|\boldsymbol{v}\|^2. \end{aligned}$$

Hence, using the fact that $\eta\beta \le 1$, which implies $-(2-\eta\beta) \le -1$.

D.2. Proof of Lemma 5

To prove Lemma 5, it suffices to show the following:

$$\|\boldsymbol{x} - \Phi(\boldsymbol{x})\| \le \sqrt{\frac{2(f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})))}{\alpha}} \le \frac{1}{\alpha}\|\nabla f(\boldsymbol{x})\| \le \frac{\beta}{\alpha}\|\boldsymbol{x} - \Phi(\boldsymbol{x})\|.$$

The proof essentially follows that of (Arora et al., 2022, Lemma B.6). We provide a proof below nevertheless to be self-contained. Since $\boldsymbol{x}$ is within ζ-neighborhood of $\mathcal{X}^\star$, Assumption 1 implies that Φ is well-defined, and hence letting $\boldsymbol{x}(t)$ be the iterate at time t of a gradient flow starting at $\boldsymbol{x}$, we have

$$\|\boldsymbol{x} - \Phi(\boldsymbol{x})\| = \left\|\int_{t=0}^{\infty}\nabla f(\boldsymbol{x}(t))\mathrm{d}t\right\| \le \int_{t=0}^{\infty}\|\nabla f(\boldsymbol{x}(t))\|\,\mathrm{d}t.$$

Now due to the Polyak–Łojasiewicz inequality, it holds that $\|\nabla f(\boldsymbol{x}_t)\|^2 \ge 2\alpha(f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})))$. Thus, we have

$$\begin{aligned} \int_{t=0}^{\infty}\|\nabla f(\boldsymbol{x}(t))\|\,\mathrm{d}t &\le \int_{t=0}^{\infty}\frac{\|\nabla f(\boldsymbol{x}(t))\|^2}{\sqrt{2\alpha(f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})))}}\mathrm{d}t \overset{(a)}{=} -\int_{t=0}^{\infty}\frac{\frac{\mathrm{d}}{\mathrm{d}t}[f(\boldsymbol{x}(t)) - f(\Phi(\boldsymbol{x}))]}{\sqrt{2\alpha(f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})))}}\mathrm{d}t \\ &= -\sqrt{\frac{2}{\alpha}}\int_{t=0}^{\infty}\frac{\mathrm{d}}{\mathrm{d}t}\sqrt{f(\boldsymbol{x}(t)) - f(\Phi(\boldsymbol{x}))}\mathrm{d}t = \sqrt{\frac{2}{\alpha}(f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})))}, \end{aligned}$$

where (a) follows from the fact

$$\frac{\mathrm{d}}{\mathrm{d}t}[f(\boldsymbol{x}(t)) - f(\Phi(\boldsymbol{x}))] = \langle \nabla f(\boldsymbol{x}(t)), \dot{\boldsymbol{x}}(t) \rangle = - \|\nabla f(\boldsymbol{x}(t))\|^2 .$$

Hence, we obtain

$$\|\boldsymbol{x} - \Phi(\boldsymbol{x})\| \leq \sqrt{\frac{2}{\alpha}(f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})))} \leq \frac{\|\nabla f(\boldsymbol{x})\|}{\alpha} ,$$

where the last inequality is due to the PL condition. Lastly, we have

$$\frac{1}{\alpha} \|\nabla f(\boldsymbol{x})\| = \frac{1}{\alpha} \|\nabla f(\boldsymbol{x}) - \nabla f(\Phi(\boldsymbol{x}))\| \leq \frac{\beta}{\alpha} \|\boldsymbol{x} - \Phi(\boldsymbol{x})\| ,$$

where the last inequality is due to β-Lipschitz gradients of f. This completes the proof.

D.3. Proof of Lemma 7

We first prove the first bullet point. From the smoothness of Φ, we obtain

$$\begin{aligned}\Phi(\boldsymbol{x}^+) - \Phi(\boldsymbol{x}) &= \partial\Phi(\boldsymbol{x})(-\eta(\nabla f(\boldsymbol{x}) + \boldsymbol{v})) + \mathcal{O}\left(L_{\partial\Phi} \left\|\boldsymbol{x}^+ - \boldsymbol{x}\right\|^2\right) \\ &\overset{(a)}{=} -\eta\partial\Phi(\boldsymbol{x})\boldsymbol{v} + \mathcal{O}\left(L_{\partial\Phi}\eta^2(\|\nabla f(\boldsymbol{x})\| + \|\boldsymbol{v}\|^2)\right) ,\end{aligned}$$

where in (a), we used the fact $\partial\Phi(\boldsymbol{x})\nabla f(\boldsymbol{x}) = 0$ from Lemma 6. This, in particular, implies that

$$\begin{aligned}\left\|\Phi(\boldsymbol{x}^+) - \Phi(\boldsymbol{x})\right\|^2 &\leq 3L_{\partial\Phi}^2\eta^2 \|\boldsymbol{v}\|^2 + 3L_{\partial\Phi}^2\eta^4 \|\nabla f(\boldsymbol{x})\|^4 + 3L_{\partial\Phi}^2\eta^4 \|\boldsymbol{v}\|^4 \\ &\leq 4L_{\partial\Phi}^2\eta^2 \|\boldsymbol{v}\|^2 + 3L_{\partial\Phi}^2\eta^4 \|\nabla f(\boldsymbol{x})\|^4 ,\end{aligned}$$

as long as η is sufficiently small since $\eta^4 \|\boldsymbol{v}\|^4$ is a lower order term than $\eta^2 \|\boldsymbol{v}\|^2$.

Next, we prove the second bullet point. From the smoothness of $f(\Phi)$, we have

$$\begin{aligned}&f(\Phi(\boldsymbol{x}^+)) - f(\Phi(\boldsymbol{x})) = f(\Phi(\Phi(\boldsymbol{x}^+))) - f(\Phi(\Phi(\boldsymbol{x}))) \\ &\leq \left\langle \partial\Phi(\Phi(\boldsymbol{x}))\nabla f(\Phi(\boldsymbol{x})), \Phi(\boldsymbol{x}^+) - \Phi(\boldsymbol{x}) \right\rangle + \mathcal{O}\left((L_{\partial^2\Phi}L_{\nabla f} + L_{\partial\Phi}L_{\nabla^2 f}) \left\|\Phi(\boldsymbol{x}^+) - \Phi(\boldsymbol{x})\right\|^2\right) \\ &= \mathcal{O}\left(\eta^2(L_{\partial^2\Phi}L_{\nabla f} + L_{\partial\Phi}L_{\nabla^2 f})L_{\partial\Phi}^2(\|\boldsymbol{v}\|^2 + \eta^2 \|\nabla f(\boldsymbol{x})\|^4)\right)\end{aligned}$$

where (a) used the fact $\partial\Phi(\Phi(\boldsymbol{x}))\nabla f(\Phi(\boldsymbol{x})) = 0$ from Lemma 6. And the same argument applies for $f(\Phi(\boldsymbol{x})) - f(\Phi(\boldsymbol{x}^+))$, so we get the conclusion.

D.4. Proof of Lemma 8

By Lemma 1, we have

$$f(\boldsymbol{x}^+) \leq f(\boldsymbol{x}) - \frac{1}{2}\eta \|\nabla f(\boldsymbol{x})\|^2 + \frac{\beta\eta^2}{2} \|\boldsymbol{v}\|^2 .$$

Now we consider two different cases:

1. First, if $\|\nabla f(\boldsymbol{x})\|^2 \leq 2\beta\eta \|\boldsymbol{v}\|^2$, then Lemma 5 implies that

$$f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})) \leq \frac{1}{2\alpha} \|\nabla f(\boldsymbol{x})\|^2 \leq \frac{\beta}{\alpha}\eta \|\boldsymbol{v}\|^2 .$$

   Hence, it follows that

$$\begin{aligned}f(\boldsymbol{x}^+) - f(\Phi(\boldsymbol{x}^+)) &\leq f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x}^+)) + \frac{\beta\eta^2}{2} \|\boldsymbol{v}\|^2 \\ &\leq f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})) + \frac{\beta\eta^2}{2} \|\boldsymbol{v}\|^2 + \mathcal{O}\left(\eta^2 \|\boldsymbol{v}\|^2\right) \\ &\leq \frac{\beta}{\alpha}\eta \|\boldsymbol{v}\|^2 + \mathcal{O}\left(\eta^2 \|\boldsymbol{v}\|^2\right) \leq \frac{2\beta}{\alpha}\eta \|\boldsymbol{v}\|^2 ,\end{aligned}$$

   as long as η is sufficiently small.

2. On the other hand if $\|\nabla f(\boldsymbol{x})\|^2 \geq 2\beta\eta\|\boldsymbol{v}\|^2$, then we have $f(\boldsymbol{x}^+) - f(\boldsymbol{x}) \leq -\frac{1}{2}\beta\eta\|\boldsymbol{v}\|^2$. Next, from Lemma 7, it holds that

$$|f(\Phi(\boldsymbol{x}^+)) - f(\Phi(\boldsymbol{x}))| = \mathcal{O}\left(\eta^2\|\boldsymbol{v}\|^2\right),$$

as $\eta^4\|\nabla f(\boldsymbol{x})\|^2 = \mathcal{O}\left(\eta^5\|\boldsymbol{v}\|^2\right)$ and $\eta^4\|\boldsymbol{v}\|^4$ are both lower order terms. Thus, it follows that

$$\begin{aligned} f(\boldsymbol{x}^+) - f(\Phi(\boldsymbol{x}^+)) &\leq f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})) - \frac{1}{2}\beta\eta\|\boldsymbol{v}\|^2 + \mathcal{O}\left(\eta^2\|\boldsymbol{v}\|^2\right) \\ &\leq f(\boldsymbol{x}) - f(\Phi(\boldsymbol{x})) - \frac{1}{4}\beta\eta\|\boldsymbol{v}\|^2, \end{aligned}$$

as long as η is sufficiently small.

Combining these two cases, we get the desired conclusion.

D.5. Proof of Lemma 9

Note that by Taylor expansion, we have

$$\nabla f(\boldsymbol{x}_t + \rho\boldsymbol{g}_t) = \nabla f(\boldsymbol{x}_t) + \rho\nabla^2 f(\boldsymbol{x}_t)\boldsymbol{g}_t + \frac{1}{2}\rho^2\nabla^3 f(\boldsymbol{x}_t)[\boldsymbol{g}_t, \boldsymbol{g}_t] + \mathcal{O}\left(\frac{1}{6}L_{\nabla^4 f}\rho^3\right)$$

This implies that

$$\begin{aligned} \boldsymbol{v}_t &= \mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}\nabla f(\boldsymbol{x}_t + \rho\boldsymbol{g}_t) = \mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}\left[\nabla f(\boldsymbol{x}_t + \rho\boldsymbol{g}_t) - \nabla f(\boldsymbol{x}_t)\right] \\ &= \mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)}\left[\rho\nabla^2 f(\boldsymbol{x}_t)\boldsymbol{g}_t + \frac{1}{2}\rho^2\nabla^3 f(\boldsymbol{x}_t)[\boldsymbol{g}_t, \boldsymbol{g}_t] + \mathcal{O}\left(\frac{1}{6}L_{\nabla^4 f}\rho^3\right)\right]. \end{aligned}$$

Now from Lemma 6, $\partial\Phi(\boldsymbol{x})\nabla f(\boldsymbol{x}) = \mathbf{0}$ for any $\boldsymbol{x}$ in the ζ-neighborhood of $\mathcal{X}^\star$, it follows

$$\begin{aligned} \partial\Phi(\boldsymbol{x}_t)\mathbb{E}\boldsymbol{v}_t &= \frac{1}{2}\rho^2\partial\Phi(\boldsymbol{x}_t)\mathbb{E}\nabla^3 f(\boldsymbol{x}_t)[\boldsymbol{g}_t, \boldsymbol{g}_t] + \mathcal{O}\left(\frac{1}{6}L_{\partial\Phi}L_{\nabla^4 f}\rho^3\right) \\ &\overset{(a)}{=} \frac{1}{2}\rho^2\partial\Phi(\boldsymbol{x}_t)\nabla\mathbb{E}\,\mathrm{tr}\left(\nabla^2 f(\boldsymbol{x}_t)\boldsymbol{g}_t\boldsymbol{g}_t^\top\right) + \mathcal{O}\left(\frac{1}{6}L_0\rho^3\right) \\ &\overset{(b)}{=} \frac{1}{2}\rho^2\partial\Phi(\boldsymbol{x}_t)\nabla\,\mathrm{tr}\left(\nabla^2 f(\boldsymbol{x}_t)\frac{1}{d}\mathbf{I}_d\right) + \mathcal{O}\left(\frac{1}{6}L_0\rho^3\right) \\ &= \frac{1}{2}\rho^2\partial\Phi(\boldsymbol{x}_t)\nabla\overline{\mathsf{tr}}(\boldsymbol{x}_t) + \mathcal{O}\left(\frac{1}{6}L_0\rho^3\right) \end{aligned}$$

where (a) is due to the fact that $\nabla^3 f(\boldsymbol{x})[\boldsymbol{g}_t, \boldsymbol{g}_t] = \nabla(\nabla^2 f(\boldsymbol{x})[\boldsymbol{g}_t, \boldsymbol{g}_t]) = \nabla\,\mathrm{tr}\left(\nabla^2 f(\boldsymbol{x})\boldsymbol{g}_t\boldsymbol{g}_t^\top\right)$ for any $\boldsymbol{x}$, and (b) uses the fact that $\mathbb{E}[\boldsymbol{g}_t\boldsymbol{g}_t^\top] = \frac{1}{d}\mathbf{I}_d$. Now due to L_0-Lipschitzness of $\partial\Phi(\cdot)\nabla\overline{\mathsf{tr}}(\cdot)$, we have

$$\begin{aligned} \partial\Phi(\boldsymbol{x}_t)\mathbb{E}\boldsymbol{v}_t &= \frac{1}{2}\rho^2\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) + \mathcal{O}\left(\frac{1}{2}\rho^2 L_0\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\|\right) + \mathcal{O}\left(\frac{1}{6}L_0\rho^3\right) \\ &= \frac{1}{2}\rho^2\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) + \mathcal{O}\left(\frac{1}{3}L_0\rho^3\right), \end{aligned}$$

where the last line is due to (9), which implies $L_0\rho^2\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\| = o\left(L_0\rho^3\right)$ as $\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\| = \mathcal{O}(\epsilon_0) = o(\rho)$. This completes the proof.

D.6. Proof of Lemma 10

First, note from Lemma 7 and (9) that

$$\begin{aligned} \Phi(\boldsymbol{x}_{t+1}) - \Phi(\boldsymbol{x}_t) &= -\eta\partial\Phi(\boldsymbol{x}_t)\boldsymbol{v}_t + \mathcal{O}\left(L_{\partial\Phi}\beta^2\eta^2\rho^2\right), \\ \|\Phi(\boldsymbol{x}_{t+1}) - \Phi(\boldsymbol{x}_t)\|^2 &\leq 6L_{\partial\Phi}^2\beta^2\eta^2\rho^2. \end{aligned}$$

Throughout the proof, we will use the notation $\boldsymbol{\nabla}_t \coloneqq \partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))$. Then from the L_0-smoothness of $\overline{\mathsf{tr}}(\Phi)$, and the fact that $\Phi(\Phi(\boldsymbol{x}_t)) = \Phi(\boldsymbol{x}_t)$ it follows that

$$\begin{aligned}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) &= \overline{\mathsf{tr}}(\Phi(\Phi(\boldsymbol{x}_{t+1}))) - \overline{\mathsf{tr}}(\Phi(\Phi(\boldsymbol{x}_t))) \\ &\le \langle \boldsymbol{\nabla}_t, \Phi(\boldsymbol{x}_{t+1}) - \Phi(\boldsymbol{x}_t)\rangle + \frac{1}{2}L_0 \left\|\Phi(\boldsymbol{x}_{t+1}) - \Phi(\boldsymbol{x}_t)\right\|^2 \\ &\le \langle \boldsymbol{\nabla}_t, -\eta\partial\Phi(\boldsymbol{x}_t)\boldsymbol{v}_t\rangle + 3L_0L_{\partial\Phi}^2\beta^2\eta^2\rho^2 \,.\end{aligned}$$

Applying Lemma 9, we obtain

$$\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \le -\frac{\eta\rho^2}{2}\left\|\boldsymbol{\nabla}_t\right\|^2 + \frac{1}{3}L_0\left\|\boldsymbol{\nabla}_t\right\|\eta\rho^3 + 3L_0L_{\partial\Phi}^2\beta^2\eta^2\rho^2 \,.$$

Now for a constant $c > 0$, consider the following the parameter choice (8):

$$\eta = \frac{1}{cL_0L_{\partial\Phi}^2\beta^2}\delta\epsilon, \qquad \rho = \frac{1}{cL_0}\delta\sqrt{\epsilon}\,.$$

From this choice, it follows that

$$\begin{aligned}-\frac{\eta\rho^2}{2}\left\|\boldsymbol{\nabla}_t\right\|^2 &= -\frac{1}{2c^3L_0^3L_{\partial\Phi}^2\beta^2}\delta^3\epsilon^2\left\|\boldsymbol{\nabla}_t\right\|^2 \\ L_0\left\|\boldsymbol{\nabla}_t\right\|\eta\rho^3 &= \frac{1}{c^4L_0^3L_{\partial\Phi}^2\beta^2}\delta^4\epsilon^{2.5}\left\|\boldsymbol{\nabla}_t\right\|, \\ L_0L_{\partial\Phi}^2\beta^2\eta^2\rho^2 &= \frac{1}{c^4L_0^3L_{\partial\Phi}^2\beta^2}\delta^4\epsilon^3\end{aligned}$$

Hence, by choosing the constant c appropriately large, one can thus ensure that

$$\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \le \frac{1}{2c^3L_0^3L_{\partial\Phi}^2\beta^2}\delta^3\epsilon^2\left(-\left\|\boldsymbol{\nabla}_t\right\|^2 + \frac{1}{4}\delta\epsilon^{1/2}\left\|\boldsymbol{\nabla}_t\right\| + \frac{1}{4}\delta\epsilon\right).$$

This completes the proof of Lemma 10.

D.7. Proof of Lemma 11

For simplicity, let $\boldsymbol{g}_{i,t} \coloneqq \frac{\nabla f_i(\boldsymbol{x}_t)}{\|\nabla f_i(\boldsymbol{x}_t)\|}$. Note that by Taylor expansion, we have

$$\nabla f_i(\boldsymbol{x}_t + \rho\boldsymbol{g}_{i,t}) = \nabla f_i(\boldsymbol{x}_t) + \rho\nabla^2 f_i(\boldsymbol{x}_t)\sigma_t\boldsymbol{g}_{i,t} + \frac{1}{2}\rho^2\nabla^3 f_i(\boldsymbol{x}_t)\left[\boldsymbol{g}_{i,t}, \boldsymbol{g}_{i,t}\right] + \mathcal{O}\left(L_{\nabla^4 f_i}\rho^3\right)$$

Using the facts that $\partial\Phi(\boldsymbol{x}_t)\nabla f(\boldsymbol{x}_t) = \mathbf{0}$ (Lemma 6), we have $\partial\Phi(\boldsymbol{x}_t)\mathrm{Proj}^{\perp}_{\nabla f(\boldsymbol{x}_t)} = \partial\Phi(\boldsymbol{x}_t)$, so the above equation implies that

$$\partial\Phi(\boldsymbol{x}_t)\boldsymbol{v}_t = \partial\Phi(\boldsymbol{x}_t)\left[\nabla f_i(\boldsymbol{x}_t) + \rho\nabla^2 f_i(\boldsymbol{x}_t)\sigma_t\boldsymbol{g}_{i,t} + \frac{1}{2}\rho^2\nabla^3 f_i(\boldsymbol{x}_t)\left[\boldsymbol{g}_{i,t}, \boldsymbol{g}_{i,t}\right] + \mathcal{O}\left(L_{\nabla^4 f_i}\rho^3\right)\right].$$

Taking expectation on both sides, we have the first two terms above vanish because $\mathbb{E}\partial\Phi(\boldsymbol{x}_t)\nabla f_i(\boldsymbol{x}_t) = \partial\Phi(\boldsymbol{x}_t)\nabla f(\boldsymbol{x}_t) = \mathbf{0}$ and $\mathbb{E}[\sigma_t] = 0$. Thus, using the L_0-Lipschitzness of $\partial\Phi(\cdot)\nabla\,\mathrm{tr}(\nabla^2 f_i(\cdot)\boldsymbol{g}\boldsymbol{g}^\top)$ for a unit vector $\boldsymbol{g}$, we obtain

$$\begin{aligned}\partial\Phi(\boldsymbol{x}_t)\mathbb{E}\boldsymbol{v}_t &= \frac{1}{2}\rho^2\partial\Phi(\boldsymbol{x}_t)\mathbb{E}\nabla^3 f_i(\boldsymbol{x}_t)\left[\boldsymbol{g}_{i,t}, \boldsymbol{g}_{i,t}\right] + \mathcal{O}\left(L_{\partial\Phi}L_{\nabla^4 f_i}\rho^3\right) \\ &= \frac{1}{2}\rho^2\partial\Phi(\boldsymbol{x}_t)\nabla\mathbb{E}\,\mathrm{tr}\left(\nabla^2 f_i(\boldsymbol{x}_t)\boldsymbol{g}_{i,t}\boldsymbol{g}_{i,t}^\top\right) + \mathcal{O}\left(L_0\rho^3\right) \\ &= \frac{1}{2}\rho^2\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\mathbb{E}\,\mathrm{tr}\left(\nabla^2 f_i(\Phi(\boldsymbol{x}_t))\boldsymbol{g}_{i,t}\boldsymbol{g}_{i,t}^\top\right) + \mathcal{O}\left(\frac{1}{2}L_0\rho^2\left\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\right\|\right) + \mathcal{O}\left(L_0\rho^3\right) \\ &= \frac{1}{2}\rho^2\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\mathbb{E}\,\mathrm{tr}\left(\nabla^2 f_i(\Phi(\boldsymbol{x}_t))\boldsymbol{g}_{i,t}\boldsymbol{g}_{i,t}^\top\right) + \mathcal{O}\left(L_0\rho^3\right), \qquad (13)\end{aligned}$$

where the last line is due to (9), which implies $L_0\rho^2 \|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\| = o\left(L_0\rho^3\right)$ as $\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\| = \mathcal{O}\left(\epsilon_0\right) = o\left(\rho\right)$. As we discussed in Subsection 4.2, now the punchline of the proof is that at a minimum $\boldsymbol{x}^\star \in \mathcal{X}^\star$, the Hessian is given as

$$\nabla^2 f(\boldsymbol{x}^\star) = \frac{1}{n}\sum_{i=1}^n \left[\frac{\partial^2 \ell(z, y_i)}{\partial^2 z}\Big|_{z=p_i(\boldsymbol{x}^\star)} \nabla p_i(\boldsymbol{x}^\star)\nabla p_i(\boldsymbol{x}^\star)^\top\right] .$$

Hence, using the notations

$$\boldsymbol{u}_i(\boldsymbol{x}) \coloneqq \frac{\nabla p_i(\boldsymbol{x})}{\|\nabla p_i(\boldsymbol{x})\|} \quad \text{and} \quad \lambda_i(\boldsymbol{x}) = \frac{1}{n}\cdot\frac{\partial^2 \ell(z, y_i)}{\partial^2 z}\Big|_{z=p_i(\boldsymbol{x})} \cdot \|\nabla p_i(\boldsymbol{x})\|^2 ,$$

one can write the Hessians at a minimum $\boldsymbol{x}^\star \in \mathcal{X}^\star$ as

$$\nabla^2 f(\boldsymbol{x}^\star) = \sum_{i=1}^n \lambda_i(\boldsymbol{x}^\star)\boldsymbol{u}_i(\boldsymbol{x}^\star)\boldsymbol{u}_i(\boldsymbol{x}^\star)^\top \quad \text{and} \quad \nabla^2 f_i(\boldsymbol{x}^\star) = n\lambda_i(\boldsymbol{x}^\star)\boldsymbol{u}_i(\boldsymbol{x}^\star)\boldsymbol{u}_i(\boldsymbol{x}^\star)^\top, \forall i . \tag{14}$$

In particular, it follows that

$$\operatorname{tr}(\nabla^2 f(\boldsymbol{x}^\star)) = \sum_{i=1}^n \operatorname{tr}\left(\lambda_i(\boldsymbol{x}^\star)\boldsymbol{u}_i(\boldsymbol{x}^\star)\boldsymbol{u}_i(\boldsymbol{x}^\star)^\top\right) = \sum_{i=1}^n \lambda_i(\boldsymbol{x}^\star) . \tag{15}$$

Note that since $\nabla f_i(\boldsymbol{x}_t) = \frac{\partial \ell(z,y_i)}{\partial z}|_{z=p_i(\boldsymbol{x}_t)}\nabla p_i(\boldsymbol{x}_t)$, we have $\boldsymbol{g}_{i,t} = \frac{\nabla f_i(\boldsymbol{x}_t)}{\|\nabla f_i(\boldsymbol{x}_t)\|} = \frac{\nabla p_i(\boldsymbol{x}_t)}{\|\nabla p_i(\boldsymbol{x}_t)\|} = \boldsymbol{u}_i(\boldsymbol{x}_t)$. Using this fact together with the above expressions for the Hessians (14), one can further manipulate the expression for $\partial\Phi(\boldsymbol{x}_t)\mathbb{E}\boldsymbol{v}_t$ in (13) as follows:

$$\begin{aligned}
\partial\Phi(\boldsymbol{x}_t)\mathbb{E}\boldsymbol{v}_t &= \frac{1}{2}\rho^2\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\mathbb{E}\operatorname{tr}\left(n\lambda_i(\Phi(\boldsymbol{x}_t))\boldsymbol{u}_i(\Phi(\boldsymbol{x}_t))\boldsymbol{u}_i(\Phi(\boldsymbol{x}_t))^\top\boldsymbol{u}_i(\boldsymbol{x}_t)\boldsymbol{u}_i(\boldsymbol{x}_t)^\top\right) + \mathcal{O}\left(L_0\rho^3\right) \\
&\overset{(a)}{=} \frac{1}{2}\rho^2\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\mathbb{E}\left[n\lambda_i(\Phi(\boldsymbol{x}_t))(1 + L_1\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\|)^2\right] + \mathcal{O}\left(L_0\rho^3\right) \\
&= \frac{1}{2}\rho^2\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\left[\sum_{i=1}^n \lambda_i(\Phi(\boldsymbol{x}_t))(1 + L_1\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\|)^2\right] + \mathcal{O}\left(L_0\rho^3\right) \\
&= \frac{1}{2}d\rho^2\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\left[\frac{1}{d}\sum_{i=1}^n \lambda_i(\Phi(\boldsymbol{x}_t))(1 + L_1\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\|)^2\right] + \mathcal{O}\left(L_0\rho^3\right) \\
&\overset{(b)}{=} \frac{1}{2}d\rho^2\partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) + \mathcal{O}\left(L_1L_2d\rho^2\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\|\right) + \mathcal{O}\left(L_0\rho^3\right) ,
\end{aligned}$$

where in (a), we use the fact $\boldsymbol{u}_i(\boldsymbol{x}_t) = \boldsymbol{u}_i(\Phi(\boldsymbol{x}_t)) + \mathcal{O}\left(L_1\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\|\right)$, and $\boldsymbol{u}_i(\Phi(\boldsymbol{x}_t))$ is well-defined since we assumed that $\nabla p_i(\boldsymbol{x}) \neq \mathbf{0}$ for $\boldsymbol{x} \in \mathcal{X}^\star$, $\forall i = 1, \ldots, n$, and (b) is due to (15). This completes the proof since $\|\boldsymbol{x}_t - \Phi(\boldsymbol{x}_t)\| = \mathcal{O}\left(\epsilon_0\right)$ from the condition (12).

D.8. Proof of Lemma 12

Throughout the proof, we will use the notation $\boldsymbol{\nabla}_t \coloneqq \partial\Phi(\Phi(\boldsymbol{x}_t))\nabla\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t))$. Similarly to Subsection D.6, we have

$$\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \leq \langle \boldsymbol{\nabla}_t, -\eta\partial\Phi(\boldsymbol{x}_t)\boldsymbol{v}_t\rangle + \mathcal{O}\left(L_0L_{\partial\Phi}^2\beta^2\eta^2\rho^2\right) .$$

Applying Lemma 11, we then obtain

$$\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \leq -\frac{d\eta\rho^2}{2}\|\boldsymbol{\nabla}_t\|^2 + \mathcal{O}\left(L_1L_2d\eta\rho^2\epsilon_0\|\boldsymbol{\nabla}_t\| + L_0\|\boldsymbol{\nabla}_t\|\eta\rho^3 + L_0L_{\partial\Phi}^2\beta^2\eta^2\rho^2\right) .$$

Now for a constant $c > 0$, consider the following the parameter choice (10):

$$\eta = \frac{1}{cL_0L_{\partial\Phi}^2\beta^2}\nu\delta\epsilon, \qquad \rho = \frac{1}{cL_0}\nu\delta\sqrt{\epsilon}, \qquad \epsilon_0 = \frac{\beta^{1.5}}{c^3\alpha L_0^{1.5}L_{\partial\Phi}}\nu^{1.5}\delta^{1.5}\epsilon$$

From this choice, together with the fact $\nu^{1.5} = \min\{d, \epsilon^{-1/3}\}^{1.5} \leq \epsilon^{-1/2}$, it follows that

$$
\begin{aligned}
-\frac{\eta\rho^2}{2}\left\|\boldsymbol{\nabla}_t\right\|^2 &= -\frac{1}{2c^3L_0^3L_{\partial\Phi}^2\beta^2}d\nu^3\delta^3\epsilon^2\left\|\boldsymbol{\nabla}_t\right\|^2 \\
L_1L_2d\eta\rho^2\epsilon_0\left\|\boldsymbol{\nabla}_t\right\| &= \mathcal{O}\left(\frac{1}{c^6}d\nu^{4.5}\delta^{4.5}\epsilon^3\left\|\boldsymbol{\nabla}_t\right\|\right) = \mathcal{O}\left(\frac{1}{c^6}d\nu^3\delta^{4.5}\epsilon^{2.5}\left\|\boldsymbol{\nabla}_t\right\|\right) \\
L_0\left\|\boldsymbol{\nabla}_t\right\|\eta\rho^3 &= \frac{1}{c^4L_0^3L_{\partial\Phi}^2\beta^2}\nu^4\delta^4\epsilon^{2.5}\left\|\boldsymbol{\nabla}_t\right\|, \\
L_0L_{\partial\Phi}^2\beta^2\eta^2\rho^2 &= \frac{1}{c^4L_0^3L_{\partial\Phi}^2\beta^2}\nu^4\delta^4\epsilon^3
\end{aligned}
$$

Hence, using the fact that $\nu = \min\{d, \epsilon^{-1/3}\} \leq d$ and by choosing the constant c appropriately large, one can thus ensure that

$$
\mathbb{E}\overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_{t+1})) - \overline{\mathsf{tr}}(\Phi(\boldsymbol{x}_t)) \leq \frac{1}{2c^3L_0^3L_{\partial\Phi}^2\beta^2}d\nu^3\delta^3\epsilon^2\left(-\left\|\boldsymbol{\nabla}_t\right\|^2 + \frac{1}{4}\delta\epsilon^{0.5}\left\|\boldsymbol{\nabla}_t\right\| + \frac{1}{4}\delta\epsilon\right).
$$

This completes the proof of Lemma 12.