Noise Stability Optimization for Finding Flat Minima: A Hessian-based Regularization Approach

Anonymous authors
Paper under double-blind review

Abstract

The training of overparameterized neural networks has received much study in recent literature. An important consideration is the regularization of overparametrized networks due to the highly nonconvex geometry of these networks. In this paper, we consider noise injection algorithms, which can regularize the Hessian of the loss, leading to flat loss surfaces. Specifically, by injecting isotropic Gaussian noise into the weight matrices of a network, we will obtain an approximately unbiased estimate of the trace of the Hessian. However, naively implementing the noise injection, such as adding noise to the weights before back-propagation, presents limited empirical improvement. To address this limitation, we design a two-point noise injection scheme, which injects noise to weights along both positive and negative directions of the random noise. We show that this form of Hessian-based regularization can improve generalization by proving a PAC-Bayes bound that depends on the trace of the Hessian and the radius of the fine-tuning region.

Extensive experiments validate that our approach can effectively regularize the Hessian, thereby improving generalization. First, our algorithm can outperform prior sharpness-reducing training, achieving up to a 1.8% increase in test accuracy for fine-tuning pretrained ResNets on six image classification datasets. The trace of the Hessian is reduced by 17.7%, and the largest eigenvalue is reduced by 12.8%. Second, the noise injection algorithm can be combined with alternative regularization methods such as weight decay and data augmentation. Third, our approach can be used to improve generalization in pretraining CLIP models and chain-of-thought fine-tuning.

Lastly, we analyze the convergence of our algorithm. Our analysis expands on a connection between minimizing noise-injected functions and stochastic optimization, leading to sharp convergence rates of the above noise-injection algorithm.

1 Introduction

The loss landscape and its geometry properties are a recurring theme in the study of neural networks (Keskar et al., 2017; Dinh et al., 2017; Hochreiter & Schmidhuber, 1997). Recently, the design of training methods such as sharpness-aware minimization and stochastic weight averaging has led to improved empirical performance in a wide range of settings (Izmailov et al., 2018; Foret et al., 2021; Wortsman et al., 2022). The theoretical study of these training methods has also been explored (Andriushchenko & Flammarion, 2022). For instance, it has been shown that the sharpness-aware minimization algorithm (Foret et al., 2021) has an implicit bias to surface regions whose largest eigenvalue of the Hessian is small (Wen et al., 2023; Bartlett et al., 2023). In this paper, we study methods that have *explicit* regularization of the Hessian, with provable generalization guarantees. More formally, given an input function $f : \mathbb{R}^d \to \mathbb{R}$ that represents the empirical risk of a neural network and a d-dimensional distribution $\mathcal{P}$ with mean zero, we consider minimizing the noise-perturbed function

$$F(W) := \mathop{\mathbb{E}}_{U \sim \mathcal{P}} \left[f(W + U) \right].$$

Minimizing this perturbed function can improve the resilience of the neural network to noise injection, thus leading to flatter loss surfaces and improved regularization (Nagarajan & Kolter, 2020; Dziugaite &

Roy, 2017). For instance, using PAC-Bayes analysis, one can identify a measure of the "sharpness" of loss surfaces based on the trace of the Hessian (Tsuzuku et al., 2020; Ju et al., 2022). While this approach is theoretically motivated, its practical performance is not always evident (Hinton & Van Camp, 1993; An, 1996; Graves, 2011). To motivate our study, we start by conducting several empirical studies to compare the performance of standard SGD and weight-perturbed SGD (WP-SGD), which first injects random noise into the weights of the neural network before computing its gradient in SGD. For this empirical study, we fine-tune pretrained ResNets on three image classification tasks. To ensure the validity of the analysis, we vary both the distribution of $\mathcal{P}$ and the variance of U and find that WP-SGD does not offer clear benefits over SGD. Our study is consistent with recent studies such as Orvieto et al. (2023). However, we hypothesize that these results may be due to the randomness of the noise injection rather than the ineffectiveness of the Hessian regularization.

Our approach to mitigate the randomness of the noise injection involves two parts. First, we add a negative perturbation along $W - U$ to cancel out the first-order expansion term of $W + U$ (recall that U is a random sample from $\mathcal{P}$). Meanwhile, the second-order expansion term remains the same after this cancellation. We term this modification a two-point noise injection scheme, analogous to using two-point gradient estimates in zeroth-order optimization Duchi et al. (2015). Second, we sample multiple perturbations $U_1, U_2, \ldots, U_k$ at each epoch and take their averaged two-point gradients.

A major advantage of our approach compared to prior approaches is that we can provide an approximately unbiased estimate of the Hessian, and we empirically validate this claim across three real-world settings (see Figure 2, Section 2.3 for an illustration). By utilizing this property, we show a PAC-Bayes bound that depends on the trace of the Hessian and the radius of the fine-tuning region. We briefly describe this result, leaving the formal statement to Theorem 2.1. Let α be an upper bound on the trace of the Hessian measured within the hypothesis space. Let r be the radius of the fine-tuning region, measured in Euclidean geometry. Suppose there are n empirical samples from an unknown distribution. We show a generalization bound that scales as $O\big(\sqrt{\frac{\alpha r^2}{n}}\big)$. The proof utilizes a linear PAC-Bayes bound (Catoni, 2007; McAllester, 2013), but we optimize the variance of the prior and posterior distributions to derive this result. A detailed proof sketch is presented in Section 2.4.

Next, we validate our approach through comprehensive experiments. First, when fine-tuning pretrained ResNets on six image classification data sets, our algorithm reduces the trace and the largest eigenvalue of the loss's Hessian matrix by 17.7% and 12.8%, respectively, compared to existing sharpness-reducing training methods. This results in up to 1.8% improvement in test accuracy over these existing methods. Second, combining our algorithm with other regularization techniques, such as data augmentation and distance-based regularization (Gouk et al., 2022), can further achieve a reduction of 13.6% in Hessian trace and 16.3% in test loss on average. Third, we apply our algorithm to multimodal model pretraining and chain-of-thought fine-tuning. Our algorithm consistently yields a lower Hessian trace and improved test performance than SAM in the applications.

Lastly, we analyze the convergence of our algorithm. In particular, we study the optimization properties of minimizing noise-perturbed function $F(W)$ using techniques from the stochastic optimization literature (Ghadimi & Lan, 2013; Lan, 2020; Zhang, 2023; Carmon et al., 2020; Drori & Shamir, 2020). Altogether, we can provide matching upper and lower bounds on the norm of the gradient of the iterates. Our analysis also raises several new questions, which may be interesting for future work. For instance, can accelerated gradient descent methods be applied to design flat-minima optimizers? Can recent advances in zeroth-order optimization be leveraged to better regularize the training of transformer neural networks?

In summary, the contributions of this paper are three-fold: 1) Presenting an algorithm that provides explicit regularization of the trace of the Hessian, along with a PAC-Bayes bound that guarantees its generalization effect. 2) Conducting experiments in a wide range of settings to validate our approach, compared to prior sharpness-reducing training methods, and alternative regularization methods. 3) Analyzing the convergence of the proposed algorithm based on techniques from the stochastic optimization literature. In Table 1, we highlight the key aspects of our approach as compared to prior approaches.

Table 1: Comparison between our approach (NSO) and SAM (Foret et al., 2021). In particular, the inductive bias of SAM is taken from Wen et al. (2023). Here is a list of notations: α is the trace norm, taken over the maximum of the entire data distribution; r is the radius of the fine-tuning region measured via Euclidean distance; n is the number of samples in the training dataset; T is the total number of iterations run by our algorithm.

Methods	Inductive Bias	Generalization Guarantee	Convergence Rate
Sharpness-Aware Minimization (SAM)	$\lambda_1[\nabla^2 \ell]$	-	-
Noise Stability Optimization (NSO)	$\mathrm{Tr}[\nabla^2 \ell]$	$\sqrt{\frac{\alpha r^2}{n}}$ (Theorem 2.1)	$O(\sqrt{\frac{1}{T}})$ (Theorem 4.2)

Organization: The rest of this paper is organized as follows. In Section 2, we will present our approach. We will start by presenting the motivating experiments. Then, we describe our algorithm and a PAC-Bayes bound that depends on the Hessian. In Section 3, we present our experiments for validating the proposed approach. In Section 4, we present an analysis of the convergence of our algorithm. In Section 5, we provide a preliminary study of the Hessian-based regularization effect in an overparameterized matrix sensing problem. In Section 6, we discuss the related works. Finally, in Section 7, we state the conclusion. In Appendix A and Appendix B, we provide complete proofs of our theoretical results. In Appendix C, we provide additional experimental results left from the main text.

2 Our Approach

In this section, we present our approach. First, to set up the stage, we will study the straightforward implementation of noise injection by directly adding noise to the weights of the network before computing gradients in backpropagation. We term this procedure as weight-perturbed SGD (or WP-SGD in short). Then, we describe our approach, along with a PAC-Bayes generalization bound to justify our approach. Lastly, we empirically measure the theoretical bound and compare the measurements with the true generalization gaps observed in practice.

2.1 Motivating Experiments

In this subsection, we will compare WP-SGD with standard SGD for fine-tuning pretrained models. We focus on this setting because overfitting has been commonly observed (Wortsman et al., 2022). Thus, developing training methods to improve generalization would be crucial. We consider fine-tuning a pre-trained ResNet-34 on image classification datasets, including an aircraft recognition task (Aircraft) (Maji et al., 2013), indoor scene recognition (Caltech-256) (Griffin et al., 2007), and medical image classification (retina images for diabetic retinopathy classification) (Pachade et al., 2021). In WP-SGD, we sample a perturbation vector from $\mathcal{P}$ and add it to the model weights in each iteration before computing the gradient. For WP-SGD, we will sample the perturbation from an isotropic Gaussian distribution. Then, we will set the standard deviation of U via cross-validation, choosing between $0.008, 0.01$, and 0.012.

We report our findings in Table 2. We observe that the performance gap between SGD and WP-SGD is less than 0.5%, within 0.75 standard deviations of the independent tests on average. Furthermore, varying the type of noise distribution does not change the result. In particular, we test choices of $\mathcal{P}$ with Laplace distribution, uniform distribution, and Binomial distribution. Similar to the Gaussian, we set their standard deviations between $0.008, 0.01$, and 0.012 using a validation set. Lastly, using the Laplace or Uniform distribution achieves a performance comparable to Gaussian. However, WP-SGD struggles to converge using the Binomial distribution, resulting in significantly lower training and test results.

2.2 Description of Our Algorithm

The above experiment suggests that the straightforward implementation of noise injection does not bring apparent benefits compared to SGD. In our approach, we make two modifications:

Table 2: Comparing WP-SGD with standard SGD across four types of perturbation distributions, measured over three image classification datasets. The results and their standard deviations are averaged over five independent seeds. Recall that WP-SGD refers to normal weight perturbation (without the paired perturbation). Note that the description of NSO will be presented below; However, we include the results in this Table for ease of comparison.

		Aircraft		Indoor		Retina Disease	
	$\mathcal{P}$	**Train Acc.**	**Test Acc.**	**Train Acc.**	**Test Acc.**	**Train Acc.**	**Test Acc.**
SGD	None	100.0% ± 0.0	59.8% ± 0.7	100.0% ± 0.0	76.0% ± 0.4	100.0% ± 0.0	61.7% ± 0.8
WP-SGD	Gaussian	98.4% ± 0.2	60.4% ± 0.1	99.0% ± 0.3	76.3% ± 0.0	100.0% ± 0.0	62.3% ± 0.5
WP-SGD	Laplace	98.3% ± 0.1	60.3% ± 0.3	98.9% ± 0.1	76.4% ± 0.3	100.0% ± 0.0	62.0% ± 0.1
WP-SGD	Uniform	98.6% ± 0.3	60.3% ± 0.5	98.6% ± 0.3	76.6% ± 0.1	100.0% ± 0.0	62.3% ± 0.0
WP-SGD	Binomial	19.6% ± 0.1	11.3% ± 0.1	18.2% ± 0.9	10.7% ± 0.1	58.1% ± 0.1	57.1% ± 0.0
NSO	Gaussian	95.8% ± 0.4	62.3% ± 0.3	95.7% ± 0.2	77.4% ± 0.3	100.0% ± 0.0	66.6% ± 0.7
NSO	Laplace	96.5% ± 0.3	61.9% ± 0.3	96.1% ± 0.3	77.1% ± 0.1	100.0% ± 0.0	65.9% ± 0.1
NSO	Uniform	96.4% ± 0.4	61.9% ± 0.5	96.4% ± 0.2	76.8% ± 0.2	100.0% ± 0.0	65.7% ± 0.1
NSO	Binomial	20.1% ± 0.1	14.3% ± 0.3	22.8% ± 0.1	17.9% ± 0.2	59.2% ± 0.1	57.8% ± 0.1

- *Two-point noise injection:* During the noise injection, we add the perturbation from both the positive and negative directions. This is shown in Line 5.
- *Averaging multiple perturbations to stabilize the gradient:* To stabilize the stochasticity of the noise injection, we average over multiple noise injections. This is described in Line 7.

To justify the first modification, recall that $\mathcal{P}$ is a symmetric distribution. We use Taylor's expansion on both $f(W+U)$ and $f(W-U)$:

$$f(W+U) = f(W) + \langle U, \nabla f(W)\rangle + \frac{1}{2}U^\top \nabla^2 f(W) U + O(\|\Sigma\|_2^{\frac{3}{2}}),$$
$$f(W-U) = f(W) - \langle U, \nabla f(W)\rangle + \frac{1}{2}U^\top \nabla^2 f(W) U + O(\|\Sigma\|_2^{\frac{3}{2}}).$$

By definition, $\mathbb{E}[U] = 0$, and $\mathbb{E}\left[UU^\top\right] = \Sigma$. Thus, by taking the average of the above two equations, we can get that

$$\mathop{\mathbb{E}}_{U\sim\mathcal{P}}\left[\frac{1}{2}(f(W+U)+f(W-U))\right] = F(W) = f(W) + \frac{1}{2}\langle \Sigma, \nabla^2 f(W)\rangle + O\left(\|\Sigma\|_2^{\frac{3}{2}}\right). \tag{1}$$

The second modification reduces the variance of the gradient, using the fact that each perturbation is independent from the others. The entire procedure is summarized in Algorithm 1. As a remark, two-point gradient estimators are commonly used in zeroth-order convex optimization (Duchi et al., 2015). The use to design flat-minima optimizer appears novel to our knowledge.

Algorithm 1 Noise stability optimization (NSO) for regularizing the Hessian of neural networks

Input: Initialization $W_0 \in \mathbb{R}^d$, a function $f : \mathbb{R}^d \to \mathbb{R}$
Require: An estimator $g : \mathbb{R}^d \to \mathbb{R}^d$ that for any W, returns $g(W)$ s.t. $\mathbb{E}[g(W)] = \nabla f(W)$
Parameters: # perturbations k, # epochs T, step sizes $\eta_0, \ldots, \eta_{T-1}$

1: **for** $i = 0, 1, \ldots, T-1$ **do**
2: /* *Compute the two-point averaged gradient over each independent noise injection* */
3: **for** $j = 0, 1, \ldots, k-1$ **do**
4: Sample $U_i^{(j)}$ independently from $\mathcal{P}$
5: Let $G_i^{(j)} = g\big(W_i + U_i^{(j)}\big) + g\big(W_i - U_i^{(j)}\big)$
6: **end for**
7: Update iterates according to $W_{i+1} = W_i - \frac{\eta_i}{2k}\sum_{j=1}^k G_i^{(j)}$
8: **end for**

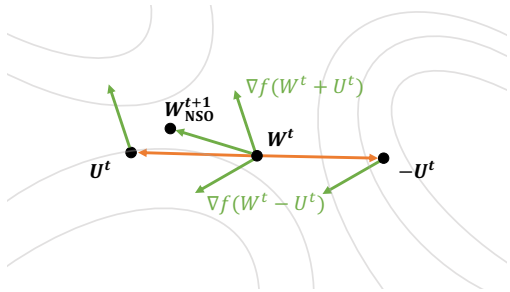


Figure 1: Illustration of the parameter update in our algorithm.

Generalization Guarantees: Next, we present a PAC-Bayes bound, which depends on the trace of the Hessian as part of the bound on the generalization gap. As a remark, the trace norm has been studied by earlier work in the setting of matrix recovery (Srebro & Shraibman, 2005).

Concretely, we have a pretrained model in the fine-tuning setting, which can be viewed as the prior in PAC-Bayes analysis. Once we have learned a hypothesis, it can be viewed as the posterior. Let $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$ be an unknown data distribution, supported on the feature space $\mathcal{X}$ and the label space $\mathcal{Y}$. Given n random samples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ drawn from $\mathcal{D}$, the empirical loss (measured by loss function ℓ) applied to a model f_W (with $W \in \mathbb{R}^p$) is:

$$\hat{L}(W) = \frac{1}{n} \sum_{i=1}^{n} \ell(f_W(x_i), y_i).$$

The population loss is $L(W) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\ell(f_W(x), y)\right]$. It is sufficient to think that the empirical loss is less than the population loss, and the goal is to bound the gap from above (Shalev-Shwartz & Ben-David, 2014).

Let W be any learned hypothesis within the hypothesis space, denoted as $\mathcal{H}$. The generalization bound will apply uniformly to W within the hypothesis space, assuming that this space, centered at the pretrained initialization, has a bounded radius of $r > 0$. We state the result as follows.

Theorem 2.1. *Assume that the loss function is bounded between* 0 *and* C *for a fixed constant* C*. Suppose that* $\ell(f_W(\cdot), \cdot)$ *is twice-differentiable in* W *and the Hessian matrix* $\nabla^2[\ell(f_W(\cdot), \cdot)]$ *is Lipschitz continuous within the hypothesis space. With probability at least* $1 - \delta$ *for any* $\delta > 0$*, the following must hold, for any* ϵ *close to zero:*

$$L(W) \leq (1+\epsilon)\hat{L}(W) + (1+\epsilon)\sqrt{\frac{C\alpha r^2}{n}} + O\Big(n^{-\frac{3}{4}} \log(\delta^{-1})\Big). \tag{2}$$

where the trace norm of the hypothesis space taken over the data distribution $\mathcal{D}$ *is given by*

$$\alpha := \max_{W \in \mathcal{H}} \max_{(x,y)\sim\mathcal{D}} \operatorname{Tr}\left[\nabla^2 \ell(f_W(x), y)\right].$$

2.3 Measurements of Hessian and Generalization

Next, we provide several empirical examples to validate the above theoretical bounds. Following the experimental setup described earlier, we fine-tune several pretrained models on one downstream task. We test on three different modalities of data, including images, texts, and graphs. After fine-tuning, we set the fine-tuned model weight at the last epoch as W for taking all the measurements. We summarize the empirical findings below, leaving experimental details to Appendix C. First, we show that Taylor's expansion of the noise injection is numerically accurate. We add perturbations to model weights by injecting isotropic Gaussian noise. We then compute the perturbed loss minus the original loss value, averaged over 100 independent runs, and we measure the trace of the Hessian as the average over the training dataset.

In Figure 2, we find that the trace of the Hessian provides an accurate approximation to the gap between $\ell_{\mathcal{Q}}$ and ℓ (recall that $\ell_{\mathcal{Q}}$ is defined in equation (3)). After fine-tuning, we add random noise injections to the

fine-tuned model weight. We do this for 100 times and again measure the perturbed loss $\ell_{\mathcal{Q}}$ on the training set. We take the gap between $\ell_{\mathcal{Q}}$ and ℓ and report that along with the magnitude of σ in the Table. We also compute the trace of the Hessian using Hessian-vector product computation libraries. Our measurements show that the error between the actual gap and the Hessian approximation is within 3%. As a remark, the range of σ^2 differs across architectures because of the differing scales of their weights.

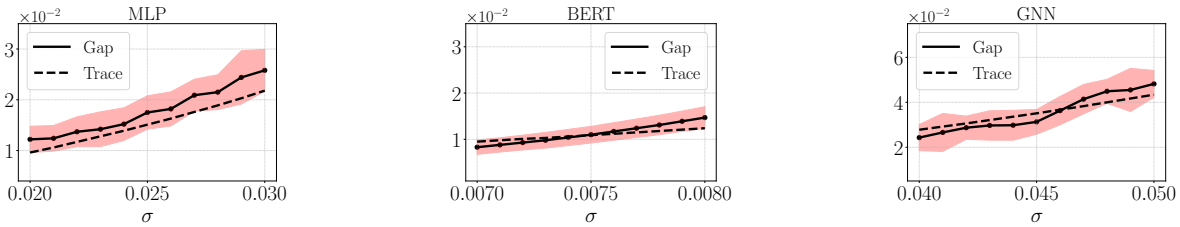


Figure 2: Illustration of the approximation to the gap between the perturbed loss $\ell_{\mathcal{Q}}$ and ℓ using the trace of the Hessian. The measurements are taken over the fine-tuned model weight W at the last epoch. σ refers to the standard deviation of the Gaussian noise injected to the model weights.

2.4 Proof Sketch of Theorem 2.1

We provide a high-level illustration of the ideas behind Theorem 2.1 without belaboring too much on the technical details. Let $\mathcal{Q}$ denote the *posterior* distribution. Specifically, we consider $\mathcal{Q}$ as being centered at the learned hypothesis W (which could be anywhere within the hypothesis space), given by a Gaussian distribution $\mathcal{N}(W, \sigma^2 \operatorname{Id}_p)$, where Id_p denotes the p by p identify matrix. Given a sample $U \sim \mathcal{N}(0, \sigma^2 \operatorname{Id}_p)$, let the perturbed loss be given by

$$\ell_{\mathcal{Q}}(f_W(x), y) = \mathop{\mathbb{E}}_{U}\left[\ell(f_{W+U}(x), y)\right]. \tag{3}$$

Then, let $\hat{L}_{\mathcal{Q}}(W)$ be the averaged value of $\ell_{\mathcal{Q}}(f_W(\cdot), \cdot)$, taken over the n empirical samples. Likewise, let $L_{\mathcal{Q}}(W)$ be the population average of $\ell_{\mathcal{Q}}(f_W(\cdot), \cdot)$.

Having introduced the notations, we start with the PAC-Bayes bound (Catoni, 2007; McAllester, 2013; Alquier, 2021) (see Theorem A.1 for reference), stated as follows:

$$L_{\mathcal{Q}}(W) \leq \frac{1}{\beta}\hat{L}_{\mathcal{Q}}(W) + \frac{C(KL(\mathcal{Q}||\mathcal{P}) + \log(\delta^{-1}))}{2\beta(1-\beta)n}, \tag{4}$$

where β is a parameter chosen between $(0, 1)$, and $\mathcal{P}$ is a *prior* distribution. For the fine-tuning setting, $\mathcal{P}$ can be viewed as centered at the pretrained initialization, with variance $\sigma^2 \operatorname{Id}_p$ similar to $\mathcal{Q}$.

Next, by Taylor's expansion of $\ell_{\mathcal{Q}}$ (see Lemma A.4 for the full result), we show that:

$$L_{\mathcal{Q}}(W) = L(W) + \frac{\sigma^2}{2} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\operatorname{Tr}\left[\nabla^2 \ell(f_W(x), y)\right]\right] + O(\sigma^3), \text{ and}$$

$$\hat{L}_{\mathcal{Q}}(W) = \hat{L}(W) + \frac{\sigma^2}{2n}\sum_{i=1}^{n} \operatorname{Tr}\left[\nabla^2 \ell(f_W(x_i), y_i)\right] + O(\sigma^3).$$

Since the Hessian operator is Lipschitz continuous by the assumptions of Theorem 2.1, we can bound the gap between the above two using uniform convergence (see Lemma A.5 for the result). By plugging in the above two results back to the PAC-Bayes bound of equation (4), and making up the difference between $1/\beta$ and 1 between the left and right sides by α, we get:

$$L(W) \leq \frac{1}{\beta}\hat{L}(W) + \frac{\sigma^2(1-\beta)\alpha}{2\beta} + \frac{Cr^2/2\sigma^2}{2\beta(1-\beta)n} + O\left(\sigma^3 + \frac{\sigma^2\sqrt{p}}{\sqrt{n}} + \frac{\log(\delta^{-1})}{n}\right).$$

In particular, the above uses the fact that the hypothesis space is uniformly bounded in a ball of radius r, and the derivation of the KL divergence can be found in Proposition A.2. By choosing σ^2 and β to minimize the above bound, we thus obtain the result of equation (2). This summarizes the high-level proof idea. The complete proof can be found in Appendix A.1.

3 Experiments

We now turn to the empirical validation of our proposed algorithm. Through extensive experiments, we show that our algorithm can indeed improve generalization, and this improvement can be explained by the regularization of the Hessian.

First, we apply our approach to fine-tune pretrained ResNets on various image classification datasets. We find that NSO can regularize the Hessian of the loss surface much more significantly. We note reductions in the trace and the largest eigenvalue of the loss Hessian by **17.7**% and **12.8**%, respectively. We notice that NSO can outperform four previous sharpness-reducing methods by up to **1.8**%. We control the amount of computation in the experiments to allow for a fair comparison. We justify each step of the algorithm design through ablation analysis.

Our method is compatible with alternative regularization techniques, including distance-based regularization and data augmentation. Combining these methods with our approach leads to even more significant improvement in both the Hessian regularization and the test performance.

Lastly, we show that our algorithm can also regularize the Hessian trace and improve the generalization when applied to pretraining contrastive language-image models and fine-tuning language models on chain-of-thought reasoning datasets.

3.1 Comparison with Sharpness Reducing Training Methods

We now compare Algorithm 1 with five sharpness-reducing training methods, including Sharpness-Aware Minimization (SAM) (Foret et al., 2021), Unnormalized SAM (USAM) (Agarwala & Dauphin, 2023), Adaptive SAM (ASAM) (Kwon et al., 2021), Random SAM (RSAM) (Liu et al., 2022), and Bayesian SAM (BSAM) (Möllenhoff & Khan, 2023). During comparison, we control for the same amount of computation by setting the number of sampled injections $k = 1$. Thus, all of these methods will cost twice the computation of SGD. For NSO, we sample perturbation from an isotropic Gaussian distribution and tune σ between $0.008, 0.01$, and 0.012 using a validation split. For SAM, we tune the ℓ_2 norm of the perturbation between $0.01, 0.02$, and 0.05. Since each other training method involves its own set of hyper-parameters, we ensure they are carefully selected. The details are tedious; See Appendix C for the range of values used for each hyper-parameter. We include SGD and Label Smoothing (LS) to calibrate these results, as they are both widely used in practice.

We report the overall comparison in Table 3. Notably, in the upper tables, NSO can significantly reduce the trace of Hessian compared to the baselines by **17.7**% on average. NSO also reduces the largest eigenvalue of the Hessian by **12.8**% on average, as reported in Table 10. Moreover, NSO performs competitively with all the baseline variants. Across these six datasets, NSO can achieve up to **1.8**% accuracy gain, with an average test accuracy improvement of **0.9**%, relative to the best-performing baselines. The results are aggregated over five independent runs, suggesting our findings are statistically significant.

In addition, we compare the measurements between SGD and NSO across three settings in Figure 3. Curiously, we find that the trace of the Hessian also decreases for SGD. While both SGD and NSO reduce the trace of the Hessian, our approach indeed penalizes the Hessian more significantly than SGD. Compared with SGD, the generalization gap of the fine-tuned model also lowers by over **20**%. The test loss of the fine-tuned model using our approach is also lower than SGD.

In the above experiments, we do not add momentum and weight decay in NSO. Next, we compare NSO and SAM when they are both used. The results are reported in Table 4. We find that NSO can still reduce the trace of the Hessian by an average of 23% than SAM. Moreover, NSO achieves better 1.4% test accuracy than SAM in this setting.

Table 3: Comparison between NSO, SGD, Label Smoothing (LS), SAM, Unnormalized SAM (USAM), Adaptive SAM (ASAM), Random-SAM (RSAM), and Bayesian SAM (BSAM) on six image classification datasets, by fine-tuning a pretrained ResNet-34 neural network using each method. In this Table, we report the test accuracy and the trace of the Hessian (for model weight found in the last epoch of each training algorithm). Lower trace values indicate wider loss surfaces. In all test cases, we report the averaged result over five random seeds and the standard deviation across these five runs. The results indicate that NSO outperforms the baselines in the three metrics.

		CIFAR-10	**CIFAR-100**	**Aircrafts**	**Caltech-256**	**Indoor**	**Retina**
Basic Stats	Train	45,000	45,000	3,334	7,680	4,824	1,396
	Val.	5,000	5,000	3,333	5,120	536	248
	Test	10,000	10,000	3,333	5,120	1,340	250
	Classes	10	100	100	256	67	5
Trace (↓)	SGD	4740 ± 64	14493 ± 359	6218 ± 63	4129 ± 94	4078 ± 78	30433 ± 217
	LS	2924 ± 81	11350 ± 499	6332 ± 76	3827 ± 83	4196 ± 36	19219 ± 119
	SAM	2827 ± 97	10225 ± 428	5034 ± 59	3849 ± 71	3789 ± 49	16411 ± 161
	USAM	2614 ± 88	7854 ± 216	4830 ± 98	3567 ± 55	3269 ± 99	13262 ± 372
	ASAM	2896 ± 64	10596 ± 339	5191 ± 32	3890 ± 97	3124 ± 73	14745 ± 131
	RSAM	2755 ± 80	10386 ± 577	5579 ± 79	3550 ± 57	4162 ± 60	19945 ± 365
	BSAM	3076 ± 51	10323 ± 567	5625 ± 61	3975 ± 12	3553 ± 72	18245 ± 318
	NSO	**2228** ± 94	**5934** ± 74	**4193** ± 46	**3354** ± 94	**2991** ± 32	**11554** ± 77
Test Accuracy (↑)	SGD	95.5% ± 0.1	82.3% ± 0.1	59.8% ± 0.7	75.5% ± 0.1	76.0% ± 0.4	61.7% ± 0.8
	LS	96.7% ± 0.1	83.8% ± 0.1	58.5% ± 0.2	76.0% ± 0.2	75.9% ± 0.3	63.6% ± 0.7
	SAM	96.6% ± 0.4	83.5% ± 0.1	61.5% ± 0.8	76.3% ± 0.1	76.6% ± 0.5	64.4% ± 0.6
	USAM	96.5% ± 0.0	83.2% ± 0.2	61.4% ± 0.6	76.1% ± 0.0	76.3% ± 0.3	62.8% ± 0.1
	ASAM	96.7% ± 0.1	83.8% ± 0.1	62.0% ± 0.6	76.7% ± 0.2	76.7% ± 0.3	64.8% ± 0.3
	RSAM	96.4% ± 0.1	83.7% ± 0.2	60.5% ± 0.5	75.8% ± 0.2	76.1% ± 0.7	65.4% ± 0.3
	BSAM	96.4% ± 0.0	83.5% ± 0.2	60.5% ± 0.5	76.3% ± 0.3	75.7% ± 0.7	64.9% ± 0.0
	NSO	**97.1%** ± 0.2	**84.3%** ± 0.2	**62.3%** ± 0.3	**77.4%** ± 0.3	**77.4%** ± 0.5	**66.6%** ± 0.7

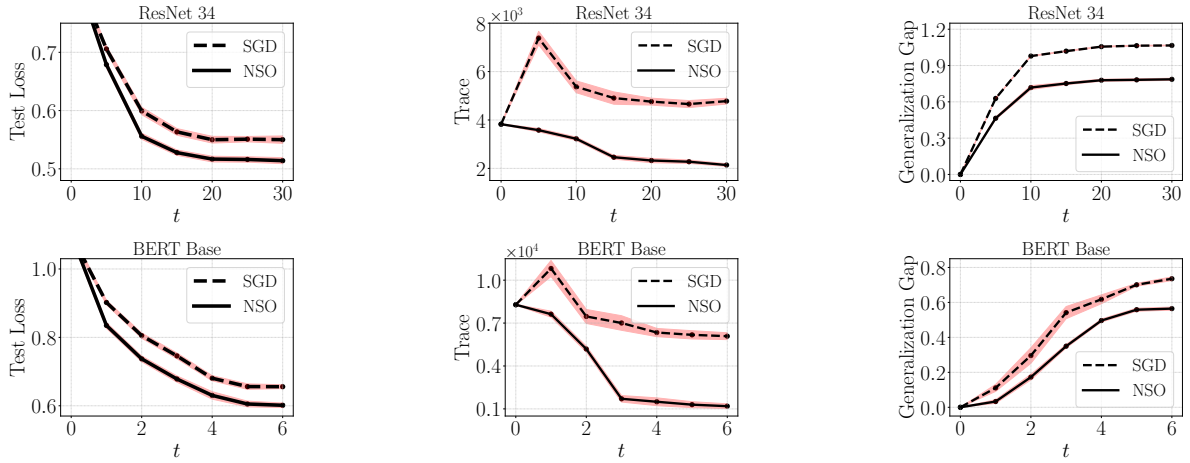


Figure 3: Comparison between SGD and NSO, for fine-tuning ResNet-34 and BERT-Base, respectively, on an image and a text classification dataset. We report the test loss, the trace of the Hessian, and the generalization gap for W taken at the last epoch. For NSO, we sample random perturbations using isotropic Gaussian distribution with standard deviation $\sigma = 0.01$ for both settings.

3.1.1 Dissecting the Design of Algorithm 1

Next, we conduct ablation studies of two components in NSO, i.e., using negative perturbations and sampling multiple perturbations in each iteration, showing both are essential.

Table 4: Comparison between NSO and SAM after using both momentum and weight decay. We fine-tune the ResNet-34 network on six image classification datasets and report the test accuracy and the trace of Hessian using the model in the last epoch of training. The results are averaged over five random seeds.

		CIFAR-10	**CIFAR-100**	**Aircrafts**	**Caltech-256**	**Indoor**	**Retina**
Trace	SAM	2429 ± 87	9227 ± 286	4499 ± 70	3285 ± 95	3159 ± 75	15444 ± 173
(↓)	NSO	1728 ± 79	5244 ± 89	3678 ± 83	2958 ± 77	2737 ± 90	10970 ± 146
Test Acc.	SAM	97.0% ± 0.2	84.0% ± 0.4	62.3% ± 0.3	77.0% ± 0.4	77.2% ± 0.3	65.0% ± 0.3
(↑)	NSO	97.6% ± 0.4	84.9% ± 0.3	63.2% ± 0.3	78.1% ± 0.5	78.2% ± 0.3	67.0% ± 0.4

Comparing using vs. not using negative perturbations, after controlling computation costs: Recall that our algorithm uses negative perturbations to zero out the first-order order in Taylor's expansion of $F(W)$, leading to a better estimation of $\nabla F(W)$. We validate this by comparing the performance between using and not using the negative perturbation. To ensure that both use the same amount of computation, we sample two independent perturbations when not using negative perturbations. We find that using negative perturbations achieves a **3.6**% improvement in test accuracy on average over the one without negative perturbations.

Increasing the number of noise injections k: Recall that increasing the number of perturbations k can reduce the variance of the estimated gradient. Thus, we consider increasing k in NSO and compare that with WP-SGD with comparable computation. We find that using $k = 2$ perturbations improves the test accuracy by **1.2**% on average compared to $k = 1$. However, in our experiments, increasing k over 3 brings no obvious improvement, but adds more computation costs.

Discussion of noise variance scheduling. Besides setting σ as a constant value, one can gradually increase the regularization strength by increasing the noise level in NSO during training. Analogous to learning rate schedules, we tested two schedules for adjusting σ. The first schedule is linearly increasing σ to a specified value. The second is exponential increasing σ to reach a specified value. In our experiments with image classification datasets, we found that neither schedule offered any performance improvement over NSO with a constant noise variance.

3.1.2 A More Detailed Comparison to Sharpness-Aware Minimization (SAM)

Varying the radius of SAM: Next, we show that the range of perturbation radius of SAM is comprehensively tuned in our experiments. To illustrate the effect of the radius, we vary the radius of SAM at 0.001, 0.002, 0.005, 0.01, 0.02, and 0.05. We report the test accuracy and the trace of the Hessian for both SAM and unnormalized SAM on an image classification dataset. The results are shown in Table 5. We observe that using a smaller radius (less than 0.01) results in larger values of Hessian trace and lower test accuracy. Thus, in the other experiments, we search for the radius between 0.01, 0.02, and 0.05.

Table 5: Results of varying the perturbation radius of SAM and unnormalized SAM when fine-tuning the ResNet-34 network on image classification datasets. We report the test accuracy and the trace of Hessian using the model in the last epoch of training. The results are averaged over five random seeds.

	ρ	0.001	0.002	0.005	0.01	0.02	0.05
Trace	SAM	4920 ± 158	4347 ± 166	4016 ± 80	3918 ± 94	3789 ± 49	**3658** ± 48
(↓)	Unnormalized SAM	4352 ± 169	3990 ± 70	3723 ± 87	3427 ± 57	3258 ± 39	**3538** ± 64
Test Accuracy	SAM	73.6 ± 0.2	74.4 ± 0.4	74.8 ± 0.6	75.2 ± 0.3	**76.6** ± 0.5	73.8 ± 0.7
(↑)	Unnormalized SAM	74.1 ± 0.1	74.1 ± 0.7	74.7 ± 0.5	74.6 ± 0.3	**76.3** ± 0.3	73.1 ± 0.6

Varying batch size: Further, we evaluate the sensitivity of NSO to batch size. We vary batch size between 8, 16, 32, and 64 in fine-tuning ResNet-34 on two image classification datasets. Table 6 reports the Hessian traces and test accuracies for both NSO and SAM. We use the same number of gradient update steps for

each batch size. We observe that NSO is less sensitive to batch size variations than SAM. Moreover, across different batch sizes, NSO consistently achieves lower Hessian traces and better test performance than SAM. The best performance was observed with a batch size of 32, which we used in the experiments.

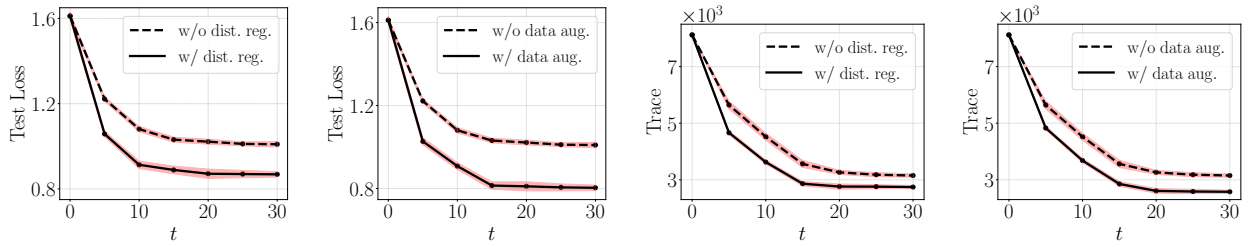
Table 6: Results of varying the batch size of NSO and SAM when fine-tuning the ResNet-34 network on two image classification datasets. We report the test accuracy and the trace of Hessian using the model from the last epoch of training. The results are averaged over five random seeds.

Dataset: Indoor	Batch size	8	16	32	64
Trace	SAM	3213 ± 94	3521 ± 64	3789 ± 49	4441 ± 62
(↓)	NSO	2757 ± 55	2888 ± 57	2991 ± 32	3325 ± 89
Test Accuracy	SAM	69.7 ± 0.6	73.9 ± 0.5	76.6 ± 0.5	73.4 ± 0.3
(↑)	NSO	70.9 ± 0.2	74.2 ± 0.3	77.4 ± 0.5	75.3 ± 0.5
Dataset: Aircrafts	Batch size	8	16	32	64
Trace	SAM	4453 ± 87	4643 ± 97	5034 ± 59	6591 ± 63
(↓)	NSO	3835 ± 82	4186 ± 84	4193 ± 46	4458 ± 61
Test Accuracy	SAM	57.1 ± 0.4	60.4 ± 0.7	61.5 ± 0.8	58.3 ± 0.5
(↑)	NSO	58.4 ± 0.1	61.9 ± 0.5	62.3 ± 0.3	59.5 ± 0.3

3.1.3 Combining Hessian regularization with Alternative Regularization Methods

In this section, we show that the regularization of the Hessian can complement alternative regularization methods. To validate this, we combine NSO with another regularization method. For distance-based regularization, we penalize the ℓ_2 distance from the fine-tuned model to the pre-trained initialization (Gouk et al., 2022). For data augmentation, we use a popular scheme that sequentially applies random horizontal flipping and random cropping to each training image.

The results are shown in Figure 4. We confirm that combining our algorithm with each regularization method further reduces the trace of the loss Hessian matrix by **13.6**% on average. Quite strikingly, this further leads to **16.3**% lower test loss of the fine-tuned neural network on average, suggesting that our method can be used on top of these alternative regularization methods.



(a) Test loss, Distance-based regularization

(b) Test loss, Data augmentation

(c) Hessian, Distance-based regularization

(d) Hessian, Data augmentation

Figure 4: The Hessian regularization is compatible with ℓ_2 distance-based regularization and data augmentation. We illustrate this for fine-tuning a pre-trained ResNet-34 neural network on an image classification dataset. Combining each regularization method with ours generally leads to lower test losses and lowers the trace of the Hessian of the loss surface. Note that the shaded area indicates the deviation across five independent runs, suggesting the statistical significance of these findings.

3.2 Results for Pretraining

Next, we apply our approach to pretraining randomly-initialized models from scratch. We apply NSO to pretraining contrastive language-image (CLIP) models (Radford et al., 2021) on a dataset of image-caption pairs. We use the Conceptual Caption dataset (Sharma et al., 2018), which contains 3.3 million image caption pairs. Each caption briefly describes the corresponding image, with ten tokens on average. We use a 12-layer Vision Transformer as the image encoder and a 12-layer GPT-2-style transformer as the text encoder. We train the encoders jointly to maximize the cosine similarity between the embedding of image caption pairs following Radford et al. (2021).

Table 7 compares NSO with SAM and SGD in pretraining the CLIP model. For each method, we evaluate the trace of the loss's Hessian and recall scores of the top-10 scored images in retrieving images from texts on the development set. The results show that NSO reduces the trace of the Hessian by **17**% compared to both SAM and SGD. Accordingly, NSO achieves **1.4**% higher recall scores in image retrieval than SAM and SGD.

Table 7: Results of comparing NSO with SAM and SGD in pretraining CLIP on the Conceptual Caption dataset. We report the recall score of image retrieval and the trace of Hessian using the model in the last epoch of training. The results are averaged over five random seeds.

	Trace (↓)	$\boldsymbol{\lambda_1}$ (↓)	**Recall@10** (↑)
SGD	220 ± 24	41 ± 2.8	36.1% ± 0.3
SAM	144 ± 20	30 ± 1.1	36.9% ± 0.4
NSO	**119** ± 34	**22** ± 1.2	**37.5**% ± 0.3

3.3 Results for Chain-of-thought Fine-tuning

Lastly, we apply our algorithm to fine-tuning pretrained language models on chain-of-thought reasoning datasets. The task is to generate the reasoning process, i.e., a chain of thoughts and the answer for a given commonsense reasoning question. We fine-tune pretrained GPT-2 models on Commonsense QA and Strategy QA datasets, using LLM-generated chain-of-thoughts during training (Ho et al., 2023).

Table 8 compares NSO with SAM and SGD in chain-of-thought fine-tuning. For each method, we evaluate the trace of the loss's Hessian and the test accuracy of the fine-tuned language model. The results also show that NSO yields **25**% lower Hessian traces than SAM and SGD and achieves **5.3**% higher test accuracy on average.

Table 8: Results of comparing NSO with SAM and SGD in fine-tuning GPT-2 on Commonsense QA and Strategy QA chain-of-thought dataset. We report the test accuracy and the trace of Hessian using the model in the last training epoch. The results are averaged over five random seeds.

CommonsenseQA	**Trace** (↓)	$\boldsymbol{\lambda_1}$ (↓)	**Test Accuracy** (↑)
SGD	372 ± 34	19 ± 0.8	27.7% ± 1.8
SAM	288 ± 15	15 ± 0.3	32.7% ± 1.4
NSO	**208** ± 31	**13** ± 0.6	**39.2**% ± 1.4
StrategyQA	**Trace** (↓)	$\boldsymbol{\lambda_1}$ (↓)	**Test Accuracy** (↑)
SGD	294 ± 13	44 ± 1.5	68.9% ± 1.0
SAM	249 ± 33	42 ± 2.6	71.1% ± 1.2
NSO	**193** ± 31	**33** ± 1.8	**75.2**% ± 1.2

4 Convergence Analysis

We now study the convergence of Algorithm 1. Recall that our algorithm minimizes $f(W)$ plus a regularization term on the trace of Hessian. As is typical with regularization, the penalty is usually small relative to the loss value. Thus, our goal is to find a stationary point of $F(W)$ instead of $f(W)$ because otherwise, we would not have the desired Hessian regularization. We state the convergence to an ϵ-approximate stationary point such that $\|\nabla F(W)\| \leq \epsilon$, for any small values of $\epsilon > 0$. The analysis builds on standard assumptions from the literature (Ghadimi & Lan, 2013; Duchi et al., 2015; Lan, 2020; Zhang, 2023).

Assumption 4.1. *Given a random seed z, let $g_z : \mathbb{R}^d \to \mathbb{R}^d$ be a continuous function that gives an unbiased estimate of the gradient: $\mathbb{E}_z\left[g_z(W)\right] = \nabla f(W)$, for any $W \in \mathbb{R}^d$. Additionally, the variance is bounded in the sense that $\mathbb{E}_z\left[\|g_z(W) - \nabla f(W)\|^2\right] \leq \sigma^2$.*

To help understand the above assumption, suppose there is a dataset of size n. Then, in SGD, the stochastic gradient would be an unbiased estimate of the gradient of the entire dataset. As for the variance of the gradient estimator, we note that as long as the gradient remains bounded, which holds in practice, then the condition of the assumption will be satisfied. We now state an upper bound on the norm of the gradient of the returned solution.

Theorem 4.2. *Let $\mathcal{P}$ be a distribution that is symmetric at zero. Let C and D be fixed, positive constants. Let $W_0 \in \mathbb{R}^d$ denote the initialization. Suppose Assumption 4.1 holds. Suppose $F(W_0) - \min_{W\in\mathbb{R}^d} F(W) \leq D^2$, and f is Lipschitz-continuous. Let $H(\mathcal{P}) = \mathbb{E}[\|U\|^2]$. There exists a fixed learning rate $\eta < C^{-1}$ such that if we run Algorithm 1 with $\eta_i = \eta$ for all i, arbitrary number of perturbations k, for T steps, the algorithm returns W_t, where t is a random integer between $1, 2, \ldots, T$, such that in expectation over the randomness of W_t:*

$$\mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \leq \sqrt{\frac{2CD^2(\sigma^2 + C^2H(\mathcal{P}))}{kT}} + \frac{2CD^2}{T}, \tag{5}$$

Recall that each iteration involves two sources of randomness stemming from g_z and $\{U_i^{(j)}\}_{j=1}^k$, respectively. Let us define

$$\delta_i = \frac{1}{2k}\sum_{j=1}^{k}\left(\nabla f\big(W_i + U_i^{(j)}\big) + \nabla f\big(W_i - U_i^{(j)}\big)\right) - \nabla F(W_i),$$
$$\xi_i = \frac{1}{2k}\sum_{j=1}^{k}\left(G_i^{(j)} - \nabla f\big(W_i + U_i^{(j)}\big) - \nabla f\big(W_i - U_i^{(j)}\big)\right),$$

for $i = 0, \ldots, T-1$. One can see that both δ_i and ξ_i have mean zero. The former is by the symmetry of $\mathcal{P}$. The latter is because g_z is unbiased under Assumption 4.1. The following result gives their variance.

Lemma 4.3. *In the setting of Theorem 4.2, for any $i = 1, \ldots, T$, we have*

$$\mathbb{E}\left[\|\xi_i\|^2\right] \leq \frac{\sigma^2}{k} \quad \textit{and} \quad \mathbb{E}\left[\|\delta_i\|^2\right] \leq \frac{C^2H(\mathcal{P})}{k}. \tag{6}$$

The last step uses smoothness to show that $\|\nabla F(W_t)\|$ keeps reducing. For details, see Appendix B.1. As a remark, existing sharpness-reducing methods such as SAM (Foret et al., 2021) seem to suffer from issues of oscillation (Bartlett et al., 2023) around the local basin, leaving a convergence analysis challenging to achieve. By contrast, our approach can be analyzed with standard techniques from stochastic optimization (Ghadimi & Lan, 2013).

Next, we construct an example to match the rate of the above analysis, essentially showing that the gradient norm bounds are tight (under the current assumptions). We use an example from the work of Drori & Shamir (2020). The difference here, in particular, is that we have to deal with the perturbations that have

been added to the objective. For $t = 0, 1, \ldots, d-1$, let $e_t \in \mathbb{R}^d$ be the basis vector in dimension d, whose t-th coordinate is 1, while the remaining coordinates are all zero. Let $f : \mathbb{R}^d \to \mathbb{R}$ be defined as

$$f(W) = \frac{1}{2G}\langle W, e_0\rangle^2 + \sum_{i=0}^{T-1} h_i(\langle W, e_{i+1}\rangle), \tag{7}$$

where h_i is a piece-wise quadratic function parameterized by α_i, defined as follow:

$$h_i(x) = \begin{cases} \frac{C\alpha_i^2}{4} & |x| \le \alpha_i, \\ -\frac{C\left(|x|-\alpha_i\right)^2}{2} + \frac{C\alpha_i^2}{4} & \alpha_i \le |x| \le \frac{3}{2}\alpha_i, \\ \frac{C\left(|x|-2\alpha_i\right)^2}{2} & \frac{3}{2}\alpha_i \le |x| \le 2\alpha_i, \\ 0 & 2\alpha_i \le |x|. \end{cases}$$

One can verify that for each piece above, ∇h_i is C-Lipschitz. As a result, provided that $G \le C^{-1}$, ∇f is C-Lipschitz, based on the definition of f in equation (7).

The stochastic function F requires setting the perturbation distribution $\mathcal{P}$. We set $\mathcal{P}$ by truncating an isotropic Gaussian $N(0, \sigma^2 \operatorname{Id}_d)$ so that the i-th coordinate is at most $2^{-1}\alpha_{i-1}$, for $i = 1, \ldots, T$. Additionally, we set the initialization W_0 to satisfy $\langle W_0, e_i\rangle = 0$ for any $i \ge 1$ while $\langle W_0, e_0\rangle \ne 0$. Finally, we choose the gradient oracle to satisfy that the i-th step's gradient noise $\xi_i = \langle \xi_i, e_{i+1}\rangle e_{i+1}$, which means that ξ_i is along the direction of the basis vector e_{i+1}. In particular, this implies only coordinate $i+1$ is updated in step i, as long as $\langle \xi_i, e_{i+1}\rangle \le 2^{-1}\alpha_i$.

Theorem 4.4. *Let the learning rates $\eta_0, \ldots, \eta_{T-1}$ be at most C^{-1}. Let $D > 0$ be a fixed value. When they either satisfy $\sum_{i=0}^{T-1} \eta_i \lesssim \sqrt{kT}$, or $\eta_i = \eta < C^{-1}$ for any epoch i, then for the above construction, the following must hold*

$$\min_{1 \le t \le T} \mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \ge D\sqrt{\frac{C\sigma^2}{32k \cdot T}}. \tag{8}$$

We remark that the above construction requires $T \le d$. Notice that this is purely for technical reasons. It is an interesting question whether this condition can be removed or not. We briefly illustrate the key ideas of the result. At step i, the gradient noise ξ_i plus the perturbation noise is less than $2^{-1}\alpha_i + 2^{-1}\alpha_i = \alpha_i$ at coordinate $i+1$ (by triangle inequality). Thus, $h_i'(\langle W_t, e_{i+1}\rangle) = 0$, which holds for all prior update steps. This implies

$$\nabla f(W_i) = G^{-1}\langle W_i, e_0\rangle.$$

Recall that $F(W_0) \le D^2$. This condition imposes how large the α_i's can be. In particular, we will set $\alpha_i = 2\eta_i\sigma/\sqrt{k}$ in the proof. Then, based on the definition of $f(W_0)$,

$$h_i(\langle W_0, e_{i+1}\rangle) = \frac{C\alpha_i^2}{4}, \text{ since } \langle W_0 + U, e_{i+1}\rangle \le \alpha_i.$$

In Lemma B.2, we then argue that the learning rates in this case must satisfy $\sum_{i=0}^{T-1} \eta_i \le O(\sqrt{T})$.

When the learning rate is fixed and at least $\Omega(T^{-1/2})$, we construct a piece-wise quadratic function (similar to equation (7)), now with a fixed α. This is described in Lemma B.3. In this case, the gradient noise grows by $1 - C^{-1}\eta$ up to T steps. We then carefully set α to lower bound the norm of the gradient. Combining these two cases, we conclude the proof of Theorem 4.4. For details, see Appendix B.2. As is typical in lower-bound constructions, our result holds for a specific instance covering a particular learning rate range. It may be interesting to examine a broader range of instances for future work.

The proof can also be extended to adaptive learning rate schedules. Notice that the above construction holds for arbitrary learning rates defined as a function of previous iterates. Then, we set the width of each function h_t, α_t, proportional to $\eta_t > 0$, for any η_t that may depend on previous iterates, as long as they satisfy the constraint that $\sum_{i=0}^{T-1} \eta_i \le O(\sqrt{T})$.

We can show a similar lower bound for the momentum update rule. Recall this is defined as

$$M_{i+1} = \mu M_i - \eta_i G_i, \text{ and } W_{i+1} = W_i + M_{i+1}, \tag{9}$$

for $i = 0, 1, \ldots, T-1$, where G_i is the specific gradient at step i. To handle this case, we will need a more fine-grained control on the gradient, so we consider a quadratic function as $f(W) = \frac{C}{2}\|W\|^2$. We leave the result and its proof to Appendix B.3.

5 Dissecting Hessian: A Case Study in Overparameterized Matrix Sensing

Before proceeding, let us give an example to better understand the regularization effect of the Hessian. We consider the matrix sensing problem, whose generalization properties are particularly well-understood in the nonconvex factorization setting (Li et al., 2018). Let there be an unknown, rank-r positive semi-definite matrix $X^\star = U^\star U^{\star\top} \in \mathbb{R}^{d\times d}$. The input consists of a list of d by d Gaussian measurement matrix $A_1, A_2, \ldots, A_n$. The labels are given by $y_i = \langle A_i, X^\star\rangle$, for every $i = 1, 2, \ldots n$. The empirical loss is

$$\hat{L}(W) = \frac{1}{2n}\sum_{i=1}^{n}\left(\langle A_i, WW^\top\rangle - y_i\right)^2, \text{ where } W \in \mathbb{R}^{d\times d}. \tag{10}$$

When the loss reaches near zero (which implies the gradient also reaches near zero), it is known that multiple local minimum solutions exist (Li et al., 2018), and the Hessian becomes

$$\frac{1}{n}\sum_{i=1}^{n}\|A_i W\|_F^2 \approx d\|W\|_F^2 = d\left\|WW^\top\right\|_\star.$$

By prior results (Recht et al., 2010), among all $X = WW^\top$ such that $\hat{L}(W) = 0$, $X^\star$ has the lowest nuclear norm. Thus, the regularization placed on $\hat{L}(W)$ is similar to nuclear norm regularization under interpolation. We formalize this and state the proof below for completeness.

Proposition 5.1. *In the setting above, for any W that satisfies $\hat{L}(W) = 0$, the following must hold with high probability:*

$$\mathrm{Tr}\left[\nabla^2[\hat{L}(U^\star)]\right] \le \mathrm{Tr}\left[\nabla^2[\hat{L}(W)]\right] + O(n^{-\frac{1}{2}}). \tag{11}$$

A similar statement holds if the trace operator is replaced by the largest eigenvalue of the Hessian in equation (11). To see this, we look at the quadratic form of the Hessian to find the maximum eigenvalue. Let u be a d^2 dimension vector with length equal to one, $\|u\| = 1$. One can derive that:

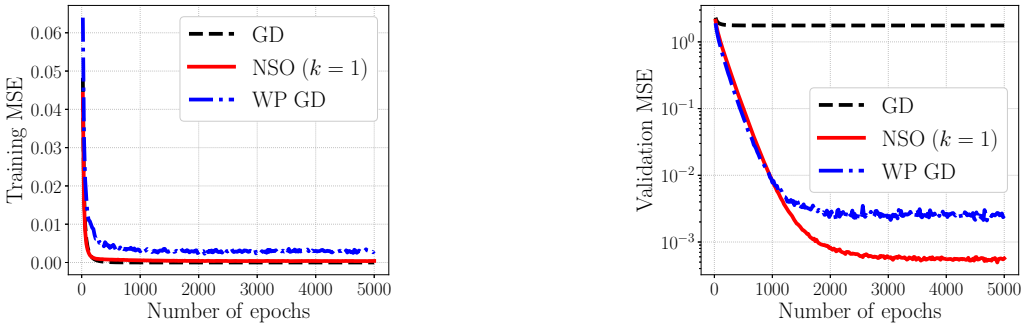
$$\lambda_1(\nabla^2\hat{L}(W)) = \max_{u\in\mathbb{R}^{d^2}:\|u\|=1} u^\top\nabla^2\hat{L}(W)u = \max_{u\in\mathbb{R}^{d^2}:\|u\|=1}\frac{1}{n}\sum_{i=1}^{n}\langle A_i W, u\rangle^2 \ge \frac{1}{d^2 n}\sum_{i=1}^{n}\|A_i W\|_F^2.$$

The last step is by setting $u = d^{-1}\mathbf{1}_{d^2}$, whose length is equal to one. The detailed proof of Proposition 5.1 and derivations for the above step are deferred in Appendix A.2.

Simulation: We conduct a numerical simulation to verify the above result. We generate a low-rank matrix $U^\star \in \mathbb{R}^{d\times r}$ from the isotropic Gaussian. We set $d = 100$ and $r = 5$. Then, we test three algorithms: gradient descent (GD), weight-perturbed gradient descent (WP-GD), and Algorithm 1 (NSO). We use an initialization $U_0 \in \mathbb{R}^{d\times d}$ where each matrix entry is sampled independently from $\mathcal{N}(0, 1)$ (the standard Gaussian).

Recall that WP-GD and NSO require setting σ. We choose σ between $0.001, 0.002, 0.004, 0.008, 0.0016$. NSO additionally requires setting the number of sampled perturbations k. We set $k = 1$ for faster computation.

Our findings are illustrated in Figure 5. We can see that all three algorithms can reduce the training MSE to near zero, as shown in Figure 5a. Regarding the validation loss, GD suffers from overfitting the training data, while both WP GD and NSO can generalize to the validation samples. Moreover, NSO manages to reduce this validation loss further.



(a) Training loss

(b) Validation loss

Figure 5: Comparing the training and validation losses between GD, NSO, and WP-GD.

6 Discussions and Related Work

As mentioned in Section 1, noise injection has been studied since very early machine learning research (Hinton & Van Camp, 1993; An, 1996). We now elaborate more on the findings from this literature. Graves (2011) develop a variational inference approach to test different priors and posteriors (e.g., Delta, Laplace, Uniform, Gaussian) on recurrent neural networks. Camuto et al. (2020) propose a layer-wise regularization scheme motivated by adaptation patterns of weights through deeper layers. Bisla et al. (2022) conduct empirical studies on the connection between sharpness and generalization. Orvieto et al. (2023) analyze Taylor's expansion of the stochastic objective after noise injection, examining the induced regularization in various neural network training settings, and found that layer-wise perturbation can improve generalization and test accuracy.

The connection between Hessian and sharpness has also been studied through the Edge of Stability (Cohen et al., 2021), which is inverse to the operator norm of the Hessian matrix. Long & Bartlett (2023) identify the edge of stability regime for the SAM algorithm, highlighting differences from gradient descent. The work of Agarwala & Dauphin (2023) presents a detailed study of the gradient dynamics of SAM. They first analyze the full batch gradient descent with unnormalized SAM in a quadratic regression model. This analysis suggests that at initialization, full batch SAM presents limited suppression of the largest eigenvalue of the Hessian. Besides, they also show that as the batch size decreases, the regularization of SAM becomes stronger. This work underscores the intricate dynamics of SAM due to its connection to the min-max problem, which is computationally intractable (Daskalakis et al., 2021). Dauphin et al. (2024) provide an in-depth comparison between SAM and weight noise by examining the structure of the Hessian during training. We note that our results in Section 2.1, which show that weight noise remains ineffective in fine-tuning, are consistent with the findings of this work. However, we also find that a modified weight noise scheme can perform well in practice.

Additionally, Gaussian smoothing has been used to estimate gradients in zeroth-order optimization (Nesterov & Spokoiny, 2017). Besides, recent research has investigated the query complexity of finding stationary points of nonconvex functions (Carmon et al., 2020; Arjevani et al., 2023). These results provide a fine-grained characterization of the iteration complexity of iterative methods under different orders of gradient oracles. By now there is a growing body of work showing that geometric measures such as sharpness and the generalization of neural networks are strongly connected. We hope further studies of this connection will lead to better optimization methods for training neural networks.

Lastly, we mention several questions for future work. For instance, can the newly developed techniques be used to study transformer networks? Can we better understand the dynamics of the Hessian during training? More broadly, the geometric properties of large models seem poorly understood. Both theoretical modeling and empirical measurements are needed to better understand their working mechanisms.

7 Conclusion

This paper examines the injection of noise into the weights of a neural network. We begin by observing that the natural approach of injecting noise into the weight before running SGD does not work well in practice. Through extensive experiments, we show that a two-point noise injection scheme can effectively regularize the Hessian, improving upon SGD, WP-SGD, and SAM. Moreover, we show a generalization bound for model fine-tuning using PAC-Bayes analysis. Our approach yields statistically significant improvements over many datasets compared to four sharpness-reducing methods. Lastly, we provide the convergence rates of our algorithm.

References

Atish Agarwala and Yann Dauphin. Sam operates far from home: eigenvalue regularization as a dynamical phenomenon. In *International Conference on Machine Learning*, pp. 152–168. PMLR, 2023. 7, 15

Pierre Alquier. User-friendly introduction to pac-bayes bounds. *arXiv preprint arXiv:2110.11216*, 2021. 6

Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996. 2, 15

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *ICML*, 2022. 1

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):165–214, 2023. 15

Francis Bach. Learning theory from first principles. *Online version*, 2021. 25

Peter L Bartlett, Philip M Long, and Olivier Bousquet. The dynamics of sharpness-aware minimization: Bouncing across ravines and drifting towards wide minima. *Journal of Machine Learning Research*, 24 (316):1–36, 2023. 1, 12

Devansh Bisla, Jing Wang, and Anna Choromanska. Low-pass filtering sgd for recovering flat optima in the deep learning optimization landscape. In *International Conference on Artificial Intelligence and Statistics*, pp. 8299–8339. PMLR, 2022. 15

Alexander Camuto, Matthew Willetts, Umut Simsekli, Stephen J Roberts, and Chris C Holmes. Explicit regularisation in gaussian noise injections. *Advances in Neural Information Processing Systems*, 33:16603–16614, 2020. 15

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. 2, 15

Olivier Catoni. Pac-bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*, 2007. 2, 6

Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. *ICLR*, 2021. 15

Constantinos Daskalakis, Stratis Skoulakis, and Manolis Zampetakis. The complexity of constrained min-max optimization. In *Symposium on Theory of Computing*, 2021. 15

Yann N Dauphin, Atish Agarwala, and Hossein Mobahi. Neglected hessian component explains mysteries in sharpness regularization. *arXiv preprint arXiv:2401.10809*, 2024. 15

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017. 1

Yoel Drori and Ohad Shamir. The complexity of finding stationary points with stochastic gradient descent. In *ICML*, 2020. 2, 12

John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 2015. 2, 4, 12

Gintare Karolina Dziugaite and Daniel Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *UAI*, 2017. 1

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *ICLR*, 2021. 1, 3, 7, 12

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. 2, 12, 25

Henry Gouk, Timothy M Hospedales, and Massimiliano Pontil. Distance-based regularisation of deep networks for fine-tuning. In *Ninth International Conference on Learning Representations 2021*, 2022. 2, 10

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011. 2, 15

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007. 3

Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pp. 5–13, 1993. 2, 15

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. *ACL*, 2023. 11

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

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *UAI*, 2018. 1

Haotian Ju, Dongyue Li, and Hongyang R Zhang. Robust fine-tuning of deep neural networks with hessian-based generalization guarantees. *ICML*, 2022. 2

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *ICLR*, 2017. 1

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *ICML*, 2021. 7

Guanghui Lan. *First-order and stochastic optimization methods for machine learning*, volume 1. Springer, 2020. 2, 12

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pp. 2–47. PMLR, 2018. 14

Yong Liu, Siqi Mai, Minhao Cheng, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Random sharpness-aware minimization. *Advances in Neural Information Processing Systems*, 2022. 7

Philip M Long and Peter L Bartlett. Sharpness-aware minimization and the edge of stability. *arXiv preprint arXiv:2309.12488*, 2023. 15

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 3

David McAllester. A pac-bayesian tutorial with a dropout bound. *arXiv preprint arXiv:1307.2118*, 2013. 2, 6, 19

Thomas Möllenhoff and Mohammad Emtiyaz Khan. Sam as an optimal relaxation of bayes. In *International Conference on Learning Representations*, 2023. 7

Vaishnavh Nagarajan and J Zico Kolter. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. *ICLR*, 2020. 1

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017. 15

Antonio Orvieto, Anant Raj, Hans Kersting, and Francis Bach. Explicit regularization in overparametrized models via noise injection. *AISTATS*, 2023. 2, 15

Samiksha Pachade, Prasanna Porwal, Dhanshree Thulkar, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, Luca Giancardo, Gwenolé Quellec, and Fabrice Mériaudeau. Retinal fundus multi-disease image dataset (rfmid): A dataset for multi-disease detection research. *Data*, 6(2):14, 2021. 3

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 11

Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010. 14, 23

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. 5

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 11

Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *International conference on computational learning theory*, pp. 545–560. Springer, 2005. 5

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*, pp. 9636–9647. PMLR, 2020. 2

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019. 22, 23

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How does sharpness-aware minimization minimize sharpness? *ICLR*, 2023. 1, 3

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022. 1, 3

Tong Zhang. *Mathematical analysis of machine learning algorithms*. Cambridge University Press, 2023. 2, 12

A Omitted Proofs from Section 2

Notations: We state a few standard notations first. Given two matrices X, Y having the same dimension, let $\langle X, Y\rangle = \mathrm{Tr}[X^\top Y]$ denote the matrix inner product of X and Y. Let $\|X\|_2$ denote the spectral norm (largest singular value) of X, and let $\|X\|_F$ denote the Frobenius norm of X. We use the big-O notation $f(x) = O(g(x))$ to indicate that there exists a fixed constant C independent of x such that $f(x) \leq C \cdot g(x)$ for large enough values of x.

A.1 Proof of Hessian-based PAC-Bayes Bound

We will use the following PAC-Bayes bound (for reference, see, e.g., Theorem 2, McAllester (2013)).

Theorem A.1. *Suppose the loss function $\ell(f_W(x), y)$ lies in a bounded range $[0, C]$ given any $x \in \mathcal{X}$ with label y. For any $\beta \in (0, 1)$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following holds:*

$$L_{\mathcal{Q}}(W) \leq \frac{1}{\beta}\hat{L}_{\mathcal{Q}}(W) + \frac{C\big(KL(\mathcal{Q}||\mathcal{P}) + \log\frac{1}{\delta}\big)}{2\beta(1-\beta)n}. \tag{12}$$

This result provides flexibility in setting β. Our results will set β to balance the perturbation error of $\mathcal{Q}$ and the KL divergence between $\mathcal{P}$ and $\mathcal{Q}$. We will need the KL divergence between the prior $\mathcal{P}$ and the posterior $\mathcal{Q}$ in the PAC-Bayesian analysis. This is stated in the following result.

Proposition A.2. *Suppose $\mathcal{P} = N(X, \Sigma)$ and $\mathcal{Q} = N(Y, \Sigma)$ are both Gaussian distributions with mean vectors given by $X \in \mathbb{R}^p, Y \in \mathbb{R}^p$, and population covariance matrix $\Sigma \in \mathbb{R}^{p\times p}$. The KL divergence between $\mathcal{P}$ and $\mathcal{Q}$ is equal to*

$$KL(\mathcal{Q}||\mathcal{P}) = \frac{1}{2}(X - Y)^\top \Sigma^{-1}(X - Y).$$

Specifically, if $\Sigma = \sigma^2 \,\mathrm{Id}_p$, then the above simplifies to

$$KL(\mathcal{Q}||\mathcal{P}) = \frac{\|X - Y\|_2^2}{2\sigma^2}.$$

We will use Taylor's expansion on the perturbed loss. This is stated precisely as follows.

Claim A.3. *Let f_W be twice-differentiable, parameterized by weight vector $W \in \mathbb{R}^p$. Let $U \in \mathbb{R}^p$ be another vector with dimension p. For any W and U, the following identity holds*

$$\ell(f_{W+U}(x), y) = \ell(f_W(x), y) + U^\top \nabla \ell(f_W(x), y) + U^\top[\nabla^2 \ell(f_W(x), y)]U + R_2(\ell(f_W(x), y)),$$

where $R_2(\ell(f_W(x), y)))$ is a second-order error term in Taylor's expansion.

Proof. The proof follows by the fact that $\ell \circ f_W$ is twice-differentiable. From the mean value theorem, let $\eta \in \mathbb{R}^p$ be a vector that has the same dimension as W and U. There must exist an η between W and $U + W$ such that the following equality holds:

$$R_2(\ell(f_W(x), y)) = U^\top \Big(\nabla^2[\ell(f_\eta(x), y)] - \nabla^2[\ell(f_W(x), y)]\Big)U.$$

This completes the proof of the claim. □

Based on the above, we provide Taylor's expansion of the gap between $\ell_{\mathcal{Q}}$ and ℓ.

Lemma A.4. *In the setting of Theorem 2.1, suppose each parameter is perturbed by an independent noise drawn from $N(0, \sigma^2)$. Let $\ell_{\mathcal{Q}}(f_W(x), y)$ be the perturbed loss with noise perturbation injection vector on W. There exist some fixed value C_1 that do not grow with n and $1/\delta$ such that*

$$\left|\ell_{\mathcal{Q}}(f_W(x), y) - \ell(f_W(x), y) - \frac{1}{2}\sigma^2 \,\mathrm{Tr}\left[\nabla^2[\ell(f_W(x), y)]\right]\right| \leq C_1\sigma^3.$$

Proof. We take the expectation over U for both sides of the equation in Claim A.3. The result becomes

$$\mathop{\mathbb{E}}_{U}[\ell(f_{W+U}(x),y)] = \mathop{\mathbb{E}}_{U}\left[\ell(f_W(x),y) + U^\top \nabla \ell(f_W(x),y) + U^\top \nabla^2[\ell(f_W(x),y)]U + R_2(\ell(f_W(x),y))\right].$$

Then, we use the perturbation distribution $\mathcal{Q}$ on $\mathbb{E}_U[\ell(f_{W+U}(x),y)]$, and get

$$\ell_{\mathcal{Q}}(f_W(x),y) = \mathop{\mathbb{E}}_{U}[\ell(f_W(x),y)] + \mathop{\mathbb{E}}_{U}\left[U^\top \nabla \ell(f_W(x),y)\right] + \mathop{\mathbb{E}}_{U}\left[U^\top \nabla^2[\ell(f_W(x),y)]U\right] + \mathop{\mathbb{E}}_{U}[R_2(\ell(f_W(x),y))].$$

Since $\mathbb{E}[U]=0$, the first-order term will be zero in expectation. The second-order term becomes equal to

$$\mathop{\mathbb{E}}_{U}\left[U^\top[\nabla^2 \ell(f_W(x),y)]U\right] = \sigma^2 \operatorname{Tr}\left[\nabla^2[\ell(f_W(x),y)]\right]. \tag{13}$$

The expectation of the error term $R_2(\ell(f_W(x),y))$ be

$$\begin{aligned}
\mathop{\mathbb{E}}_{U}[R_2(\ell(f_W(x),y))] &= \mathop{\mathbb{E}}_{U}\left[U^\top\left(\nabla^2[\ell(f_\eta(x),y)] - \nabla^2[\ell(f_W(x),y)]\right)U\right] \\
&\le \mathop{\mathbb{E}}_{U}\left[\|U\|_2^2 \cdot \left\|\nabla^2[\ell(f_\eta(x),y)] - \nabla^2[\ell(f_W(x),y)]\right\|_F\right] \\
&\lesssim \mathop{\mathbb{E}}_{U}\left[\|U\|_2^2 \cdot C_1\|U\|_2\right] \lesssim C_1\sigma^3.
\end{aligned}$$

Thus, the proof is complete. □

The last piece we will need is the uniform convergence of the Hessian operator. The result uses the fact that the Hessian matrix is Lipschitz continuous.

Lemma A.5. *In the setting of Theorem 2.1, there exist some fixed values C_2, C_3 that do not grow with n and $1/\delta$, such that with probability at least $1-\delta$ for any $\delta > 0$, over the randomness of the n training examples, we have*

$$\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla^2[\ell(f_W(x_i),y_i)] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\nabla^2[\ell(f_W(x),y)]\right]\right\|_F \le \frac{C_2\sqrt{\log(C_3 n/\delta)}}{\sqrt{n}}. \tag{14}$$

The proof will be deferred to Section A.1.2. With these results ready, we will now state the proof of the Hessian-based generalization bound.

A.1.1 Proof of Theorem 2.1

Proof of Theorem 2.1. First, we separate the gap of $L(W)$ and $\frac{1}{\beta}\hat{L}(W)$ into three parts:

$$L(W) - \frac{1}{\beta}\hat{L}(W) = L(W) - L_{\mathcal{Q}}(W) + L_{\mathcal{Q}}(W) - \frac{1}{\beta}\hat{L}_{\mathcal{Q}}(W) + \frac{1}{\beta}\hat{L}_{\mathcal{Q}}(W) - \frac{1}{\beta}\hat{L}(W).$$

By Lemma A.4, we can bound the difference between $L(W)$ and $L_{\mathcal{Q}}(W)$ by the Hessian trace plus an error:

$$\begin{aligned}
L(W) - \frac{1}{\beta}\hat{L}(W) \le &- \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\frac{\sigma^2}{2}\operatorname{Tr}\left[\nabla^2[\ell(f_W(x),y)]\right]\right] + C_1\sigma^3 + \left(L_{\mathcal{Q}}(W) - \frac{1}{\beta}\hat{L}_{\mathcal{Q}}(W)\right) \\
&+ \frac{1}{\beta}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{\sigma^2}{2}\operatorname{Tr}\left[\nabla^2[\ell(f_W(x_i),y_i)]\right] + C_1\sigma^3\right).
\end{aligned}$$

After re-arranging the terms, we can get the following:

$$\begin{aligned}
L(W) - \frac{1}{\beta}\hat{L}(W) \le &\underbrace{- \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\frac{\sigma^2}{2}\operatorname{Tr}\left[\nabla^2[\ell(f_W(x),y)]\right]\right] + \frac{1}{n\beta}\sum_{i=1}^{n}\frac{\sigma^2}{2}\operatorname{Tr}\left[\nabla^2[\ell(f_W(x_i),y_i)]\right]}_{E_1} \\
&+ \frac{1+\beta}{\beta}C_1\sigma^3 + \underbrace{L_{\mathcal{Q}}(W) - \frac{1}{\beta}\hat{L}_{\mathcal{Q}}(W)}_{E_2}.
\end{aligned} \tag{15}$$

We will examine E_1 by separating it into two parts:

$$E_1 = \frac{1}{\beta}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{\sigma^2}{2}\operatorname{Tr}\left[\nabla^2[\ell(f_{\hat{W}}(x_i), y_i)]\right] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\frac{\sigma^2}{2}\operatorname{Tr}\left[\nabla^2[\ell(f_W(x), y)]\right]\right]\right) \tag{16}$$

$$+ \frac{1-\beta}{\beta}\frac{\sigma^2}{2}\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\operatorname{Tr}\left[\nabla^2\ell(f_W(x), y)\right]\right]. \tag{17}$$

We can use the uniform convergence result of Lemma A.5 to bound equation (16), leading to:

$$\begin{aligned}
&\frac{\sigma^2}{2\beta}\left(\frac{1}{n}\sum_{i=1}^{n}\operatorname{Tr}\left[\nabla^2\ell(f_W(x_i), y_i)\right] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\operatorname{Tr}\left[\nabla^2\ell(f_W(x), y))\right]\right]\right) \\
\leq &\frac{\sigma^2}{2\beta}\cdot\sqrt{p}\cdot\left\|\frac{1}{n}\sum_{i=1}^{n}\operatorname{Tr}\left[\nabla^2[\ell(f_W(x_i), y_i)]\right] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\operatorname{Tr}\left[\nabla^2[\ell(f_W(x), y)]\right]\right]\right\|_F \qquad \text{(by Cauchy-Schwarz)} \\
\leq &\frac{\sigma^2\sqrt{p}\cdot C_2\sqrt{\log(C_3 n/\delta)}}{2\beta\sqrt{n}}.
\end{aligned} \tag{18}$$

As for equation (17), we recall that

$$\alpha := \max_{(x,y)\sim\mathcal{D}}\operatorname{Tr}\left[\nabla^2\ell(f_W(x), y)\right].$$

Combined with equation (18), we have shown that

$$E_1 \leq \frac{\sigma^2\sqrt{p}\cdot C_2\sqrt{\log(C_3 n/\delta)}}{2\beta\sqrt{n}} + \frac{1-\beta}{\beta}\frac{\sigma^2}{2}\cdot\alpha. \tag{19}$$

As for E_2, we will use the PAC-Bayes bound of Theorem A.1. In particular, we set the prior distribution $\mathcal{P}$ as the distribution of U and we set the posterior distribution $\mathcal{Q}$ as the distribution of $W+U$. Thus,

$$E_2 \leq \frac{C\big(KL(\mathcal{Q}||\mathcal{P}) + \log\frac{1}{\delta}\big)}{2\beta(1-\beta)n} \leq \frac{C\Big(\frac{\|W\|_2^2}{2\sigma^2} + \log\frac{1}{\delta}\Big)}{2\beta(1-\beta)n} \leq \frac{C(\frac{r^2}{2\sigma^2} + \log\delta^{-1})}{2\beta(1-\beta)n}. \tag{20}$$

The last step is because $\|W\|_2 \leq r$ by assumption of the hypothesis space. Combining equations (15), (19), (20), we claim that with probability at least $1-2\delta$, the following must be true:

$$L(W) - \frac{1}{\beta}\hat{L}(W) \leq \frac{\sigma^2\sqrt{p}\cdot C_2\sqrt{\log(C_3 n/\delta)}}{2\beta\sqrt{n}} + \frac{1-\beta}{\beta}\frac{\sigma^2}{2}\alpha + \frac{1+\beta}{\beta}C_1\sigma^3 + \frac{C(\frac{r^2}{2\sigma^2} + \log\frac{1}{\delta})}{2\beta(1-\beta)n}. \tag{21}$$

Thus, we will now choose σ and $\beta \in (0,1)$ to minimize the term above. In particular, we will set σ such that:

$$\sigma^2 = \frac{r}{1-\beta}\sqrt{\frac{C}{\alpha n}}. \tag{22}$$

By plugging in this setting to equation (21) and re-arranging terms, the gap between $L(W)$ and $\hat{L}(W)/\beta$ becomes:

$$L(W) - \frac{1}{\beta}\hat{L}(W) \leq \frac{1}{\beta}\sqrt{\frac{C\alpha r^2}{n}} + \frac{C_2\sqrt{2p\log(C_3 n/\delta)}}{2\beta\sqrt{n}}\sigma^2 + \frac{1+\beta}{\beta}C_1\sigma^3 + \frac{C}{2\beta(1-\beta)n}\log\frac{1}{\delta}.$$

Let β be a fixed value close to 1 and independent of N and δ^{-1}, and let $\epsilon = (1-\beta)/\beta$. We get

$$L(W) \leq (1+\epsilon)\hat{L}(W) + (1+\epsilon)\sqrt{\frac{C\alpha r^2}{n}} + \xi, \text{ where}$$

$$\xi = \frac{C_2\sqrt{2p\log(C_3 n/\delta)}}{2\beta\sqrt{n}}\sigma^2 + \Big(1+\frac{1}{\beta}\Big)C_1\sigma^3 + \frac{C}{2\beta(1-\beta)n}\log\frac{1}{\delta}.$$

Notice that ξ is of order $O(n^{-\frac{3}{4}} + n^{-\frac{3}{4}} + \log(\delta^{-1})n^{-1}) \leq O(\log(\delta^{-1})n^{-\frac{3}{4}})$. Therefore, we have finished the proof of equation (2). □

Discussions: In the case that f is a strongly convex function, the lowest eigenvalue of the Hessian is bounded from below. Once the algorithm reaches the global minimizer, our result from Theorem 2 can be used to provide a generalization bound based on the trace of the Hessian. Notice that the noise injection will add some bias to this minimizer, leading to a sub-optimal empirical loss. To remedy this issue, one can place the regularization of the Hessian as a constraint, similar to how ℓ_2-regularization can be implemented as a constraint.

A.1.2 Proof of Lemma A.5

In this section, we provide the proof of Lemma A.5, which shows the uniform convergence of the loss Hessian.

Proof of Lemma A.5. Let C, $\epsilon > 0$, and let $S = \{W \in \mathbb{R}^p : \|W\|_2 \leq C\}$. There exists an ϵ-cover of S with respect to the ℓ_2-norm at most $\max\left(\left(\frac{3C}{\epsilon}\right)^p, 1\right)$ elements; see, e.g., Example 5.8 (Wainwright, 2019). Let $T \subseteq S$ denote the set of this cover. Recall that the Hessian $\nabla^2[\ell(f_W(x), y)]$ is C_1-Lipschitz for all $(W + U) \in S, W \in S$. Then we have

$$\left\|\nabla^2[\ell(f_{W+U}(x), y)] - \nabla^2[\ell(f_W(x), y)]\right\|_F \leq C_1 \|U\|_2 .$$

For parameters $\delta, \epsilon > 0$, let $\mathcal{N}$ be the ϵ-cover of S with respect to the ℓ_2-norm. Define the event

$$E = \left\{\forall W \in T, \left\|\frac{1}{n}\sum_{i=1}^{n} \nabla^2[\ell(f_W(x_i), y_i)] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\nabla^2[\ell(f_W(x), y)]\right]\right\|_F \leq \delta\right\}.$$

By the matrix Bernstein inequality, we have

$$\Pr[E] \geq 1 - 4 \cdot |\mathcal{N}| \cdot p \cdot \exp\left(-\frac{n\delta^2}{2\alpha^2}\right).$$

Next, for any $W \in S$, we can pick some $W + U \in T$ such that $\|U\|_2 \leq \epsilon$. We have

$$\left\|\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\nabla^2[\ell(f_{W+U}(x), y)]\right] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\nabla^2[\ell(f_W(x), y)]\right]\right\|_F \leq C_1 \|U\|_2 \leq C_1\epsilon$$

$$\left\|\frac{1}{n}\sum_{j=1}^{n} \nabla^2[\ell(f_{W+U}(x_j), y_j)] - \frac{1}{n}\sum_{j=1}^{n} \nabla^2[\ell(f_W(x_j), y_j)]\right\|_F \leq C_1 \|U\|_2 \leq C_1\epsilon.$$

Therefore, for any $W \in S$, we obtain:

$$\left\|\frac{1}{n}\sum_{j=1}^{n} \nabla^2[\ell(f_W(x_j), y_j)] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\nabla^2[\ell(f_W(x), y)]\right]\right\|_F \leq 2C_1\epsilon + \delta.$$

We will also set the value of δ and ϵ. First, set $\epsilon = \delta/(2C_1)$ so that conditional on E,

$$\left\|\frac{1}{n}\sum_{j=1}^{n} \nabla^2[\ell(f_W(x_j), y_j)] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\nabla^2[\ell(f_W(x), y)]\right]\right\|_F \leq 2\delta.$$

The event E happens with a probability of at least:

$$1 - 4|T|p \cdot \exp\left(-\frac{n\delta^2}{2\alpha^2}\right) = 1 - 4p \cdot \exp\left(\log|T| - \frac{n\delta^2}{2\alpha^2}\right).$$

We have $\log|T| \leq p\log(3B/\epsilon) = p\log(6CC_1/\delta)$. If we set

$$\delta = \sqrt{\frac{4p\alpha^2\log(3\tau CC_1 n/\alpha)}{n}}$$

so that $\log(3\tau CC_1 n/\alpha) \geq 1$ (because $n \geq \frac{e\alpha}{3C_1}$ and $\tau \geq 1$), then we get

$$\begin{aligned}
p\log(6CC_1/\delta) - n\delta^2/(2\alpha^2) =& p\log\left(\frac{6CC_1\sqrt{n}}{\sqrt{4p\alpha^2\log(3\tau CC_1 n/\alpha)}}\right) - 2p\log\left(3\tau CC_1 n/\alpha\right) \\
=& p\log\left(\frac{3CC_1\sqrt{n}}{\alpha\sqrt{p\log(3\tau CC_1 n/\alpha)}}\right) - 2p\log\left(3\tau CC_1 n/\alpha\right) \\
\leq& p\log\left(3\tau CC_1 n/\alpha\right) - 2p\log\left(3\tau CC_1 n/\alpha\right) \qquad (\tau \geq 1, \log(3\tau CC_1 n/\alpha) \geq 1) \\
=& -p\log\left(3\tau CC_1 n/\alpha\right) \leq -p\log(e\tau). \qquad (3CC_1 n/\alpha \geq e)
\end{aligned}$$

Therefore, with a probability greater than

$$1 - 4|\mathcal{N}|p \cdot \exp(-n\delta^2/(2\alpha^2)) \geq 1 - 4p(e\tau)^{-p},$$

the following estimate holds:

$$\left\| \frac{1}{n}\sum_{j=1}^{n} \nabla^2[\ell(f_W(x_j), y_j)] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\nabla^2[\ell(f_W(x), y)]\right] \right\|_F \leq \sqrt{\frac{16p\alpha^2\log(3\tau CC_1 n/\alpha)}{n}}.$$

Denote $\delta' = 4p(e\tau)^{-p}$, $C_2 = 4\alpha\sqrt{p}$, and $C_3 = 12pCC_1/(e\alpha)$. With probability greater than $1 - \delta'$, the final result is:

$$\left\| \frac{1}{n}\sum_{i=1}^{n} \nabla^2[\ell(f_W(x_i), y_i)] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\nabla^2[\ell(f_W(x), y)]\right] \right\|_F \leq C_2\sqrt{\frac{\log(C_3 n/\delta')}{n}}.$$

This completes the proof of Lemma A.5. □

A.2 Proof of Proposition 5.1

Proof of Proposition 5.1. We can calculate the gradient as

$$\nabla\hat{L}(W) = \frac{1}{n}\sum_{i=1}^{n}(\langle A_i, WW^\top\rangle - y_i)A_i W. \tag{23}$$

For a particular entry $W_{j,k}$ of W, for any $1 \leq j, k \leq d$, the derivative of the above gradient with respect to $W_{j,k}$ is

$$\frac{1}{n}\sum_{i=1}^{n}\left([A_i W]_{j,k} A_i W + \left(\langle A_i, WW^\top\rangle - y_i\right)\frac{\partial(A_i W)}{\partial W_{j,k}}\right). \tag{24}$$

When $\hat{L}(W)$ is zero, the second term of equation (24) above must be zero, because $\langle A_i, WW^\top\rangle$ is equal to y_i, for any $i = 1, \ldots, n$.

Now, we use the assumption that A_i is a random Gaussian matrix, in which every entry is drawn from a normal distribution with mean zero and variance one. Notice that the expectation of $\|A_i W\|_F^2$ satisfies:

$$\mathbb{E}\left[\|A_i W\|_F^2\right] = \mathbb{E}\left[\mathrm{Tr}\left[W^\top A_i^\top A_i W\right]\right] = \mathrm{Tr}\left[W^\top(d \cdot \mathrm{Id}_{d\times d})W^\top\right] = d \cdot \mathrm{Tr}\left[W^\top W\right] = d\,\|W\|_F^2.$$

Thus, by concentration inequality for χ^2 random variables (e.g., Wainwright (2019, equation (2.19))), the following holds for any $0 < \epsilon < 1$,

$$\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n}\|A_i W\|_F^2 - d\,\|W\|_F^2\right| \geq \epsilon d\,\|W\|_F^2\right] \leq 2\exp\left(-\frac{n\epsilon^2}{8}\right). \tag{25}$$

This implies that ϵ must be smaller than $O(n^{-1/2})$ with high probability. As a result, the average of $\|A_i W\|_F^2$ must be $d\,\|W\|_F^2$ plus some deviation error that scales with $n^{-1/2}$ times the expectation.

By Theorem 3.2, Recht et al. (2010), the minimum Frobenius norm ($\|W\|_F^2$) solution that satisfies $\hat{L}(W) = 0$ (for Gaussian random matrices) is precisely $U^\star$. Thus, we conclude that equation (11) holds. □

B Omitted Proofs from Section 4

We say that f is C-Lipschitz continuous, if for any $W_1 \in \mathbb{R}^d$ and $W_2 \in \mathbb{R}^d$, we have $\|\nabla f(W_2) - \nabla f(W_1)\| \leq C \|W_2 - W_1\|$. A corollary is that $\nabla F(W)$ is also C-Lipschitz.

B.1 Proof of Theorem 4.2

First, let us show that ∇F is C-Lipschitz. To see this, we apply the Lipschitz condition of the gradient inside the expectation of $F(W)$. For any $W_1, W_2 \in \mathbb{R}^d$, by definition,

$$\begin{aligned}
\|\nabla F(W_1) - \nabla F(W_2)\| &= \left\| \nabla \mathop{\mathbb{E}}_{U\sim\mathcal{P}} [f(W_1+U)] - \nabla \mathop{\mathbb{E}}_{U\sim\mathcal{P}} [f(W_2+U)] \right\| \\
&= \left\| \mathop{\mathbb{E}}_{U\sim\mathcal{P}} [\nabla f(W_1+U) - \nabla f(W_2+U)] \right\| \\
&\leq \mathop{\mathbb{E}}_{U\sim\mathcal{P}} [\|\nabla f(W_1+U) - \nabla f(W_2+U)\|] \leq C \|W_1 - W_2\|.
\end{aligned}$$

Next, we provide the proof for bounding the variance of δ_i and ξ_i for $i = 0, 1, \ldots, T-1$.

Proof. First, we can see that

$$\begin{aligned}
\mathop{\mathbb{E}}_{U_i^1,\ldots,U_i^k} \left[\|\delta_i\|^2 \right] &= \mathop{\mathbb{E}}_{U_i^1,\ldots,U_i^k} \left[\left\| \frac{1}{2k} \sum_{j=1}^{k} \Big(\nabla f(W_i + U_i^j) + \nabla f(W_i - U_i^j) - 2\nabla F(W_i) \Big) \right\|^2 \right] \\
&= \frac{1}{k^2} \sum_{j=1}^{k} \mathop{\mathbb{E}}_{U_i^j} \left[\left\| \frac{1}{2} \Big(\nabla f(W_i + U_i^j) + \nabla f(W_i - U_i^j) - 2\nabla F(W_i) \Big) \right\|^2 \right] \qquad (26) \\
&= \frac{1}{k} \mathop{\mathbb{E}}_{U_i^1} \left[\left\| \frac{1}{2} \Big(\nabla f(W_i + U_i^1) + \nabla f(W_i - U_i^1) \Big) - \nabla F(W_i) \right\|^2 \right] \qquad (27)
\end{aligned}$$

where in the second line we use that $U_i^{j_1}$ and $U_i^{j_2}$ are independent when $j_1 \neq j_2$, in the last line we use fact that $U_i^1, \ldots, U_i^k$ are identically distributed. In the second step, we use the fact that for two independent random variables U, V, and any continuous functions $h(U), g(V)$, $h(U)$ and $g(V)$ are still independent (recall that f is continuous since it is twice-differentiable). We include a short proof of this fact for completeness. If U and V are independent, we have $\Pr[U \in A, V \in B] = \Pr[U \in A] \cdot \Pr[V \in B]$, for any $A, B \in \text{Borel}(\mathbb{R})$. Thus, if h and g are continuous functions, we obtain

$$\begin{aligned}
\Pr[h(U) \in A, g(V) \in B] &= \Pr[U \in h^{-1}(A), V \in g^{-1}(B)] \\
&= \Pr[U \in h^{-1}(A)] \cdot \Pr[V \in g^{-1}(B)] = \Pr[h(U) \in A] \cdot \Pr[g(V) \in B].
\end{aligned}$$

Thus, we have shown that

$$\mathbb{E} \left[\|\delta_i\|^2 \right] = \frac{1}{k} \mathop{\mathbb{E}}_{U\sim\mathcal{P}} \left[\left\| \frac{1}{2} \Big(\nabla f(W_i + U) + f(W_i - U) \Big) - \nabla F(W_i) \right\|^2 \right]. \qquad (28)$$

Next, we deal with the variance of the two-point stochastic gradient. We will show that

$$\mathop{\mathbb{E}}_{U} \left[\left\| \frac{1}{2} \Big(\nabla f(W + U) + \nabla f(W - U) \Big) - \nabla F(W) \right\|^2 \right] \leq C^2 H(\mathcal{P}). \qquad (29)$$

We mainly use the Lipschitz continuity of the gradient of F. The left-hand side of equation (29) is equal to

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{U}\left[\left\|\frac{1}{2}\Big(\nabla f(W+U)-\nabla F(W)\Big)+\frac{1}{2}\Big(\nabla f(W-U)-\nabla F(W)\Big)\right\|^2\right]\\
\leq&\mathop{\mathbb{E}}_{U}\left[\frac{1}{2}\left\|\nabla f(W+U)-\nabla F(W)\right\|^2+\frac{1}{2}\left\|\nabla f(W-U)-\nabla F(W)\right\|^2\right] && \text{(by Cauchy-Schwartz)}\\
=&\frac{1}{2}\mathop{\mathbb{E}}_{U}\left[\left\|\nabla f(W+U)-\nabla F(W)\right\|^2\right] && \text{(by symmetry of } \mathcal{P} \text{ since it has mean zero)}\\
=&\frac{1}{2}\mathop{\mathbb{E}}_{U}\left[\left\|\mathop{\mathbb{E}}_{U'\sim\mathcal{P}}[\nabla f(W+U)-\nabla f(W+U')]\right\|^2\right]\\
\leq&\frac{1}{2}\mathop{\mathbb{E}}_{U}\left[\mathop{\mathbb{E}}_{U'\sim\mathcal{P}}\left[\left\|\nabla f(W+U)-\nabla f(W+U')\right\|^2\right]\right]\\
\leq&\frac{1}{2}\mathop{\mathbb{E}}_{U,U'}\left[C^2\left\|U-U'\right\|^2\right]=\frac{1}{2}C^2\mathop{\mathbb{E}}_{U,U'}\left[\left\|U\right\|^2+\left\|U'\right\|^2\right]=C^2H(\mathcal{P}) && \text{(by equation (31))}
\end{aligned}
$$

As for the variance of ξ_i, we note that $U_i^{(1)},\ldots,U_i^{(j)}$ are all independent from each other. Therefore,

$$
\begin{aligned}
\mathop{\mathbb{E}}_{\left\{U_i^{(j)},z_i^{(j)}\right\}_{j=1}^k}\left[\left\|\xi_i\right\|^2\right]=&\frac{1}{4k}\mathop{\mathbb{E}}_{U,z}\left[\left\|g_z(W+U)-\nabla f(W+U)+g_z(W-U)-f(W-U)\right\|^2\right]\\
\leq&\frac{1}{2k}\mathop{\mathbb{E}}_{U,z}\left[\left\|g_z(W+U)-\nabla f(W+U)\right\|^2+\left\|g_z(W-U)-\nabla f(W-U)\right\|^2\right]\\
\leq&\frac{\sigma^2}{k}.
\end{aligned}
$$

The first step uses the fact that both $g_z(\cdot)$ and $f(\cdot)$ are continuous functions The second step above uses Cauchy-Schwartz inequality. The last step uses the variance bound of $g_z(\cdot)$, Thus, the proof is finished. □

Next, we show the convergence of the gradient, which is based on the classical work of Ghadimi & Lan (2013).

Lemma B.1. *In the setting of Theorem 4.2, for any $\eta_0,\cdots,\eta_{T-1}$ less than C^{-1} and a random variable according to a distribution $\Pr[t=j]=\frac{\eta_j}{\sum_{i=0}^{T-1}\eta_i}$, for any $j=0,\ldots,T-1$, the following holds:*

$$
\mathbb{E}\left[\left\|\nabla F(W_t)\right\|^2\right]\leq\frac{2C}{\sum_{i=0}^{T-1}\eta_i}D^2+\frac{C\sum_{i=0}^{T-1}\eta_i^2\big(\mathbb{E}\left[\left\|\delta_i\right\|^2\right]+\mathbb{E}\left[\left\|\xi_i\right\|^2\right]\big)}{\sum_{i=0}^{T-1}\eta_i}. \tag{30}
$$

Proof. The smoothness condition on f implies the following domination inequality:

$$
\left|F(W_2)-F(W_1)-\langle\nabla F(W_1),W_2-W_1\rangle\right|\leq\frac{C}{2}\left\|W_2-W_1\right\|^2. \tag{31}
$$

See, e.g., Bach (2021, Chapter 5). Here, we use the fact that $\nabla F(W)$ is L-Lipschitz continuous. Based on the above smoothness inequality, we have

$$
\begin{aligned}
&F(W_{i+1})\\
\leq&F(W_i)+\langle\nabla F(W_i),W_{i+1}-W_i\rangle+\frac{C}{2}\eta_i^2\left\|\frac{1}{2}\Big(\nabla f(W_i+U_i)+\nabla f(W_i-U_i)\Big)+\xi_i\right\|^2\\
=&F(W_i)-\eta_i\langle\nabla F(W_i),\delta_i+\xi_i+\nabla F(W_i)\rangle+\frac{C\eta_i^2}{2}\left\|\delta_i+\xi_i+\nabla F(W_i)\right\|^2\\
=&F(W_i)-\left(\eta_i-\frac{C\eta_i^2}{2}\right)\left\|\nabla F(W_i)\right\|^2-\left(\eta_i-C\eta_i^2\right)\langle\nabla F(W_i),\delta_i+\xi_i\rangle+\frac{C\eta_i^2}{2}\left\|\delta_i+\xi_i\right\|^2.
\end{aligned}
$$

Summing up the above inequalities for $i = 0, 1, \ldots, T-1$, we obtain

$$\sum_{i=0}^{T-1} F(W_{i+1}) \leq \sum_{i=0}^{T-1} F(W_i) - \sum_{i=0}^{T-1} \left(\eta_i - \frac{C\eta_i^2}{2}\right) \|\nabla F(W_i)\|^2 - \sum_{i=0}^{T-1} \left(\eta_i - C\eta_i^2\right) \langle \nabla F(W_i), \delta_i + \xi_i \rangle + \sum_{i=0}^{T-1} \frac{C\eta_i^2}{2} \|\delta_i + \xi_i\|^2,$$

which implies that

$$\sum_{i=0}^{T-1} \left(\eta_i - \frac{C\eta_i^2}{2}\right) \|\nabla F(W_i)\|^2 \tag{32}$$

$$\leq F(W_0) - F(W_T) - \sum_{i=0}^{T-1} \left(\eta_i - C\eta_i^2\right) \langle \nabla F(W_i), \delta_i + \xi_i \rangle + \frac{C}{2} \sum_{i=0}^{T-1} \eta_i^2 \|\delta_i + \xi_i\|^2$$

$$\leq D^2 - \sum_{i=0}^{T-1} \left(\eta_i - C\eta_i^2\right) \langle \nabla F(W_i), \delta_i + \xi_i \rangle + \frac{C}{2} \sum_{i=0}^{T-1} \eta_i^2 \|\delta_i + \xi_i\|^2. \tag{33}$$

where in the last step, we use the fact that

$$F(W_0) - F(W_T) \leq F(W_0) - \min_{W \in \mathbb{R}^d} F(W) \leq D^2.$$

For any $t = 0, 1, \ldots, T-1$, notice that as long as $0 < \eta_t \leq \frac{1}{C}$, then

$$\eta_t \leq 2\eta_t - C\eta_t^2.$$

Hence, we have

$$\frac{1}{2} \sum_{t=0}^{T-1} \eta_t \|\nabla F(W_t)\|^2 \leq \sum_{t=0}^{T-1} \left(\eta_t - \frac{C\eta_t^2}{2}\right) \|\nabla F(W_t)\|^2,$$

which implies that

$$\frac{1}{2} \sum_{i=0}^{T-1} \eta_i \|\nabla F(W_i)\|^2 \leq D^2 - \sum_{i=0}^{T-1} \left(\eta_i - C\eta_i^2\right) \langle \nabla F(W_i), \delta_i + \xi_i \rangle + \frac{C}{2} \sum_{i=0}^{T-1} \eta_i^2 \|\delta_i + \xi_i\|^2. \tag{34}$$

Additionally, since U_t is drawn from a distribution with mean zero. Hence, by symmetry, we get that

$$\mathop{\mathbb{E}}_{U_t}[\delta_t] = \frac{1}{2} \mathop{\mathbb{E}}_{U_t}[\nabla f(W_t - U_t) - \nabla f(W_t + U_t)] = 0. \tag{35}$$

Thus, if we take the expectation over $U_0, U_1, \ldots, U_{T-1}, \xi_0, \xi_1, \ldots, \xi_{T-1}$, then

$$\mathbb{E}[\langle \nabla F(W_i), \delta_i + \xi_i \rangle] = 0.$$

Recall that t is a random variable whose probability mass is specified in Lemma B.1. We can write equation (34) equivalently as (below, we take expectation over all the random variables along the update since W_t is a function of the previous gradient updates, for each $t = 0, 1, \ldots, T-1$, recalling that $\Pr[t = i] = \frac{\eta_i}{\sum_{j=0}^{T-1} \eta_j}$)

$$\begin{aligned}
\mathop{\mathbb{E}}_{t;\ U_0,\ldots,U_{T-1},\xi_0,\xi_1,\ldots,\xi_{T-1}} \left[\|\nabla F(W_t)\|^2\right] &= \frac{\sum_{i=0}^{T-1} \eta_i \mathbb{E}\left[\|\nabla F(W_i)\|^2\right]}{\sum_{i=0}^{T-1} \eta_i} \\
&\leq \frac{2D^2 + C\sum_{i=0}^{T-1} \eta_i^2 \mathbb{E}\left[\|\delta_i + \xi_i\|^2\right]}{\sum_{i=0}^{T-1} \eta_i} \\
&= \frac{2D^2 + C\sum_{i=0}^{T-1} \eta_i^2 \left(\mathbb{E}\left[\|\delta_i\|^2\right] + \mathbb{E}\left[\|\xi_i\|^2\right]\right)}{\sum_{i=0}^{T-1} \eta_i}.
\end{aligned}$$

where we use the fact that δ_i and ξ_i are independent for any i. Hence, we have finished the proof of equation (30). □

Based on the above result, we now finish the proof of the upper bound in Proposition 4.2.

Proof. Let the step sizes be equal to a fixed η for all epochs. Thus, Eq. (30) becomes

$$\mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \leq \frac{2}{T\eta}D^2 + \frac{C\eta}{T}\sum_{i=0}^{T-1}\left(\mathbb{E}\left[\|\delta_i\|^2\right] + \mathbb{E}\left[\|\xi_i\|^2\right]\right). \tag{36}$$

By Lemma 4.3,

$$\sum_{i=0}^{T-1}\left(\mathbb{E}\left[\|\delta_i\|^2\right] + \mathbb{E}\left[\|\xi_i\|^2\right]\right) \leq T \cdot \frac{\sigma^2 + C^2 H(\mathcal{P})}{k}. \tag{37}$$

For simplicity, let us denote $\Delta = \frac{\sigma^2 + C^2 H(\mathcal{P})}{k}$. The proof is divided into two cases.

Case 1: Δ is large. More precisely, suppose that $\Delta \geq 2CD^2/T$. Then, minimizing over η above leads us to the following upper bound on the right-hand side of equation (36):

$$\sqrt{\frac{2CD^2\Delta}{T}}, \tag{38}$$

which is obtained by setting

$$\eta = \sqrt{\frac{2D^2}{C\Delta T}}.$$

One can verify that this step size is less than $\frac{1}{C}$ since Δ is at least $2CD^2$. Thus, we conclude that equation (36) must be less than

$$\sqrt{\frac{2CD^2\Delta}{T}} = \sqrt{\frac{2CD^2(\sigma^2 + C^2 H(\mathcal{P})))}{kT}}. \tag{39}$$

Case 2: Δ is small. In this case, suppose $\Delta < 2CD^2/T$. Then, the right-hand side of equation (36) must be less than

$$\frac{2D^2}{T\eta} + \frac{2C^2D^2\eta}{T} \leq \frac{2CD^2}{T}. \tag{40}$$

Thus, combining equations (39) and (40), we have completed the proof of equation (5).

□

B.2 Proof of Theorem 4.4

Recall our construction from Section 4 as follows. Let e_t be the basis vector for the t-th dimension, for $t = 0, 1, \ldots, T-1$. Define $f(W)$ as

$$f(W) = \frac{1}{2G}\langle W, e_0\rangle^2 + \sum_{i=0}^{T-1} h_i(\langle W, e_{i+1}\rangle),$$

where h_i a quadratic function parameterized by α_i, defined as follow:

$$h_i(x) = \begin{cases} \frac{C\alpha_i^2}{4} & |x| \leq \alpha_i \\ -\frac{C(|x|-\alpha_i)^2}{2} + \frac{C\alpha_i^2}{4} & \alpha_i \leq |x| \leq \frac{3}{2}\alpha_i \\ \frac{C(|x|-2\alpha_i)^2}{2} & \frac{3}{2}\alpha_i \leq |x| \leq 2\alpha_i \\ 0 & 2\alpha_i \leq |x|. \end{cases}$$

For technical reasons, we define a truncated perturbation distribution $\mathcal{P}$ as follows. Given a sample U from a d-dimensional isotropic Gaussian $N(0, \mathrm{Id}_d)$, we truncate the i-th coordinate of U so that $\tilde{U}_i = \min(U_i, a_i)$, for some fixed $a_i > 0$ that we will specify below, for all $i = 0, 1, \ldots, d-1$. We let $\mathcal{P}$ denote the distribution of $\tilde{U}$.

The proof of Theorem 4.4 is divided into two cases. In the first, we examine the case when the averaged learning rate is $O(T^{-1/2})$.

Lemma B.2. *In the setting of Theorem 4.4, suppose the learning rates satisfy that* $\sum_{i=0}^{T-1} \eta_i \leq \sqrt{\frac{D^2 kT}{2\sigma^2 C}}$, *consider the function* $f(W)$ *constructed in equation* (7), *we have*

$$\min_{1 \leq t \leq T} \mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \geq D\sqrt{\frac{C\sigma^2}{32kT}}.$$

Proof. We start by defining a gradient oracle by choosing the noise vectors $\{\xi_t\}_{t=0}^{T-1}$ to be independent random variables such that

$$\xi_t = \langle \xi_t, e_{t+1} \rangle e_{t+1} \text{ and } |\langle \xi_t, e_{t+1} \rangle| \leq \frac{\sigma}{\sqrt{k}}, \tag{41}$$

where e_{t+1} is a basis vector whose $(t+1)$-th entry is one and otherwise is zero. In other words, only the $(t+1)$-th coordinate of ξ_t is nonzero, otherwise the rest of the vector remains zero. We use $\bar{\xi}_t$ to denote the averaged noise variable as

$$\bar{\xi}_t = \frac{1}{k} \sum_{i=1}^{k} \xi_t^{(i)},$$

where $\xi_t^{(i)}$ is defined following the condition specified in equation (41). Thus, we can also conclude that

$$|\langle \bar{\xi}_t, e_{t+1} \rangle| \leq \frac{\sigma}{\sqrt{k}}.$$

We consider the objective function $f(W) : \mathbb{R}^d \to \mathbb{R}$ defined above (see also equation (7), Section 4), with

$$\alpha_i = \frac{2\eta_i \sigma}{\sqrt{k}}, \text{ for } i = 0, 1, \ldots, T. \tag{42}$$

We will analyze the dynamics of Algorithm 1 with the objective function $f(W)$ and the starting point $W_0 = D\sqrt{G} \cdot e_0$, where $G = \max\left\{C^{-1}, 2\sum_{i=0}^{T-1} \eta_i\right\}$. For the first iteration, we have

$$\begin{aligned} W_1 &= W_0 - \eta_0 \Big(\frac{1}{2} \sum_{i=1}^{k} \big(\nabla f(W_0 + U_0^{(i)}) + \nabla f(W_0 - U_0^{(i)}) \big) + \bar{\xi}_0 \Big) \\ &= (1 - \eta_0 G^{-1}) W_0 - \eta_0 \bar{\xi}_0, \end{aligned}$$

where U is a random draw from the truncated distribution $\mathcal{P}$ with $\langle U, e_i \rangle = \min\{\mathcal{P}_i, a_i\}$ for $a_i = \frac{\eta_{i-1}\sigma}{\sqrt{k}}$. Next, from the construction of h_1, we get

$$\begin{aligned} &\frac{1}{2}\big(\nabla f(W_1 + U) + \nabla f(W_1 - U)\big) \\ &= G^{-1} \langle W_1, e_0 \rangle e_0 + \frac{1}{2}\Big(h_0'\big(\eta_0 \langle \bar{\xi}_0, e_1 \rangle + \langle U, e_1 \rangle\big) e_1 + h_0'\big(\eta_0 \langle \bar{\xi}_0, e_1 \rangle - \langle U, e_1 \rangle\big) e_1 \Big). \end{aligned}$$

Here, using the fact that $\alpha_0 = \frac{2\eta_0 \sigma}{\sqrt{k}}$ from equation (42) above, and the truncation of U, which implies $|\langle U, e_1 \rangle| \leq \frac{\eta_0 \sigma}{\sqrt{k}}$, and $\langle \bar{\xi}_0, e_1 \rangle \leq \frac{\sigma}{\sqrt{k}}$, we obtain

$$\big|\eta_0 \langle \bar{\xi}_0, e_1 \rangle + \langle U, e_1 \rangle\big| \leq \frac{2\eta_0 \sigma}{\sqrt{k}} = \alpha_0, \text{ and similarly } \big|\eta_0 \langle \bar{\xi}_0, e_1 \rangle - \langle U, e_1 \rangle\big| \leq \frac{2\eta_0 \sigma}{\sqrt{k}} = \alpha_0,$$

which implies that

$$h_0'(\eta_0\langle\bar{\xi}_0, e_1\rangle + \langle U, e_1\rangle) = h_0'(\eta_0\langle\bar{\xi}_0, e_1\rangle - \langle U, e_1\rangle) = 0.$$

This is the first update. Then, in the next iteration,

$$\begin{aligned} W_2 &= W_1 - \eta_1\Big(G^{-1}\langle W_1, e_0\rangle + \bar{\xi}_1\Big) \\ &= -(1-\eta_1 G^{-1})(1-\eta_0 G^{-1})W_0 - \eta_0\bar{\xi}_0 - \eta_1\bar{\xi}_1. \end{aligned}$$

Similarly, we use the fact that $\alpha_i = \frac{2\eta_i\sigma}{\sqrt{k}}$ and the fact that $|\langle U, e_{i+1}\rangle| \leq \frac{\eta_i\sigma}{\sqrt{k}}$, which renders the gradient as zero similar to the above reasoning. This holds for any $i = 1, 2, \ldots, T-1$.

At the t-th iteration, suppose we have that

$$W_t = W_0\prod_{i=0}^{t-1}\Big(1-\eta_i G^{-1}\Big) - \sum_{i=0}^{t-1}\eta_i\bar{\xi}_i.$$

Then by induction, at the $(t+1)$-th iteration, we must have

$$\begin{aligned} W_{t+1} &= W_t - \eta_t\Big(G^{-1}\langle W_t, e_0\rangle + \bar{\xi}_t\Big) \\ &= W_0\prod_{i=0}^{t}\Big(1-\eta_i G^{-1}\Big) - \sum_{i=0}^{t}\eta_i\bar{\xi}_i. \end{aligned} \tag{43}$$

Next, from the definition of h_t above, we have that

$$\begin{aligned} F(W_0) - \min_{W\in\mathbb{R}^d} F(W) &= F(W_0) && \text{(the minimum can be attained at zero)} \\ &= \frac{1}{2G}(D\sqrt{G})^2 + \sum_{i=0}^{T-1}\frac{C}{4}\Big(\frac{2\eta_i\sigma}{\sqrt{k}}\Big)^2 && \text{(since } \langle W_0 + U, e_{i+1}\rangle \leq \alpha_i\text{)} \end{aligned}$$

The above must be at most D^2, which implies that we should set the learning rates to satisfy (after some calculation)

$$\frac{1}{T}\Big(\sum_{i=0}^{T-1}\eta_i\Big)^2 \leq \sum_{i=0}^{T-1}\eta_i^2 \leq \frac{kD^2}{2C\sigma^2}. \tag{44}$$

We note that for all $z \in [0,1]$, $1 - \frac{z}{2} \geq \exp(\log\frac{z}{2})$. Thus, applying this to the right-hand side of equation (43), we obtain that for any t,

$$\prod_{i=0}^{t}\Big(1-\eta_i G^{-1}\Big) \geq \frac{1}{2}, \tag{45}$$

where we recall that $G = \max\{C^{-1}, 2\sum_{i=0}^{T-1}\eta_i\}$. Essentially, our calculation so far shows that for all the h_i except h_0, the algorithm has not moved at all from its initialization at W_0 under the above gradient noise. We thus conclude that

$$\begin{aligned} \min_{1\leq i\leq T}\|\nabla F(W_i)\|^2 &= \min_{1\leq i\leq T}\Big(G^{-1}\langle W_0, e_0\rangle\Big)^2 && \text{(by the construction of } F(\cdot)\text{)} \\ &\geq \frac{1}{4}G^{-2}(D\sqrt{G})^2 && \text{(by equations (43) and (45))} \\ &= \frac{D^2}{4}\min\Big\{C, \frac{1}{2\sum_{i=0}^{T-1}\eta_i}\Big\} && \text{(recall the definition of } G \text{ above)} \\ &\geq \frac{D^2}{4}\min\Big\{C, \frac{\sqrt{2C\sigma^2}}{2D\sqrt{kT}}\Big\} && \text{(by equation (44))} \\ &\geq D\sqrt{\frac{C\sigma^2}{32kT}}. \end{aligned}$$

In the first step, we use the fact that $\langle \bar{\xi}_i, e_0 \rangle = 0$, for all $0 = 1, 2, \ldots, T-1$.

Thus, we have proved that equation (8) holds for W_i for any $i = 1, 2, \ldots, T$. The proof of Lemma B.2 is finished. □

Next, let us consider the case of large, fixed learning rates.

Lemma B.3. *In the setting of Theorem 4.4, suppose the learning rates satisfy that* $\sum_{i=0}^{T-1} \eta_i \geq \sqrt{\frac{D^2 kT}{2\sigma^2 C}}$ *and* $\eta_i = \eta$ *for some fixed* $\eta \leq C^{-1}$. *Then, consider the function from equation (7), we have that* $\min_{1 \leq t \leq T} \mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \geq D\sqrt{\frac{C\sigma^2}{32kT}}$.

Proof. We define the functions g, parametrized by a fixed, positive constants $\alpha = \frac{1-\rho^T}{1-\rho} \cdot 2c\eta\sigma$, as follows:

$$g(x) = \begin{cases} -\frac{C}{2}x^2 + \frac{C}{4}\alpha^2 & |x| \leq \frac{\alpha}{2}, \\ \frac{C}{2}(|x| - \alpha)^2 & \frac{\alpha}{2} \leq |x| \leq \alpha, \\ 0 & \alpha \leq |x|. \end{cases}$$

One can verify that g has C-Lipschitz gradient, but g is not twice-differentiable. We also consider a chain-like function:

$$f(W) = g(\langle W, e_0 \rangle) + \sum_{t=0}^{d-1} \frac{C}{2} \langle W, e_{t+1} \rangle^2. \tag{46}$$

From the definition of f, f also has C-Lipschitz gradient. Similar to equation (41), we start by defining an adversarial gradient oracle by choosing the noise vectors $\{\xi_t\}_{t=0}^{T-1}$ to be independent random variables such that

$$\xi_t = \langle \xi_t, e_{t+1} \rangle, \mathbb{E}\left[\langle \xi_t, e_{t+1} \rangle^2\right] = \sigma^2, \text{ and } |\langle \xi_t, e_{t+1} \rangle| \leq c\sigma,$$

where c is a fixed constant. We use $\bar{\xi}_t$ to denote the averaged noise variable as

$$\bar{\xi}_t = \sum_{i=1}^{k} \xi_t^{(i)}.$$

Suppose $\{\xi_t^{(i)}\}_{i=1}^k$ are i.i.d. random variables for any t, we have

$$|\langle \bar{\xi}_t, e_{t+1} \rangle| \leq c\sigma \text{ and } \mathbb{E}\left[\left\|\bar{\xi}_t\right\|^2\right] \leq \frac{\sigma^2}{k}. \tag{47}$$

Next, we analyze the dynamics of Algorithm 1 with the objective function $f(W)$ and the starting point $W_0 = \sum_{i=1}^d \sqrt{\frac{D^2}{Cd}} \cdot e_i$. In this case, by setting $\eta_i = \eta$ for all $i = 0, 1, \ldots, T-1$. Recall that $\eta < C^{-1}$. Denote by $\rho = C\eta$, which is strictly less than one.

Since h_t is an even function, its derivative h_t' is odd. For the first iteration, we have

$$\begin{aligned} W_1 &= W_0 - \eta\Big(\frac{1}{2}\big(\nabla f(W_0 + U) + \nabla f(W_0 - U)\big) + \bar{\xi}_0\Big) \\ &= (1 - C\eta)W_0 - \eta\bar{\xi}_0. \end{aligned}$$

where U is a truncate distribution of $\mathcal{P} \sim N(0, \mathrm{Id}_d)$ with $\langle U, e_0 \rangle = \min\{\mathcal{P}_0, a_0\}$ and $a_0 = c\eta\sigma$.

Using the fact that $\alpha = \frac{1-\rho^T}{1-\rho} \cdot 2c\eta\sigma$, $|\langle U, e_0 \rangle| \leq c\eta\sigma$, and $\langle \bar{\xi}_0, e_0 \rangle \leq c\sigma$, we have

$$g'(\eta\langle \bar{\xi}_0, e_0 \rangle + \langle U, e_0 \rangle) + g'(\eta\langle \bar{\xi}_0, e_0 \rangle - \langle U, e_0 \rangle) = -2C\eta\langle \bar{\xi}_0, e_0 \rangle.$$

Then, in the next iteration,

$$\begin{aligned} W_2 &= W_1 - \eta\Big(C\sum_{i=1}^{d}\langle W_1, e_i\rangle - C\eta\bar{\xi}_0 + \bar{\xi}_1\Big) \\ &= (1-C\eta)^2 W_0 - (1-C\eta)\eta\bar{\xi}_0 - \eta\bar{\xi}_1. \end{aligned}$$

Similarly, we use the fact that $\alpha = \frac{1-\rho^T}{1-\rho}\cdot 2c\eta\sigma$ and the fact that $|\langle U, e_0\rangle| \le c\eta\sigma$, which renders the gradient as $g'(x) = -Cx$, for any $i = 1, 2, \ldots, T-1$.

At the t-th iteration, suppose that

$$W_t = (1-C\eta)^t W_0 - \sum_{i=0}^{t-1}(1-C\eta)^{t-1-i}\eta\bar{\xi}_i.$$

Then by induction, at the $(t+1)$-th iteration, we have

$$\begin{aligned} W_{t+1} &= W_t - \eta\Big(C\sum_{i=1}^{d}\langle W_t, e_i\rangle - C\sum_{i=0}^{t-1}(1-C\eta)^{t-1-i}\eta\bar{\xi}_i + \bar{\xi}_t\Big) \\ &= (1-C\eta)^{t+1}W_0 - \sum_{i=0}^{t}(1-C\eta)^{t-1-i}\eta\bar{\xi}_i. \end{aligned} \tag{48}$$

Next, from the definition of F above, we have that

$$\begin{aligned} F(W_0) - \min_{W\in\mathbb{R}^d} F(W) &= F(W_0) \\ &= \frac{dC}{2}\Big(\sqrt{\frac{D^2}{Cd}}\Big)^2 + \frac{C}{4}\Big(\frac{2(1-\rho^T)c\eta\sigma}{(1-\rho)}\Big)^2, \qquad \text{(since } \langle W_0 + U, e_0\rangle \le \alpha) \end{aligned}$$

which must be at most D^2. Thus, we must have (after some calculation)

$$c^2 \le \frac{D^2(1-\rho)^2}{2\sigma^2\rho^2(1-\rho^T)^2}.$$

We conclude that

$$\begin{aligned} \min_{1\le i\le T}\mathbb{E}\left[\|\nabla F(W_i)\|^2\right] &= \min_{1\le i\le T}\mathbb{E}\left[\sum_{j=1}^{d} C^2\langle W_i, e_j\rangle^2 + C^2\langle W_i, e_0\rangle^2\right] \\ &= \min_{1\le i\le T}\Big(dC^2(1-\rho)^{2t}\Big(\sqrt{\frac{D^2}{Cd}}\Big)^2 + \frac{\sigma^2}{k}\cdot\rho^2\sum_{i=0}^{t}(1-\rho)^{2(t-1-i)}\Big) \\ &\ge \min_{1\le i\le T}\Big(CD^2(1-\rho)^{2t} + \frac{\sigma^2}{k}\frac{\rho}{2-\rho}\big(1-(1-\rho)^{2t}\big)\Big) \\ &\ge \min\Big\{CD^2, \frac{\sigma^2}{k}\frac{\rho}{2-\rho}\Big\} \\ &\ge \frac{\sigma^2}{k}C\sqrt{\frac{kD^2}{2T\sigma^2C}}\frac{1}{2-C\sqrt{\frac{kD^2}{2T\sigma^2C}}} \\ &\ge D\sqrt{\frac{C\sigma^2}{16k\cdot T}}. \qquad \text{(after some calculation)} \end{aligned}$$

Thus, we have proved this lemma. □

Taking both Lemma B.2 and B.3 together, we thus conclude the proof of Theorem 4.4.

B.3 Proof of momentum lower bound

In this section, we prove the following result.

Theorem B.4. *There exists a quadratic function f such that for the iterates $W_1, \ldots, W_T$ generated by equation (9), we must have:* $\min_{1\le t\le T} \mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \ge O\big(D\sqrt{\frac{C\sigma^2}{k\cdot T}}\big)$.

We will focus on a perturbation distribution $\mathcal{P}$ equal to the isotropic Gaussian distribution for this result. In this case, we know that $F(W) = f(W) + d$. For the quadratic function $f(W) = \frac{C}{2}\|W\|^2$, its gradient is clearly C-Lipschitz. We set the initialization $W_0 \in \mathbb{R}^d$ such that

$$F(W_0) - \min_{W\in\mathbb{R}^d} F(W) = D^2.$$

This condition can be met when we set W_0 as a vector whose Euclidean norm is equal to

$$D\sqrt{2\max\Big\{C^{-1}, 2\sum_{i=0}^{T-1}\eta_i\Big\}}.$$

The case when $\mu = 0$. We begin by considering the case when $\mu = 0$. In this case, the update reduces to SGD, and the iterate W_{t+1} evolves as follows:

$$W_{t+1} = \Big(1 - C\eta_t\Big)W_t - \eta_t\bar{\xi}_t, \tag{49}$$

where we denote $\bar{\xi}_t$ as the averaged noise $k^{-1}\sum_{j=1}^k \xi_t^{(j)}$, and the noise perturbation $U_t^{(j)}$ cancelled out between the plus and minus perturbations. The case when $\mu > 0$ builds on this simpler case, as we will describe below.

The key observation is that the gradient noise sequence $\bar{\xi}_1, \bar{\xi}_2, \ldots, \bar{\xi}_T$ forms a martingale sequence:

- For any $i = 1, 2, \ldots, T$, conditioned on the previous random variables $\xi_{i'}^{(j)}$ for any $i' < i$ and any $j = 1, 2, \ldots, k$, the expectation of $\bar{\xi}_i$ is equal to zero.
- In addition, the variance of $\bar{\xi}_i$ is equal to $k^{-1}\sigma^2$, since conditional on the previous random variables, the $\xi_i^{(j)}$s are all independent from each other.

The martingale property allows us to characterize the SGD path of $\|W_t\|^2$, as shown in the following result.

Lemma B.5. *In the setting of Theorem B.4, for any step sizes $\eta_0, \ldots, \eta_{T-1}$ less than C^{-1}, and any $t = 1, \ldots, T$, the expected gradient of W_t, $\mathbb{E}\left[\|\nabla F(W_t)\|^2\right]$, is equal to*

$$2CD^2\prod_{j=0}^{t-1}\big(1 - C\eta_j\big)^2 + \frac{C\sigma^2}{k}\sum_{i=0}^{t-1}\eta_i^2\prod_{j=i+1}^{t-1}\big(1 - C\eta_j\big)^2.$$

Proof. By iterating over equation (49), we can get

$$W_t = W_0\prod_{j=0}^{t-1}\Big(1 - C\eta_j\Big) - \sum_{i=0}^{t-1}\eta_i\bar{\xi}_i\prod_{j=i+1}^{t-1}\Big(1 - C\eta_j\Big).$$

Meanwhile,

$$\nabla F(W_t) = CW_t \Rightarrow \|\nabla F(W_t)\|^2 = C^2\|W_t\|^2.$$

Thus, by squaring the norm of W_t and taking the expectation, we can get

$$\mathbb{E}\left[\|\nabla F(W_t)\|^2\right] = C^2 \|W_0\|^2 \prod_{j=0}^{t-1} \left(1 - C\eta_j\right)^2 + C^2 \sum_{i=0}^{t-1} \mathbb{E}\Big[\Big|\Big|\eta_i \bar{\xi}_i \prod_{j=i+1}^{t-1} \left(1 - C\eta_j\right)\Big|\Big|^2\Big]. \tag{50}$$

Above, we use martingale property a), which says the expectation of $\bar{\xi}_i$ is equal to zero for all i. In addition, based on property b), equation (50) is equal to

$$C^2 \sum_{i=0}^{t-1} \eta_i^2 \left(\prod_{j=i+1}^{t-1} \left(1 - C\eta_j\right)^2 \mathbb{E}\left[\|\bar{\xi}_i\|^2\right] \right) = \frac{C^2\sigma^2}{k} \sum_{i=0}^{t-1} \eta_i^2 \prod_{j=i+1}^{t-1} \left(1 - C\eta_j\right)^2.$$

To see this, based on the martingale property of $\bar{\xi}$ again, the cross terms between $\bar{\xi}_i$ and $\bar{\xi}_j$ for different i, j are equal to zero in expectation:

$$\mathbb{E}\left[\langle \bar{\xi}_i, \bar{\xi}_j \rangle | \bar{\xi}_j\right] = 0, \text{ for all } 1 \leq j < i \leq T.$$

Additionally, the second moment of $\bar{\xi}_i$ satisfies:

$$\mathbb{E}\left[\|\bar{\xi}_i\|^2\right] = \frac{\sigma^2}{k}, \text{ for any } i = 1, \ldots, T.$$

Lastly, let W_0 be a vector such that

$$\|W_0\| = D\sqrt{2C^{-1}} \Rightarrow F(W_0) - \min_{W \in \mathbb{R}^d} F(W) \leq D^2.$$

Setting $\|W_0\| = D\sqrt{2C^{-1}}$ in equation (50) leads to

$$\mathbb{E}\left[\|\nabla F(W_t)\|^2\right] = 2CD^2 \prod_{j=0}^{t-1} \left(1 - C\eta_j\right)^2 + \frac{C^2\sigma^2}{k} \sum_{i=0}^{t-1} \eta_i^2 \prod_{j=i+1}^{t-1} \left(1 - C\eta_j\right)^2.$$

Thus, we conclude the proof of this result. □

We now present the proof for the case when $\sum_{i=0}^{T-1} \eta_i \leq O(\sqrt{T})$. For this result, we will use the following quadratic function:

$$f(W) = \frac{1}{2\kappa} \|W\|^2, \text{ where } \kappa = \max\{C^{-1}, 2\sum_{i=0}^{T-1} \eta_i\}, \tag{51}$$

Lemma B.6. *Consider f given in equation (51) above. For any step sizes $\eta_0, \ldots, \eta_{T-1}$ less than C^{-1}, the following holds for the stochastic objective F:*

$$\min_{1 \leq t \leq T} \mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \geq \frac{D^2}{2\max\{C^{-1}, 2\sum_{i=0}^{T-1} \eta_i\}}.$$

Proof. Clearly, the norm of the gradient of $F(W)$ is equal to

$$\|\nabla F(W)\| = \frac{1}{\kappa}\|W\|. \tag{52}$$

Following the update rule in NSO, similar to equation (49), W_t evolves as follows:

$$W_{t+1} = \left(1 - \frac{\eta_t}{\kappa}\right) W_t - \eta_t \bar{\xi}_t, \tag{53}$$

where $\bar{\xi}_t$ has variance equal to σ^2/k, according to the proof of Lemma B.5. By iterating equation (53) from the initialization, we can get a closed-form equation for $W_t^{(1)}$, for any $t = 1, 2, \ldots, T$:

$$W_t = W_0 \prod_{j=0}^{t-1}\left(1 - \frac{\eta_j}{\kappa}\right) - \sum_{k=0}^{t-1} \eta_k \xi_k \prod_{j=k+1}^{t-1}\left(1 - \frac{\eta_j}{\kappa}\right). \tag{54}$$

Following equation (52), we can show that

$$\|\nabla F(W)\|^2 = \kappa^{-2}\|W_t\|^2.$$

Thus, in expectation,

$$\begin{aligned}
\mathbb{E}\left[\|\nabla F(W_t)\|^2\right] &= \kappa^{-2}\,\mathbb{E}\left[\|W_t\|^2\right] \\
&= \kappa^{-2}\|W_0\|^2 \prod_{j=0}^{t-1}\left(1 - \kappa^{-1}\eta_j\right)^2 + \kappa^{-2}\sum_{i=0}^{t-1}\mathbb{E}\left[\left(\eta_i \bar{\xi}_i \prod_{j=i+1}^{t-1}\left(1 - \kappa^{-1}\eta_j\right)\right)^2\right] \\
&= \kappa^{-2}\|W_0\|^2 \prod_{j=0}^{t-1}\left(1 - \kappa^{-1}\eta_j\right)^2 + \kappa^{-2}\sum_{i=0}^{t-1}\eta_i^2 \prod_{j=i+1}^{t-1}\left(1 - \kappa^{-1}\eta_j\right)^2 \mathbb{E}\left[\|\bar{\xi}_i\|^2\right] \\
&= 2D^2\kappa^{-1}\prod_{j=0}^{t-1}\left(1 - \kappa^{-1}\eta_j\right)^2 + \frac{\sigma^2\kappa^{-2}}{k}\sum_{i=0}^{t-1}\eta_i^2 \prod_{j=i+1}^{t-1}\left(1 - \kappa^{-1}\eta_j\right)^2,
\end{aligned} \tag{55}$$

where we use the definition of initialization W_0 and the variance of $\bar{\xi}_i$ in the last step. In order to tackle equation (55), we note that for all $z \in [0, 1]$,

$$1 - \frac{z}{2} \geq \exp\left(\log\frac{1}{2}\cdot z\right). \tag{56}$$

Hence, applying equation (56) to the right-hand side of equation (55), we obtain that for any $i = 0, 1, \ldots, t-1$,

$$\begin{aligned}
&\prod_{j=i}^{t-1}\left(1 - \frac{\eta_j}{\max\{C^{-1}, 2\sum_{j=i}^{T-1}\eta_i\}}\right) \\
&\geq \exp\left(\log\frac{1}{2}\cdot\sum_{j=i}^{t-1}\frac{\eta_j}{\max\{(2C)^{-1}, \sum_{i=0}^{T-1}\eta_i\}}\right) \geq \frac{1}{2}.
\end{aligned}$$

Thus, equation (55) must be at least

$$\mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \geq \frac{2D^2\kappa^{-1}}{4} + \frac{\sigma^2\kappa^{-2}}{k}\sum_{i=0}^{t-1}\frac{\eta_i^2}{4}. \tag{57}$$

The above result holds for any $t = 1, 2, \ldots, T$. Therefore, we conclude that

$$\min_{1\leq t\leq T}\mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \geq \frac{D^2}{2\max\{C^{-1}, 2\sum_{i=0}^{T-1}\eta_i\}}.$$

Thus, the proof of Lemma B.6 is finished. □

Next we consider the other case when the learning rates are fixed.

Lemma B.7. *There exists convex quadratic functions f such that for any gradient oracle satisfying Assumption 4.1 and any distribution $\mathcal{P}$ with mean zero, if $\eta_i = \eta < C^{-1}$ for any $i = 1, \ldots, T$, or if $\sum_{i=0}^{T-1} \eta_i \lesssim \sqrt{T}$, then the following must hold:*

$$\min_{1 \le t \le T} \mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \ge D\sqrt{\frac{C\sigma^2}{32k \cdot T}}. \tag{58}$$

Proof. By Lemma B.6, there exists a function such that the left-hand side of equation (58) is at least

$$\frac{D^2}{2\max\{C^{-1}, 2\sum_{i=0}^{T-1}\eta_i\}} \ge \frac{CD^2}{2\max\{1, 2x^{-1}\sqrt{T}\}} = \frac{D^2 x}{4\sqrt{T}}, \tag{59}$$

which holds if $\sum_{i=0}^{T-1} \eta_i \le \sqrt{T}x^{-1}$ for any fixed $x > 0$.

On the other hand, if $\sum_{i=0}^{T-1} \eta_i \ge x^{-1}\sqrt{T}$ and $\eta_i = \eta$ for a fixed η, then $\eta > x^{-1}/\sqrt{T}$. By setting $\eta_i = \eta$ for all i in Lemma B.5, the left-hand side of equation (58) is equal to

$$\min_{1 \le t \le T}\Big(2CD^2(1 - C\eta)^{2t} + \frac{C^2\sigma^2}{k}\sum_{k=0}^{t-1}\eta^2(1 - C\eta)^{2(t-k-1)}\Big).$$

Recall that $\eta < C^{-1}$. Thus, $\rho = C\eta$ must be less than one. With some calculations, we can simplify the above to

$$\begin{aligned}&\min_{1 \le t \le T}\left(2CD^2(1-\rho)^{2t} + \frac{\sigma^2\rho^2}{k}\frac{1-(1-\rho)^{2t}}{1-(1-\rho)^2}\right)\\ &= \min_{1 \le t \le T}\left(\frac{\sigma^2\rho}{k(2-\rho)} + (1-\rho)^{2t}\Big(2CD^2 - \frac{\sigma^2\rho}{k(2-\rho)}\Big)\right).\end{aligned} \tag{60}$$

If $2CD^2 < \frac{\sigma^2\rho}{k(2-\rho)}$, the above is the smallest when $t = 1$. In this case, equation (60) is equal to

$$2CD^2(1-\rho)^2 + \frac{\sigma^2\rho^2}{k} \ge \frac{1}{\frac{1}{2CD^2} + \frac{k}{\sigma^2}} = O(1).$$

If $2CD^2 \ge \frac{\sigma^2\rho}{k(2-\rho)}$, the above is the smallest when $t = T$. In this case, equation (60) is at least

$$\frac{\sigma^2\rho}{k(2-\rho)} \ge \frac{\sigma^2\rho}{2k} \ge \frac{\sigma^2 C x^{-1}}{2k} \cdot \frac{1}{\sqrt{T}}. \tag{61}$$

To conclude the proof, we set x so that the right-hand side of equations (59) and (61) match each other. This leads to

$$x = \sqrt{\frac{2\sigma^2 C}{kD^2}}.$$

Thus, by combining the conclusions from both equations (59) and (61) with this value of x, we finally conclude that if $\sum_{i=0}^{T-1} \eta_i \le \sqrt{T}x^{-1}$, or for all $i = 0, \ldots, T-1$, $\eta_i = \eta < C^{-1}$, then in both cases, there exists a function f such that equation (58) holds. This completes the proof of Lemma B.7. □

The case when $\mu > 0$. In this case, since the update of W_t also depends on the update of the momentum, it becomes significantly more involved. One can verify that the update from step t to step $t+1$ is based on

$$X_u = \begin{bmatrix} 1 - C\eta_t & \mu \\ C\eta_t & \mu \end{bmatrix}. \tag{62}$$

Our analysis examines the eigenvalues of the matrix $X_u X_u^\top$ and the first entry in the corresponding eigenvectors. Particularly, we show that the two entries are bounded away from zero. Then, we apply the Hölder's inequality to reduce the case of $\mu > 0$ to the case of $\mu = 0$, Lemma B.7 in particular.

Proof. First, consider a quadratic function

$$f(W) = \frac{1}{2C} \|W\|^2 .$$

Clearly, $f(W)$ is C-Lipschitz. Further, $F(W) = f(W) + d$, for $\mathcal{P}$ being the isotropic Gaussian. Let W_0 be a vector whose Euclidean norm equals $D\sqrt{2C}$. Thus,

$$F(W_0) - \min_{W \in \mathbb{R}^d} F(W) = D^2.$$

As for the dynamic of momentum SGD, recall that

$$M_{t+1} = \mu M_t - \eta_t G_t \text{ and } \; W_{t+1} = W_t + M_{t+1}.$$

We consider the case where $\eta_t = \eta$ for all steps t. In this case, we can write the above update into a matrix notation as follows:

$$\begin{bmatrix} W_{t+1} \\ M_{t+1} \end{bmatrix} = \begin{bmatrix} 1 - C\eta & \mu \\ -C\eta & \mu \end{bmatrix} \begin{bmatrix} W_t \\ M_t \end{bmatrix} + C\eta \begin{bmatrix} \bar{\xi}_t \\ \bar{\xi}_t \end{bmatrix}.$$

Let $X_\mu = [1 - C\eta, \mu; -C\eta, \mu]$ denote the 2 by 2 matrix (that depends on μ) above. Similar to Lemma B.5, we can apply the above iterative update to obtain the formula for W_{t+1} as:

$$\begin{bmatrix} W_{t+1} \\ M_{t+1} \end{bmatrix} = X_u^t \begin{bmatrix} W_0 \\ M_0 \end{bmatrix} + \sum_{i=0}^{t} C\eta X_u^{t-i} \begin{bmatrix} \bar{\xi}_i \\ \bar{\xi}_i \end{bmatrix}. \tag{63}$$

By multiplying both sides by the vector $e_1 = [1, 0]^\top$, and then taking the Euclidean norm of the vector (notice that this now only evolves that W_{t+1} vector on the left, and the W_t vector on the right), we now obtain that, in expectation over the randomness of the $\bar{\xi}_i$'s, the following holds:

$$\mathbb{E}\left[\|W_{t+1}\|^2\right] = 2CD^2(e_1^\top X_u^t e_1)^2 + \frac{C^2\eta^2\sigma^2}{k} \sum_{i=0}^{t} \left\|e_1^\top X_u^i e\right\|^2 . \tag{64}$$

Above, similar to Lemma B.5, we have set the length of W_0 appropriately, so that its length is equal to $D\sqrt{2C^{-1}}$, which has led to the CD^2 term above. Recall that M_0 is equal to zero in the beginning. To get the first term above, we follow this calculation:

$$\begin{aligned} \left\| e_1^\top X_\mu^t \begin{bmatrix} W_0 \\ M_0 \end{bmatrix} \right\|^2 &= \operatorname{Tr}\left[e_1^\top X_\mu^t \begin{bmatrix} W_0 \\ M_0 \end{bmatrix} \begin{bmatrix} W_0 \\ M_0 \end{bmatrix}^\top {X_\mu^t}^\top e_1 \right] \\ &= \operatorname{Tr}\left[e_1^\top X_\mu^t \begin{bmatrix} CD^2 & 0 \\ 0 & 0 \end{bmatrix} {X_\mu^t}^\top e_1 \right] \\ &= 2CD^2(e_1^\top X_\mu^t e_1)^2. \end{aligned}$$

We use $e = [1, 1]^\top$ to denote the vector of ones. Now, we focus on the 2 by 2 matrix X_u (recall this is the coefficient matrix on the right side of equation (63)). Let its singular values be denoted as λ_1 and λ_2. In addition, to deal with equation (64), let α_1 and α_2 denote the first entry of X_u's left singular vectors, corresponding to a and b, respectively. Thus, we can write

$$(e_1^\top X_\mu^i e)^2 = \alpha_1^2 \lambda_1^{2i} + \alpha_2^2 \lambda_2^{2i}. \tag{65}$$

Now, one can verify that λ_1^2 and λ_2^2 are the roots of the following quadratic equation over x:

$$x^2 - ((1 - C\eta)^2 + (C\eta)^2 + 2\mu^2)x + \mu^2 = 0. \tag{66}$$

This can be checked by first taking X_u times $X_u^\top$, then using the definition of the eigenvalues by calculating the determinant of $X_u X_u^\top - x\,\mathrm{Id} = 0$. Thus, we have that λ_1 and λ_2 are equal to:

$$\lambda_1, \lambda_2 = \frac{(1-C\eta)^2 + (C\eta)^2 + 2\mu^2 \pm \sqrt{((1-C\eta)^2 + (C\eta)^2 + 2\mu^2)^2 - 4\mu^2}}{2}. \tag{67}$$

Now, α_1^2 (and α_2^2, respectively) satisfies that:

$$\alpha_1^2 = \frac{-C\eta(1-C\eta) + \mu^2}{(1-C\eta)^2 + \mu^2 - \lambda_1 + -C\eta(1-C\eta) + \mu^2}. \tag{68}$$

By enumerating the possible values of $C\eta$ between 0 and 1, one can verify that for a fixed value of μ, α_1^2 and α_2^2 are both bounded below from zero. Therefore, we can claim that from equation (65),

$$\alpha_1^2 \lambda_1^{2i} + \alpha_2^2 \lambda_2^{2i} \gtrsim \lambda_1^{2i} + \lambda_2^{2i}. \tag{69}$$

By the Hölder's inequality,

$$(\lambda_1^{2i} + \lambda_2^{2i})^{\frac{1}{2i}} (1+1)^{1-\frac{1}{2i}} \geq \lambda_1 + \lambda_2 = (1-C\eta)^2 + (C\eta)^2 + 2\mu^2 \tag{70}$$

$$\geq (1-C\eta)^2 + (C\eta)^2, \tag{71}$$

which implies that

$$\lambda_1^{2i} + \lambda_2^{2i} \geq \frac{((1-C\eta)^2 + (C\eta)^2)^i}{2^{(2i-1)}}. \tag{72}$$

Now, we consider two cases. If $C\eta < 1/2$, then the above is greater than $(1-C\eta)^{2i}$, which holds for any $i = 0, 1, \ldots, T-1$. By way of reduction, we can follow the proof of Lemma B.7 to complete this proof. If $C\eta > 1/2$, then the above is greater than $(C\eta)^{2i}$. Again by following the proof steps in Lemma B.7, we can show that

$$\min_{t=1}^{T} \mathbb{E}\left[\|W_t\|^2\right] \gtrsim D\sqrt{\frac{C\sigma^2}{k \cdot T}}.$$

This completes the proof of Theorem B.4. □

C Additional Experimental Results

Approximating perturbed loss using Hessian trace. Recall that we find that the trace of the Hessian provides an accurate approximation to the gap between the perturbed loss and the trained model loss across several neural networks. These include (1) a two-layer Multi-Layer Perceptron (MLP) trained on the MNIST digit classification data set, (2) a twelve-layer BERT-Base model trained on the MRPC sentence classification data set from the GLUE benchmark, and (3) a two-layer Graph Convolutional Network (GCN) trained on the COLLAB node classification data set from TUDataset.

In more detail, we set both MLP and GCN with a hidden dimension of 128 for model architectures and initialize them randomly. We initialize the BERT model from pretrained BERT-Base-Uncased. We train each model on the provided training set for the training process until the training loss is close to zero. Specifically, we train the MLP, BERT, and GCN models for 30, 10, and 100 epochs. We use the model of the last epoch to measure the error in the approximation. We do this for 100 times and again measure the perturbed loss $\ell_{\mathcal{Q}}$ on the training set. We take the gap between $\ell_{\mathcal{Q}}$ and ℓ and report that along with the magnitude of σ in the Table. We also compute the trace of the Hessian using Hessian-vector product computation libraries.

Table 9 reports the measurement of the Hessian trace and the empirical gap between $\ell_{\mathcal{Q}}$ and ℓ, corresponding to Figure 2. Our measurements show that the error between the actual gap and the Hessian approximation is within 3%. As a remark, the range of σ^2 differs across architectures because of the differing scales of their weights.

Table 9: We find that the trace of the Hessian provides an accurate approximation to the gap between $\ell_{\mathcal{Q}}$ (recall that $\ell_{\mathcal{Q}}$ is the perturbed loss) and ℓ. In particular, the measurements are taken over the fine-tuned model weight W at the last epoch.

Multi-Layer Perceptron (MNIST)			BERT Base (MRPC)			Graph ConvNets (COLLAB)		
$\boldsymbol{\sigma}$	**Gap**	**Measure**	$\boldsymbol{\sigma}$	**Gap**	**Measure**	$\boldsymbol{\sigma}$	**Gap**	**Measure**
0.020	**0.0122** $\pm$ 0.0027	**0.0096**	0.0070	**0.0083** $\pm$ 0.0031	**0.0095**	0.040	**0.0243** $\pm$ 0.0097	**0.0278**
0.021	**0.0124** $\pm$ 0.0026	**0.0106**	0.0071	**0.0088** $\pm$ 0.0031	**0.0098**	0.041	**0.0266** $\pm$ 0.0141	**0.0292**
0.022	**0.0137** $\pm$ 0.0042	**0.0117**	0.0072	**0.0093** $\pm$ 0.0032	**0.0101**	0.042	**0.0287** $\pm$ 0.0086	**0.0306**
0.023	**0.0142** $\pm$ 0.0049	**0.0128**	0.0073	**0.0098** $\pm$ 0.0034	**0.0103**	0.043	**0.0297** $\pm$ 0.0109	**0.0321**
0.024	**0.0152** $\pm$ 0.0046	**0.0139**	0.0074	**0.0104** $\pm$ 0.0035	**0.0106**	0.044	**0.0298** $\pm$ 0.0111	**0.0336**
0.025	**0.0175** $\pm$ 0.0047	**0.0151**	0.0075	**0.0110** $\pm$ 0.0036	**0.0109**	0.045	**0.0313** $\pm$ 0.0092	**0.0351**
0.026	**0.0182** $\pm$ 0.0038	**0.0163**	0.0076	**0.0117** $\pm$ 0.0038	**0.0112**	0.046	**0.0363** $\pm$ 0.0105	**0.0367**
0.027	**0.0209** $\pm$ 0.0035	**0.0176**	0.0077	**0.0124** $\pm$ 0.0040	**0.0115**	0.047	**0.0414** $\pm$ 0.0109	**0.0383**
0.028	**0.0215** $\pm$ 0.0049	**0.0189**	0.0078	**0.0131** $\pm$ 0.0042	**0.0118**	0.048	**0.0449** $\pm$ 0.0089	**0.0400**
0.029	**0.0244** $\pm$ 0.0075	**0.0203**	0.0079	**0.0139** $\pm$ 0.0044	**0.0121**	0.049	**0.0455** $\pm$ 0.0160	**0.0417**
0.030	**0.0258** $\pm$ 0.0059	**0.0218**	0.0080	**0.0147** $\pm$ 0.0047	**0.0124**	0.050	**0.0482** $\pm$ 0.0100	**0.0434**
RSS	**2.74%**			**1.03%**			**2.16%**	

Table 10: Comparison between NSO, SGD, Label Smoothing (LS), SAM, Unnormalized SAM (USAM), Adaptive SAM (ASAM), Random-SAM (RSAM), and Bayesian SAM (BSAM) on six image classification datasets, by fine-tuning a pretrained ResNet-34 neural network using each method. We report the test accuracy, the largest eigenvalue of the Hessian (for model weight found in the last epoch of each training algorithm). Lower values indicate wider loss surfaces. In all test cases, we report the averaged result over five random seeds and the standard deviation across these five runs.

		CIFAR-10	**CIFAR-100**	**Aircrafts**	**Caltech-256**	**Indoor**	**Retina**
$\boldsymbol{\lambda_1}$ (↓)	SGD	1568 $\pm$ 92	4936 $\pm$ 121	1239 $\pm$ 43	1132 $\pm$ 44	1169 $\pm$ 66	8996 $\pm$ 92
	LS	1410 $\pm$ 81	3568 $\pm$ 110	1339 $\pm$ 84	1033 $\pm$ 68	876 $\pm$ 81	4911 $\pm$ 73
	SAM	1442 $\pm$ 93	2854 $\pm$ 131	958 $\pm$ 60	1020 $\pm$ 35	975 $\pm$ 56	4246 $\pm$ 55
	USAM	1339 $\pm$ 28	2499 $\pm$ 183	624 $\pm$ 33	841 $\pm$ 34	797 $\pm$ 48	4056 $\pm$ 50
	ASAM	1488 $\pm$ 82	2837 $\pm$ 133	641 $\pm$ 89	874 $\pm$ 63	722 $\pm$ 56	4267 $\pm$ 58
	RSAM	1491 $\pm$ 51	3087 $\pm$ 134	973 $\pm$ 72	829 $\pm$ 61	1086 $\pm$ 82	5042 $\pm$ 30
	BSAM	1496 $\pm$ 89	3032 $\pm$ 191	1069 $\pm$ 61	927 $\pm$ 63	1021 $\pm$ 71	4387 $\pm$ 94
	NSO	**1162** $\pm$ 78	**2215** $\pm$ 49	**612** $\pm$ 44	**695** $\pm$ 44	**688** $\pm$ 58	**3916** $\pm$ 47

Comparison of the largest eigenvalue of the loss Hessian. Table 10 reports the comparison of the largest eigenvalue of the Hessian, between our algorithm and sharpness-reducing methods on the six image classification data sets. We observe that our algorithm further reduces the largest eigenvalue by **12.8**% more than the existing methods on average.

In Figures 6-8, we illustrate the comparison of the test loss, the trace, and the largest eigenvalue of the Hessian matrix, using the model at the last epoch of fine-tuning. We observe that our algorithm consistently reduces the three measurements compared with SAM and SGD.

Implementation. We use the same training hyper-parameters for the experiments in Section 3. These include a learning rate of 0.02, batch size of 32, and training epochs of 30. We reduce the learning rate by 0.1 every 10 epochs. We choose these hyper-parameters based on a grid search on the validation split. The range of hyper-parameters in which we conduct a grid search is as follows:

- Learning rate: 0.05, 0.02, 0.01, 0.005, 0.002, and 0.001;
- Epochs: 10, 20, and 30;
- Batch size: 16, 32, and 64.

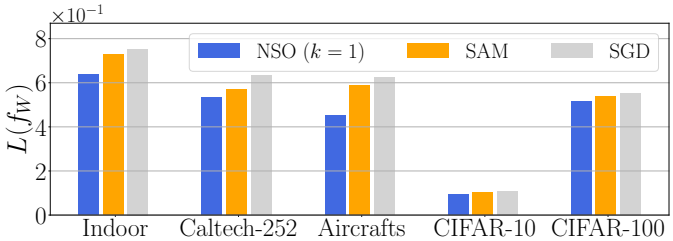


Figure 6: Illustration of the test loss measured at the last epoch of model fine-tuning. The results are run from a pretrained ResNet-34 network across five image classification tasks.

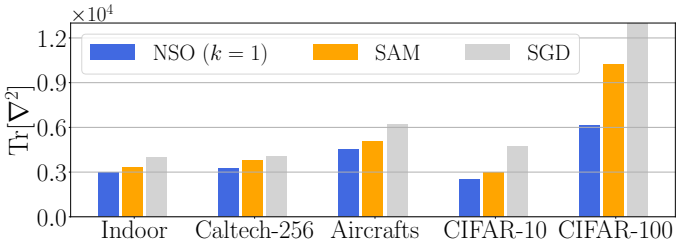


Figure 7: Illustration of the trace of the Hessian measured at the last epoch of fine-tuning ResNet-34 on five datasets.

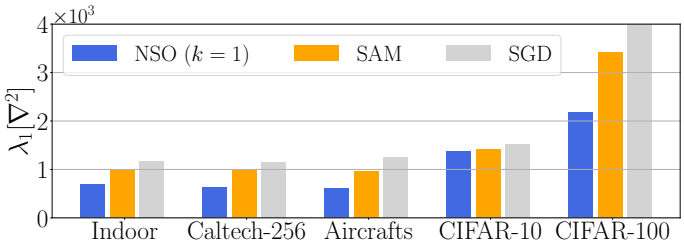


Figure 8: Reporting the λ_1 of the Hessian matrix in the last iteration of fine-tuning ResNet-34 on five datasets, comparing NSO with SAM and SGD. The results are averaged over five random seeds.

Each baseline method has its own set of hyper-parameters. We also conduct a grid search for the hyper-parameters specifically for each baseline.

- For label smoothing, we choose the weight of the loss calculated from the incorrect labels between 0.1, 0.2, and 0.3.
- For SAM and BSAM, we choose the ℓ_2 norm of the perturbation between 0.01, 0.02, and 0.05.
- For ASAM, we choose the ℓ_2 norm of the perturbation for the rescaled weights between 0.5, 1.0, and 2.0.
- For RSAM, we choose the ℓ_2 norm of the perturbation between 0.01, 0.02, and 0.05 and the standard deviation for sampling perturbation between 0.008, 0.01, and 0.012.