Noise Stability Optimization For Flat Minima With Tight Rates

Anonymous authors
Paper under double-blind review

Abstract

We consider minimizing a perturbed function $F(W) = \mathbb{E}_U[f(W + U)]$, given a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ and a random sample U from a distribution $\mathcal{P}$ with mean zero. When $\mathcal{P}$ is the isotropic Gaussian, $F(W)$ is roughly equal to $f(W)$ plus a penalty on the trace of $\nabla^2 f(W)$, scaled by the variance of $\mathcal{P}$. This penalty on the Hessian has the benefit of improving generalization, through PAC-Bayes analysis. It is useful in low-sample regimes, for instance, when a (large) pre-trained model is fine-tuned on a small data set. One way to minimize F is by adding U to W, and then run SGD. We observe, empirically, that this noise injection does not provide significant gains over SGD, in our experiments of conducting fine-tuning on three image classification data sets. We design a simple, practical algorithm that adds noise along both U and $-U$, with the option of adding several perturbations and taking their average. We analyze the convergence of this algorithm, showing tight rates on the norm of the output's gradient.

We provide a comprehensive empirical analysis to show that this modified noise injection algorithm can be competitive and even outperform sharpness-reducing training methods. First, we show that in an over-parameterized matrix sensing problem, it can find solutions with lower test loss than naive noise injection. Then, we compare our algorithm with four sharpness-reducing training methods (including the Sharpness-Aware Minimization (Foret et al., 2021)). We find that our algorithm can outperform them by up to 1.8% test accuracy, for fine-tuning ResNet on six image classification data sets. It leads to a 17.7% (and 12.8%) reduction in the trace (and largest eigenvalue) of the Hessian matrix of the loss surface. This form of regularization on the Hessian is compatible with ℓ_2 weight decay (and data augmentation), in the sense that combining both can lead to improved empirical performance.

1 Introduction

The loss landscape of neural networks and how it can affect the performance of the neural network has received much studies. Algorithmically, this can amount to searching a local region of the loss surface that is flat (Hochreiter & Schmidhuber, 1997). Recent work, notably Sharpness-Aware Minimization (Foret et al., 2021), designs an algorithm based on the principle of penalizing the largest eigenvalue of the Hessian (of the loss), to train neural networks with better generalization (give limited labeled data), and also robustness (to label noise). The algorithm by Foret et al. (2021) is based on a (constrained) min-max problem. In this paper, we examine a min-average problem, which injects noise to a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ by sampling from a d-dimensional distribution $\mathcal{P}$ (with mean zero)

$$\min_{W \in \mathbb{R}^d} F(W) := \mathop{\mathbb{E}}_{U \sim \mathcal{P}} \left[f(W + U) \right]. \tag{1}$$

The difference between the perturbed function F and the original function indicates the sensitivity or resilience of f around its local neighborhood, leading algorithms to converge to wide minima. In particular, the noise injection operates as smoothing on the loss surface, with connection to the generalization via PAC-Bayes (Nagarajan & Kolter, 2020; Dziugaite et al., 2021; Ju et al., 2022). Before proceeding, we describe an example to illustrate the regularization effect of Problem equation 1 upon the original function.

Example 1.1. Let $\mathcal{P} = \mathcal{N}(0, \sigma^2 \operatorname{Id}_d)$ denote the d-dimensional isotropic Gaussian. Then, $F(W)$ can be approximated by $f(W) + \frac{\sigma^2}{2} \operatorname{Tr}\left[\nabla^2 f(W)\right]$, plus an error term whose order scales as $O(\sigma^3)$. If σ is small, then F is roughly equal to f plus a penalty of $2^{-1}\sigma^2$ times $\operatorname{Tr}\left[\nabla^2 f\right]$.

This regularization can be further fleshed out in the matrix sensing problem. Suppose there is an unknown, rank-r positive semi-definite matrix $X^\star = U^\star {U^\star}^\top \in \mathbb{R}^{d\times d}$ that one aims to learn. One is given measurement y_i that take the matrix inner product between $X^\star$ and a Gaussian measurement matrix $A_i \in \mathbb{R}^{d\times d}$, $y_i = \langle A_i, X^\star\rangle$, for $i = 1, 2, \dots n$. Consider the Mean Squared Error parameterized by a $d \times d$ matrix W:

$$\hat{L}(W) = \frac{1}{2n} \sum_{i=1}^{n} \left(\langle A_i, WW^\top\rangle - y_i\right)^2. \tag{2}$$

Provided that $\hat{L}(W) = 0$, the Hessian is equal to (see Section A) $\frac{1}{n}\sum_{i=1}^{n} \|A_i W\|_F^2$, which is approximately $d\,\|W\|_F^2 = d\left\|WW^\top\right\|_\star$. By a well-known result, among all $X = WW^\top$ such that $\hat{L}(W) = 0$, $X^\star$ has the lowest nuclear norm (Recht et al., 2010). Thus, the regularization placed on $\hat{L}(W)$ is similar to nuclear norm regularization under interpolation.

Having discussed the regularization of Problem equation 1, the next question is how this works and whether this adds utility to directly minimizing the function itself. A naive way to instantiate this idea is to add noise to the neural network's weight before computing its gradient. This type of noise injection has been studied before (An, 1996) (which should not be confused with injecting label noise (Müller et al., 2019; Damian et al., 2021)). However, whether they will work well in practice is not always clear (Hinton & Van Camp, 1993; Graves, 2011). For example, a layer-wise regularization considering the different layer weight scaling performs better than this noise injection method (Orvieto et al., 2023).

In our empirical analysis (starting Section 4.2.1), we also observe that the naive way of perturbing the weight matrices before running SGD does not work well and only introduces marginal benefit on top of the SGD algorithm. By contrast, we design a modified noise injection scheme, which works very well through a comprehensive empirical analysis, with a detailed theoretical analysis. To our knowledge, our algorithm is the first to *perform competitively with SAM (and its variants) empirically, outperforming its regularization effect on the Hessian while permitting a clean analysis of its convergence.*

Results. We start by designing a simple, practical algorithm to solve Problem equation 1. In our design, we observe that naively perturbing the weight introduces a first-order term of $U^\top \nabla f(W)$, where U is sampled from $\mathcal{P}$ and W is the weight. Thus, although this term has a mean of zero, its standard deviation may dominate the second-order Hessian term of $U^\top \nabla^2 f(W) U$, where we recall that U is a random sample from $\mathcal{P}$. This is especially important if the algorithm is run only for a few iterations (e.g., fine-tuning transformer neural networks such as the BERT). Thus, we introduce a negative perturbation along $W - U$ to cancel out the first-order term, while keeping the second-order term unaffected. Besides, to reduce the variance of the gradient, we sample multiple perturbations at each step and take their average. See Algorithm 1 for the full procedure. In particular, when k (number of perturbations) is equal to 1, our algorithm incurs twice the computation cost of the SGD, which is the same as SAM. Then, we analyze the convergence of Algorithm 1, by showing matching upper and lower bounds on the expected gradient of its output.

Next, we conduct a comprehensive empirical analysis of our algorithm, including a comparison to existing sharpness-reducing training methods, the regularization effect on the Hessian, and interaction with existing methods, including weight decay, data augmentation, and distance-based regularization (for fine-tuning pre-trained models). We demonstrate the benefit of our algorithm compared to Perturbed SGD in the setting of Example 1.1. Then, we present a suite of experiments to demonstrate that, across various image classification data sets, our algorithm can find neural networks whose test loss is also comparably lower and whose Hessian is better regularized than several established training methods. Our experiments focus on the setting of fine-tuning a large, pre-trained neural network on a downstream task. This is an important setting of practical

Algorithm 1 Noise Stability Optimization (NSO)

```
Input: Initialization $W_0 \in \mathbb{R}^d$, a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$
Require: An estimator $g : \mathbb{R}^d \rightarrow \mathbb{R}^d$ that for any $W$, returns $g(W)$ s.t. $\mathbb{E}[g(W)] = \nabla f(W)$
Parameters: # perturbations $k$, # epochs $T$, step sizes $\eta_0, \dots, \eta_{T-1}$
for $i = 0, 1, \dots, T-1$ do
    for $j = 0, 1, \dots, k-1$ do
        Sample $U_i^{(j)}$ independently from $\mathcal{P}$
        Let $G_i^{(j)} = g\big(W_i + U_i^{(j)}\big) + g\big(W_i - U_i^{(j)}\big)$
    end for
    Update iterates according to $W_{i+1} = W_i - \frac{\eta_i}{2k} \sum_{j=1}^{k} G_i^{(j)}$
end for
```

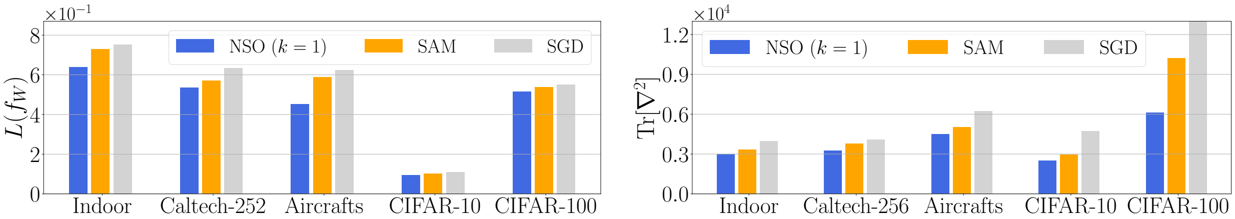


Figure 1: Comparing NSO, SAM, SGD, in terms of their test loss (left) and the trace of the Hessian of the training loss (measured at the last epoch), across five data sets, by fine-tuning ResNet-34. The results are averaged over five random seeds (their standard deviations will be included in Section 4, which also include expanded comparisons).

interest, yet it is sensitive to overfitting due to the low number of samples compared to the number of model parameters required for fine-tuning. We provide an illustration to show the comparison in Figure 1.

Summary. This paper makes the following contributions to designing regularization methods for training neural networks from algorithmic, and empirical aspects, respectively as follows:

- We revisit the injection of noise to the weight matrices in modern neural network fine-tuning, to design a simple algorithm that can also regularize the trace of the Hessian matrix of neural networks, with the strength of the regularization determined by the magnitude of the noise perturbation.

  We give matching upper and lower bounds on the gradient norms of the converged solution of our proposed algorithm. The bounds can also be extended to momentum updates.

- We conduct a comprehensive experimental evaluation of our algorithm, compared with a number of "sharpness" reducing training methods. We empirically validate the regularization of our algorithm on the Hessian matrix.

  We also show that this regularization is compatible with ℓ_2 weight decay, data augmentation, and distance-based regularization to improve empirical performance.

2 Related Work

The PAC-Bayes analysis framework gives a way to reason about the generalization of a model by postulating a prior and a posterior distribution on the hypothesis space. One may then argue about the generalization of the model, based on the KL divergence between the prior and the posterior, and the information already available in the prior (McAllester, 1999; Shawe-Taylor & Williamson, 1997). See also Alquier (2021) for references. PAC-Bayesian bounds are used to obtain measurements of the generalization gap in a "data-dependent" manner. Through optimizing the covariance matrices in both the prior and the posterior, one can obtain non-vacuous generalization bounds for realistic neural network settings (Dziugaite & Roy, 2017; Dziugaite et al., 2021). For Gaussian perturbations, one can derive a connection between PAC-Bayes bounds and the loss Hessian (Tsuzuku et al., 2020), leading to a Hessian-based generalization bound, which is also non-vacuous in various empirical settings (Ju et al., 2022; Yang et al., 2022).

The idea that injecting noise (through various forms) to neural networks can induce flatness in the found minima may be traced to very early works such as An (1996), Hinton & Van Camp (1993) (among others). However, whether or not such injection can improve empirical performance is not always clear. In particular, Graves (2011) develop a variational inference approach to test different priors and posteriors (e.g., Delta, Laplace, Uniform, Gauss) on recurrent neural networks. The work of Camuto et al. (2020) proposes a layer-wise regularization scheme motivated by how weight matrices adapt through deeper layers. Bisla et al. (2022) conduct empirical studies investigating the connection between sharpness and generalization. Orvieto et al. (2023) examine Taylor's expansion of the stochastic objective after noise injection and analyze the regularization induced by this noise (for various neural network settings). They also empirically analyze this noise injection, as well as layer-wise noise perturbation, on both convolutional and Residual networks, to find that layer-wise perturbation can lead to improved generalization and test accuracy.

A popular empirical method for training neural networks is Sharpness-Aware Minimization (SAM), which is motivated by a constrained min-max optimization problem. The problem itself is computationally intractable (Daskalakis et al., 2021), and there have been theoretical developments in the analysis of SAM (Wen et al., 2023; Andriushchenko et al., 2024). Bartlett et al. (2023), considering a convex quadratic function, finds that the stationary point of SAM oscillates locally, according to the eigenvector corresponding to the largest eigenvalue. This behavior is also noted in simulations. One insight from our empirical analysis is that examining the min-averaging formulation can lead to algorithms that perform comparably with SAM empirically. Thus, we believe this offers a new perspective to the existing literature, which may be worth further investigation in future work.

The interplay between Hessian and optimization is recently examined through the Edge of Stability (Cohen et al., 2021), which is inverse to the operator norm of the Hessian matrix. The work of Long & Bartlett (2023) identifies the Edge of Stability regime for the SAM algorithm, which has a few distinguishing properties in contrast to gradient descent. Besides, there have been studies using Gaussian smoothing to estimate gradients in zeroth-order optimization (Nesterov & Spokoiny, 2017).

The query complexity of finding stationary points of nonconvex functions has been studied in recent work; see, e.g., Carmon et al. (2020) and references therein. Our analysis is restricted to first-order oracles, and there may well be other statistical query models worth considering (Ge et al., 2015; Jin et al., 2017) and higher-order local minimum (Anandkumar & Ge, 2016). There seems to be a rich connection between generalization and optimization via the Hessian, which would be interesting to explore in future studies. For instance, it may be worth considering how one might set the covariance of the noise perturbation—A recent work by Möllenhoff & Khan (2023) sets the off-diagonals of the variance as zero to simplify the choice set.

3 Algorithm

We describe the design of the algorithm. Its regularization of the Hessian matrix can be seen through the following proposition, stated for the completeness of this paper.[1] We state a few standard notations first. Given two matrices X, Y, let $\langle X, Y\rangle = \mathrm{Tr}[X^\top Y]$ denote the matrix inner product of X and Y. Let $\|X\|$ denote the spectral norm (largest singular value) 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 $f(x) \le C \cdot g(x)$ for large enough values of x.

Proposition 3.1. *Suppose $f(W)$ is twice-differentiable in W. Let Σ denote a d by d positive semi-definite matrix. For a random vector U sampled from a Gaussian distribution $\mathcal{P} = \mathcal{N}(0, \Sigma)$, we have that*

---

[1] See also Theorem 1, Orvieto et al. (2023), who derive a similar result for the standard Gaussian, but *without* the negative perturbation of $W - U$.

$\mathbb{E}\left[\frac{1}{2}(f(W+U)+f(W-U))\right] = f(W) + \frac{1}{2}\langle \Sigma, \nabla^2 f(W)\rangle + O(\|\Sigma\|^{3/2})$ *holds w.h.p. over the randomness of the vector* U*.*

Proof. Given a random perturbation U whose magnitude is small, the Taylor's expansion implies $f(W+U) = f(W) + \langle U, \nabla f(W)\rangle + 2^{-1}U^\top \nabla^2 f(W) U + O(\|\Sigma\|^{3/2})$. Likewise, one can apply this expansion to $f(W-U)$ to get $f(W) - \langle U, \nabla f(W)\rangle + 2^{-1}U^\top \nabla^2 f(W) U$. Thus, in expectation over U, using the fact that $\mathbb{E}[U] = 0$ and $\mathbb{E}\left[UU^\top\right] = \Sigma$, we may conclude $\mathbb{E}\left[\frac{1}{2}\Big(f(W+U) + f(W-U)\Big)\right]$ is equal to $f(W) + \frac{1}{2}\langle \Sigma, \nabla^2 f(W)\rangle + O(\|\Sigma\|^{3/2})$. This completes the proof. $\square$

Notice that the above differs from (naively) injecting noise to the gradient, as we also inject noise in the negative direction.[2] However, there is still a gap between the value of $F(W)$ and $f(W)$ plus the Hessian regularization, $(\sigma^2/2)\operatorname{Tr}[\nabla^2 f(W)]$, due to the expansion error term. Next, we validate that this error term is negligible in a few real scenarios:

- A two-layer Multi-Layer Perceptron (MLP) trained on the MNIST digit classification data set;
- A twelve-layer BERT-Base model trained on the MRPC sentence classification data set (from the GLUE benchmark);
- 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 ϵ.

We first perturb the model weights by injecting isotropic Gaussian noise into them. We then compute $F(W) - f(W)$, averaged over 100 independent runs. We measure $\operatorname{Tr}[\nabla^2 f]$ as the average over the training data set. The comparisons are shown in Table 1.

Table 1: We sample 100 perturbations and compute the averaged perturbed loss values. We measure the Hessian trace using Hessian vector product computations in PyTorch (the reported values are rescaled by 10^2).

MNIST, MLP			MRPC, BERT-Base-Uncased			COLLAB, GCN		
σ	$F-f$	$\frac{\sigma^2}{2}\operatorname{Tr}[\nabla^2]$	σ	$F-f$	$\frac{\sigma^2}{2}\operatorname{Tr}[\nabla^2]$	σ	$F-f$	$\frac{\sigma^2}{2}\operatorname{Tr}[\nabla^2]$
0.020	1.22 ± 0.27	0.96	0.0070	0.83 ± 0.31	0.95	0.040	2.97 ± 0.97	2.78
0.021	1.24 ± 0.26	1.06	0.0071	0.88 ± 0.31	0.98	0.041	2.66 ± 1.41	2.92
0.022	1.37 ± 0.42	1.17	0.0072	0.93 ± 0.32	1.01	0.042	3.63 ± 0.86	3.06
0.023	1.42 ± 0.49	1.28	0.0073	0.98 ± 0.34	1.03	0.043	2.43 ± 1.09	3.21
0.024	1.52 ± 0.46	1.39	0.0074	1.04 ± 0.35	1.06	0.044	2.87 ± 1.11	3.36
0.025	1.75 ± 0.47	1.51	0.0075	1.10 ± 0.36	1.09	0.045	2.98 ± 0.92	3.51
0.026	1.82 ± 0.38	1.63	0.0076	1.17 ± 0.38	1.12	0.046	4.14 ± 1.05	3.67
0.027	2.09 ± 0.35	1.76	0.0077	1.24 ± 0.40	1.15	0.047	3.13 ± 1.09	3.83
0.028	2.15 ± 0.49	1.89	0.0078	1.31 ± 0.42	1.18	0.048	4.55 ± 0.89	4.00
0.029	2.44 ± 0.75	2.03	0.0079	1.39 ± 0.44	1.21	0.049	4.49 ± 1.60	4.17
0.030	2.58 ± 0.59	2.18	0.0080	1.47 ± 0.47	1.24	0.050	4.82 ± 1.00	4.34
Relative RSS	**2.74%**			**1.03%**			**2.16%**	

---

[2] We refer the analysis to Section 4.2.

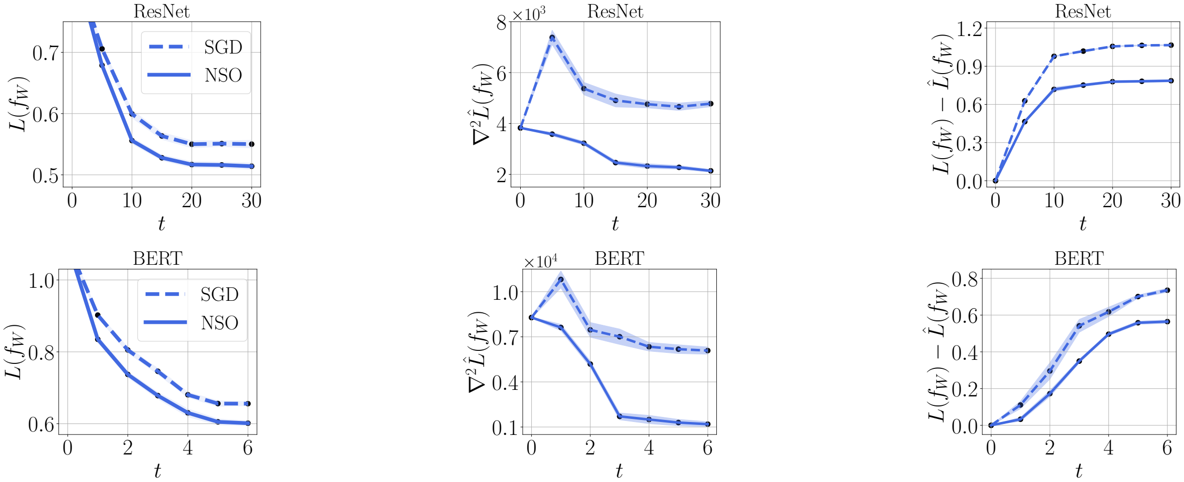


Figure 2: Comparison between SGD and NSO, for fine-tuning ResNet-34 and BERT-Base, on an image and a text classification data set, respectively, in terms of the test loss, trace of the Hessian (W taken at last epoch), and gap between test/training losses.

Remarkably, the comparison suggests that the ϵ is within 3%.[3] Based on this comparison, the next question is how to estimate this Hessian more accurately. Our algorithm thus samples multiple perturbations and takes their average to reduce the variance.

Next, we state two standard assumptions to set up the analysis of our algorithm.

Assumption 3.2. *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 $\mathbb{E}_z\left[\left\|g_z(W) - \nabla f(W)\right\|^2\right] \le \sigma^2$.*

Assumption 3.3. *Let C, D be fixed, positive constants. Let $W_0 \in \mathbb{R}^d$ denote the initialization. We require that $F(W_0) - \min_{W \in \mathbb{R}^d} F(W) \le D^2$. Let $\nabla f(W)$ denote the gradient of $f(W)$. For any $W_1 \in \mathbb{R}^d$ and $W_2 \in \mathbb{R}^d$, we have $\left\|\nabla f(W_2) - \nabla f(W_1)\right\| \le C \left\|W_2 - W_1\right\|$. A corollary is that $\nabla F(W)$ is also C-Lipschitz.*

Illustration. We illustrate a comparison between NSO and SGD in Figure 2. Notice that NSO significantly reduces $\mathrm{Tr}[\nabla^2 \hat{L}(f_W)]$, the trace of the Hessian of the empirical risk, for W taken at the last epoch, by more than 60%, while lowering the test loss. The generalization gap lowers by over 20% as well. This improvement in generalization can be explained by PAC-Bayes analysis (Nagarajan & Kolter, 2020; Dziugaite et al., 2021; Ju et al., 2022).

3.1 Rates

We now turn to analyzing our algorithm by showing tight convergence rates to stationary points of F. Notice that our algorithm can be viewed as minimizing $f(W)$ plus a regularization term on the trace of $\nabla^2 f(W)$, multiplied by $2^{-1}\sigma^2$. In our experiments, σ is set as a small value, and our goal is to find a stationary point of $F(W)$ instead of $f(W)$ because otherwise, we would not have the desired regularization added to the Hessian. Concretely, let $\epsilon > 0$ be a small value. W is an ϵ-approximate (first-order) stationary point if $\|\nabla F(W)\| \le \epsilon$. For a random sample $U \sim \mathcal{P}$, denote $\mathbb{E}[\|U\|^2]$ as $H(\mathcal{P})$.

We start by showing an upper bound on the norm of the gradient, on the solution returned by NSO. Our result may be viewed as a slight generalization of Theorem 2.1 by Ghadimi & Lan (2013). Notice that we are studying a radically different setting from that work. We state the result below.

Theorem 3.4. *Given Assumptions 3.2 and 3.3, let $\mathcal{P}$ be a distribution that is symmetric at zero. 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*

---

[3] The range of σ^2 differs across architectures because of the differing scales of their weights.

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:

$$\mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \leq \sqrt{\frac{2CD^2(\sigma^2 + C^2 H(\mathcal{P}))}{kT}} + \frac{2CD^2}{T}. \tag{3}$$

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

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

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 3.2. The next result gives their variance.

Lemma 3.5. *In the setting of Theorem 3.4, for any $i = 1, \ldots, T$, we have*

$$\mathbb{E}\left[\|\xi_i\|^2\right] \leq \frac{\sigma^2}{k} \quad \textit{and} \quad \mathbb{E}\left[\|\delta_i\|^2\right] \leq \frac{C^2 H(\mathcal{P})}{k}. \tag{4}$$

The last step (which is quite standard) is using the smoothness of F to show that $\|\nabla F(W_t)\|$ keep reducing. For details, see Appendix B.

3.2 Lower Bounds

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). In particular, the difference here is that we have to deal with the perturbations added to the objective additionally. 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

$$f(W) = \frac{1}{2G}\langle W, e_0\rangle^2 + \sum_{i=0}^{T-1} h_i(\langle W, e_{i+1}\rangle), \tag{5}$$

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

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

One can verify that for each piece above, ∇h_i is C-Lipschitz. As a result, provided that $G \leq C^{-1}$, ∇f is C-Lipschitz, based on the definition of f in equation equation 5.

The stochastic function F requires setting the perturbation distribution $\mathcal{P}$. We set $\mathcal{P}$ by truncating an isotropic Gaussian $N(0, \sigma^2\,\mathrm{Id}_d)$ 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 $i \geq 1$ while $\langle W_0, e_0\rangle \neq 0$. 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 $\langle \xi_i, e_{i+1}\rangle \leq 2^{-1}\alpha_i$.

Theorem 3.6. *Let the learning rates $\eta_0, \dots, \eta_{T-1}$ 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*

$$\min_{1 \le t \le T} \mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \ge D\sqrt{\frac{C\sigma^2}{32k \cdot T}}. \tag{6}$$

We remark that the above construction requires $T \le d$. Notice that this is purely for technical reasons due to the construction. 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 from Assumption 3.3 that $F(W_0) \le D^2$. This condition imposes how large the α_i's can be. In particular, in the proof we will set $\alpha_i = 2\eta_i\sigma/\sqrt{k}$. 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 C.1, 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 equation 5), now with a fixed α. This is described in Lemma C.2. 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 3.6. For details, see Appendix C. As is typical in lower-bound constructions, our result holds for a specific instance that covers a specific range of learning rates. It may be an interesting question 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 $\sum_{i=0}^{T-1} \eta_i \le O(\sqrt{T})$.

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

$$M_{i+1} = \mu M_i - \eta_i G_i, \text{ and } W_{i+1} = W_i + M_{i+1}, \tag{7}$$

for $i = 0, 1, \dots, T-1$, 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 $f(W) = \frac{C}{2}\|W\|^2$.[4] We state the following, whose proof can be found in Appendix D.

Theorem 3.7. *For the f function defined above, and the iterates $W_1, \dots, W_T$ generated by equation equation 7, we must have:* $\min_{1\le t\le T} \mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \ge O\big(D\sqrt{\frac{C\sigma^2}{k\cdot T}}\big)$.

4 Experiments

We now turn to the empirical analysis of our proposed algorithm. We will illustrate the simulation following Example 1.1. We find that, starting at an isotropic Gaussian initialization, gradient descent can minimize the training error to zero, but the validation error remains non-negligible, suggesting the found solution does not "generalize." By contrast, Weight-Perturbed GD (which injects noise to the gradient) and our algorithm (NSO) manage to find solutions with low training and validation errors. Further, our algorithm finds a solution with even lower validation errors than WP GD.

---

[4] We remark that a quadratic function with a generalized diagonal covariance is used in Long & Bartlett (2023) to derive the Edge of Stability of SAM.

Next, we study the behavior of our algorithm on a wide range of real-world data sets. Here, we find that on both natural and medical image classification data sets, WP SGD provides marginal gains over SGD, in terms of their test performance. By contrast, our algorithm (recall that it uses both negative perturbations and multiple noise injections to reduce the variance of the gradient) can significantly improve upon SGD.

- Across various image classification data sets, 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, detailing their incremental effect on the final performance.

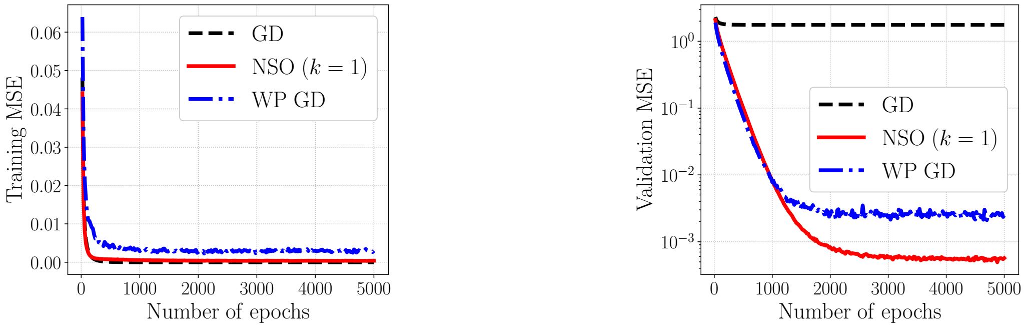
Third, we notice that NSO regularizes the Hessian of the loss surface much more significantly, by noting reductions in the trace (and the largest eigenvalue) of the (loss) Hessian by **17.7**% (and **12.8**%), respectively. Our method is compatible with existing regularization techniques, including weight decay, distance-based regularization, and data augmentation, as combining these techniques leads to even greater improvement in both the Hessian regularization and the test performance.

4.1 Simulations

As discussed briefly in Example 1.1, the Hessian regularization is similar to nuclear norm regularization in the matrix sensing problem. We now flesh this out by conducting a numerical simulation. 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 Noise Stability Optimization (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 the noise variance 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 choose $k = 1$ (for faster computation).

Our findings are illustrated in Figure 3. We can see that all three algorithms can reduce the training MSE to near zero, as shown in Figure 3a. 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 3: Comparing the training and validation losses between GD, NSO, and Weight Perturbed GD.

4.2 Training Neural Networks

Next, we study the behavior of noise injection for training neural networks. We focus on a setting concerning the fine-tuning of pre-trained neural networks, as overfitting is quite common in this setting, hence improving generalization is crucial. We consider fine-tuning a pre-trained ResNet-34 on image classification data sets. The data sets include natural image classification (namely, Aircraft, Caltech-256, Indoor, CIFAR-10, and CIFAR-100), and medical image classification (namely, retina images for diabetic retinopathy classification), downloaded online. For details, see Appendix E.

We use the same training hyper-parameters for the following experiments, including 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 value range for each hyper-parameter is described in Appendix E. We do not use momentum or weight decay (or other advanced techniques in this study), but we will discuss them in the next subsection.

4.2.1 Comparison to Perturbed SGD

We first compare the effect of Weight Perturbed SGD (WP SGD) with SGD. The former samples one perturbation to the model weights in each iteration and conducts SGD using the gradients on the perturbed weights. We compare three image classification data sets, including the aircraft recognition task, the indoor scene recognition task, and the medical image classification task. The results on other datasets are expected to be similar. Recall that this requires setting the standard deviation σ of $\mathcal{P}$. We sample perturbation from an isotropic Gaussian distribution and pick σ based on validation results between $0.008, 0.01$, and 0.012. As shown in Table 2, WP SGD only performs slightly better than SGD.

In addition, we evaluate several alternative types of distributions for sampling the noise perturbations, including Laplace, Uniform, and Binomial distributions. For each distribution, we also select its standard deviation between $0.008, 0.01$, and 0.012 using a validation set. We find that 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.

Table 2: Comparing Weight Perturbed SGD (WP SGD) to SGD, across four types of perturbation distributions (denoted as $\mathcal{P}$), measured over three image classification data sets. The results and their standard deviations are averaged over five independent seeds.

		Aircrafts		Indoor		Retina	
	$\mathcal{P}$	**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

4.2.2 Comparison to Sharpness Reducing Training Methods

We now compare Algorithm 1 (or NSO in short) with four established training methods, which are designed to reduce the sharpness of the loss surface of a neural network, including Sharpness-Aware Minimization (SAM) (Foret et al., 2021), Adaptive SAM (ASAM) (Kwon et al., 2021), Random SAM (RSAM) (Liu et al., 2022), and Bayesian SAM (BSAM) (Möllenhoff & Khan, 2023). These algorithms all compute gradient twice in each iteration. When comparing their comparison with our algorithm, we make sure to use the same amount of computation, by setting the number of sampled injections to be $k = 1$. Thus, all of these methods will use twice the cost of SGD in the end. 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 make sure they are carefully selected. The details are tedious; See Appendix E for the range of values used for each hyper-parameter. To calibrate these results, we include both SGD and Label Smoothing (LS), as they are both widely used in practice.

We report the overall comparison in Table 3. In a nutshell, NSO performs competitively with all the baseline variants. Across these six data sets, 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 that our findings are statistically significant.

Table 3: Comparison between NSO, SGD, Label Smoothing (LS), SAM (and its variants including Adaptive SAM, Random-SAM, and Bayesian SAM), on six image classification data sets, by fine-tuning a pre-trained ResNet-34 neural network on them. In this table, we report the test accuracy, the trace of the Hessian $\mathrm{Tr}[\nabla^2 \hat{L}(f_W)]$ (for W found in the last epoch of each training algorithm), and also the largest eigenvalue of the Hessian $\lambda_1[\nabla^2 \hat{L}(f_W)]$. For the latter two measures, 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. The results indicate that NSO outperforms the baselines in terms of 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
Test Acc. (↑)	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
	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
Trace $\times 10^3$ (↓)	SGD	4.7 ± 0.0	14.4 ± 0.3	6.2 ± 0.0	4.1 ± 0.0	4.1 ± 0.0	30.4 ± 0.2
	LS	2.9 ± 0.0	11.3 ± 0.4	6.3 ± 0.0	3.8 ± 0.0	4.2 ± 0.0	19.2 ± 0.1
	SAM	2.8 ± 0.0	10.2 ± 0.4	5.0 ± 0.0	3.8 ± 0.0	3.8 ± 0.0	16.4 ± 0.2
	ASAM	2.8 ± 0.0	10.5 ± 0.3	5.0 ± 0.0	3.8 ± 0.0	3.1 ± 0.0	14.7 ± 0.1
	RSAM	2.7 ± 0.0	10.3 ± 0.5	5.5 ± 0.2	3.5 ± 0.0	4.1 ± 0.0	19.9 ± 0.5
	BSAM	3.0 ± 0.1	10.3 ± 0.5	5.6 ± 0.1	3.9 ± 0.0	3.5 ± 0.0	18.2 ± 0.3
	NSO	**2.2** ± 0.0	**5.9** ± 0.0	**4.2** ± 0.0	**3.3** ± 0.0	**3.0** ± 0.0	**11.6** ± 0.0
$\boldsymbol{\lambda_1}$ $\times 10^3$ (↓)	SGD	1.5 ± 0.0	4.9 ± 0.1	1.2 ± 0.0	1.1 ± 0.0	1.2 ± 0.1	9.0 ± 0.1
	LS	1.4 ± 0.0	3.5 ± 0.1	1.3 ± 0.1	1.0 ± 0.1	0.9 ± 0.1	4.9 ± 0.0
	SAM	1.4 ± 0.0	2.8 ± 0.1	0.9 ± 0.1	1.0 ± 0.0	1.0 ± 0.1	4.2 ± 0.0
	ASAM	1.4 ± 0.1	2.8 ± 0.1	0.6 ± 0.1	0.8 ± 0.0	**0.7** ± 0.1	4.2 ± 0.0
	RSAM	1.4 ± 0.1	3.0 ± 0.1	0.9 ± 0.1	0.8 ± 0.1	1.0 ± 0.0	5.0 ± 0.0
	BSAM	1.4 ± 0.0	3.0 ± 0.1	1.0 ± 0.1	0.9 ± 0.0	1.0 ± 0.1	4.3 ± 0.2
	NSO	**1.1** ± 0.1	**2.2** ± 0.1	**0.5** ± 0.1	**0.6** ± 0.0	**0.7** ± 0.1	**3.9** ± 0.0

4.2.3 Ablation Studies

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.

Comparing using vs. not using negative perturbations. 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 **1.8**% improvement in test accuracy on average over the one without negative perturbations.

Varying k. Furthermore, increasing the number of perturbations k reduces the variance of the estimated $\nabla F(W)$. 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, increasing k over 3 brings no obvious improvement (but adds more compute cost).

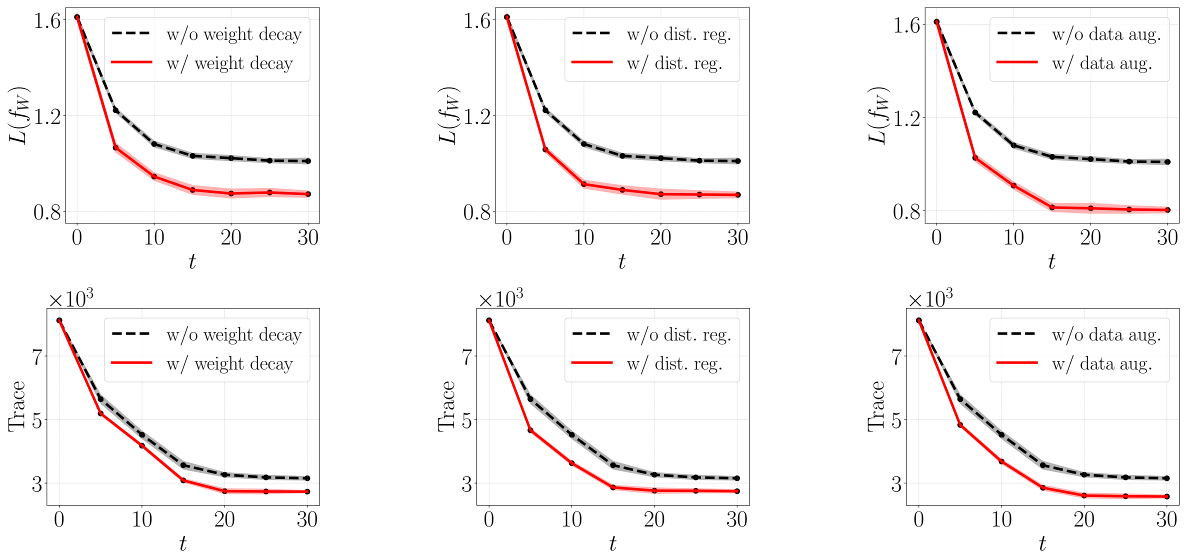


Figure 4: The Hessian regularization can be used in compatible with weight decay, ℓ_2 distance-based regularization, and data augmentation. We illustrate this for fine-tuning a pre-trained ResNet-34 neural network on an image classification data set. Combining each regularization method with ours generally 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.

4.3 Compatibility with Alternative Regularization

Table 3 also shows the regularization effect of each training method on the Hessian. We compute the trace and the λ_1 of the loss Hessian matrix using power iteration implemented by Hessian vector product operations in PyTorch. Notably, in the middle and lower tables, where lower sharpness means better, NSO can significantly reduce them compared to the baselines, averaging **17.7%** (on trace) and **12.8%** (on λ_1).

The regularization on the Hessian can serve as a complement to existing regularization methods, including weight decay, distance-based regularization, and data augmentation. We combine NSO with these methods in the same experiment setup to validate this.[5]

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. Quite strikingly, this leads to **16.3%** lower test loss of the fine-tuned neural network, suggesting that our method can be used on top of these already existing regularization methods.

5 Conclusion

This paper revisits the injection of perturbation to the weight matrices of a neural network. This approach can be rooted in the PAC-Bayes generalization framework, which provides an algorithmic lens to the search for wide minima through regularizing the Hessian. Through extensive empirical analysis in the fine-tuning of pre-trained neural networks, we demonstrate that, a modified version of the noise injection, can indeed be used to effectively regularize the Hessian, improving upon SGD and perturbed SGD. Compared with several sharpness-reducing methods, the modified algorithm uses the same compute cost (twice the cost of running SGD) and yields (statistically significant) empirical improvement. This algorithm's convergence analysis is provided.

---

[5] For distance-based regularization, we penalize the ℓ_2 distance from the fine-tuned model to the pre-trained initialization. For data augmentation, we use a popular scheme that sequentially applies random horizontal flipping and random cropping to each training image.

References

Pierre Alquier. User-friendly introduction to pac-bayes bounds. *arXiv preprint arXiv:2110.11216*, 2021. 3

Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996. 2, 4

Animashree Anandkumar and Rong Ge. Efficient approaches for escaping higher order saddle points in non-convex optimization. In *Conference on learning theory*, pp. 81–102. PMLR, 2016. 4

Maksym Andriushchenko, Dara Bahri, Hossein Mobahi, and Nicolas Flammarion. Sharpness-aware minimization leads to low-rank features. *Advances in Neural Information Processing Systems*, 36, 2024. 4

Francis Bach. Learning theory from first principles. *Online version*, 2021. 17

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. 4

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. 4

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. 4

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. 4

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. 4

Alex Damian, Tengyu Ma, and Jason D Lee. Label noise sgd provably prefers flat global minimizers. *NeurIPS*, 2021. 2

Constantinos Daskalakis, Stratis Skoulakis, and Manolis Zampetakis. The complexity of constrained min-max optimization. In *Symposium on Theory of Computing*, 2021. 4

Yoel Drori and Ohad Shamir. The complexity of finding stationary points with stochastic gradient descent. In *ICML*, 2020. 7

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *UAI*, 2017. 4

Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel Roy. On the role of data in pac-bayes bounds. In *International Conference on Artificial Intelligence and Statistics*, pp. 604–612. PMLR, 2021. 2, 4, 6

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *ICLR*, 2021. 1, 10

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pp. 797–842. PMLR, 2015. 4

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. 6, 17

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011. 2, 4

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. *Advances in neural information processing systems*, 30, 2017. 15

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, 4

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

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International conference on machine learning*, pp. 1724–1732. PMLR, 2017. 4

Haotian Ju, Dongyue Li, and Hongyang R Zhang. Robust fine-tuning of deep neural networks with hessian-based generalization guarantees. *ICML*, 2022. 2, 4, 6

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. 10

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. 15

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. 10

Philip M Long and Peter L Bartlett. Sharpness-aware minimization and the edge of stability. *arXiv preprint arXiv:2309.12488*, 2023. 4, 8

David A McAllester. Some pac-bayesian theorems. *Machine Learning*, 1999. 3

Thomas Möllenhoff and Mohammad Emtiyaz Khan. Sam as an optimal relaxation of bayes. In *International Conference on Learning Representations*, 2023. 4, 10

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *NeurIPS*, 2019. 2

Vaishnavh Nagarajan and J Zico Kolter. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. *ICLR*, 2020. 2, 6

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017. 4

Antonio Orvieto, Anant Raj, Hans Kersting, and Francis Bach. Explicit regularization in overparametrized models via noise injection. *AISTATS*, 2023. 2, 4

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. 2, 15

John Shawe-Taylor and Robert C Williamson. A pac analysis of a bayesian estimator. In *Proceedings of the tenth annual conference on Computational learning theory*, pp. 2–9, 1997. 3

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. 4

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019. 15

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How does sharpness-aware minimization minimize sharpness? *ICLR*, 2023. 4

Rubing Yang, Jialin Mao, and Pratik Chaudhari. Does the data induce capacity control in deep learning? In *International Conference on Machine Learning*, pp. 25166–25197. PMLR, 2022. 4

A Details for Example 1.1

In this section, we expand on Example 1.1. Concretely, for $i = 1, 2, \dots, n$, let $A_i \in \mathbb{R}^{d\times d}$ be a Gaussian random matrix, whose entries are drawn independently from an isotropic Gaussian with mean zero and variance one. Prior work (Gunasekar et al., 2017; Li et al., 2018) has shown that the loss surface of $\hat{L}(W)$ may have spurious solutions whose empirical risk is zero, but differ from the true solution (even after rotation). Below, we argue that among all minimizers such that $\hat{L}(W) \le \epsilon$, $U^\star$ incurs the lowest Hessian trace among all d by d matrices. For a small enough ϵ, the Taylor's expansion of $\hat{L}(W+\epsilon)$ is:

$$\hat{L}(W+\epsilon) = \hat{L}(W) + \langle \epsilon, \nabla \hat{L}(W)\rangle + \frac{1}{2}\epsilon^\top \nabla^2 \hat{L}(W)\epsilon + \mathrm{O}\left(\|\epsilon\|^3 + \|\epsilon\|^4\right). \tag{8}$$

Provided that W is a minimizer of $\hat{L}(W)$, we must have $\hat{L}(W) = 0$. Consequently, the gradient of $\hat{L}(W)$, $\nabla\hat{L}(W)$, is equal to zero:

$$\nabla \hat{L}(W) = \frac{1}{n}\sum_{i=1}^{n}\big(\langle A_i, WW^\top\rangle - y_i\big)A_i W = 0. \tag{9}$$

Therefore, equation equation 8 primarily depends on the second-order Hessian term. We formalize this and state the proof below for completeness.

Proposition A.1. *In the setting of Example 1.1, for any W that satisfies $\hat{L}(W) = 0$, the following must hold with high probability:*

$$\mathrm{Tr}\left[\nabla^2 \hat{L}(U^\star)\right] \le \mathrm{Tr}\left[\nabla^2 \hat{L}(W)\right] + O(n^{-1/2}). \tag{10}$$

Proof. We can calculate the gradient as

$$\nabla \hat{L}(W) = \frac{1}{n}\sum_{i=1}^{n}(\langle A_i, WW^\top\rangle - y_i)A_i W. \tag{11}$$

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

$$\frac{1}{n}\sum_{i=1}^{n}\left([A_i W]_{j,k} A_i W + \Big(\langle A_i, WW^\top\rangle - y_i\Big)\frac{\partial (A_i W)}{\partial W_{j,k}}\right). \tag{12}$$

When $\hat{L}(W)$ is zero, the second term of equation equation 12 above must be zero, because $\langle A_i, WW^\top\rangle$ is equal to y_i, for any $i = 1, \dots, n$.

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 $0 < \epsilon < 1$,

$$\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n}\|A_i W\|_F^2 - d\,\|W\|_F^2\right| \ge \epsilon d\,\|W\|_F^2\right] \le 2\exp\left(-\frac{n\epsilon^2}{8}\right). \tag{13}$$

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 equation 10 holds. ☐

Remark A.2. The same conclusion holds when we replace the trace with the largest eigenvalue of the Hessian in equation equation 10. To see this, we can focus on the first term of equation equation 12. We can look at the quadratic form of the Hessian in order to find the maximum eigenvalue. Let u be a d^2 dimension vector with length equal to one, $\|u\| = 1$. Based on equation equation 12, one can derive that:

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

The last step is by setting $u = d^{-1} \mathbf{1}_{d^2}$, whose length is equal to one.

B Proof of Theorem 3.4

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

$$\begin{aligned} \mathop{\mathbb{E}}_{U_i^1, \ldots, U_i^k} \left[\|\delta_i\|^2 \right] &= \mathop{\mathbb{E}}_{U_i^1, \ldots, U_i^k} \left[\left\| \frac{1}{2k} \sum_{j=1}^{k} \Big(\nabla f(W_i + U_i^j) + \nabla f(W_i - U_i^j) - 2\nabla F(W_i) \Big) \right\|^2 \right] \\ &= \frac{1}{k^2} \sum_{j=1}^{k} \mathop{\mathbb{E}}_{U_i^j} \left[\left\| \frac{1}{2} \Big(\nabla f(W_i + U_i^j) + \nabla f(W_i - U_i^j) - 2\nabla F(W_i) \Big) \right\|^2 \right] && (14) \\ &= \frac{1}{k} \mathop{\mathbb{E}}_{U_i^1} \left[\left\| \frac{1}{2} \Big(\nabla f(W_i + U_i^1) + \nabla f(W_i - U_i^1) \Big) - \nabla F(W_i) \right\|^2 \right] && (15) \end{aligned}$$

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

$$\mathbb{E}\left[\|\delta_i\|^2 \right] = \frac{1}{k} \mathop{\mathbb{E}}_{U \sim \mathcal{P}} \left[\left\| \frac{1}{2} \Big(\nabla f(W_i + U) + f(W_i - U) \Big) - \nabla F(W_i) \right\|^2 \right]. \tag{16}$$

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

$$\mathop{\mathbb{E}}_{U} \left[\left\| \frac{1}{2} \Big(\nabla f(W + U) + \nabla f(W - U) \Big) - \nabla F(W) \right\|^2 \right] \leq C^2 H(\mathcal{P}). \tag{17}$$

We mainly use the Lipschitz continuity of the gradient of F. The left-hand side of equation equation 17 is equal to

$$\begin{aligned}
&\mathop{\mathbb{E}}_{U}\left[\left\|\frac{1}{2}\Big(\nabla f(W+U)-\nabla F(W)\Big)+\frac{1}{2}\Big(\nabla f(W-U)-\nabla F(W)\Big)\right\|^2\right]\\
\le&\mathop{\mathbb{E}}_{U}\left[\frac{1}{2}\left\|\nabla f(W+U)-\nabla F(W)\right\|^2+\frac{1}{2}\left\|\nabla f(W-U)-\nabla F(W)\right\|^2\right] && \text{(by Cauchy-Schwartz)}\\
=&\frac{1}{2}\mathop{\mathbb{E}}_{U}\left[\left\|\nabla f(W+U)-\nabla F(W)\right\|^2\right] && \text{(by symmetry of } \mathcal{P} \text{ since it has mean zero)}\\
=&\frac{1}{2}\mathop{\mathbb{E}}_{U}\left[\left\|\mathop{\mathbb{E}}_{U'\sim\mathcal{P}}\left[\nabla f(W+U)-\nabla f(W+U')\right]\right\|^2\right]\\
\le&\frac{1}{2}\mathop{\mathbb{E}}_{U}\left[\mathop{\mathbb{E}}_{U'\sim\mathcal{P}}\left[\left\|\nabla f(W+U)-\nabla f(W+U')\right\|^2\right]\right]\\
\le&\frac{1}{2}\mathop{\mathbb{E}}_{U,U'}\left[C^2\left\|U-U'\right\|^2\right]=\frac{1}{2}C^2\mathop{\mathbb{E}}_{U,U'}\left[\|U\|^2+\|U'\|^2\right]=C^2H(\mathcal{P}) && \text{(by equation equation 19)}
\end{aligned}$$

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

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

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 3.4, 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* $j=0,\dots,T-1$*, the following holds:*

$$\mathbb{E}\left[\left\|\nabla F(W_t)\right\|^2\right]\le\frac{2C}{\sum_{i=0}^{T-1}\eta_i}D^2+\frac{C\sum_{i=0}^{T-1}\eta_i^2\big(\mathbb{E}\left[\|\delta_i\|^2\right]+\mathbb{E}\left[\|\xi_i\|^2\right]\big)}{\sum_{i=0}^{T-1}\eta_i}. \tag{18}$$

Proof. The smoothness condition in Assumption 3.3 implies the following domination inequality:

$$\left|F(W_2)-F(W_1)-\langle\nabla F(W_1),W_2-W_1\rangle\right|\le\frac{C}{2}\left\|W_2-W_1\right\|^2. \tag{19}$$

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

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

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

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

which implies that

$$\begin{aligned}&\sum_{i=0}^{T-1} \Big(\eta_i - \frac{C\eta_i^2}{2}\Big) \left\|\nabla F(W_i)\right\|^2 && (20)\\ \leq& F(W_0) - F(W_T) - \sum_{i=0}^{T-1} \Big(\eta_i - C\eta_i^2\Big) \langle \nabla F(W_i), \delta_i + \xi_i \rangle + \frac{C}{2} \sum_{i=0}^{T-1} \eta_i^2 \left\|\delta_i + \xi_i\right\|^2 \\ \leq& D^2 - \sum_{i=0}^{T-1} \Big(\eta_i - C\eta_i^2\Big) \langle \nabla F(W_i), \delta_i + \xi_i \rangle + \frac{C}{2} \sum_{i=0}^{T-1} \eta_i^2 \left\|\delta_i + \xi_i\right\|^2. && (21)\end{aligned}$$

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

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

which implies that

$$\frac{1}{2} \sum_{i=0}^{T-1} \eta_i \left\|\nabla F(W_i)\right\|^2 \leq D^2 - \sum_{i=0}^{T-1} \Big(\eta_i - C\eta_i^2\Big) \langle \nabla F(W_i), \delta_i + \xi_i \rangle + \frac{C}{2} \sum_{i=0}^{T-1} \eta_i^2 \left\|\delta_i + \xi_i\right\|^2. \quad (22)$$

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

$$\mathop{\mathbb{E}}_{U_t}[\delta_t] = \frac{1}{2} \mathop{\mathbb{E}}_{U_t}[\nabla f(W_t - U_t) - \nabla f(W_t + U_t)] = 0. \quad (23)$$

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

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

Recall that t is a random variable whose probability mass is specified in Lemma B.1. We can write equation equation 22 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 $\Pr[t = i] = \frac{\eta_i}{\sum_{j=0}^{T-1} \eta_j}$)

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

where we use the fact that δ_i and ξ_i are independent for any i. Hence, we have finished the proof of equation equation 18. ☐

Based on the above result, we now finish the proof of the upper bound in Proposition 3.4.

Proof. Let the step sizes be equal to a fixed η for all epochs. Thus, Eq. equation 18 becomes

$$\mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \le \frac{2}{T\eta}D^2 + \frac{C\eta}{T}\sum_{i=0}^{T-1}\left(\mathbb{E}\left[\|\delta_i\|^2\right] + \mathbb{E}\left[\|\xi_i\|^2\right]\right). \tag{24}$$

By Lemma 3.5,

$$\sum_{i=0}^{T-1}\left(\mathbb{E}\left[\|\delta_i\|^2\right] + \mathbb{E}\left[\|\xi_i\|^2\right]\right) \le T \cdot \frac{\sigma^2 + C^2 H(\mathcal{P})}{k}. \tag{25}$$

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 $\Delta \ge 2CD^2/T$. Then, minimizing over η above leads us to the following upper bound on the right-hand side of equation equation 24:

$$\sqrt{\frac{2CD^2\Delta}{T}}, \tag{26}$$

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 equation 24 must be less than

$$\sqrt{\frac{2CD^2\Delta}{T}} = \sqrt{\frac{2CD^2(\sigma^2 + C^2 H(\mathcal{P}))}{kT}}. \tag{27}$$

Case 2: Δ is small. In this case, suppose $\Delta < 2CD^2/T$. Then, the right-hand side of equation equation 24 must be less than

$$\frac{2D^2}{T\eta} + \frac{2C^2D^2\eta}{T} \le \frac{2CD^2}{T}. \tag{28}$$

Thus, combining equations equation 27 and equation 28, we have completed the proof of equation equation 3.

☐

C Proof of Theorem 3.6

Recall our construction from Section 3.2 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:

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

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 $i = 0, 1, \dots, d-1$. We let $\mathcal{P}$ denote the distribution of $\tilde{U}$.

The proof of Theorem 3.6 is divided into two cases. In the first, we examine the case when the averaged learning rate is $O(T^{-1/2})$.

Lemma C.1. *In the setting of Theorem 3.6, suppose the learning rates satisfy that* $\sum_{i=0}^{T-1} \eta_i \le \sqrt{\frac{D^2 kT}{2\sigma^2 C}}$, *consider the function* $f(W)$ *constructed in equation equation 5, we have*

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

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

$$\xi_t = \langle \xi_t, e_{t+1} \rangle e_{t+1} \text{ and } |\langle \xi_t, e_{t+1} \rangle| \le \frac{\sigma}{\sqrt{k}}, \tag{29}$$

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 equation 29. Thus, we can also conclude that

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

We consider the objective function $f(W) : \mathbb{R}^d \to \mathbb{R}$ defined above (see also equation equation 5, Section 3.1), with

$$\alpha_i = \frac{2\eta_i \sigma}{\sqrt{k}}, \text{ for } i = 0, 1, \dots, T. \tag{30}$$

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 equation 30 above, and the truncation of U, which implies $|\langle U, e_1 \rangle| \le \frac{\eta_0 \sigma}{\sqrt{k}}$, and $\langle \bar{\xi}_0, e_1 \rangle \le \frac{\sigma}{\sqrt{k}}$, we obtain

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

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 $|\langle U, e_{i+1}\rangle| \le \frac{\eta_i\sigma}{\sqrt{k}}$, 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

$$\begin{aligned} W_{t+1} &= W_t - \eta_t\Big(G^{-1}\langle W_t, e_0\rangle + \bar{\xi}_t\Big) \\ &= W_0\prod_{i=0}^{t}\Big(1-\eta_i G^{-1}\Big) - \sum_{i=0}^{t}\eta_i\bar{\xi}_i. \end{aligned} \tag{31}$$

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

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

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

$$\frac{1}{T}\Big(\sum_{i=0}^{T-1}\eta_i\Big)^2 \le \sum_{i=0}^{T-1}\eta_i^2 \le \frac{kD^2}{2C\sigma^2}. \tag{32}$$

We note that for all $z \in [0,1]$, $1 - \frac{z}{2} \ge \exp(\log\frac{z}{2})$. Thus, applying this to the right-hand side of equation equation 31, we obtain that for any t,

$$\prod_{i=0}^{t}\Big(1-\eta_i G^{-1}\Big) \ge \frac{1}{2}, \tag{33}$$

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

$$\begin{aligned} \min_{1\le i\le T}\|\nabla F(W_i)\|^2 &= \min_{1\le i\le T}\Big(G^{-1}\langle W_0, e_0\rangle\Big)^2 && \text{(by the construction of } F(\cdot)\text{)} \\ &\ge \frac{1}{4}G^{-2}(D\sqrt{G})^2 && \text{(by equations equation 31 and equation 33)} \\ &= \frac{D^2}{4}\min\Big\{C, \frac{1}{2\sum_{i=0}^{T-1}\eta_i}\Big\} && \text{(recall the definition of } G \text{ above)} \\ &\ge \frac{D^2}{4}\min\Big\{C, \frac{\sqrt{2C\sigma^2}}{2D\sqrt{kT}}\Big\} && \text{(by equation equation 32)} \\ &\ge D\sqrt{\frac{C\sigma^2}{32kT}}. \end{aligned}$$

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

Thus, we have proved that equation equation 6 holds for W_i for any $i = 1, 2, \dots, T$. The proof of Lemma C.1 is finished. ☐

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

Lemma C.2. *In the setting of Theorem 3.6, suppose the learning rates satisfy that* $\sum_{i=0}^{T-1} \eta_i \ge \sqrt{\frac{D^2 kT}{2\sigma^2 C}}$ *and* $\eta_i = \eta$ *for some fixed* $\eta \le C^{-1}$*. Then, consider the function from equation equation 5, we have that* $\min_{1\le t\le T} \mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \ge D\sqrt{\frac{C\sigma^2}{32kT}}$.

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:

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

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

$$f(W) = g(\langle W, e_0\rangle) + \sum_{t=0}^{d-1} \frac{C}{2}\langle W, e_{t+1}\rangle^2. \tag{34}$$

From the definition of f, f also has C-Lipschitz gradient. Similar to equation equation 29, 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

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

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

$$|\langle \bar{\xi}_t, e_{t+1}\rangle| \le c\sigma \text{ and } \mathbb{E}\left[\|\bar{\xi}_t\|^2\right] \le \frac{\sigma^2}{k}. \tag{35}$$

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 $i = 0, 1, \dots, T-1$. 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 $\alpha = \frac{1-\rho^T}{1-\rho}\cdot 2c\eta\sigma$, $|\langle U, e_0\rangle| \le c\eta\sigma$, and $\langle \bar{\xi}_0, e_0\rangle \le c\sigma$, 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 $i = 1, 2, \dots, T-1$.

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

$$\begin{aligned} W_{t+1} &= W_t - \eta\Big(C\sum_{i=1}^{d}\langle W_t, e_i\rangle - C\sum_{i=0}^{t-1}(1 - C\eta)^{t-1-i}\eta\bar{\xi}_i + \bar{\xi}_t\Big) \\ &= (1 - C\eta)^{t+1} W_0 - \sum_{i=0}^{t}(1 - C\eta)^{t-1-i}\eta\bar{\xi}_i. \end{aligned} \tag{36}$$

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

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

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

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

Thus, we have proved this lemma. □

Taking both Lemma C.1 and C.2 together, we thus conclude the proof of Theorem 3.6.

D Proof of Theorem 3.7

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 $f(W) = \frac{C}{2}\left\|W\right\|^2$, 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

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

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:

$$W_{t+1} = \Big(1 - C\eta_t\Big)W_t - \eta_t\bar{\xi}_t, \tag{37}$$

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 $\bar{\xi}_1, \bar{\xi}_2, \dots, \bar{\xi}_T$ forms a martingale sequence:

- For any $i = 1, 2, \dots, T$, conditioned on the previous random variables $\xi_{i'}^{(j)}$ for any $i' < i$ and any $j = 1, 2, \dots, k$, 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 $\left\|W_t\right\|^2$, as shown in the following result.

Lemma D.1. *In the setting of Theorem 3.7, for any step sizes $\eta_0, \dots, \eta_{T-1}$ less than C^{-1}, and any $t = 1, \dots, T$, the expected gradient of W_t, $\mathbb{E}\left[\left\|\nabla F(W_t)\right\|^2\right]$, is equal to*

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

Proof. By iterating over equation equation 37, 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,

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

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

$$\begin{aligned}\mathbb{E}\left[\left\|\nabla F(W_t)\right\|^2\right] = {} & C^2\left\|W_0\right\|^2\prod_{j=0}^{t-1}\left(1 - C\eta_j\right)^2 \\ & + C^2\sum_{i=0}^{t-1}\mathbb{E}\left[\left\|\eta_i\bar{\xi}_i\prod_{j=i+1}^{t-1}\left(1 - C\eta_j\right)\right\|^2\right].\end{aligned} \tag{38}$$

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 equation 38 is equal to

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

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:

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

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

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

Lastly, let W_0 be a vector such that

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

Setting $\|W_0\|=D\sqrt{2C^{-1}}$ in equation equation 38 leads to

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

Thus, we conclude the proof of this result. □

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

$$f(W)=\frac{1}{2\kappa}\|W\|^2,\ \text{where } \kappa=\max\{C^{-1},2\sum_{i=0}^{T-1}\eta_i\},\tag{39}$$

Lemma D.2. *Consider f given in equation equation 39 above. For any step sizes $\eta_0,\dots,\eta_{T-1}$ less than C^{-1}, the following holds for the stochastic objective F:*

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

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

$$\|\nabla F(W)\|=\frac{1}{\kappa}\|W\|.\tag{40}$$

Following the update rule in NSO, similar to equation equation 37, W_t evolves as follows:

$$W_{t+1}=\left(1-\frac{\eta_t}{\kappa}\right)W_t-\eta_t\bar{\xi}_t,\tag{41}$$

where $\bar{\xi}_t$ has variance equal to σ^2/k, according to the proof of Lemma D.1. By iterating equation equation 41 from the initialization, we can get a closed-form equation for $W_t^{(1)}$, for any $t = 1, 2, \ldots, T$:

$$W_t = W_0 \prod_{j=0}^{t-1} \left(1 - \frac{\eta_j}{\kappa}\right) - \sum_{k=0}^{t-1} \eta_k \xi_k \prod_{j=k+1}^{t-1} \left(1 - \frac{\eta_j}{\kappa}\right). \tag{42}$$

Following equation equation 40, we can show that

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

Thus, in expectation,

$$\begin{aligned}
\mathbb{E}\left[\|\nabla F(W_t)\|^2\right] &= \kappa^{-2}\, \mathbb{E}\left[\|W_t\|^2\right] \\
&= \kappa^{-2} \|W_0\|^2 \prod_{j=0}^{t-1} \left(1 - \kappa^{-1}\eta_j\right)^2 + \kappa^{-2} \sum_{i=0}^{t-1} \mathbb{E}\left[\left(\eta_i \bar{\xi}_i \prod_{j=i+1}^{t-1} \left(1 - \kappa^{-1}\eta_j\right)\right)^2\right] \\
&= \kappa^{-2} \|W_0\|^2 \prod_{j=0}^{t-1} \left(1 - \kappa^{-1}\eta_j\right)^2 + \kappa^{-2} \sum_{i=0}^{t-1} \eta_i^2 \prod_{j=i+1}^{t-1} \left(1 - \kappa^{-1}\eta_j\right)^2 \mathbb{E}\left[\|\bar{\xi}_i\|^2\right] \\
&= 2D^2\kappa^{-1} \prod_{j=0}^{t-1} \left(1 - \kappa^{-1}\eta_j\right)^2 + \frac{\sigma^2\kappa^{-2}}{k} \sum_{i=0}^{t-1} \eta_i^2 \prod_{j=i+1}^{t-1} \left(1 - \kappa^{-1}\eta_j\right)^2,
\end{aligned} \tag{43}$$

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 equation 43, we note that for all $z \in [0, 1]$,

$$1 - \frac{z}{2} \geq \exp\left(\log\frac{1}{2} \cdot z\right). \tag{44}$$

Hence, applying equation equation 44 to the right-hand side of equation equation 43, 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 equation 43 must be at least

$$\mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \geq \frac{2D^2\kappa^{-1}}{4} + \frac{\sigma^2\kappa^{-2}}{k} \sum_{i=0}^{t-1} \frac{\eta_i^2}{4}. \tag{45}$$

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 D.2 is finished. □

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

Lemma D.3. *There exists convex quadratic functions f such that for any gradient oracle satisfying Assumption 3.2 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:*

$$\min_{1\leq t\leq T} \mathbb{E}\left[\|\nabla F(W_t)\|^2\right] \geq D\sqrt{\frac{C\sigma^2}{32k \cdot T}}. \tag{46}$$

Proof. By Lemma D.2, there exists a function such that the left-hand side of equation equation 46 is at least

$$\frac{D^2}{2\max\{C^{-1}, 2\sum_{i=0}^{T-1}\eta_i\}} \geq \frac{CD^2}{2\max\{1, 2x^{-1}\sqrt{T}\}} = \frac{D^2 x}{4\sqrt{T}}, \tag{47}$$

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

On the other hand, if $\sum_{i=0}^{T-1}\eta_i \geq x^{-1}\sqrt{T}$ and $\eta_i = \eta$ for a fixed η, then $\eta > x^{-1}/\sqrt{T}$. By setting $\eta_i = \eta$ for all i in Lemma D.1, the left-hand side of equation equation 46 is equal to

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

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

$$\begin{aligned}
&\min_{1\leq t\leq T}\left(2CD^2(1-\rho)^{2t} + \frac{\sigma^2\rho^2}{k}\frac{1-(1-\rho)^{2t}}{1-(1-\rho)^2}\right)\\
&= \min_{1\leq t\leq T}\left(\frac{\sigma^2\rho}{k(2-\rho)} + (1-\rho)^{2t}\Big(2CD^2 - \frac{\sigma^2\rho}{k(2-\rho)}\Big)\right).
\end{aligned} \tag{48}$$

If $2CD^2 < \frac{\sigma^2\rho}{k(2-\rho)}$, the above is the smallest when $t = 1$. In this case, equation equation 48 is equal to

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

If $2CD^2 \geq \frac{\sigma^2\rho}{k(2-\rho)}$, the above is the smallest when $t = T$. In this case, equation equation 48 is at least

$$\frac{\sigma^2\rho}{k(2-\rho)} \geq \frac{\sigma^2\rho}{2k} \geq \frac{\sigma^2 C x^{-1}}{2k}\cdot\frac{1}{\sqrt{T}}. \tag{49}$$

To conclude the proof, we set x so that the right-hand side of equations equation 47 and equation 49 match each other. This leads to

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

Thus, by combining the conclusions from both equations equation 47 and equation 49 with this value of x, we finally conclude that if $\sum_{i=0}^{T-1}\eta_i \leq \sqrt{T}x^{-1}$, 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 equation 46 holds. This completes the proof of Lemma D.3. ☐

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

$$X_u = \begin{bmatrix} 1 - C\eta_t & \mu \\ C\eta_t & \mu \end{bmatrix}. \tag{50}$$

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 D.3 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 D.1, we can apply the above iterative update to obtain the formula for W_{t+1} as:

$$\begin{bmatrix} W_{t+1} \\ M_{t+1} \end{bmatrix} = X_u^t \begin{bmatrix} W_0 \\ M_0 \end{bmatrix} + \sum_{i=0}^{t} C\eta X_u^{t-i} \begin{bmatrix} \bar{\xi}_i \\ \bar{\xi}_i \end{bmatrix}. \tag{51}$$

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:

$$\mathbb{E}\left[\left\|W_{t+1}\right\|^2\right] = 2CD^2(e_1^\top X_u^t e_1)^2 + \frac{C^2\eta^2\sigma^2}{k} \sum_{i=0}^{t} \left\|e_1^\top X_u^i e\right\|^2. \tag{52}$$

Above, similar to Lemma D.1, 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 equation 51). Let its singular values be denoted as λ_1 and λ_2. In addition, to deal with equation equation 52, 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

$$(e_1^\top X_\mu^i e)^2 = \alpha_1^2 \lambda_1^{2i} + \alpha_2^2 \lambda_2^{2i}. \tag{53}$$

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

$$x^2 - ((1 - C\eta)^2 + (C\eta)^2 + 2\mu^2)x + \mu^2 = 0. \tag{54}$$

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:

$$\lambda_1, \lambda_2 = \frac{(1 - C\eta)^2 + (C\eta)^2 + 2\mu^2 \pm \sqrt{((1 - C\eta)^2 + (C\eta)^2 + 2\mu^2)^2 - 4\mu^2}}{2}. \tag{55}$$

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

$$\alpha_1^2 = \frac{-C\eta(1 - C\eta) + \mu^2}{(1 - C\eta)^2 + \mu^2 - \lambda_1 + -C\eta(1 - C\eta) + \mu^2}. \tag{56}$$

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 equation 53,

$$\alpha_1^2\lambda_1^{2i} + \alpha_2^2\lambda_2^{2i} \gtrsim \lambda_1^{2i} + \lambda_2^{2i}. \tag{57}$$

By the Hölder's inequality,

$$\begin{aligned}(\lambda_1^{2i} + \lambda_2^{2i})^{\frac{1}{2i}}(1+1)^{1-\frac{1}{2i}} &\geq \lambda_1 + \lambda_2 = (1 - C\eta)^2 + (C\eta)^2 + 2\mu^2 &(58)\\ &\geq (1 - C\eta)^2 + (C\eta)^2, &(59)\end{aligned}$$

which implies that

$$\lambda_1^{2i} + \lambda_2^{2i} \geq \frac{((1 - C\eta)^2 + (C\eta)^2)^i}{2^{(2i-1)}}. \tag{60}$$

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 D.3 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 D.3, 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 3.7. □

E Omitted Experiments

Below, we report the comparison of the λ_1 of the Hessian matrix (in the same setting as Figure 1), at the last epoch of fine-tuning.

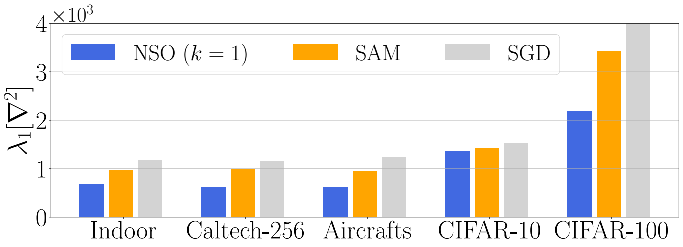


Figure 5: Reporting the λ_1 of the Hessian matrix in the last iteration of fine-tuning ResNet-34 on five data sets, comparing NSO with SAM and SGD. The results are averaged over five random seeds.

We describe the range of hyper-parameters in which we search for each algorithm. We conduct a grid search with a validation split.

- 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.

We also choose 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.

The datasets in our experiments can also be found online.