Fisher SAM: Information Geometry and Sharpness Aware Minimisation

Minyoung Kim [1] **Da Li** [1] **Shell Xu Hu** [1] **Timothy M. Hospedales** [1 2]

Abstract

Recent sharpness-aware minimisation (SAM) is known to find flat minima which is beneficial for better generalisation with improved robustness. SAM essentially modifies the loss function by reporting the maximum loss value within the small neighborhood around the current iterate. However, it uses the Euclidean ball to define the neighborhood, which can be inaccurate since loss functions for neural networks are typically defined over probability distributions (e.g., class predictive probabilities), rendering the parameter space non Euclidean. In this paper we consider the information geometry of the model parameter space when defining the neighborhood, namely replacing SAM's Euclidean balls with ellipsoids induced by the Fisher information. Our approach, dubbed Fisher SAM, defines more accurate neighborhood structures that conform to the intrinsic metric of the underlying statistical manifold. For instance, SAM may probe the worst-case loss value at either a too nearby or inappropriately distant point due to the ignorance of the parameter space geometry, which is avoided by our Fisher SAM. Another recent Adaptive SAM approach stretches/shrinks the Euclidean ball in accordance with the scale of the parameter magnitudes. This might be dangerous, potentially destroying the neighborhood structure. We demonstrate improved performance of the proposed Fisher SAM on several benchmark datasets/tasks.

1. Introduction

Contemporary deep learning models achieve state of the art generalisation performance on a wide variety of tasks. These models are often massively overparameterised, and capable of memorizing the entire training set (Zhang et al., 2017). The training loss landscape of such models is complex and non-convex with multiple local and global minima of highly varying generalisation performance. Good performance is therefore obtained by exploiting various explicit and implicit regularisation schemes during learning to find local minima in the training loss that actually generalise well. Methods such as dropout (Srivastava et al., 2014), weight-decay, and data augmentation have been developed to provide explicit regularisation, while the dynamics of optimisers such as SGD can provide implicit regularisation, by finding solutions with low-norm weights (Chaudhar et al., 2017; Zhang et al., 2017).

A number of studies have linked the flatness of a given training minima to generalisation quality (Keskar et al., 2017; Chaudhar et al., 2017). Searching for flat minima of the loss function is intuitively appealing, as it is obviously beneficial for finding models resilient to data noise and/or model parameter corruption/perturbation. This has led to an increasing number of optimisation methods (Chaudhar et al., 2017; Foret et al., 2021; Sun et al., 2021) designed to explicitly search for flat minima. Despite this variety of noteworthy theoretical and empirical work, existing approaches have yet to scalably solve this problem, as developing computationally efficient methods for finding flat minima is non-trivial.

A seminal method in this area is known as sharpness-aware minimisation (SAM) (Foret et al., 2021). SAM is a minimax type algorithm that essentially modifies the loss function to report the maximum loss value within the small neighborhood around the current iterate. Optimising with SAM thus prefers flatter minima than conventional SGD. However, one of the main drawbacks of SAM is that it uses a Euclidean ball to define the neighborhood, which is inaccurate since loss functions for neural networks are typically defined over probability distributions (e.g., class predictive probabilities), rendering the parameter space non Euclidean. Another recent approach called Adaptive SAM (ASAM) (Kwon et al., 2021) stretches/shrinks the Euclidean ball in accordance with the scales of the parameter magnitudes. However, this approach to determining the flatness ellipsoid of interest is heuristic and might severely degrade the neighborhood structure. Although SAM and ASAM are successful in many empirical tasks, ignorance of the underlying geometry of the model parameter space may lead to suboptimal results.

[1]Samsung AI Center, Cambridge, UK [2]University of Edinburgh. Correspondence to: Minyoung Kim <mikim21@gmail.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

In this paper we build upon the ideas of SAM, but address the issue of a principled approach to determining the ellipsoid of interest by considering information geometry (Amari, 1998; Murray & Rice, 1993) of the model parameter space when defining the neighborhood. Specifically, we replace SAM's Euclidean balls with ellipsoids induced by the Fisher information. Our approach, dubbed Fisher SAM, defines more accurate neighborhood structures that conform to the intrinsic metric of the underlying statistical manifold. By way of comparison, SAM may probe the worst-case loss value at either a too nearby or too far point due to using a spherical neighborhood. In contrast Fisher SAM avoids this by probing the worst-case point within the ellipsoid derived from the Fisher information at the current point – thus providing a more principled and optimisation objective, and improving empirical generalisation performance.

Our main contributions are as follows:

1. We propose a novel information geometry and sharpness aware loss function which addresses the above-mentioned issues of the existing flat-minima optimisation approaches.
2. Our Fisher SAM is as efficient as SAM, only requiring double the cost of that of vanilla SGD, using the gradient magnitude approximation for Fisher information matrix. We also justify this approximation.
3. We provide a theoretical generalisation bound similar to SAM's using the prior covering proof technique in PAC-Bayes, in which we extend SAM's spherical Gaussian prior set to an ellipsoidal full-covariance set.
4. We demonstrate improved empirical performance of the proposed FSAM on several benchmark datasets and tasks: image classification, ImageNet overtraining, finetuning; and label-noise robust learning; and robustness to parameter perturbation during inference.

2. Background

Although flatness/sharpness of the loss function can be formally defined using the Hessian, dealing with (optimizing) the Hessian function is computationally prohibitive. As a remedy, the sharpness-aware minimisation (SAM for short) (Foret et al., 2021) introduced a novel robust loss function, where the new loss at the current iterate is defined as the maximum (worst-case) possible loss within the neighborhood at around it. More formally, considering a γ-ball neighborhood, the robust loss l^γ is defined as:

$$l^\gamma(\theta) = \max_{\|\epsilon\| \le \gamma} l(\theta + \epsilon), \tag{1}$$

where θ is the model parameters (iterate), and $l(\theta)$ is the original loss function. Using the first-order Taylor (linear) approximation of $l(\theta + \epsilon)$, (1) becomes the famous dual-norm problem (Boyd & Vandenberghe, 2004), admitting a closed-form solution. In the Euclidean (L2) norm case, the solution becomes the normalised gradient,

$$\epsilon^*_{SAM}(\theta) = \gamma \frac{\nabla l(\theta)}{\|\nabla l(\theta)\|}. \tag{2}$$

Plugging (2) into (1) defines the SAM loss, while its gradient can be further simplified by ignoring the (higher-order) gradient terms in $\nabla \epsilon^*(\theta)$ for computational tractability:

$$l^\gamma_{SAM}(\theta) = l(\theta'), \;\; \nabla l^\gamma_{SAM}(\theta) = \frac{\partial l(\theta)}{\partial \theta}\Big|_{\theta=\theta'} \tag{3}$$
$$\text{where } \; \theta' = \theta + \epsilon^*_{SAM}(\theta).$$

In terms of computational complexity, SAM incurs only twice the forward/backward cost of the standard SGD: one forward/backward for computing $\epsilon^*_{SAM}(\theta)$ and the other for evaluating the loss and gradient at $\theta' = \theta + \epsilon^*_{SAM}(\theta)$.

More recently, a drawback of SAM, related to the model parameterisation, was raised by (Kwon et al., 2021), in which SAM's fixed-radius γ-ball can be sensitive to the parameter re-scaling, weakening the connection between sharpness and generalisation performance. To address the issue, they proposed what is called Adaptive SAM (ASAM for short), which essentially re-defines the neighborhood γ-ball with the magnitude-scaled parameters. That is,

$$l^\gamma_{ASAM}(\theta) = \max_{\|\epsilon/|\theta|\| \le \gamma} l(\theta + \epsilon), \tag{4}$$

where $\epsilon/|\theta|$ is the elementwise operation (i.e., $\epsilon_i/|\theta_i|$ for each axis i). It leads to the following maximum (worst-case) probe direction within the neighborhood,

$$\epsilon^*_{ASAM}(\theta) = \gamma \frac{\theta^2 \nabla l(\theta)}{\|\theta \nabla l(\theta)\|} \quad \text{(elementwise ops.)}. \tag{5}$$

The loss and gradient of ASAM are defined similarly as (3) with $\theta' = \theta + \epsilon^*_{ASAM}(\theta)$.

3. Our Method: Fisher SAM

ASAM's γ-neighborhood structure is a function of θ, thus not fixed but adaptive to parameter scales in a quite intuitive way (e.g., more perturbation allowed for larger θ_i, and vice versa). However, ASAM's parameter magnitude-scaled neighborhood choice is rather ad hoc, not fully reflecting the underlying geometry of the parameter manifold.

Note that the loss functions for neural networks are typically dependent on the model parameters θ only through the predictive distributions $p(y|x, \theta)$ where y is the target variable (e.g., the negative log-likelihood or cross-entropy loss, $l(\theta) = \mathbb{E}_{x,y}[-\log p(y|x, \theta)]$). Hence the geometry of the

parameter space is not Euclidean but a *statistical manifold* induced by the Fisher information metric of the distribution $p(y|x,\theta)$ (Amari, 1998; Murray & Rice, 1993).

The intuition behind the Fisher information and statistical manifold can be informally stated as follows. When we measure the distance between two neural networks with parameters θ and θ', we should compare the underlying distributions $p(y|x,\theta)$ and $p(y|x,\theta')$. The Euclidean distance of the parameters $\|\theta-\theta'\|$ does not capture this distributional divergence because two distributions may be similar even though θ and θ' are largely different (in L2 sense), and vice versa. For instance, even though $p(x|\theta)=\mathcal{N}(\mu, 1+0.001\sigma)$ and $p(x|\theta')=\mathcal{N}(\mu', 1+0.001\sigma')$ with $\theta=(\mu=1,\sigma=10)$ and $\theta'=(\mu'=1,\sigma'=20)$ have large L2 distance, the underlying distributions are nearly the same. That is, the Euclidean distance is not a good metric for the parameters of a distribution family. We need to use *statistical divergence* instead, such as the Kullback-Leibler (KL) divergence, from which the Fisher information metric can be derived.

Based on the idea, we propose a new SAM algorithm that fully reflects the underlying geometry of the statistical manifold of the parameters. In (1) we replace the Euclidean distance by the KL divergence[1]:

$$l^{\gamma}_{FSAM}(\theta) = \max_{d(\theta+\epsilon,\theta)\leq\gamma^2} l(\theta+\epsilon) \;\text{ where} \tag{6}$$
$$d(\theta',\theta) = \mathbb{E}_x[\mathrm{KL}(p(y|x,\theta')||p(y|x,\theta))],$$

which we dub Fisher SAM (FSAM for short). For small ϵ, it can be shown that $d(\theta+\epsilon,\theta)\approx\epsilon^\top F(\theta)\epsilon$ (See Appendix B for details), where $F(\theta)$ is the Fisher information matrix,

$$F(\theta) = \mathbb{E}_x\mathbb{E}_\theta\Big[\nabla\log p(y|x,\theta)\nabla\log p(y|x,\theta)^\top\Big]. \tag{7}$$

That is, our Fisher SAM loss function can be written as:

$$l^{\gamma}_{FSAM}(\theta) = \max_{\epsilon^\top F(\theta)\epsilon\leq\gamma^2} l(\theta+\epsilon). \tag{8}$$

We solve (8) using the first-order approximated objective $l(\theta+\epsilon)\approx l(\theta)+\nabla l(\theta)^\top\epsilon$, leading to a quadratic constrained linear programming problem. The Lagrangian is

$$\mathcal{L}(\epsilon,\lambda) = l(\theta)+\nabla l(\theta)^\top\epsilon-\lambda(\epsilon^\top F(\theta)\epsilon-\gamma^2), \tag{9}$$

and solving $\frac{\partial\mathcal{L}}{\partial\epsilon}=0$ yields $\epsilon^*=\frac{1}{2\lambda}F(\theta)^{-1}\nabla l(\theta)$. Plugging this into the ellipsoidal constraint (from the KKT conditions) determines the optimal λ, resulting in:

$$\epsilon^*_{FSAM}(\theta) = \gamma\frac{F(\theta)^{-1}\nabla l(\theta)}{\sqrt{\nabla l(\theta)F(\theta)^{-1}\nabla l(\theta)}}. \tag{10}$$

[1]To be more rigorous, one can consider the *symmetric* Jensen-Shannon divergence, $d(\theta',\theta)=0.5\cdot\mathbb{E}_x[\mathrm{KL}(\theta'||\theta)+\mathrm{KL}(\theta||\theta')]$. For $\theta'\approx\theta$, however, the latter KL term vanishes (easily verified using the derivations similar to those in Appendix B), and it coincides with the KL divergence in (6) (up to a constant factor).

The loss and gradient of Fisher SAM are defined similarly as (3) with $\theta'=\theta+\epsilon^*_{FSAM}(\theta)$.

Approximating Fisher. Dealing with a large dense matrix $F(\theta)$ (and its inverse) is prohibitively expensive. Following the conventional practice, we consider the empirical diagonalised minibatch approximation,

$$F(\theta)\approx\frac{1}{|B|}\sum_{i\in B}\mathrm{Diag}\big(\nabla\log p(y_i|x_i,\theta)\big)^2, \tag{11}$$

for a minibatch $B=\{(x_i,y_i)\}$. $\mathrm{Diag}(v)$ is a diagonal matrix with vector v embedded in the diagonal entries. However, it is still computationally cumbersome to handle *instance-wise* gradients in (11) using the off-the-shelf auto-differentiation numerical libraries such as PyTorch (Paszke et al., 2019), Tensorflow (Abadi et al., 2015) or JAX (Bradbury et al., 2018) that are especially tailored for the *batch sum* of gradients for the best efficiency. The sum of squared gradients in (11) has a similar form as the Generalised Gauss-Newton (GGN) approximation for a Hessian (Schraudolph, 2002; Graves, 2011; Martens, 2014). Motivated from the *gradient magnitude* approximation of Hessian/GGN (Bottou et al., 2018; Khan et al., 2018), we replace the sum of gradient squares with the square of the batch gradient sum,

$$\hat{F}(\theta)=\mathrm{Diag}\bigg(\frac{1}{|B|}\sum_{i\in B}\nabla\log p(y_i|x_i,\theta)\bigg)^2. \tag{12}$$

Note that (12) only requires the gradient of the batch sum of the logits (prediction scores), a very common form efficiently done by the off-the-shelf auto-differentiation libraries. If we adopt the negative log-loss (cross-entropy), it further reduces to $\hat{F}(\theta)=\mathrm{Diag}(\nabla l_B(\theta))^2$ where $l_B(\theta)$ is the minibatch estimate of $l(\theta)$. For the inverse of the Fisher information in (10), we add a small positive regulariser to the diagonal elements before taking the reciprocal.

Although this gradient magnitude approximation can introduce unwanted bias to the original $F(\theta)$ (the amount of bias being dependent on the degree of cross correlation between $\nabla\log p(y_i|x_i,\theta)$ terms), it is a widely adopted technique for learning rate scheduling also known as average squared gradients in modern optimisers such as RMSprop, Adam, and AdaGrad. Furthermore, the following theorem from (Khan et al., 2018) justifies the gradient magnitude approximation by relating the squared sum of vectors and the sum of squared vectors.

Theorem 3.1 (Rephrased from Theorem 1 (Khan et al., 2018))**.** *Let $\{v_1,\ldots,v_N\}$ be the population vectors, and $B\subset\{1\ldots N\}$ be a uniformly sampled (w/ replacement) minibatch with $M=|B|$. Denoting the minibatch and population averages by $\overline{v}(B)=\frac{1}{M}\sum_{i\in B}v_i$ and $\overline{v}=$*

Algorithm 1 Fisher SAM.

Input: Training set $S = \{(x_i, y_i)\}$, neighborhood size γ, regulariser η for inverse Fisher, and learning rate α.
for $t = 1, 2, \dots$ **do**
1) Sample a batch $B \sim S$.
2) Compute the gradient of the batch loss $\nabla l_B(\theta)$.
3) Compute the approximate Fisher info $\hat{F}(\theta)$ as per (12).
4) Compute $\epsilon^*_{FSAM}(\theta)$ using (10).
5) Compute gradient approximation for the Fisher SAM loss, $\nabla l^\gamma_{FSAM}(\theta) = \frac{\partial l_B(\theta)}{\partial \theta}\Big|_{\theta+\epsilon^*_{FSAM}(\theta)}$.
6) Update: $\theta \leftarrow \theta - \alpha \nabla l^\gamma_{FSAM}(\theta)$.
end for

$\frac{1}{N}\sum_{i=1}^N v_i$, *respectively, for some constant* α,

$$\frac{1}{N}\sum_{i=1}^{N} v_i v_i^\top = \alpha \mathbb{E}_B[\overline{v}(B)\overline{v}(B)^\top] + (1-\alpha)\overline{v}\overline{v}^\top. \quad (13)$$

Although it is proved in (Khan et al., 2018), we provide full proof here for self-containment.

Proof. We denote by $\mathbb{V}_i(v_i)$ and $\mathbb{V}_B(\cdot)$ the population variance and variance over B, respectively. Let A be the LHS of (13). Then $\mathbb{V}_i(v_i) = A - \overline{v}\overline{v}^\top$. Also $\mathbb{V}_B(\overline{v}(B)) = \mathbb{E}_B[\overline{v}(B)\overline{v}(B)^\top] - \overline{v}\overline{v}^\top$ since $\mathbb{E}_B[\overline{v}(B)] = \overline{v}$. From Theorem 2.2 of (Cochran, 1977), $\mathbb{V}_B(\overline{v}(B)) = \frac{N-M}{M(N-1)}\mathbb{V}_i(v_i)$. By arranging the terms, with $\alpha = \frac{M(N-1)}{N-M}$, we have $A = \alpha \mathbb{E}_B[\overline{v}(B)\overline{v}(B)^\top] + (1-\alpha)\overline{v}\overline{v}^\top$. □

The theorem essentially implies that the sum of squared gradients (LHS of (13)) gets close to the squared sum of gradients ($\overline{v}(B)\overline{v}(B)^\top$ or $\overline{v}\overline{v}^\top$) if the batch estimate $\overline{v}(B)$ is close enough to its population version $\overline{v}$ [2].

The Fisher SAM algorithm[3] is summarized in Alg. 1. Now we state our main theorem for generalisation bound of Fisher SAM. Specifically we bound the expectation of the generalisation loss over the Gaussian perturbation that aligns with the Fisher information geometry.

Theorem 3.2 (Generalisation bound of Fisher SAM)**.** *Let $\Theta \subseteq \mathbb{R}^k$ be the model parameter space that satisfies the regularity conditions in Appendix A. For any $\theta \in \Theta$, with probability at least $1-\delta$ over the choice of the training set S ($|S| = n$), the following holds.*

$$\mathbb{E}_\epsilon[l_D(\theta+\epsilon)] \le l^\gamma_{FSAM}(\theta; S) + \sqrt{\frac{O(k + \log\frac{n}{\delta})}{n-1}}, \quad (14)$$

where $l_D(\cdot)$ is the generalisation loss, $l^\gamma_{FSAM}(\cdot; S)$ is the empirical Fisher SAM loss as in (8), and the expectation is over $\epsilon \sim \mathcal{N}(0, \rho^2 F(\theta)^{-1})$ for $\rho \propto \gamma$.

Remark 3.3. Compared to SAM's generalisation bound in Appendix A.1 of (Foret et al., 2021), the complexity term is asymptotically identical (only some constants are different). However, the expected generalisation loss in the LHS of (14) is different: we have perturbation of θ aligned with the Fisher geometry of the model parameter space (i.e., $\epsilon \sim \mathcal{N}(0, \rho^2 F(\theta)^{-1})$), while in SAM they bound the generalisation loss averaged over *spherical* Gaussian perturbation, $\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\rho^2 I)}[l_D(\theta+\epsilon)]$. The latter might be an inaccurate measure for the average loss since the perturbation does not conform to the geometry of the underlying manifold.

The full proof is provided in Appendix A, and we describe the proof sketch here.

Proof (sketch). Our proof is an extension of (Foret et al., 2021)'s proof, in which the PAC-Bayes bound (McAllester, 1999) is considered for a pre-defined set of prior distributions, among which the one closest to the posterior is chosen to tighten the bound. In (Foret et al., 2021), the posterior is a spherical Gaussian (corresponding to a Euclidean ball) with the variance being *independent* of the current model θ. Then the prior set can be pre-defined as a series of spherical Gaussians with increasing variances so that there always exists a member in the prior set that matches the posterior by only small error. In our case, however, the posterior is a Gaussian with Fisher-induced ellipsoidal covariance, thus covariance being *dependent* on the current θ. This implies that the prior set needs to be pre-defined more sophisticatedly to adapt to a not-yet-seen posterior. Our key idea is to partition the model parameter space Θ into many small Fisher ellipsoids $R_j \triangleq \{\theta \mid (\theta - \overline{\theta_j})^\top F(\overline{\theta_j})(\theta - \overline{\theta_j}) \le r_j^2\}, j = 1, \dots, J$, for some fixed points $\{\overline{\theta_j}\}$, and we define the priors to be aligned with these ellipsoids. Then it can be shown that under some regularity conditions, any Fisher-induced ellipsoidal covariance of the posterior can match one of the R_j's with small error, thus tightening the bound. □

3.1. Fisher SAM Illustration: Toy 2D Experiments

We devise a synthetic setup with 2D parameter space to illustrate the merits of the proposed FSAM against previous SAM and ASAM. The model we consider is a univariate Gaussian, $p(x;\theta) = \mathcal{N}(x; \mu, \sigma^2)$ where $\theta = (\mu, \sigma) \in \mathbb{R} \times \mathbb{R}_+ \subset \mathbb{R}^2$. For the loss function, we aim to build a one with two local minima, one with sharp curvature, the other flat. We further confine the loss to be a function of the model likelihood $p(x;\theta)$ so that the the parameter space becomes a manifold with the Fisher information metric. To this end, we define the loss function as a negative log-mixture of two

[2] For instance, the two terms in the RHS of (13) can be approximately merged into a single squared sum of gradients.

[3] In the current version, we take the vanilla gradient update (step 6 in Alg. 1). However, it is possible to take the natural gradient update instead (by pre-multiplying the update vector by the inverse Fisher information), which can be beneficial for other methods SGD and SAM, likewise. Nevertheless, we leave it and related further extensive study as future work.

KL-driven energy models. More specifically,

$$l(\theta) = -\log\left(\alpha_1 e^{-E_1(\theta)/\beta_1^2} + \alpha_2 e^{-E_2(\theta)/\beta_2^2}\right), \quad (15)$$
$$\text{where } E_i(\theta) = \mathrm{KL}\big(p(x;\theta)||N(x;m_i,s_i^2)\big),\ i=1,2.$$

We set constants as: $(m_1, s_1, \alpha_1, \beta_1) = (20, 30, 0.7, 1.8)$ and $(m_2, s_2, \alpha_2, \beta_2) = (-20, 10, 0.3, 1.2)$. Since β_i determines the component scale, we can guess that the flat minimum is at around (m_1, s_1) (larger β_1), and the sharp one at around (m_2, s_2) (smaller β_2). The contour map of $l(\theta)$ is depicted in Fig. 1, where the two minima numerically found are: $\theta^{flat} = (19.85, 29.95)$ and $\theta^{sharp} = (-15.94, 13.46)$ as marked in the figure. We prefer the flat minimum (marked as star/blue) to the sharp one (dot/magenta) even though θ^{sharp} attains slightly lower loss.

Comparing the neighborhood structures at the current iterate (μ, σ), SAM has a circle, $\{(\epsilon_1, \epsilon_2) \mid \epsilon_1^2 + \epsilon_2^2 \le \gamma^2\}$, whereas FSAM has an ellipse, $\{(\epsilon_1, \epsilon_2) \mid \epsilon_1^2/\sigma^2 + \epsilon_2^2/(\sigma^2/2) \le \gamma^2\}$ since the Fisher information for Gaussian is $F(\mu, \sigma) = \mathrm{Diag}(1/\sigma^2, 2/\sigma^2)$. Note that the latter is the intrinsic metric for the underlying parameter manifold. Thus when σ is large (away from 0), it is a valid strategy to explore more aggressively to probe the worst-case loss in both axes (as FSAM does). On the other hand, SAM is unable to adapt to the current iterate and probes relatively too little, which hinders finding a sensible robust loss function. This scenario is illustrated in Fig. 2. The initial iterate (diamond/green) has a large σ value, and FSAM makes aggressive exploration in both axes, helping us move toward the flat minimum. However, SAM makes too narrow exploration, merely converging to the relatively nearby sharp minimum. For ASAM, the neighborhood at current iterate (μ, σ) is the magnitude-scaled ellipse, $\{(\epsilon_1, \epsilon_2) \mid \epsilon_1^2/\mu^2 + \epsilon_2^2/\sigma^2 \le \gamma^2\}$. Thus when μ is close to 0, for instance, ϵ_1 is not allowed to perturb much, hindering effective exploration of the parameter space toward robustness, as illustrated in Fig. 3.

4. Related Work

Interest in flat minima for neural networks dates back to at least (Hochreiter & Schmidhuber, 1995; 1997), who characterise flatness as the size of the region around which the training loss remains low. Many studies have since investigated the link between flat minima and generalisation performance (Keskar et al., 2017; Neyshabur et al., 2017; Dziugaite & Roy, 2017; Dinh et al., 2017). In particular, the correlation between sharpness and generalisation performance was studied with diverse measures empirically on large-scale experiments: (Jiang et al., 2020). Beyond the IID setting, (Cha et al., 2021) analyse the impact of flat minima on generalisation performance under domain-shift.

Motivated by these analyses, several methods have been proposed to visualise and optimise for flat minima, with the aim of improving generalisation. (Li et al., 2018) developed methods for visualising loss surfaces to inspect minima shape. (Zhu et al., 2019) analyse how the noise in SGD promotes preferentially discovering flat minima over sharp minima, as a potential explanation for SGD's strong generalisation. Entropy-SGD (Chaudhar et al., 2017) biases gradient descent toward flat minima by regularising local entropy. Stochastic Weight Averaging (SWA) (Izmailov et al., 2018) was proposed as an approach to find flatter minima by parameter-space averaging of an ensemble of models' weights. The state-of-the-art SAM (Foret et al., 2021) and ASAM (Kwon et al., 2021) find flat minima by reporting the worst-case loss within a ball around the current iterate. Our Fisher SAM builds upon these by providing a principled approach to defining a non-Euclidean ball within which to compute the worst-case loss.

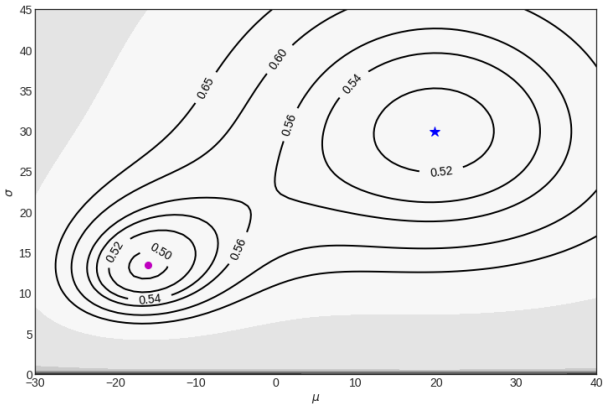

Figure 1. (Toy experiment) Contour plot of the loss function. The flat minimum is shown as star/blue $\theta^{flat} = (19.85, 29.95)$ ($l = 0.51$, $H = 0.001$), and the sharp one as circle/magenta $\theta^{sharp} = (-15.94, 13.46)$ ($l = 0.49$, $H = 0.006$), with their loss values and Hessian traces shown in parentheses.

5. Experiments

We empirically demonstrate generalisation performance (Sec. 5.1–5.3) and noise robustness (Sec. 5.4, 5.5) of the proposed Fisher SAM method. As competing approaches, we consider the vanilla (non-robust) optimisation (**SGD**) as a baseline, and two latest SAM approaches: **SAM** (Foret et al., 2021) that uses Euclidean-ball neighborhood and **ASAM** (Kwon et al., 2021) that employs parameter-scaled neighborhood. Our approach, forming Fisher-driven neighborhood, is denoted by **FSAM**.

For the implementation of Fisher SAM in the experiments, instead of simply adding a regulariser to each diagonal entry f_i of the Fisher information matrix $\hat{F}(\theta)$, we take $1/(1 + \eta f_i)$ as the diagonal entry of the inverse Fisher. Hence η serves as anti-regulariser (e.g., small η diminishes or regularises the Fisher impact). We find this implementation performs better than simply adding a regulariser. In most of our experiments, we set $\eta = 1.0$. Certain

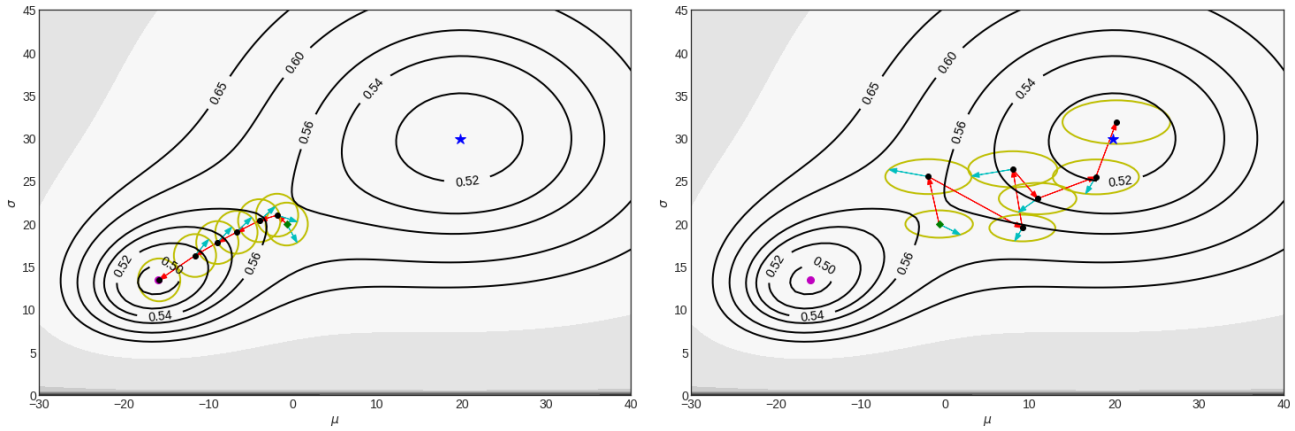

Figure 2. (Toy experiment) **SAM vs. FSAM**. X-axis is μ and Y-axis is σ. (**Left**=**SAM**) SAM failed due to the inaccurate neighborhood structure of Euclidean ball. (**Right**=**FSAM**) FSAM finds the flat minimum due to the accurate neighborhood structure from Fisher information metric. Initial iterate shown as diamond/green; the neighborhood ball is depicted as yellow circle/ellipse; the worst-case probe within the neighborhood is indicated by cyan arrow, update direction is shown as red arrow. The sizes of circles/ellipses are adjusted for better visualisation.

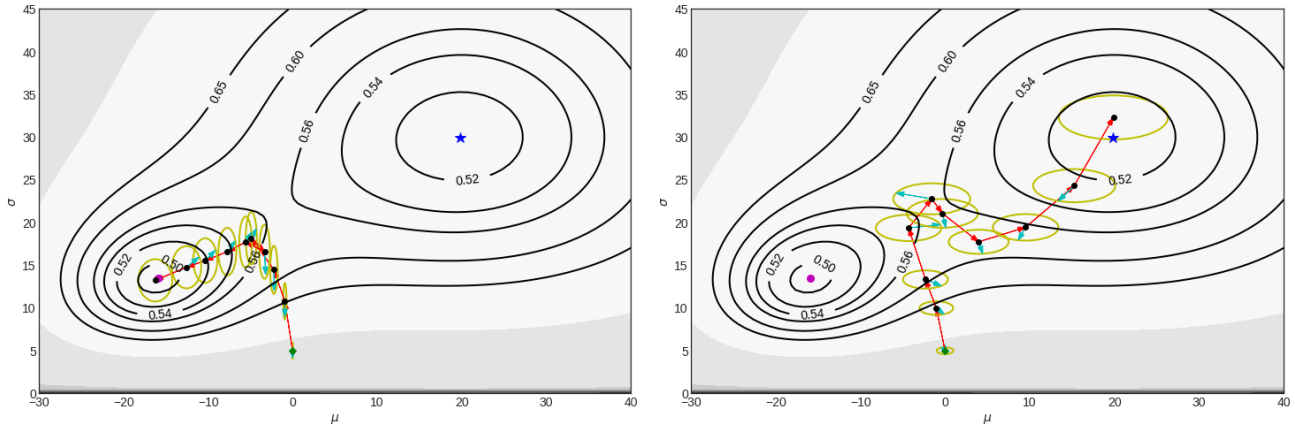

Figure 3. (Toy experiment) **ASAM vs. FSAM**. X-axis is μ and Y-axis is σ. (**Left**=**ASAM**) ASAM failed due to the inaccurate neighborhood structure. Especially when the magnitude of a particular parameter value, in this case μ, is small (close to 0), it overly penalises perturbation along the axis. The parameter μ has small magnitude initially, which forms an incorrect neighborhood ellipse overly shrunk along the X-axis, preventing it from finding an effective worst-case probe through X-axis perturbation. (**Right**=**FSAM**) FSAM finds the flat minimum due to the accurate neighborhood structure from Fisher information metric. Initial iterate shown as diamond/green; the neighborhood ball is depicted as yellow circle/ellipse; the worst-case probe within the neighborhood is indicated by cyan arrow, update direction is shown as red arrow.

Table 1. Test accuracies on CIFAR-10 and CIFAR-100.

	CIFAR-10				CIFAR-100			
	SGD	SAM	ASAM	FSAM	SGD	SAM	ASAM	FSAM
DenseNet-121	$91.83^{\pm 0.13}$	$92.44^{\pm 0.28}$	$92.70^{\pm 0.30}$	$\mathbf{92.81}^{\pm 0.17}$	$71.26^{\pm 0.15}$	$72.83^{\pm 0.01}$	$73.10^{\pm 0.23}$	$\mathbf{73.15}^{\pm 0.33}$
ResNet-20	$92.91^{\pm 0.13}$	$92.99^{\pm 0.16}$	$92.92^{\pm 0.15}$	$\mathbf{93.18}^{\pm 0.11}$	$68.24^{\pm 0.34}$	$68.61^{\pm 0.26}$	$68.68^{\pm 0.11}$	$\mathbf{69.04}^{\pm 0.30}$
ResNet-56	$95.37^{\pm 0.06}$	$95.59^{\pm 0.14}$	$95.63^{\pm 0.07}$	$\mathbf{95.71}^{\pm 0.08}$	$75.52^{\pm 0.27}$	$76.44^{\pm 0.26}$	$76.32^{\pm 0.14}$	$\mathbf{76.86}^{\pm 0.16}$
VGG-19-BN	$95.70^{\pm 0.09}$	$96.11^{\pm 0.09}$	$95.97^{\pm 0.10}$	$\mathbf{96.17}^{\pm 0.07}$	$73.45^{\pm 0.32}$	$77.25^{\pm 0.24}$	$74.36^{\pm 0.19}$	$\mathbf{77.86}^{\pm 0.22}$
ResNeXt-29-32x4d	$96.55^{\pm 0.15}$	$97.27^{\pm 0.10}$	$97.29^{\pm 0.06}$	$\mathbf{97.47}^{\pm 0.05}$	$79.36^{\pm 0.19}$	$82.63^{\pm 0.16}$	$82.41^{\pm 0.31}$	$\mathbf{82.92}^{\pm 0.15}$
WRN-28-2	$95.56^{\pm 0.22}$	$96.28^{\pm 0.14}$	$96.25^{\pm 0.07}$	$\mathbf{96.51}^{\pm 0.08}$	$78.85^{\pm 0.25}$	$79.87^{\pm 0.13}$	$80.17^{\pm 0.14}$	$\mathbf{80.22}^{\pm 0.26}$
WRN-28-10	$97.12^{\pm 0.10}$	$97.56^{\pm 0.06}$	$97.63^{\pm 0.04}$	$\mathbf{97.89}^{\pm 0.07}$	$83.47^{\pm 0.21}$	$\mathbf{85.60}^{\pm 0.05}$	$85.20^{\pm 0.18}$	$\mathbf{85.60}^{\pm 0.11}$
PyramidNet-272	$97.73^{\pm 0.04}$	$97.91^{\pm 0.02}$	$97.91^{\pm 0.01}$	$\mathbf{97.93}^{\pm 0.04}$	$83.46^{\pm 0.02}$	$85.19^{\pm 0.04}$	$85.05^{\pm 0.11}$	$\mathbf{86.93}^{\pm 0.14}$

multi-GPU/TPU gradient averaging heuristics called the m-sharpness trick empirically improves the generalisation performance of SAM and ASAM (Foret et al., 2021). However, since the trick is theoretically less justified, we do not use the trick in our experiments for fair comparison.

5.1. Image Classification

The goal of this section is to empirically compare generalisation performance of the competing algorithms: SGD, SAM, ASAM, and our FSAM on image classification problems. Following the experimental setups suggested in (Foret et al., 2021; Kwon et al., 2021), we employ several ResNet (He et al., 2016)-based backbone networks including WideResNet (Zagoruyko & Komodakis, 2016), VGG (Simonyan & Zisserman, 2015), DenseNet (Huang et al., 2017), ResNeXt (Xie et al., 2017), and PyramidNet (Han et al., 2017), on the CIFAR-10/100 datasets (Krizhevsky, 2009). Similar to (Foret et al., 2021; Kwon et al., 2021), we use the SGD optimiser with momentum 0.9, weight decay 0.0005, initial learning rate 0.1, cosine learning rate scheduling (Loshchilov & Hutter, 2016), for up to 200 epochs (400 for SGD) with batch size 128. For the PyramidNet, we use batch size 256, initial learning rate 0.05 trained up to 900 epochs (1800 for SGD). We also apply Autoaugment (Cubuk et al., 2019), Cutout (DeVries & Taylor, 2017) data augmentation, and the label smoothing (Müller et al., 2019) with factor 0.1 is used for defining the loss function.

We perform the grid search to find best hyperparameters (γ, η) for FSAM, and they are $(\gamma = 0.1, \eta = 1.0)$ for both CIFAR-10 and CIFAR-100 across all backbones except for PyramidNet. For the PyramidNet on CIFAR-100, we set $(\gamma = 0.5, \eta = 0.1)$. For SAM and ASAM, we follow the best hyperparameters reported in their papers: (SAM) $\gamma = 0.05$ and (ASAM) $\gamma = 0.5, \eta = 0.01$ for CIFAR-10 and (SAM) $\gamma = 0.1$ and (ASAM) $\gamma = 1.0, \eta = 0.1$ for CIFAR-100. For the PyramidNet, (SAM) $\gamma = 0.05$ and (ASAM) $\gamma = 1.0$. The results are summarized in Table 1, where Fisher SAM consistently outperforms SGD and previous SAM approaches for all backbones. This can be attributed to FSAM's correct neighborhood estimation that respects the underlying geometry of the parameter space.

5.2. Extra (Over-)Training on ImageNet

For a large-scale experiment, we consider the ImageNet dataset (Deng et al., 2009). We use the DeiT-base (denoted by DeiT-B) vision transformer model (Touvron et al., 2021) as a powerful backbone network. Instead of training the DeiT-B model from the scratch, we use the publicly available[4] ImageNet pre-trained parameters as a warm start, and perform finetuning with the competing loss functions. Since the same dataset is used for pre-training and finetuning, it can be better termed extra/over-training.

[4] https://github.com/facebookresearch/deit

Table 2. Extra-training results (test accuracies) on ImageNet. Before extra-training starts, 81.94% (Top-1) and 95.63% (Top-5).

	SGD	SAM	ASAM	FSAM
Top-1	$81.97^{\pm 0.01}$	$81.99^{\pm 0.01}$	$82.03^{\pm 0.04}$	$\mathbf{82.17}^{\pm 0.01}$
Top-5	$95.61^{\pm 0.01}$	$95.71^{\pm 0.03}$	$95.83^{\pm 0.04}$	$\mathbf{95.90}^{\pm 0.01}$

Table 3. Test accuracies for transfer learning.

	SGD	SAM	ASAM	FSAM
CIFAR	$87.97^{\pm 0.12}$	$87.99^{\pm 0.09}$	$87.97^{\pm 0.08}$	$\mathbf{88.39}^{\pm 0.13}$
Cars	$92.85^{\pm 0.31}$	$93.29^{\pm 0.01}$	$93.28^{\pm 0.02}$	$\mathbf{93.42}^{\pm 0.01}$
Flowers	$94.53^{\pm 0.20}$	$95.05^{\pm 0.06}$	$95.08^{\pm 0.10}$	$\mathbf{95.26}^{\pm 0.03}$

The main goal of this experimental setup is to see if robust sharpness-aware loss functions in the extra training stage can further improve the test performance. First, we measure the test performance of the pre-trained DeiT-B model, which is 81.94% (Top-1) and 95.63% (Top-5). After three epochs of extra training, the test accuracies of the competing approaches are summarized in Table 2. For extra training, we use hyperparameters: SAM ($\gamma = 0.05$), ASAM ($\gamma = 1.0, \eta = 0.01$), and FSAM ($\gamma = 0.5, \eta = 0.1$) using the SGD optimiser with batch size 256, weight decay 0.0001, initial learning rate 10^{-5} and the cosine scheduling. Although the improvements are not very drastic, the sharpness-aware loss functions appear to move the pre-trained model further toward points that yield better generalisation performance, and our FSAM attains the largest improvement among other SAM approaches.

5.3. Transfer Learning by Finetuning

One of the powerful features of the deep neural network models trained on extremely large datasets, is the transferability, that is, the models tend to adapt easily and quickly to novel target datasets and/or downstream tasks by finetuning. We use the vision transformer model ViT-base with 16×16 patches (ViT-B/16) (Dosovitskiy et al., 2021) pre-trained on the ImageNet (with publicly available checkpoints), and finetune the model on the target datasets: CIFAR-100, Stanford Cars (Krause et al., 2013), and Flowers (Nilsback & Zisserman, 2008). We run SGD, SAM ($\gamma = 0.05$), ASAM ($\gamma = 0.1, \eta = 0.01$), and FSAM ($\gamma = 0.1, \eta = 1.0$) with the SGD optimiser for 200 epochs, batch size 256, weight decay 0.05, initial learning rate 0.0005 and the cosine scheduling. As the results in Table 3 indicate, FSAM consistently improves the performance over the competing methods.

5.4. Robustness to Adversarial Parameter Perturbation

Another important benefit of the proposed approach is robustness to parameter perturbation. In the literature, the

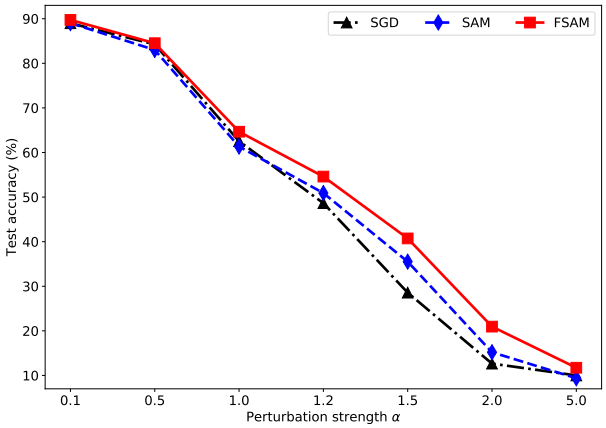

Figure 4. Adversarial parameter perturbation.

generalisation performance of the corrupted models is measured by injecting artificial noise to the learned parameters, which serves as a measure of vulnerability of neural networks (Chen et al., 2017; Arora et al., 2018; Dai et al., 2019; Gu et al., 2019; Nagel et al., 2019; Rakin & Fan, 2020; Weng et al., 2020). Although it is popular to add Gaussian random noise to the parameters, recently the *adversarial* perturbation (Sun et al., 2021) was proposed where they consider the worst-case scenario under parameter corruption, which amounts to perturbation along the gradient direction. More specifically, the perturbation process is: $\theta \to \theta + \alpha \frac{\nabla l(\theta)}{\|\nabla l(\theta)\|}$ where $\alpha > 0$ is the perturbation strength that can be chosen. It turns out to be a more effective perturbation measure than the random (Gaussian noise) corruption.

We apply this adversarial parameter perturbation process to ResNet-34 models trained by SGD, SAM ($\gamma = 0.05$), and FSAM ($\gamma = 0.1, \eta = 1.0$) on CIFAR-10, where we vary the perturbation strength α from 0.1 to 5.0. The results are depicted in Fig. 4. While we see performance drop for all models as α increases, eventually reaching nonsensical models (pure random prediction accuracy 10%) after $\alpha \geq 5.0$, the proposed Fisher SAM exhibits the least performance degradation among the competing methods, proving the highest robustness to the adversarial parameter corruption.

5.5. Robustness to Label Noise

In the previous works, SAM and ASAM are shown to be robust to label noise in training data. Similarly as their experiments, we introduce symmetric label noise by random flipping with corruption levels 20, 40, 60, and 80%, as introduced in (Rooyen et al., 2015). The results on ResNet-32 networks on the CIFAR-10 dataset are summarized in Table 4. Our Fisher SAM shows high robustness to label noise comparable to SAM while exhibiting significantly large improvements over SGD and ASAM.

Table 4. Label noise. Test accuracies on CIFAR-10. The hyperparameters are: (SAM) $\gamma = 0.1$, (ASAM) $\gamma = 0.5, \eta = 0.01$, and (FSAM) $\gamma = 0.1, \eta = 0.001$.

Rates	SGD	SAM	ASAM	FSAM
0.2	$87.97^{\pm 0.04}$	$\mathbf{93.12}^{\pm 0.24}$	$92.26^{\pm 0.33}$	$93.03^{\pm 0.11}$
0.4	$83.60^{\pm 0.59}$	$90.54^{\pm 0.19}$	$88.47^{\pm 0.06}$	$\mathbf{90.95}^{\pm 0.17}$
0.6	$76.97^{\pm 0.31}$	$85.39^{\pm 0.52}$	$82.32^{\pm 0.55}$	$\mathbf{85.76}^{\pm 0.21}$
0.8	$66.32^{\pm 0.27}$	$74.31^{\pm 1.02}$	$70.56^{\pm 0.27}$	$\mathbf{74.66}^{\pm 0.67}$

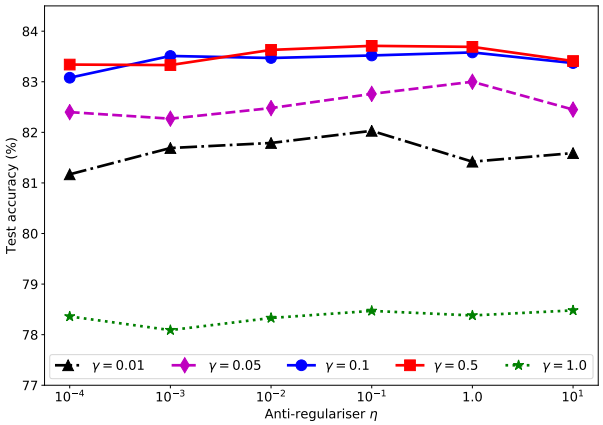

Figure 5. Hyperparameter sensitivity of Fisher SAM.

5.6. Hyperparameter Sensitivity

In our Fisher SAM, there are two hyperparameters: $\gamma =$ the size of the neighborhood and $\eta =$ the anti-regulariser for the Fisher impact. We demonstrate the sensitivity of Fisher SAM to these hyperparameters. To this end, we train WRN-28-10 backbone models trained with the FSAM loss on the CIFAR-100 dataset for different hyperparameter combinations: $(\gamma, \eta) \in \{0.01, 0.05, 0.1, 0.5, 1.0\} \times \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1.0, 10\}$. In Fig. 5 we plot the test accuracy of the learned models[5]. The results show that unless γ is chosen too large (e.g., $\gamma = 1.0$), the learned models all perform favorably well, being less sensitive to the hyperparameter choice. But the best performance is attained when γ lies in between 0.1 and 0.5, with some moderate values for the Fisher impact η in between 0.01 and 1.0.

5.7. Computational Overhead of FSAM

Compared to SAM, our FSAM requires only extra cost of element-wise vector product under our diagonal gradient-magnitude approximation schemes. In practice, the difference is negligible: the per-batch (batch size 128) times for CIFAR10/WRN28-10 are: 0.2322 seconds (SAM), 0.2334 seconds (FSAM) on a single RTX 2080 Ti machine.

[5]Note that there are discrepancies from Table 1 that may arise from the lack of data augmentation.

6. Conclusion

In this paper we have proposed a novel sharpness aware loss function that incorporates the information geometry of the underlying parameter manifold, which defines a more accurate intrinsic neighborhood structure, addressing the issues of the previous flat-minima optimisation methods. The proposed algorithm remains computationally efficient via a theoretically justified gradient magnitude approximation for the Fisher information matrix. By proving the theoretical generalisation bound, and through diverse experiments on image classification, extra-training/finetuning, data corruption, and model perturbation, we demonstrated improved generalisation performance and robustness of the proposed Fisher SAM. Several research questions remain: natural gradient updates combined with the Fisher SAM loss, investigation of distributed update averaging schemes (m-sharpness), and discovering the relation to the proximal gradients, among others, and we leave them as future work.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Amari, S. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. Stronger Generalization Bounds for Deep Nets via a Compression Approach. In *International Conference on Machine Learning*, 2018.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge: Cambridge University Press, 2004.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Cha, J., Chun, S., Lee, K., Cho, H.-C., Park, S., Lee, Y., and Park, S. Swad: Domain generalization by seeking flat minima. *NeurIPS*, 2021.

Chatterji, N. S., Neyshabur, B., and Sedghi, H. The intriguing role of module criticality in the generalization of deep networks. In *International Conference on Learning Representations*, 2020.

Chaudhar, P., Choromansk, A., Soatt, S., LeCun, Y., Baldass, C., Borg, C., Chays, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. In *ICLR*, 2017.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. In *arXiv preprint arXiv:1712.05526*, 2017.

Cochran, W. G. *Sampling Techniques*. Wiley, Palo Alto, CA, 1977.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Dai, J., Chen, C., and Li, Y. A Backdoor Attack Against LSTM-Based Text Classification Systems. In *IEEE Access 7*, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, 2017.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *UAI*, 2017.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Graves, A. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, 2011.

Gu, T., Liu, K., Dolan-Gavitt, B., and Garg, S. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. In *IEEE Access 7*, 2019.

Han, D., Kim, J., and Kim, J. Deep pyramidal residual networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Hochreiter, S. and Schmidhuber, J. Simplifying neural nets by discovering flat minima. In *Advances in Neural Information Processing Systems*, 1995.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *CVPR*, 2017.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *ICLR*, 2020.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017.

Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. Fast and Scalable Bayesian Deep Learning by Weight-Perturbation in Adam. In *International Conference on Machine Learning*, 2018.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

Krizhevsky, A. Learning multiple layers of features from tiny images. In *Tech. Report*, 2009.

Kwon, J., Kim, J., Park, H., and Choi, I. K. ASAM: Adaptive Sharpness-Aware Minimization for Scale-Invariant Learning of Deep Neural Networks. In *International Conference on Machine Learning*, 2021.

Langford, J. and Caruana, R. (Not) Bounding the true error. In *Advances in Neural Information Processing Systems*, 2002.

Laurent, B. and Massart, P. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, 28(5):1302–1338, 2000.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In *NIPS*. 2018.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *arXiv preprint arXiv:1608.03983*, 2016.

Martens, J. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

McAllester, D. A. PAC-Bayesian model averaging. In *In Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, 1999.

Müller, R., Kornblith, S., and Hinton, G. E. When does label smoothing help? In *Advances in Neural Information Processing Systems*, 2019.

Murray, M. K. and Rice, J. W. Differential geometry and statistics. *Monographs on Statistics and Applied Probability*, (48), 1993.

Nagel, M., van Baalen, M., Blankevoort, T., and Welling, M. Data-Free Quantization Through Weight Equalization and Bias Correction. In *IEEE/CVF International Conference on Computer Vision*, 2019.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, 2017.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. 2019.

Rakin, A. S.; He, Z. and Fan, D. TBT: Targeted Neural Network Attack With Bit Trojan. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Rooyen, B. v., Menon, A. K., and Williamson, R. C. Learning with symmetric label noise: the importance of being unhinged. In *Advances in Neural Information Processing Systems*, 2015.

Schraudolph, N. N. Fast curvature matrix-vector products for second-order gradient descent. *Neural computation*, 14(7):1723–1738, 2002.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014.

Sun, X., Zhang, Z., Ren, X., Luo, R., and Li, L. Exploring the Vulnerability of Deep Neural Networks: A Study of Parameter Corruption. In *Proc. of AAAI Conference on Artificial Intelligence*, 2021.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jegou, H. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 2021.

Weng, T., Zhao, P., Liu, S., Chen, P., Lin, X., and Daniel, L. Towards Certificated Model Robustness Against Weight Perturbations. In *AAAI Conference on Artificial Intelligence*, 2020.

Xie, S.and Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *BMVC*, 2016.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Zhu, Z., Wu, J., Yu, B., Wu, L., and Ma, J. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In *International Conference on Machine Learning*, 2019.

A. Theorem 3.2 and Proof

We first state several regularity conditions and assumptions under which the theorem can be proved formally.

Assumption A.1. Let k be the dimensionality of the model parameters θ. We consider the following J ellipsoids centered at some fixed points $\overline{\theta_j} \in \mathbb{R}^k$ with elliptic axes determined by the Fisher information $F(\overline{\theta_j})$ and sizes r_j:

$$R_j \triangleq \{\theta \in \mathbb{R}^k \mid (\theta - \overline{\theta_j})^\top F(\overline{\theta_j})(\theta - \overline{\theta_j}) \le r_j^2\}, \quad j = 1, \dots, J. \tag{16}$$

We choose $\overline{\theta_j}$ properly such that $F(\overline{\theta_j})$ are strictly positive[6] (all eigenvalues greater than some constant $\lambda_{min} > 0$). Our model parameter space Θ is assumed to be contained in these ellipsoids, i.e., $\Theta \subseteq \cup_{j=1}^J R_j$. We also assume Θ has a bounded diameter B, that is, $B \ge \mathrm{diam}(\Theta) = \max_{\theta,\theta' \in \Theta} \|\theta - \theta'\|$ (thus $\|\theta\| \le B$). Note that since $\mathrm{vol}(R_j) \propto r_j^k \cdot |F(\overline{\theta_j})|^{-1/2}$, we have $J = O(\max_j \mathrm{diam}(\Theta)^k / r_j^k)$, and thus $\log J = O(k)$.

The following two assumptions are regularity conditions regarding smoothness of the Fisher information matrix $F(\theta)$ as a function of θ, that are assumed to hold for θ in each ellipsoid R_j. Intuitively these conditions can be met by adjusting r_j sufficiently small, but specific conditions are provided below.

Assumption A.2. Let $\lambda_i(A)$ be the i-th largest eigenvalue of the (positive definite) matrix A. Then for $j = 1, \dots, J$,

$$\frac{\lambda_i(F(\theta))}{\lambda_i(F(\overline{\theta_j}))} = 1 + c_{ij}, \;\; c_{ij} \in [-\epsilon_{min}, \epsilon_{min}], \;\; \forall \theta \in R_j, \; \forall i = 1, \dots, k, \tag{17}$$

where ϵ_{min} is a small positive constant. For instance, if $\lambda_i(\theta) \triangleq \lambda_i(F(\theta))$ is Lipschitz continuous with constant C_1, that is,

$$|\lambda_i(\theta) - \lambda_i(\overline{\theta_j})| \le C_1 \|\theta - \overline{\theta_j}\|_{F(\overline{\theta_j})}, \;\; \forall \theta \in R_j \tag{18}$$

where $\|x\|_A = (x^\top A x)^{1/2}$, then (17) holds by adjusting r_j properly. More specifically,

$$\left| \frac{\lambda_i(\theta)}{\lambda_i(\overline{\theta_j})} - 1 \right| = \lambda_i(\overline{\theta_j})^{-1} |\lambda_i(\theta) - \lambda_i(\overline{\theta_j})| \le \lambda_i(\overline{\theta_j})^{-1} C_1 \|\theta - \overline{\theta_j}\|_{F(\overline{\theta_j})} \le \lambda_i(\overline{\theta_j})^{-1} C_1 r_j. \tag{19}$$

Hence we can choose $r_j \le \lambda_i(\overline{\theta_j}) \epsilon_{min} / C_1$ to make (17) hold.

Assumption A.3. We assume that $F(\theta)$ is non-singular, and for $j = 1, \dots, J$,

$$F(\overline{\theta_j}) F(\theta)^{-1} = I + A^j, \;\; A^j_{i,i'} \in [-\epsilon_{min}, \epsilon_{min}], \;\; \forall \theta \in R_j \tag{20}$$

For instance, if $F(\theta)^{-1}$ is Lipschitz continuous with constant C_2, that is,

$$\|F(\theta)^{-1} - F(\overline{\theta_j})^{-1}\| \le C_2 \|\theta - \overline{\theta_j}\|_{F(\overline{\theta_j})}, \;\; \forall \theta \in R_j \tag{21}$$

where the matrix norm in the RHS is the max-norm, i.e., $\|B\| = \max_{i,i'} |B_{i,i'}|$, then we have

$$\|F(\overline{\theta_j}) F(\theta)^{-1} - I\| \;=\; \|F(\overline{\theta_j})\big(F(\theta)^{-1} - F(\overline{\theta_j})^{-1}\big)\| \;\le\; \|F(\overline{\theta_j})\| \|F(\theta)^{-1} - F(\overline{\theta_j})^{-1}\| \tag{22}$$

$$\le\; \|F(\overline{\theta_j})\| C_2 \|\theta - \overline{\theta_j}\|_{F(\overline{\theta_j})} \;\le\; \|F(\overline{\theta_j})\| C_2 r_j. \tag{23}$$

Choosing $r_j \le \|F(\overline{\theta_j})\|^{-1} \epsilon_{min} / C_2$ is sufficient to satisfy (20).

Now we re-state our main theorem.

Theorem A.4 (Generalisation bound of Fisher SAM)**.** *Let $\Theta \subseteq \mathbb{R}^k$ be the model parameter space as described in the above assumptions. For any $\theta \in \Theta$, with probability at least $1 - \delta$ over the choice of the training set S ($|S| = n$),*

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho^2 F(\theta)^{-1})}[l_D(\theta + \epsilon)] \;\le\; l^\gamma_{FSAM}(\theta; S) + \sqrt{\frac{O(k + \log \frac{n}{\delta})}{n - 1}}, \tag{24}$$

where $l_D(\cdot)$ is the generalisation loss, $l^\gamma_{FSAM}(\cdot; S) = \max_{\epsilon^\top F(\theta) \epsilon \le \gamma^2} l_S(\theta + \epsilon)$ is the empirical Fisher SAM loss on S, and $\rho = (\sqrt{k} + \sqrt{\log n})^{-1} \gamma$.

[6] The strict positive definiteness of the Fisher can be assured for non-redundant parametrisation, and can be mildly assumed.

Proof. Motivated from (Foret et al., 2021), we use the PAC-Bayes theorem (McAllester, 1999) to derive the bound. According to the PAC-Bayes generalisation bound of (McAllester, 1999; Dziugaite & Roy, 2017), for any prior distribution $P(\theta)$, with probability at least $1-\delta$ over the choice of the training set S, it holds that

$$\mathbb{E}_{Q(\theta)}[l_D(\theta)] \leq \mathbb{E}_{Q(\theta)}[l_S(\theta)] + \sqrt{\frac{\mathrm{KL}(Q(\theta)||P(\theta)) + \log\frac{n}{\delta}}{2(n-1)}} \tag{25}$$

for any posterior distribution $Q(\theta)$ that may be dependent on the training data S. In (25), $l_D(\cdot)$ and $l_S(\cdot)$ are generalisation and empirical losses, respectively. We choose $Q(\theta) = \mathcal{N}(\theta_0, \rho^2 F(\theta_0)^{-1})$, a Gaussian centered at θ_0 with the covariance aligned with the Fisher information metric at θ_0. One can choose θ_0 arbitrarily from Θ, and it can be dependent on S (in which sense, a more accurate notation would be $\theta_{|S}$, however, we use θ_0 for simplicity). To minimise the bound of (25), we aim to choose the prior $P(\theta)$ that minimises the KL divergence term, which coincides with $Q(\theta)$. However, this would violate the PAC-Bayes assumption where the prior should be independent of S. The idea, inspired by the covering approach (Langford & Caruana, 2002; Chatterji et al., 2020; Foret et al., 2021), is to have a pre-defined set of (data independent) prior distributions for all of which the PAC-Bayes bounds hold, and we select the one that is closest to $Q(\theta)$ from the pre-defined set.

Specifically, we define J prior distributions $\{P_j(\theta)\}_{j=1}^J$ as $P_j(\theta) = \mathcal{N}(\overline{\theta_j}, \rho^2 F(\overline{\theta_j})^{-1})$ sharing the centers and covariances with the ellipsoids defined as above. Then applying the PAC-Bayes bound (25) for each j makes the following hold for $P_j(\theta)$ with probability at least $1-\delta_j$ over the choice of the training set S,

$$\forall Q(\theta),\ \ \mathbb{E}_{Q(\theta)}[l_D(\theta)] \leq \mathbb{E}_{Q(\theta)}[l_S(\theta)] + \sqrt{\frac{\mathrm{KL}(Q(\theta)||P_j(\theta)) + \log\frac{n}{\delta_j}}{2(n-1)}}. \tag{26}$$

By having the intersection of the training sets for which (26) holds, we can say that (26) holds for *all* $P_j(\theta)$ ($\forall j = 1, \ldots, J$) over the intersection. By the union bound theorem, the probability over the choice of the intersection is at least $1 - \sum_{j=1}^J \delta_j$. By letting $\delta_j = \frac{\delta}{J}$, we thus have the following bound (statement): For all $P_j(\theta)$ ($\forall j = 1, \ldots, J$), with probability at least $1-\delta$ over the choice of the training set S, the folowing holds:

$$\forall Q(\theta),\ \ \mathbb{E}_{Q(\theta)}[l_D(\theta)] \leq \mathbb{E}_{Q(\theta)}[l_S(\theta)] + \sqrt{\frac{\mathrm{KL}(Q(\theta)||P_j(\theta)) + \log\frac{n}{\delta} + \log J}{2(n-1)}}, \quad \forall j = 1, \ldots, J. \tag{27}$$

Now, we choose the prior $P_j(\theta)$ from the prior set that is as close to the posterior $Q(\theta)$ as possible (in KL divergence). Since

$$\mathrm{KL}(Q||P_j) = \frac{1}{2}\left(\mathrm{Tr}\big(F(\overline{\theta_j})F(\theta_0)^{-1}\big) + \frac{1}{\rho^2}(\theta_0 - \overline{\theta_j})^\top F(\overline{\theta_j})(\theta_0 - \overline{\theta_j}) + \log\frac{|F(\theta_0)|}{|F(\overline{\theta_j})|} - k\right), \tag{28}$$

if we choose j^* such that $\theta_0 \in R_{j^*}$, using $\log\frac{|F(\theta_0)|}{|F(\overline{\theta_j})|} = \sum_i \log\frac{\lambda_i(F(\theta_0))}{\lambda_i(F(\overline{\theta_j}))} \leq \sum_i \log(1+c_{ij}) \leq \sum_i c_{ij} \leq k\epsilon_{min}$ (from Assumption A.2) and $\mathrm{Tr}(F(\overline{\theta_j})F(\theta_0)^{-1}) = \mathrm{Tr}(I + A^j) \leq k + k\epsilon_{min}$ (from Assumption A.3), we have the following:

$$\mathrm{KL}(Q||P_{j^*}) \leq \frac{1}{2}\left(k + k\epsilon_{min} + \frac{r_{j^*}^2}{\rho^2} + k\epsilon_{min} - k\right) = \frac{r_{j^*}^2}{2\rho^2} + k\epsilon_{min} \leq \frac{r^2}{2\rho^2} + k\epsilon_{min}, \tag{29}$$

where $r \triangleq \max_{1\leq j\leq J} r_j$. From (27) which holds for $\forall j = 1, \ldots, J$, we take only the inequality corresponding to $j = j^*$. By slightly rephrasing $\mathbb{E}_{Q(\theta)}[f(\theta)]$ as $\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\rho^2F(\theta_0)^{-1})}[f(\theta_0+\epsilon)]$ and replacing θ_0 with θ, we have the following bound that holds with probability at least $1-\delta$:

$$\forall\theta\in\Theta,\ \ \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\rho^2F(\theta)^{-1})}[l_D(\theta+\epsilon)] \ \leq\ \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\rho^2F(\theta)^{-1})}[l_S(\theta+\epsilon)] + \sqrt{\frac{\frac{r^2}{2\rho^2} + k\epsilon_{min} + \log\frac{n}{\delta} + \log J}{2(n-1)}}. \tag{30}$$

The next step is to bound the expectation in RHS of (30) by the worst-case loss, similarly as (Foret et al., 2021). We make use of the following result from (Laurent & Massart, 2000):

$$z\sim\mathcal{N}(0,\rho^2 I) \implies \|z\|^2 \leq k\rho^2\left(1+\sqrt{\frac{\log n}{k}}\right)^2 \text{ with probability at least } 1-\frac{1}{\sqrt{n}}. \tag{31}$$

Since we have non-spherical Gaussian ϵ, we cannot directly apply (31), but need some transformation to whiten the correlation among the variables in ϵ. By letting $u = F(\theta)^{1/2}\epsilon$, we have $u \sim \mathcal{N}(0, \rho^2 I)$. Then applying (31) leads to $\|u\|^2 = \epsilon^\top F(\theta)\epsilon \leq k\rho^2(1 + \sqrt{(\log n)/k})^2$ with probability at least $1 - 1/\sqrt{n}$. We denote the rightmost term by γ^2, that is, $\gamma = \rho \cdot (\sqrt{k} + \sqrt{\log n})$. We then upper-bound $\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho^2 F(\theta)^{-1})}[l_S(\theta + \epsilon)]$ by partitioning the ϵ space into those with $\epsilon^\top F(\theta)\epsilon \leq \gamma^2$ and the rest $\epsilon^\top F(\theta)\epsilon > \gamma^2$. By taking the maximum loss for the former while choosing the loss bound (constant) l_{max} for the latter, we have:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho^2 F(\theta)^{-1})}[l_S(\theta + \epsilon)] \;\leq\; (1 - 1/\sqrt{n}) \max_{\epsilon^\top F(\theta)\epsilon \leq \gamma^2} l_S(\theta + \epsilon) + \frac{l_{max}}{\sqrt{n}} \;\leq\; \max_{\epsilon^\top F(\theta)\epsilon \leq \gamma^2} l_S(\theta + \epsilon) + \frac{l_{max}}{\sqrt{n}}. \tag{32}$$

Plugging (32) and γ into (30) yields: With probability at least $1 - \delta$, $\forall \theta \in \Theta$, the following holds

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho^2 F(\theta)^{-1})}[l_D(\theta + \epsilon)] \;\leq\; \max_{\epsilon^\top F(\theta)\epsilon \leq \gamma^2} l_S(\theta + \epsilon) + \frac{l_{max}}{\sqrt{n}} + \sqrt{\frac{\frac{r^2(\sqrt{k} + \sqrt{\log n})^2}{2\gamma^2} + k\epsilon_{min} + \log\frac{n}{\delta} + \log J}{2(n-1)}} \tag{33}$$

$$= \max_{\epsilon^\top F(\theta)\epsilon \leq \gamma^2} l_S(\theta + \epsilon) + \sqrt{\frac{O(k + \log\frac{n}{\delta})}{n-1}}. \tag{34}$$

□

B. Approximate Equality of KL Divergence and Fisher Quadratic

In this section we prove that $d(\theta + \epsilon, \theta) \approx \epsilon^\top F(\theta)\epsilon$ when ϵ is small, where

$$d(\theta', \theta) = \mathbb{E}_x\big[\mathrm{KL}(p(y|x, \theta')||p(y|x, \theta))\big] \quad \text{and} \quad F(\theta) = \mathbb{E}_x\mathbb{E}_\theta\big[\nabla \log p(y|x, \theta) \nabla \log p(y|x, \theta)^\top\big]. \tag{35}$$

Note that $\mathbb{E}_\theta[\cdot]$ indicates expectation over $p(y|x, \theta)$. First, we let $d(\theta', \theta; x) = \mathrm{KL}(p(y|x, \theta')||p(y|x, \theta))$ and $F(\theta; x) = \mathbb{E}_\theta\big[\nabla \log p(y|x, \theta) \nabla \log p(y|x, \theta)^\top\big]$. From the definition of the KL divergence,

$$d(\theta + \epsilon, \theta; x) = \sum_y p(y|x, \theta + \epsilon) \cdot \log \frac{p(y|x, \theta + \epsilon)}{p(y|x, \theta)}. \tag{36}$$

Regarding $p(y|x, \theta + \epsilon)$ and $\log p(y|x, \theta + \epsilon)$ as functions of θ, we apply the first-order Taylor expansion at θ to each function as follows:

$$p(y|x, \theta + \epsilon) \approx p(y|x, \theta) + \nabla_\theta p(y|x, \theta)^\top \epsilon \quad \text{and} \quad \log p(y|x, \theta + \epsilon) \approx \log p(y|x, \theta) + \nabla_\theta \log p(y|x, \theta)^\top \epsilon. \tag{37}$$

Plugging these approximates to (36) leads to:

$$d(\theta + \epsilon, \theta; x) \;\approx\; \sum_y \Big(p(y|x, \theta) + \epsilon^\top \nabla_\theta p(y|x, \theta)\Big) \cdot \nabla_\theta \log p(y|x, \theta)^\top \epsilon \tag{38}$$

$$= \mathbb{E}_\theta\big[\nabla_\theta \log p(y|x, \theta)\big]^\top \epsilon \;+\; \epsilon^\top \mathbb{E}_\theta\big[\nabla_\theta \log p(y|x, \theta) \nabla_\theta \log p(y|x, \theta)^\top\big]\epsilon, \tag{39}$$

where in (39) we use $\nabla_\theta p(y|x, \theta) = p(y|x, \theta)\nabla_\theta \log p(y|x, \theta)$. The first term of (39) equals 0 since

$$\mathbb{E}_\theta\big[\nabla_\theta \log p(y|x, \theta)\big] = \sum_y p(y|x, \theta) \cdot \frac{\nabla_\theta p(y|x, \theta)}{p(y|x, \theta)} = \sum_y \nabla_\theta p(y|x, \theta) = \frac{\partial}{\partial \theta} \sum_y p(y|x, \theta) = 0. \tag{40}$$

Lastly, taking expectation over x in (39) completes the proof.