GA-SAM: Gradient-Strength based Adaptive Sharpness-Aware Minimization for Improved Generalization

Zhiyuan Zhang[1], Ruixuan Luo[2], Qi Su[3,1], Xu Sun[1]
[1]MOE Key Laboratory of Computational Linguistics, School of Computer Science, Peking University
[2]Center for Data Science, Peking University
[3]School of Foreign Languages, Peking University
{zzy1210,luoruixuan97,sukia,xusun}@pku.edu.cn

Abstract

Recently, Sharpness-Aware Minimization (SAM) algorithm has shown state-of-the-art generalization abilities in vision tasks. It demonstrates that flat minima tend to imply better generalization abilities. However, it has some difficulty implying SAM to some natural language tasks, especially to models with drastic gradient changes, such as RNNs. In this work, we analyze the relation between the flatness of the local minimum and its generalization ability from a novel and straightforward theoretical perspective. We propose that the shift of the training and test distributions can be equivalently seen as a virtual parameter corruption or perturbation, which can explain why flat minima that are robust against parameter corruptions or perturbations have better generalization performances. On its basis, we propose a Gradient-Strength based Adaptive Sharpness-Aware Minimization (GA-SAM) algorithm to help to learn algorithms find flat minima that generalize better. Results in various language benchmarks validate the effectiveness of the proposed GA-SAM algorithm on natural language tasks.

1 Introduction

Recently, researchers (Wu et al., 2020; Weng et al., 2020; Sun et al., 2021; Zhang et al., 2021; Foret et al., 2020; Liu et al., 2021) propose that for better generalization ability, learning algorithms should find flat minima that have better robustness resistant to parameter corruptions or perturbations. Many learning algorithms that take the flatness or sharpness of the parameters into consideration are motivated by the observation that flat minima tend to imply better generalization abilities. Among them, Sharpness-Aware Minimization (SAM) (Foret et al., 2020) algorithm has achieved state-of-the-art generalization abilities in vision tasks. It adopts virtual adversarial parameter corruptions or perturbations during training and lowers the risk after parameter corruptions. However, traditional SAM algorithms usually adopt fixed strengths of parameter corruptions and constraint the corruptions with L_2-norm or $L_{+\infty}$-norm balls. It cannot conduct flexible strengths of parameter corruptions for different parameters, or during different stages of training. Thus, it is difficult to apply SAM to some natural language tasks, especially to models with drastic gradient changes, such as RNNs. To settle this issue, many adaptive SAM algorithms (Kwon et al., 2021; Liu et al., 2021) are proposed empirically. In this work, we propose a gradient-strength based adaptive solution based on our theoretical framework.

Existing studies (Wu et al., 2020; Zhang et al., 2021) try to explain the relation between the flatness of the local minimum and its generalization ability according to Probably Approximately Correct (PAC) Bayesian generalization bounds (Neyshabur et al., 2017). In this work, we propose a novel theoretical framework to analyze this relation from a more intuitive and direct perspective. In the Distributionally Robust Optimization (DRO) (Rahimian and Mehrotra, 2019) field, the elementary assumption is that there exists a shift between the distributions of the training set and the test set. We propose that a small distribution shift can be equivalently seen as a virtual parameter corruption or perturbation on the loss function. We conduct analytic trials to verify our theoretical account and the results show that it fits the simulation well and can therefore explain why flat minima that are robust against parameter corruptions or perturbations have better generalization performances. We also analyze the strength of the parameter corruption within this framework, based on which we propose a Gradient-Strength based Adaptive Sharpness-Aware Minimization (GA-SAM) algorithm, which can set flexible strengths of parameter corruptions for different parameter groups, during different training stages.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3888 - 3903

To validate the effectiveness of the proposed GA-SAM algorithm, we choose several natural language models and benchmarks, including Convolution Neural Networks (CNN) (Kim, 2014) on text classification, Long Short-term Memory (LSTM) (Merity et al., 2017) networks on language modeling, and Transformer (Vaswani et al., 2017) on neural machine translation. We also compare our proposed GA-SAM algorithm with the traditional SAM algorithm (Foret et al., 2020) and its multiple variants, including multi-step adversarial parameter defense algorithm (Zhang et al., 2021), adaptive SAM (Kwon et al., 2021), layer-wise SAM (Liu et al., 2021) and other possible variants of our proposed algorithm. Experimental results show that our proposed GA-SAM gains better generalization compared to the traditional SAM algorithm and other variants.

Our contributions can be summarized as follows:

- We propose a novel theoretical framework to analyze the relation between the flatness of the local minimum and its generalization ability. Under our proposed theoretical framework, the shift of the training and test distributions can be equivalently seen as a virtual parameter corruption or perturbation. Thus, the flatness or the robustness against parameter corruptions can indicate the generalization ability.

- On the basis of our novel framework, we further propose a Gradient-Strength based Adaptive Sharpness-Aware Minimization (GA-SAM) algorithm to set flexible strengths of parameter corruptions for different parameter groups, during different stages of training for an improvement over generalization ability.

- Experimental results show the effectiveness of the GA-SAM algorithm compared to the traditional SAM algorithm and its variants.

2 Proposed Theoretical Framework

In this section, we propose a novel theoretical framework to reveal the relation between distribution shifts and parameter corruptions from an intuitive and direct theoretical perspective.

2.1 Preliminary

Let us consider a neural network with the parameter vector $\mathbf{w} \in \mathbb{R}^n$. For a data instance $\mathbf{z} = (\mathbf{x}, y)$, denote $\ell(\mathbf{w}; \mathbf{z})$ as the loss of the data instance, $\mathcal{L}(\mathbf{w}; \mathcal{D})$ as the average loss of a dataset $\mathcal{D}$, and $p(\mathbf{z})$ as the probability distribution of $\mathcal{D}$, we have:

$$\mathcal{L}(\mathbf{w}; \mathcal{D}) = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[\ell] = \int_{\mathbf{z}} p(\mathbf{z}) \ell(\mathbf{w}; \mathbf{z}) d\mathbf{z}. \quad (1)$$

Denote $\boldsymbol{\theta}$ as the optimal parameter:

$$\boldsymbol{\theta} = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}; \mathcal{D}), \quad (2)$$

and the Hessian matrix on $\boldsymbol{\theta}$ is $\boldsymbol{H} = \nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathcal{D})$.

Similarly, denote $\mathcal{D}^*$ and $p^*(\mathbf{z})$ as the test set and its distribution, $\boldsymbol{\theta}^*$ as its optimal parameter, and the Hessian matrix on $\boldsymbol{\theta}^*$ is $\boldsymbol{H}^*$. Define the parameter shift of the test and training minima as $\boldsymbol{\delta} = \boldsymbol{\theta}^* - \boldsymbol{\theta}$.

Suppose n parameters are divided into l groups and the i-th group has $n_{(i)}$ parameters (*e.g.*, $l = 1, n = n_{(1)}$ when the whole model adopt the same strength and we call it model-wise, $n = l, n_{(i)} = 1$ when element-wise, l is the layer number when layer-wise, l is the filter number when filter-wise, *etc.*), $\mathbf{w} = [\mathbf{w}_{(1)}^{\mathrm{T}}, \cdots, \mathbf{w}_{(l)}^{\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{g} = \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}; \mathcal{D}) = [\mathbf{g}_{(1)}^{\mathrm{T}}, \cdots, \mathbf{g}_{(l)}^{\mathrm{T}}]^{\mathrm{T}}$, and $\boldsymbol{\delta} = [\boldsymbol{\delta}_{(1)}^{\mathrm{T}}, \boldsymbol{\delta}_{(2)}^{\mathrm{T}}, \cdots, \boldsymbol{\delta}_{(i)}^{\mathrm{T}}, \cdots, \boldsymbol{\delta}_{(l)}^{\mathrm{T}}]^{\mathrm{T}}$.

2.2 The Distribution Shift between the Training and Test Sets

The elementary assumption in the Distributionally Robust Optimization (DRO) (Rahimian and Mehrotra, 2019) field is that there exists a small distributional shift between the training and test sets. Previous studies usually assume that the divergence or the distance of the training and test distributions is bounded by a constant, *e.g.*, the Kullback-Leibler divergence (Kullback and Leibler, 1951), $\mathrm{KL}(p(\mathbf{z}) || p^*(\mathbf{z})) \leq$ Constant. In this work, more generally, we assume that the f-divergence (Rényi, 1961) D_f of the distributions is bounded by C_f:

$$D_f\left(p^*(\mathbf{z}) || p(\mathbf{z})\right) = \int_{\mathbf{z}} p(\mathbf{z}) f\left(\frac{p^*(\mathbf{z})}{p(\mathbf{z})}\right) \leq C_f, \quad (3)$$

where the function f is convex and $f(1) = 0$, its Taylor expansion should also satisfy $f(1+x) = a_1 x + a_2 x^2 + o(x^2), a_2 \neq 0$. For example, for the Kullback-Leibler divergence (KL-div), $f(1+x) = (1+x)\log(1+x) = x + x^2/2 + o(x^2)$.

2.3 Parameter Corruptions as Results of Distribution Shifts

We propose a novel theoretical framework to analyze this relation from a more intuitive and direct

perspective. The main theoretical motivation is Theorem 1. Proofs and details are in Appendix.

Theorem 1. *The distribution shifts of datasets $\mathcal{D}$ and $\mathcal{D}^*$ can be equivalently treated as a parameter corruption near the corresponding minima,*

$$\mathcal{L}(\boldsymbol{\theta}^* + \mathbf{v}; \mathcal{D}^*) \approx \mathcal{L}(\boldsymbol{\theta} + \mathbf{v}; \mathcal{D}) + \textit{Constant}, \quad (4)$$

where Constant=$\mathcal{L}(\boldsymbol{\theta}^*; \mathcal{D}^*) - \mathcal{L}(\boldsymbol{\theta}; \mathcal{D})$. Let $\mathbf{a} = -\boldsymbol{\delta}$, when $\mathbf{w}$ is near $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$, we have

$$\mathcal{L}(\mathbf{w}; \mathcal{D}^*) \approx \mathcal{L}(\mathbf{w} + \mathbf{a}; \mathcal{D}) + \textit{Constant}. \quad (5)$$

It shows that the distribution shifts will cause a parameter corruption or parameter shift. Therefore, optimizing the parameter corruption risk $\mathcal{L}(\mathbf{w} + \mathbf{a}; \mathcal{D})$ can help optimize the loss on the test set $\mathcal{L}(\mathbf{w}; \mathcal{D}^*)$. Define S as the possible corruption constraint set of potential corruptions $\mathbf{a}$. Since $\mathbf{a} = -\boldsymbol{\delta}$ is determined by the invisible distribution shifts, we optimize the risk under potential corruptions $\mathbf{a}$ instead, which is exactly the SAM optimization,

$$\boldsymbol{\theta}_{\text{SAM}} = \arg\min_{\mathbf{w}} \max_{\mathbf{a} \in S} \mathcal{L}(\mathbf{w} + \mathbf{a}; \mathcal{D}). \quad (6)$$

Thus, we reveal why flat minima that are robust against potential parameter corruptions or perturbations have better generalization performances in our theoretical framework. Traditional SAM algorithms adopt fixed strengths of parameter corruptions and constraint the corruptions with L_2-norm or $L_{+\infty}$-norm balls, namely $S = \{\mathbf{a} : \|\mathbf{a}\|_2 \leq \epsilon\}$ or $S = \{\mathbf{a} : \|\mathbf{a}\|_{+\infty} \leq \epsilon\}$. However, in Theorem 2, it reveals that the potential parameter corruption $\boldsymbol{\delta}$ is determined by the distribution shifts and the local geometry near the local minimum in the loss basin. Based on this, we have Proposition 1.

Theorem 2. *Define $r(\mathbf{z}) = p^*(\mathbf{z})/p(\mathbf{z}) - 1$. When the distribution shift is small enough, namely $r(\mathbf{z})$ is small, we can estimate the parameter shift $\boldsymbol{\delta}$ as,*

$$\boldsymbol{\delta} = -\boldsymbol{H}^{-1} \mathbb{E}_p[r(\mathbf{z}) \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{z})] + o(\|\boldsymbol{\delta}\|_2). \quad (7)$$

Proposition 1. *Suppose the loss $\mathcal{L}(\mathbf{w}; \mathcal{D})$ is μ-strongly convex*[1]*, and $D_f(p^*||p) \leq C_f$, there exists*

$$C_{\boldsymbol{\delta}} = \frac{1 + o(1)}{\mu} \sqrt{\frac{C_f}{a_2} \mathbb{E}_{p(\mathbf{z})} \left[\|\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{z})\|_2^2 \right]} \quad (8)$$

such that $\|\boldsymbol{\delta}\|_2 \leq C_{\boldsymbol{\delta}}$, namely $C_{\boldsymbol{\delta}}$ is a upper bound.

[1]Note that $\mathcal{L}$ is only required to be μ-strongly convex in the neighborhood of the loss basin including $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$, instead of the entire $\mathbb{R}^n$.

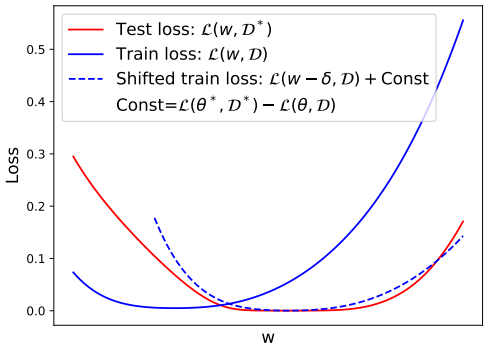

Figure 1: Visualizations of the training loss, test loss, and the shifted training loss. The shifted training loss is similar to the test loss near the local minimum.

2.4 Verification of the Theoretical Framework

In this section, we conduct several analytic trials[2] to verify our theoretical explanations and analyze the corruption strength. Results show that our theoretical framework fits the simulation.

Visualization of Distribution Shift Effects. Fig. 1 visualizes the training loss, test loss and the shifted training loss. The shifted training loss is similar to the test loss near the local minimum, namely we have $\mathcal{L}(\mathbf{w}; \mathcal{D}^*) \approx \mathcal{L}(\mathbf{w} - \boldsymbol{\delta}; \mathcal{D})$+Constant, which validates Theorem 1. This phenomenon can also be observed in visualizations of training and test loss landscapes in other studies.

Relation between Corruption Strength and Distribution Shift Strength. We conduct analytic trials to reveal the relation between the corruption strength and the distribution shift to verify our theoretical framework. Suppose $\mathcal{D}$ is the training set and $\mathcal{D}^*$ is the test set. We can construct a mixed dataset $\mathcal{D}^{\text{mix}}$, mixed with $(1 - \eta)$ of the training data from $\mathcal{D}$ and η of the test data from $\mathcal{D}^*$. We have $p^{\text{mix}} = (1 - \eta)p + \eta p^*$ approximately. Define η as the relatively distribution shift strength between $\mathcal{D}^{\text{mix}}$ and $\mathcal{D}$. Proposition 2 reveals that the corruption strength is proportional to the distribution shift strength, which fits both the simulation results of analytic trails in Fig. 2 and the intuition.

Proposition 2. *Suppose the mixed distribution of $\mathcal{D}^{mix}$ is $p^{mix} = (1 - \eta)p + \eta p^*$, then we have $D_f(p^{mix}||p) \leq C_f^{mix} = \eta^2 C_f$. Denote $\boldsymbol{\theta}^{mix}$ as the optimal parameter on $\mathcal{D}^{mix}$, $\boldsymbol{\delta}^{mix} = \boldsymbol{\theta}^{mix} - \boldsymbol{\theta}$, then*

[2]The details of trials are in Appendix.

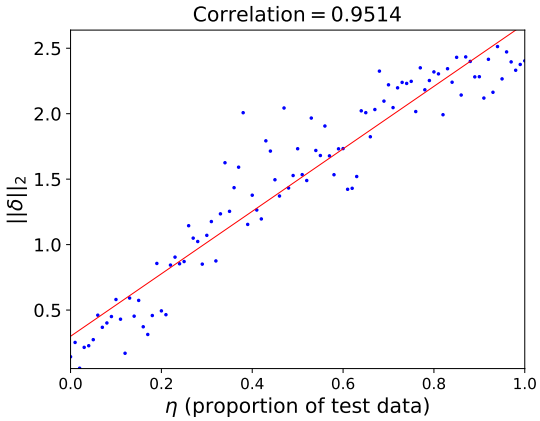

Figure 2: Results of $\|\boldsymbol{\delta}\|_2$ of 100 trials with the same training set $\mathcal{D}$ and different mixed test sets $\mathcal{D}^{\text{mix}}$ (mixed with $(1-\eta)$ of the training data from $\mathcal{D}$ and η of the test data from $\mathcal{D}^*$). η can be utilized to measure the strength of the distribution shift between $\mathcal{D}$ and $\mathcal{D}^{\text{mix}}$. Results show that there exists an approximately linear relationship between $\|\boldsymbol{\delta}\|_2$ and distribution shift strengths.

we have:

$$\frac{\|\boldsymbol{\delta}^{mix}\|_2}{\|\boldsymbol{\delta}\|_2} = \frac{C_{\boldsymbol{\delta}}^{mix}}{C_{\boldsymbol{\delta}}} = \eta + o(1). \tag{9}$$

3 Gradient-Strength based Adaptive Sharpness-Aware Minimization

In this section, we propose a Gradient-Strength based Adaptive Sharpness-Aware Minimization (GA-SAM) algorithm based on the proposed theoretical framework, which conducts flexible strengths of parameter corruptions for different parameter groups for better generalization abilities.

3.1 Adaptive Sharpness-Aware Minimization

As illustrated in Section 2.3, the SAM (Foret et al., 2020) optimization objective is exactly the risk under potential corruption as a result of distribution shift,

$$\boldsymbol{\theta} = \arg\min_{\mathbf{w}} \max_{\mathbf{a}\in S} \mathcal{L}(\mathbf{w}+\mathbf{a}; \mathcal{D}), \tag{10}$$

where the constraint S is $S = \{\mathbf{a} : \|\mathbf{a}\|_p \leq \epsilon\}$.

Adaptive SAM algorithms (Kwon et al., 2021; Liu et al., 2021) are proposed to set flexible strengths of parameter corruptions, which set the constraint S as $S = \{\mathbf{a} : \|\boldsymbol{T}^{-1}\mathbf{a}\|_p \leq \epsilon\}$, where $\boldsymbol{T}$ is the transformation matrix (usually diagonal) controlling the corruption strengths of corresponding parameter groups. Define $\boldsymbol{T} = \text{diag}\{T_{(1)}\boldsymbol{I}_{n_{(1)}}, \cdots, T_{(l)}\boldsymbol{I}_{n_{(l)}}\}$, where $T_{(i)}$ controls the corruption strengths of group i. Adaptive SAM (ASAM) (Kwon et al., 2021) empirically adopts $T_{(i)} = \|\mathbf{w}_{(i)}\|_2$ element-wisely or filter-wisely in CNNs, and layer-wise SAM (Layer-SAM) (Liu et al., 2021) empirically adopts $T_{(i)} = \|\mathbf{w}_{(i)}\|_2/\|\mathbf{g}_{(i)}\|_2$ layer-wisely.

3.2 The Relation between the Corruption Strength and the Gradient Strength

We hope to derive $\boldsymbol{T}$ from the theoretical framework instead of setting $\boldsymbol{T}$ empirically.

In Theorem 2, besides the term $r(\mathbf{z})$ determined by the distribution shift, corruption strength is also determined by the local geometry near the local minimum $(\boldsymbol{H}, \nabla_{\boldsymbol{\theta}}\ell(\theta; \mathbf{z}))$. Suppose n is the parameter number, $G = \mathbb{E}_{p(\mathbf{z})}[\|\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}; \mathbf{z})\|_2]/\sqrt{n}$ is the average gradient strength. Suppose the Fisher information matrix assumption (Pascanu and Bengio, 2013) holds, namely $\boldsymbol{H} = \mathbb{E}_{p(\mathbf{z})}[\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}; \mathbf{z})\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}; \mathbf{z})^{\mathrm{T}}]$, the local geometry is determined by the gradient strength G.

In Proposition 3, we analyze the relation between the corruption strength and the gradient strength. We have $\|\boldsymbol{\delta}\|_2 \propto \sqrt{n}/G$ and $\|\boldsymbol{\delta}_{(i)}\|_2 \propto \sqrt{n_{(i)}}/G_{(i)}$, corruption strengths and scales $\boldsymbol{T}$ should be determined by gradient strengths G.

Proposition 3. *Define the average gradient strength as* $G = \mathbb{E}_{p(\mathbf{z})}[\|\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}; \mathbf{z})\|_2]/\sqrt{n}$, *and the average gradient strength of group* i *as* $G_{(i)} = E_{p(\mathbf{z})}[\|\nabla_{\boldsymbol{\theta}_{(i)}}\ell(\boldsymbol{\theta}; \mathbf{z})\|_2]/\sqrt{n_{(i)}}$, *then*

$$\|\boldsymbol{\delta}\|_2 \propto \frac{\sqrt{n}}{G}, \quad \|\boldsymbol{\delta}_{(i)}\|_2 \propto \frac{\sqrt{n_{(i)}}}{G_{(i)}}. \tag{11}$$

3.3 Proposed Algorithm

Our proposed algorithm adopts the constraint $S = \{\mathbf{a} : \|\boldsymbol{T}^{-1}\mathbf{a}\|_p \leq \epsilon\}$ in SAM learning, where $\boldsymbol{T}$ are adaptive scales derived from the theoretical framework, and we adopt the multi-step implementation.

In Proposition 3, we have $\|\boldsymbol{\delta}_{(i)}\|_2 \propto \sqrt{n_{(i)}}/G_{(i)}$. To make the scale of the virtual corruptions $\mathbf{a}_{(i)}$ propotional to the scale of $\boldsymbol{\delta}_{(i)}$, we should ensure that $\|\mathbf{a}_{(i)}\|_2 \propto \|\boldsymbol{\delta}_{(i)}\|_2$. Suppose $a = \|\mathbf{a}_{(i)}\|_2/\sqrt{n_{(i)}}$ is the average scale of $\mathbf{a}_{(i)}$, we have $a \propto T_{(i)}\epsilon$, we should ensure that

$$\|\mathbf{a}_{(i)}\|_2 = a\sqrt{n_{(i)}} \propto \|\boldsymbol{\delta}_{(i)}\|_2 \propto \frac{\sqrt{n_{(i)}}}{G_{(i)}}, \tag{12}$$

and we can set $T_{(i)}\epsilon\sqrt{n_{(i)}} \propto \sqrt{n_{(i)}}/G_{(i)}$, therefore $T_{(i)} \propto 1/G_{(i)} = \sqrt{n_{(i)}}/\mathbb{E}_{p(\mathbf{z})}[\|\nabla_{\boldsymbol{\theta}_{(i)}}\ell(\boldsymbol{\theta}; \mathbf{z})\|_2]$.

We use $\|\mathbf{g}_{(i)}\|_2$ to replace $\mathbb{E}_{p(\mathbf{z})}[\|\nabla_{\boldsymbol{\theta}_{(i)}}\ell(\boldsymbol{\theta};\mathbf{z})\|_2]$, then our proposal is derived,

$$T_{(i)} = \frac{\sqrt{n_{(i)}}}{\|\mathbf{g}_{(i)}\|_2\sqrt{n}}. \tag{13}$$

Existing algorithms are mainly single-step based, we adopt the multi-step implementation inspired by multi-step adversarial parameter defense algorithm (Zhang et al., 2021), which optimizes

$$\boldsymbol{\theta} = \arg\min_{\mathbf{w}} \mathbb{E}_{\mathcal{B}}\left[\sum_{k=0}^{K}\frac{\mathcal{L}(\mathbf{w}+\mathbf{a}_k;\mathcal{B})}{K+1}\right], \tag{14}$$

where $\mathbf{a}_0 = \mathbf{0}$. Suppose the k-th update of the corruption is $\mathbf{u}_k = \arg\max_{\|\boldsymbol{T}^{-1}\mathbf{u}\|_p\le\eta}\mathbf{u}^{\mathrm{T}}\mathbf{g}_{k-1}$, which is generated based on $\mathbf{a}_{k-1}$ and $\mathbf{g}_{k-1} = \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}+\mathbf{a}_{k-1};\mathcal{B})$, where η is the step size and following Zhang et al. (2021), we set $\eta = 1.5\epsilon/K$, then

$$\mathbf{u}_k = \frac{\eta\big(\boldsymbol{T}\mathrm{sgn}(\mathbf{g}_{k-1})\big)\odot|\boldsymbol{T}\mathbf{g}_{k-1}|^{\frac{1}{p-1}}}{\||\boldsymbol{T}\mathbf{g}_{k-1}|^{\frac{1}{p-1}}\|_p}. \tag{15}$$

To get the k-th corruption $\mathbf{a}_k$, we project the updated corruption $\mathbf{a}_{k-1}+\mathbf{u}_k$ into the set S,

$$\mathbf{a}_k = \Pi_S(\mathbf{a}_{k-1}+\mathbf{u}_k), \tag{16}$$

and the solutions to the commonly adopted L_2-norm and $L_{+\infty}$-norm constraints are:

$$\Pi_{\|\boldsymbol{T}^{-1}\mathbf{v}\|_2\le\epsilon}(\mathbf{v}) = \frac{\min\{\|\boldsymbol{T}^{-1}\mathbf{v}\|_2,\epsilon\}\mathbf{v}}{\|\boldsymbol{T}^{-1}\mathbf{v}\|_2}; \tag{17}$$

$$\Pi_{\|\boldsymbol{T}^{-1}\mathbf{v}\|_{+\infty}\le\epsilon}(\mathbf{v}) = \boldsymbol{T}\mathrm{clip}(\boldsymbol{T}^{-1}\mathbf{v},-\epsilon,\epsilon). \tag{18}$$

To summarize, we adopt the adaptive scale $T_{(i)}$ according to the gradient strength and a multi-step implementation. It should be noted that even when $K=1$, our multi-step implementation, which optimizes $(\mathcal{L}(\mathbf{w})+\mathcal{L}(\mathbf{w}+\mathbf{a}_1))/2$, is different from the single-step SAM implementation, which optimizes $\mathcal{L}(\mathbf{w}+\mathbf{a}_1)$. However, when $K=1$, they have similar speeds since they both require to generate $\mathbf{a}_1$ and need two backward propagation processes. Our multi-step implementation, however, allows setting a larger K for better generalization.

We name the proposed algorithm as the Gradient-Strength based Adaptive SAM (**GA-SAM**). The algorithm is shown in Algorithm 1. The proofs and theoretical details are in Appendix.

4 Experiments

In this section, we report the tasks, datasets, and implementation details. Main results are in Table 1.

Algorithm 1 GA-SAM Algorithm

Require: Parameters $\mathbf{w}$; loss $\mathcal{L}$ and dataset $\mathcal{D}$; steps K; training iterations; batch size $|\mathcal{B}|$.

Prepare batches $\{\mathcal{B}\}$ and initialize $\mathbf{w}$.
Calculate $\boldsymbol{T}$ with $T_{(i)} = \frac{\sqrt{n_{(i)}}}{\|\mathbf{g}_{(i)}\|_2\sqrt{n}}$.
while Training **do**
  $\mathbf{a}_0 \leftarrow \mathbf{0}$.
  Calculate the initial loss: $\mathcal{L}(\mathbf{w};\mathcal{B})$.
  for $k=1$ to K **do**
    Get $\mathbf{u}_k = \frac{\eta\big(\boldsymbol{T}\mathrm{sgn}(\mathbf{g}_{k-1})\big)\odot|\boldsymbol{T}\mathbf{g}_{k-1}|^{\frac{1}{p-1}}}{\||\boldsymbol{T}\mathbf{g}_{k-1}|^{\frac{1}{p-1}}\|_p}$.
    Get $\mathbf{a}_k \leftarrow \Pi_S(\mathbf{a}_{k-1}+\mathbf{u}_k)$ as Eq. (16).
    Calculate the risk: $\mathcal{L}(\mathbf{w}+\mathbf{a}_k;\mathcal{B})$.
  end for
  Update $\mathbf{w}$ as minimizing Eq. (14).
end while

4.1 Tasks and Datasets

We adopt three typical neural networks and natural language tasks to validate the effectiveness of the proposed GA-SAM algorithm on NLP tasks.

On the **text classification** task, we adopt Convolution Neural Networks (**CNN**) (Kim, 2014) on the IMDb movie reviews dataset (**IMDB**) (Maas et al., 2011) with the accuracy (**ACC**) evaluation metric. On the **language modeling** task, we adopt Long Short-term Memory (**LSTM**) (Merity et al., 2017) networks on the English Penn TreeBank (**PTB-LM**) (Marcus et al., 1993) dataset with the perplexity (**PPL**) evaluation metric. On the neural machine **translation** task, we adopt the **Transformer** (Vaswani et al., 2017) model based on the Fairseq implementation (Ott et al., 2019) on IWSLT 15 English-Vietnamese (**En-Vi**) (Cettolo et al., 2015) and IWSLT 14 German-English (**De-En**) (Cettolo et al., 2014) datasets with the **BLEU** score evaluation metric. Compared with the classification and language modeling tasks, the datasets of the machine translation task are relatively large. Other details are in Appendix.

4.2 Implementations

We implement our proposed GA-SAM algorithm with a multi-step risk minimization, and set layer-wise adaptive scales $T_{(i)} = \sqrt{n_{(i)}}/(\|\mathbf{g}_{(i)}\|_2\sqrt{n})$. We also compare our proposed GA-SAM with other existing algorithms. The traditional SAM algorithm (Foret et al., 2020) adopts a single-step risk implementation and sets fixed scales. ASAM (Kwon et al., 2021) adopts a single-step

Dataset	Approach	IMDB (ACC)	PTB-LM (PPL)	En-Vi (BLEU)	De-En (BLEU)
	Base Model	CNN	LSTM	Transformer	Transformer
Baseline	w/o SAM	84.42±0.12	86.70±0.54	30.60±0.21	35.41±0.13
Single-step	SAM	84.75±0.31 (+0.33)	89.66±0.25 (+2.96)	30.79±0.15 (+0.19)	35.61±0.18 (+0.20)
	ASAM	85.05±0.22 (+0.63)	90.08±0.24 (+3.38)	30.81±0.24 (+0.21)	35.56±0.17 (+0.15)
	Layer-SAM	85.27±0.19 (+0.83)	89.82±0.10 (+3.12)	30.70±0.27(+0.10)	35.78±0.08 (+0.37)
Multi-step	Multi-step Defense	84.87±0.15 (+0.45)	84.74±0.42 (-1.96)	30.95±0.12 (+0.35)	35.86±0.13 (+0.45)
	Proposed GA-SAM	**86.11±0.22 (+1.69)**	**84.52±0.26 (-2.18)**	**31.15±0.29 (+0.55)**	**35.95±0.15 (+0.54)**

Table 1: Results of baselines and different SAM algorithms. Results show the effectiveness of GA-SAM.

risk implementation and sets element-wise adaptive scales $T_{(i)} = |w_{(i)}|$. Layer-SAM (Liu et al., 2021) adopts a single-step risk implementation and sets layer-wise adaptive scales $T_{(i)} = \|\mathbf{w}_{(i)}\|_2 / \|\mathbf{g}_{(i)}\|_2$. The multi-step adversarial parameter defense algorithm (Zhang et al., 2021) adopts a multi-step risk implementation and sets fixed scales. We also try to combine these techniques and implement other possible variants of GA-SAM, and we conduct an ablation study to compare GA-SAM with these variants.

For a fair comparison, the settings of different SAM algorithms and variants are the same as the base models except for the SAM settings $(K, \epsilon, L_p, T_{(i)})$. We grid search the optimal SAM hyper-parameters for each algorithm. The details of base models and hyper-parameters are in Appendix.

4.3 Main Results

The main results are shown in Table 1. Evidently, our proposed GA-SAM leads to significant performance gains over the base models.

Single-step SAM algorithms cannot improve the performance of the LSTM base model on the language modeling task. SAM algorithms that set adaptive scales cannot improve the performance of the traditional SAM algorithm consistently. But it may help deal with drastic changes in gradient scales of different parameters or different learning phases in NLP tasks. The multi-step risk minimization generally outperforms SAM and helps improve the stability of learning and the generalization abilities of models. GA-SAM can both achieve the adaptive scales deduced from the theoretical framework and help improve the stability and the generalization via the multi-step implementation. Thus, GA-SAM outperforms base models and other SAM algorithms.

Approach	IMDB (ACC)	PTB-LM (PPL)
Baseline	84.42±0.12	86.70±0.54
SAM	84.75±0.31 (+0.33)	89.66±0.25 (+2.96)
GA-SAM	**86.11±0.22 (+1.69)**	**84.52±0.26 (-2.18)**
Single-step	85.86±0.25 (+1.44)	89.97±0.27 (+3.27)
Element-wise	84.87±0.14 (+0.45)	84.69±0.38 (-2.01)
Model-wise	85.32±0.21 (+0.90)	84.55±0.39 (-2.15)
Variants with other scales $T_{(i)}$:		
1	84.87±0.15 (+0.45)	84.74±0.42 (-1.96)
$\|\mathbf{w}_{(i)}\|_2 / \|\mathbf{g}_{(i)}\|_2$	85.82±0.06 (+1.40)	84.58±0.27 (-2.12)
$1/\|\mathbf{g}_{(i)}\|_2$	85.81±0.23 (+1.39)	84.90±0.32 (-1.80)
$\|\mathbf{w}_{(i)}\|_2 / \sqrt{n_{(i)}}$	85.06±0.07 (+0.64)	85.04±0.15 (-1.66)
$\|\mathbf{w}_{(i)}\|_2$	85.42±0.10 (+1.00)	85.03±0.57 (-1.67)

Table 2: Results of the ablation study. GA-SAM ($K = 1$) is compared to multiple variants. Results show that GA-SAM outperforms other variants, and gradient-strength based adaptive scales usually outperform other scales.

5 Analysis

In this section, we first conduct an ablation study and analyze the hyper-parameters. Then we illustrate the sharpness and Hessian spectra with GA-SAM and analyze the difference between CV and NLP learning to explain why NLP tasks need GA-SAM.

5.1 Ablation Study

We implement the single-step variants, element-wise or model-wise variants, and variants with other scales $T_{(i)}$ on IMDB and PTB-LM. The results of the ablation study are reported in Table 2.

Approach	IMDB (ACC)	PTB-LM (PPL)
Baseline	84.42±0.12	86.70±0.54
GA-SAM w/ diff. (with different) K:		
K = 1	**86.11±0.22 (+1.69)**	84.52±0.26 (-2.18)
K = 2	85.83±0.23 (+1.41)	84.48±0.52 (-2.22)
K = 3	84.97±0.23 (+0.55)	84.14±0.29 (-2.56)
K = 4	77.55±3.25 (-6.87)	**84.05±0.23 (-2.65)**
K = 5	70.49±5.10 (-13.9)	84.59±0.36 (-2.11)
w/ diff. ϵ (L_2):	$\times 10^{-2}$	$\times 10^{-3}$
ϵ = 0.1×	84.93±0.27 (+0.51)	85.24±0.40 (-1.46)
ϵ = 0.5×	85.11±0.22 (+0.69)	84.90±0.09 (-1.80)
ϵ = 1×	85.36±0.26 (+0.94)	84.91±0.17 (-1.79)
ϵ = 5×	85.62±0.70 (+1.20)	**84.69±0.40 (-2.01)**
ϵ = 10×	**85.77±0.11 (+1.35)**	85.38±0.41 (-1.32)
ϵ = 50×	64.14±10.3 (-20.3)	507.0±243 (+420)
w/ diff. ϵ $(L_{+\infty})$:	$\times 10^{-4}$	$\times 10^{-5}$
ϵ = 0.2×	84.91±0.89 (+0.49)	85.12±0.55 (-1.58)
ϵ = 0.5×	85.54±0.18 (+1.12)	85.19±0.86 (-1.51)
ϵ = 0.8×	85.63±0.15 (+1.21)	**84.52±0.26 (-2.18)**
ϵ = 1×	**86.11±0.22 (+1.69)**	85.06±0.51 (-1.64)
ϵ = 2×	85.74±0.18 (+1.32)	85.20±0.94 (-0.96)
ϵ = 5×	65.96±2.74 (-18.5)	85.87±1.19 (-0.83)

Table 3: Analysis of hyper-parameters. We implement GA-SAM with different K and ϵ (under L_2 and $L_{+\infty}$).

Experimental results show that layer-wise implementation outperforms element-wise or model-wise implementations and GA-SAM with the multi-step implementation ($K = 1$) outperforms GA-SAM with the single-step implementation.

For adaptive scales, the multi-step adversarial parameter defense algorithm (Zhang et al., 2021) adopts $T_{(i)} = 1$. We also adopt $T_{(i)} = \|\mathbf{w}_{(i)}\|_2/\|\mathbf{g}_{(i)}\|_2$ following Liu et al. (2021), and $T_{(i)} = \|\mathbf{w}_{(i)}\|_2$ following Kwon et al. (2021). We also try other variants with similar formulas. Experimental results show that GA-SAM outperforms other variants. Gradient-strength ($\|\mathbf{g}_{(i)}\|_2$) based adaptive scales usually outperform other adaptive scales, and adaptive scales can enhance the multi-step adversarial parameter defense algorithm generally, which validates our theoretical framework.

5.2 Analysis of Hyper-parameters

We analyze the influence of hyper-parameters and settings in GA-SAM learning in Table 3.

On IMDB, adopting larger K cannot improve the accuracy, while on PTB-LM, larger K can achieve better PPL. The reason might be that on PTB-LM, the LSTM model with more drastic gradient changes needs more steps for better stability.

Corruption	Baseline	GA-SAM
w/o Corruption	30.60±0.21	31.15±0.29
L_2, ϵ = 0.05	30.26 (-0.34)	30.90 (-0.25)
L_2, ϵ = 0.1	29.36 (-1.24)	30.26 (-0.89)
L_2, ϵ = 0.2	6.46 (-24.14)	17.87 (-13.28)
$L_{+\infty}$, ϵ = 0.0001	30.30 (-0.30)	30.72 (-0.43)
$L_{+\infty}$, ϵ = 0.0002	29.95 (-0.65)	30.67 (-0.48)
$L_{+\infty}$, ϵ = 0.0005	5.33 (-25.27)	29.14 (-2.01)

Table 4: The parameter robustness of baselines and GA-SAM on En-Vi. Minima with GA-SAM are more robust.

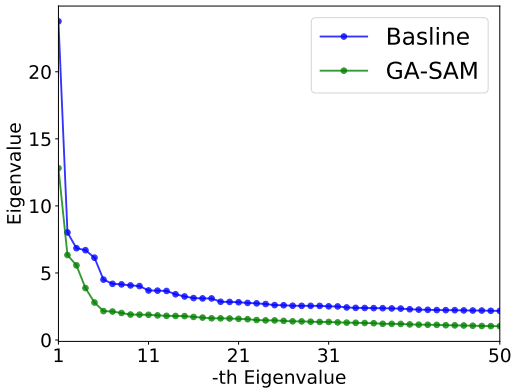

Figure 3: Top 50 eigenvalues of Hessians of baselines and GA-SAM. GA-SAM can help find flat minima.

Both under L_2 and $L_{+\infty}$ constraints, on both datasets, the performance can often be improved substantially with small ϵ. However, when ϵ grows too large, it may harm the learning and the performance will drop. We also find that in this work, the best performance is achieved under $L_{+\infty}$.

5.3 Sharpness and Parameter Robustness

As analyzed in our theoretical framework, flat minima tend to imply better generalization abilities. In this section, we validate that GA-SAM can help find flat minima, which tends to help improve the generalization abilities of models.

The sharpness near the local minima can be evaluated by the parameter robustness against parameter corruptions via the multi-step adversarial parameter corruption algorithm (Zhang et al., 2021). In Table 4, we evaluate the robustness of the baselines and models with GA-SAM against the L_2 or $L_{+\infty}$ constrained parameter corruptions on En-Vi. We can see that models with GA-SAM are more robust than baselines, which implies that GA-SAM can help find flat minima. We also adopt the Fisher

Approach	En-Vi (BLEU)	De-En (BLEU)
Baseline	30.60±0.21	35.41±0.13
SAM	30.79±0.15 (+0.19)	35.61±0.18 (+0.20)
FreeLB	30.91±0.09 (+0.31)	35.49±0.11 (+0.08)
GA-SAM	**31.15±0.29 (+0.55)**	**35.95±0.15 (+0.54)**

Table 5: Comparisons to FreeLB.

information matrix assumption (Pascanu and Bengio, 2013) to estimate the top 50 eigenvalues of the Hessian matrix to evaluate the sharpness of minima. As shown in Fig. 3, the eigenvalues of models with GA-SAM are lower, which illustrates that GA-SAM can help find flat minima.

5.4 Comparisons to Adversarial Training.

Some adversarial training algorithms improve the generalization ability of neural networks by optimizing the loss of virtual adversarial examples. In this section, we compare GA-SAM with FreeLB (Zhu et al., 2019), an existing high-performance algorithm for NLP tasks, on En-Vi and De-En. Detailed settings are in Appendix.

In Table 5, we can see that FreeLB (Zhu et al., 2019) can improve the accuracy of NLP models. The reasons that FreeLB (Zhu et al., 2019) works may lie in two aspects: (1) FreeLB adopts a multi-step minimization that is helpful for training stability; and (2) FreeLB only involves attacks on word embeddings while the ideal attack strengths for parameters in different layers vary a lot due to gradient vanishing and explosion in NLP models. Therefore, FreeLB does not need flexible scales as necessarily as SAM. However, GA-SAM can still outperform FreeLB.

5.5 Why NLP Tasks Need GA-SAM

Compared with the traditional SAM algorithm, GA-SAM adopts the multi-step risk minimization and gradient-strength based adaptive corruption strengths. From the ablation study, we can see that the multi-step risk minimization can enhance the traditional SAM algorithm on NLP tasks. It shows that NLP tasks do need GA-SAM for better stability and generalization with the multi-step risk minimization algorithm. The gradient-strength based adaptive corruption strengths can also enhance SAM algorithms since gradient strengths change drastically during different learning phases and gradient strengths vary in different layers.

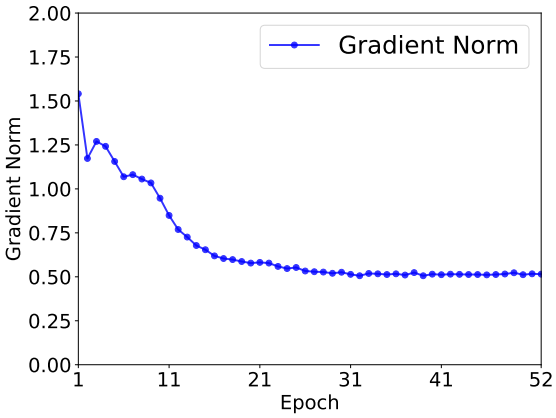

Figure 4: Gradient norms in different learning phases.

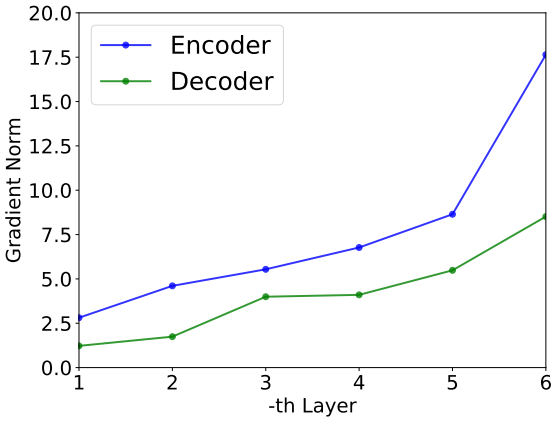

Figure 5: Gradient norms of different layers in Transformer encoder and decoder. 6-th layer is the highest layer in the encoder or decoder, which has the largest gradients.

In this section, we further visualize the gradient strengths during different learning phases and different layers on En-Vi to illustrate why NLP tasks need GA-SAM. As shown in Fig. 4, during the early phase of learning, the gradient norms are much larger and we should conduct parameter corruptions with smaller strengths. As shown in Fig. 5, we can see that higher layers of the Transformer encoder or decoder have larger gradients and need smaller parameter corruptions. To conclude, we need gradient-strength based adaptive corruption strengths since gradient strengths vary in different learning phases and different layers, while the traditional SAM algorithm fails to conduct flexible strengths of parameter corruptions for different parameters, or during different stages of training.

5.6 Computational Complexity

Single-step implementations usually involve two forward and backward propagations, including one

for generating the corruption $\mathbf{a}$ and another for optimizing the $\mathcal{L}(\mathbf{w}+\mathbf{a})$ loss function or the $\mathcal{L}(\mathbf{w})+\lambda\mathbf{a}^{\mathrm{T}}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w})$ loss function). Multi-step implementations involve $K+1$ forward and backward propagations, including $K+1$ times for forwarding and backwarding $\{\mathcal{L}(\mathbf{w}+\mathbf{a}_k)\}_{0\leq k\leq K}$.

Therefore, when $K=1$, the neural network forward and backward propagation cost (involving two forward and backward propagations) of GA-SAM is the same as SAM or GA-SAM with the single-step implementation. In Table 2, we conduct a fair comparison between GA-SAM ($K=1$) and SAM or GA-SAM with the single-step implementation (all involve two forward and backward propagations). GA-SAM still outperforms SAM and GA-SAM with the single-step implementation, which illustrates the effectiveness of our proposed GA-SAM and the multi-step implementation.

6 Related Work

6.1 Parameter Corruptions or Perturbations

Besides adversarial examples (Szegedy et al., 2014; Kurakin et al., 2017; Carlini and Wagner, 2017) and adversarial training (Goodfellow et al., 2015; Wang et al., 2019; Madry et al., 2018; Zhang et al., 2019; Zhu et al., 2019) concerning adversarial examples, existing studies also concerns small changes on neural network parameters, namely parameter corruptions (Sun et al., 2021; Zhang et al., 2021) or perturbations (Garg et al., 2020; Wu et al., 2020).

Existing studies research parameter corruptions or perturbations mainly for better generalization ability (Zheng et al., 2020; Foret et al., 2020; Kwon et al., 2021; Liu et al., 2021; Du et al., 2021; Sun et al., 2021; Zhang et al., 2021), safety issue (Garg et al., 2020; Rakin et al., 2020), analyzing the loss change allocation to parameters (Lan et al., 2019), analyzing the compression (Arora et al., 2018) or parameter quantization (Nagel et al., 2019; Migacz, 2017; Alizadeh et al., 2020).

6.2 Generalization and Flat Minima

Existing studies (Dinh et al., 2017; Keskar et al., 2017; Chaudhari et al., 2017; Xie et al., 2020; Wu et al., 2020; Sun et al., 2021; Zhang et al., 2021) show that local minima that are robust against to parameter corruptions are usually flat minima, which tends to have better generalization ability. A line of Sharpness-aware Minimization (SAM) (Foret et al., 2020) algorithms drive parameters away from sharp minima via virtual parameter corruptions. Other researches acquire flat minima via adopting adaptive scales of parameter corruptions (Kwon et al., 2021; Liu et al., 2021), rescaling parameter corruptions (Liu et al., 2021; Du et al., 2021), adopting a multi-step implementation (Zhang et al., 2021) or sharpness-aware learning rates (Yue et al., 2020).

7 Limitation and Broader Impact

Limitation. One limitation of our work is that, similar to other SAM learning, the hyper-parameters tuning, especially ϵ, involves many numerical experiments, which is time costly and environmentally unfriendly. To settle this issue, we recommend researchers binary search the proper order of magnitude of ϵ first, and then grid search ϵ in a small range for a faster hyper-parameter search, instead of directly grid searching ϵ in a large range.

Broader Impact. Our work is beneficial for the security of NLP models since our work can help improve the robustness of NLP models against parameter corruptions, which can occur as random noises at the hardware level, quantization, or model compression. Our work also has negative social impacts. Our proposed GA-SAM can be utilized to enhance NLP models and improve the accuracy of base models. However, the hyper-parameters tuning, especially ϵ, involves many numerical experiments, which is also a limitation of our work, and it is environmentally unfriendly.

8 Conclusion

In this paper, we propose a novel theoretical framework to analyze the relation between parameter corruptions and generalization abilities. Based on our proposed framework, we propose a Gradient-Strength based Adaptive Sharpness-Aware Minimization (GA-SAM) algorithm. Experimental results validate the effectiveness of GA-SAM compared to the traditional SAM algorithm and its variants. Further analyses also show that GA-SAM can help find flat minima and improve the generalization ability of neural networks.

Acknowledgement

The authors would like to thank the reviewers for their helpful comments. This work is supported by Natural Science Foundation of China (NSFC) No. 62176002 and Beijing Natural Science Foundation of China (4192057). Xu Sun is the corresponding author.

References

Milad Alizadeh, Arash Behboodi, Mart van Baalen, Christos Louizos, Tijmen Blankevoort, and Max Welling. 2020. Gradient ℓ_1 regularization for quantization robustness. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. 2018. Stronger generalization bounds for deep nets via a compression approach. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 254–263.

Nicholas Carlini and David A. Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2015. The iwslt 2015 evaluation campaign. In *IWSLT 2015, International Workshop on Spoken Language Translation*.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. The iwslt 2015 evaluation campaign. In *IWSLT 2014, International Workshop on Spoken Language Translation*.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Levent Sagun, and Riccardo Zecchina. 2017. Entropy-sgd: Biasing gradient descent into wide valleys. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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

Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent YF Tan. 2021. Efficient sharpness-aware minimization for improved training of neural networks. *arXiv preprint arXiv:2110.03141*.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2020. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*.

Siddhant Garg, Adarsh Kumar, Vibhor Goel, and Yingyu Liang. 2020. Can adversarial weight perturbations inject neural backdoors. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2029–2032. ACM.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2017. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. 2021. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. *arXiv preprint arXiv:2102.11600*.

Janice Lan, Rosanne Liu, Hattie Zhou, and Jason Yosinski. 2019. LCA: loss change allocation for neural network training. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 3614–3624.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. 2021. Sharpness-aware minimization in large-batch training: Training vision transformer in minutes.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and Optimizing LSTM Language Models. *arXiv preprint arXiv:1708.02182*.

Szymon Migacz. 2017. 8-bit inference with tensorrt. In *GPU technology conference*, volume 2, page 5.

Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. 2019. Data-free quantization through weight equalization and bias correction. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 1325–1334. IEEE.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. 2017. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5947–5956.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Razvan Pascanu and Yoshua Bengio. 2013. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*.

Hamed Rahimian and Sanjay Mehrotra. 2019. Distributionally robust optimization: A review. *CoRR*, abs/1908.05659.

Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. 2020. TBT: targeted neural network attack with bit trojan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 13195–13204. IEEE.

Alfréd Rényi. 1961. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561. University of California Press.

Xu Sun, Zhiyuan Zhang, Xuancheng Ren, Ruixuan Luo, and Liangyou Li. 2021. Exploring the vulnerability of deep neural networks: A study of parameter corruption. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 11648–11656. AAAI Press.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

J. Wang, Tianyun Zhang, S. Liu, Pin-Yu Chen, Jiacen Xu, M. Fardad, and B. Li. 2019. Towards a unified min-max framework for adversarial exploration and robustness. *arXiv: Learning*.

Tsui-Wei Weng, Pu Zhao, Sijia Liu, Pin-Yu Chen, Xue Lin, and Luca Daniel. 2020. Towards certificated model robustness against weight perturbations. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 6356–6363. AAAI Press.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. 2020. Adversarial weight perturbation helps robust generalization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zeke Xie, Issei Sato, and Masashi Sugiyama. 2020. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. *arXiv preprint arXiv:2002.03495*.

Xubo Yue, Maher Nouiehed, and Raed Al Kontar. 2020. Salr: Sharpness-aware learning rates for improved generalization. *arXiv preprint arXiv:2011.05348*.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. 2019. You only propagate once: Accelerating adversarial training via maximal principle. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 227–238.

Zhiyuan Zhang, Ruixuan Luo, Xuancheng Ren, Qi Su, Liangyou Li, and Xu Sun. 2021. Adversarial parameter defense by multi-step risk minimization. *Neural Networks*, 144:154–163.

Yaowei Zheng, Richong Zhang, and Yongyi Mao. 2020. Regularizing neural networks via adversarial model perturbation. *CoRR*, abs/2010.04925.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for language understanding. *CoRR*, abs/1909.11764.

A Theoretical Details

A.1 Proofs of Theorem 1

Theorem 1. *The distribution shifts of datasets $\mathcal{D}$ and $\mathcal{D}^*$ can be equivalently treated as a parameter corruption near the corresponding minima,*

$$\mathcal{L}(\boldsymbol{\theta}^* + \mathbf{v}; \mathcal{D}^*) \approx \mathcal{L}(\boldsymbol{\theta} + \mathbf{v}; \mathcal{D}) + \textit{Constant}, \tag{19}$$

where Constant=$\mathcal{L}(\boldsymbol{\theta}^*; \mathcal{D}^*) - \mathcal{L}(\boldsymbol{\theta}; \mathcal{D})$. Let $\mathbf{a} = -\boldsymbol{\delta}$, when $\mathbf{w}$ is near $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$, we have

$$\mathcal{L}(\mathbf{w}; \mathcal{D}^*) \approx \mathcal{L}(\mathbf{w} + \mathbf{a}; \mathcal{D}) + \textit{Constant}. \tag{20}$$

Proof. Define $f(\mathbf{v}) = \mathcal{L}(\boldsymbol{\theta}^* + \mathbf{v}; \mathcal{D}^*) - \mathcal{L}(\boldsymbol{\theta} + \mathbf{v}; \mathcal{D})$, namely $\mathcal{L}(\boldsymbol{\theta}^* + \mathbf{v}; \mathcal{D}^*) = \mathcal{L}(\boldsymbol{\theta} + \mathbf{v}; \mathcal{D}) + f(\mathbf{v})$. First we prove there exists $C_H = \rho C_f^{\frac{1}{2}} + LC_{\boldsymbol{\delta}} = o(1)$ such that $\|\boldsymbol{H}^* - \boldsymbol{H}\|_2 \leq C_H$, where $\rho = \mathbb{E}_{p(\mathbf{z})}[\|\nabla^2_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{z})\|_2^2]^{\frac{1}{2}}$, and $\|\nabla^2_{\boldsymbol{\theta}^*} \ell(\boldsymbol{\theta}^*; \mathbf{z}) - \nabla^2_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{z})\|_2 \leq L\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|_2$. We have,

$$\begin{aligned}
&\|\boldsymbol{H}^* - \boldsymbol{H}\|_2 && (21)\\
&= \Big\| \int_{\mathbf{z}} \big\{ \big(p^*(\mathbf{z}) - p(\mathbf{z})\big) \nabla^2_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{z}) && (22)\\
&+ p^*(\mathbf{z}) (\nabla^2_{\boldsymbol{\theta}^*} \ell(\boldsymbol{\theta}^*; \mathbf{z}) - \nabla^2_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{z}) \big\} d\mathbf{z} \Big\|_2 && (23)\\
&\leq \Big\| \mathbb{E}_{p(\mathbf{z})} \big[\big(\frac{p^*(\mathbf{z})}{p(\mathbf{z})} - 1 \big) \nabla^2_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{z}) \big] \Big\|_2 && (24)\\
&+ \Big\| \mathbb{E}_{p^*(\mathbf{z})} \big[\nabla^2_{\boldsymbol{\theta}^*} \ell(\boldsymbol{\theta}^*; \mathbf{z}) - \nabla^2_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{z}) \big] \Big\|_2 && (25)\\
&\leq \Big\| \mathbb{E}_{p(\mathbf{z})} \big[r(\mathbf{z}) \nabla^2_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{z}) \big] \Big\|_2 + L\|\boldsymbol{\delta}\|_2 && (26)\\
&\leq \mathbb{E}_{p(\mathbf{z})} [|r(\mathbf{z})|^2]^{\frac{1}{2}} \mathbb{E}_{p(\mathbf{z})} \big[\|\nabla^2_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{z})\|_2^2 \big]^{\frac{1}{2}} && (27)\\
&+ LC_{\boldsymbol{\delta}} && (28)\\
&= \rho C_f^{\frac{1}{2}} + LC_{\boldsymbol{\delta}} = C_H. && (29)
\end{aligned}$$

Consider the gradients,

$$\begin{aligned}
&\nabla_{\mathbf{v}} f(\mathbf{v}) && (30)\\
&= \nabla_{\mathbf{v}} \mathcal{L}(\boldsymbol{\theta}^* + \mathbf{v}; \mathcal{D}^*) - \nabla_{\mathbf{v}} \mathcal{L}(\boldsymbol{\theta} + \mathbf{v}; \mathcal{D}) && (31)\\
&= (\boldsymbol{H}^* - \boldsymbol{H})\mathbf{v} + o(\|\mathbf{v}\|_2), && (32)\\
&\nabla_{\mathbf{v}} \mathcal{L}(\boldsymbol{\theta} + \mathbf{v}; \mathcal{D}) = \boldsymbol{H}\mathbf{v} + o(\|\mathbf{v}\|_2), && (33)\\
&\frac{\|\nabla_{\mathbf{v}} f(\mathbf{v})\|_2}{\|\nabla_{\mathbf{v}} \mathcal{L}(\boldsymbol{\theta} + \mathbf{v}; \mathcal{D})\|_2} && (34)\\
&= \frac{\|(\boldsymbol{H}^* - \boldsymbol{H})\mathbf{v}\|_2 + o(\|\mathbf{v}\|_2)}{\|\boldsymbol{H}\mathbf{v}\|_2 + o(\|\mathbf{v}\|_2)} && (35)\\
&\leq \frac{C_H + o(1)}{\mu + o(1)} = o(1). && (36)
\end{aligned}$$

Consider the function f, $\nabla_{\mathbf{v}} f(\mathbf{0}) = \mathbf{0}, \nabla^2_{\mathbf{v}} f(\mathbf{0}) = \boldsymbol{H}^* - \boldsymbol{H}$,

$$\begin{aligned}
&|f(\mathbf{v}) - f(\mathbf{0})| && (37)\\
&= \frac{1}{2} \mathbf{v}^{\mathrm{T}} (\nabla^2_{\mathbf{v}} f(\mathbf{0})) \mathbf{v} + o(\|\mathbf{v}\|_2^2) && (38)\\
&\leq \frac{1}{2} C_H \|\mathbf{v}\|_2^2 + o(\|\mathbf{v}\|_2^2), && (39)\\
&|\mathcal{L}(\boldsymbol{\theta} + \mathbf{v}; \mathcal{D}) - \mathcal{L}(\boldsymbol{\theta}; \mathcal{D})| && (40)\\
&= |\mathbf{v}^{\mathrm{T}} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) + \frac{1}{2} \mathbf{v}^{\mathrm{T}} \boldsymbol{H} \mathbf{v}| + o(\|\mathbf{v}\|_2^2) && (41)\\
&\geq \frac{1}{2} \mu \|\mathbf{v}\|_2^2 + o(\|\mathbf{v}\|_2^2), && (42)\\
&\frac{|f(\mathbf{v}) - f(\mathbf{0})|}{|\mathcal{L}(\boldsymbol{\theta} + \mathbf{v}; \mathcal{D}) - \mathcal{L}(\boldsymbol{\theta}; \mathcal{D})|} && (43)\\
&\leq \frac{\frac{1}{2} C_H \|\mathbf{v}\|_2^2 + o(\|\mathbf{v}\|_2^2)}{\frac{1}{2} \mu \|\mathbf{v}\|_2^2 + o(\|\mathbf{v}\|_2^2)} && (44)\\
&= \frac{C_H + o(1)}{\mu + o(1)} = o(1). && (45)
\end{aligned}$$

Therefore, the change in the term $f(\mathbf{v})$ in the loss function can be omitted compared to the change in the loss function,

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}^* + \mathbf{v}; \mathcal{D}^*) && (46)\\
&= \mathcal{L}(\boldsymbol{\theta} + \mathbf{v}; \mathcal{D}) + f(\mathbf{v}) && (47)\\
&\approx \mathcal{L}(\boldsymbol{\theta} + \mathbf{v}; \mathcal{D}) + f(\mathbf{0}) && (48)\\
&= \mathcal{L}(\boldsymbol{\theta} + \mathbf{v}; \mathcal{D}) + \text{Constant}. && (49)
\end{aligned}$$

Let $\mathbf{a} = -\boldsymbol{\delta}, \mathbf{w} = \boldsymbol{\theta}^* + \mathbf{v}$, we have,

$$\mathcal{L}(\mathbf{w}; \mathcal{D}^*) \approx \mathcal{L}(\mathbf{w} + \mathbf{a}; \mathcal{D}) + \text{Constant}. \tag{50}$$

□

A.2 Proofs of Theorem 2

Theorem 2. *Define $r(\mathbf{z}) = p^*(\mathbf{z})/p(\mathbf{z}) - 1$. When the distribution shift is small enough, namely $r(\mathbf{z})$*

is small, we can estimate the parameter shift $\boldsymbol{\delta}$ as,

$$\boldsymbol{\delta} = -\boldsymbol{H}^{-1}\mathbb{E}_p[r(\mathbf{z})\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})] + o(\|\boldsymbol{\delta}\|_2). \quad (51)$$

Proof. With the change-of-measure technique, we have

$$\mathbb{E}_{p(\mathbf{z})}\big[r(\mathbf{z})\big] = 0. \quad (52)$$

According to the definition,

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta};\mathcal{D}) = \nabla_{\boldsymbol{\theta}^*}\mathcal{L}(\boldsymbol{\theta}^*;\mathcal{D}^*) = \mathbf{0}. \quad (53)$$

Conduct Taylor Expansion, we have,

$$\begin{aligned}
\mathbf{0} &= \nabla_{\boldsymbol{\theta}^*}\mathcal{L}(\boldsymbol{\theta}^*;\mathcal{D}^*) - \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta};\mathcal{D}) && (54)\\
&= \nabla_{\boldsymbol{\theta}^*}\mathcal{L}(\boldsymbol{\theta}^*;\mathcal{D}^*) - && (55)\\
&\big(\nabla_{\boldsymbol{\theta}^*}\mathcal{L}(\boldsymbol{\theta}^*;\mathcal{D}) - \boldsymbol{H}(-\boldsymbol{\delta}) + o(\|\boldsymbol{\delta}\|_2)\big). && (56)
\end{aligned}$$

Solve it, we have,

$$\begin{aligned}
\boldsymbol{\delta} &= -\boldsymbol{H}^{-1}\big(\nabla_{\boldsymbol{\theta}^*}\mathcal{L}(\boldsymbol{\theta}^*;\mathcal{D}^*) && (57)\\
&\quad - \nabla_{\boldsymbol{\theta}^*}\mathcal{L}(\boldsymbol{\theta}^*;\mathcal{D})\big) + o(\|\boldsymbol{\delta}\|_2). && (58)
\end{aligned}$$

Consider $\nabla_{\boldsymbol{\theta}^*}\mathcal{L}(\boldsymbol{\theta}^*;\mathcal{D}^*) - \nabla_{\boldsymbol{\theta}^*}\mathcal{L}(\boldsymbol{\theta}^*;\mathcal{D})$,

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}^*}\mathcal{L}(\boldsymbol{\theta}^*;\mathcal{D}^*) - \nabla_{\boldsymbol{\theta}^*}\mathcal{L}(\boldsymbol{\theta}^*;\mathcal{D}) && (59)\\
&= \int_{\mathbf{z}} \big(p^*(\mathbf{z}) - p(\mathbf{z})\big)\nabla_{\boldsymbol{\theta}^*}\ell(\boldsymbol{\theta}^*;\mathbf{z})d\mathbf{z} && (60)\\
&= \int_{\mathbf{z}} \big(\frac{p^*(\mathbf{z})}{p(\mathbf{z})} - 1\big)p(\mathbf{z})\nabla_{\boldsymbol{\theta}^*}\ell(\boldsymbol{\theta}^*;\mathbf{z})d\mathbf{z} && (61)\\
&= \mathbb{E}_{p(\mathbf{z})}[r(\mathbf{z})\nabla_{\boldsymbol{\theta}^*}\ell(\boldsymbol{\theta}^*;\mathbf{z})] && (62)\\
&= \mathbb{E}_{p(\mathbf{z})}[r(\mathbf{z})\big(\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z}) + && (63)\\
&\nabla^2_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\boldsymbol{\delta} + o(\|\boldsymbol{\delta}\|_2)\big)] && (64)\\
&= \mathbb{E}_{p(\mathbf{z})}[r(\mathbf{z})\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})] + o(\|\boldsymbol{\delta}\|_2), && (65)
\end{aligned}$$

where the term $r(\mathbf{z})\nabla^2_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\boldsymbol{\delta} = o(\|\boldsymbol{\delta}\|_2)$ since the distribution shift $r(\mathbf{z}) = o(1)$. To conclude,

$$\boldsymbol{\delta} = -\boldsymbol{H}^{-1}\mathbb{E}_p[r(\mathbf{z})\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})] + o(\|\boldsymbol{\delta}\|_2). \quad (66)$$

□

A.3 Proofs of Propositions

Proposition 1. *Suppose the loss $\mathcal{L}(\mathbf{w};\mathcal{D})$ is μ-strongly convex*[3]*, and $D_f(p^*||p) \le C_f$, there exists*

$$C_{\boldsymbol{\delta}} = \frac{1+o(1)}{\mu}\sqrt{\frac{C_f}{a_2}\mathbb{E}_{p(\mathbf{z})}\big[\|\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\|_2^2\big]} \quad (67)$$

such that $\|\boldsymbol{\delta}\|_2 \le C_{\boldsymbol{\delta}}$, namely $C_{\boldsymbol{\delta}}$ is a upper bound.

[3]Note that $\mathcal{L}$ is only required to be μ-strongly convex in the neighborhood of the loss basin including $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$, instead of the entire $\mathbb{R}^n$.

Proof. With the change-of-measure technique, we have:

$$\mathbb{E}_{p(\mathbf{z})}\big[r(\mathbf{z})\big] = 0. \quad (68)$$

Then

$$\begin{aligned}
D_f(p^*||p) &= \mathbb{E}_{p(\mathbf{z})}\big[f(1+r(\mathbf{z}))\big] && (69)\\
&= \mathbb{E}_{p(\mathbf{z})}\big[a_1 r + a_2|r|^2 + o(r)^2\big] && (70)\\
&= (a_2 + o(1))\mathbb{E}_{p(\mathbf{z})}\big[|r|^2\big] \le C_f. && (71)
\end{aligned}$$

Therefore,

$$\mathbb{E}_{p(\mathbf{z})}\big[|r|^2\big] \le \frac{(1+o(1))C_f}{a_2}. \quad (72)$$

According to the upper bound of $\mathbb{E}_{p(\mathbf{z})}\big[|r|^2\big]$,

$$\begin{aligned}
&\big\|\mathbb{E}_{p(\mathbf{z})}\big[r(\mathbf{z})\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\big]\big\|_2 && (73)\\
&\le \mathbb{E}_{p(\mathbf{z})}\big[|r(\mathbf{z})|^2\big]^{\frac{1}{2}}\mathbb{E}_{p(\mathbf{z})}\big[\|\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\|^2\big]^{\frac{1}{2}} && (74)\\
&\le \sqrt{\frac{(1+o(1))C_f}{a_2}\mathbb{E}_{p(\mathbf{z})}\big[\|\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\|^2\big]}. && (75)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mu\|\boldsymbol{\delta}\|_2 \le \|\boldsymbol{H}\boldsymbol{\delta}\|_2 && (76)\\
&= \big\|\mathbb{E}_{p(\mathbf{z})}\big[r(\mathbf{z})\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\big]\big\|_2 && (77)\\
&\le \sqrt{\frac{(1+o(1))C_f}{a_2}\mathbb{E}_{p(\mathbf{z})}\big[\|\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\|^2\big]}. && (78)
\end{aligned}$$

There exists

$$C_{\boldsymbol{\delta}} = \frac{1+o(1)}{\mu}\sqrt{\frac{C_f}{a_2}\mathbb{E}_{p(\mathbf{z})}\big[\|\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\|_2^2\big]} \quad (79)$$

such that $\|\boldsymbol{\delta}\|_2 \le C_{\boldsymbol{\delta}}$.

□

Proposition 2. *Suppose the mixed distribution of $\mathcal{D}^{mix}$ is $p^{mix} = (1-\eta)p + \eta p^*$, then we have $D_f(p^{mix}||p) \le C_f^{mix} = \eta^2 C_f$. Denote $\boldsymbol{\theta}^{mix}$ as the optimal parameter on $\mathcal{D}^{mix}$, $\boldsymbol{\delta}^{mix} = \boldsymbol{\theta}^{mix} - \boldsymbol{\theta}$, then we have:*

$$\frac{\|\boldsymbol{\delta}^{mix}\|_2}{\|\boldsymbol{\delta}\|_2} = \frac{C_{\boldsymbol{\delta}}^{mix}}{C_{\boldsymbol{\delta}}} = \eta + o(1). \quad (80)$$

Proof.

$$r^{\text{mix}}(\mathbf{z}) = \frac{p^{\text{mix}}(\mathbf{z})}{p(\mathbf{z})} - 1 = \eta \times r(\mathbf{z}). \quad (81)$$

□

Proposition 3. *Define the average gradient strength as* $G = \mathbb{E}_{p(\mathbf{z})}[\|\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\|_2]/\sqrt{n}$*, and the average gradient strength of group* i *as* $G_{(i)} = E_{p(\mathbf{z})}[\|\nabla_{\boldsymbol{\theta}_{(i)}}\ell(\boldsymbol{\theta};\mathbf{z})\|_2]/\sqrt{n_{(i)}}$*, then*

$$\|\boldsymbol{\delta}\|_2 \propto \frac{\sqrt{n}}{G}, \quad \|\boldsymbol{\delta}_{(i)}\|_2 \propto \frac{\sqrt{n_{(i)}}}{G_{(i)}}. \tag{82}$$

Proof. Suppose λ denotes the average values of eigenvalues of the Hessian matrix, according to the Fisher information matrix assumption (Pascanu and Bengio, 2013),

$$\boldsymbol{H} = \mathbb{E}_{p(\mathbf{z})}[\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})^{\mathrm{T}}], \tag{83}$$

$$\lambda = \frac{\mathrm{tr}(\boldsymbol{H})}{n} \tag{84}$$

$$= \frac{\mathbb{E}_{p(\mathbf{z})}[\mathrm{tr}(\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})^{\mathrm{T}})]}{n} \tag{85}$$

$$= \frac{\mathbb{E}_{p(\mathbf{z})}[\|\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})\|_2^2]}{n} \propto G^2. \tag{86}$$

We have,

$$\boldsymbol{H}\boldsymbol{\delta} = -\mathbb{E}_{p(\mathbf{z})}[r(\mathbf{z})\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})] + o(\|\boldsymbol{\delta}\|_2), \tag{87}$$

$$\|\boldsymbol{H}\boldsymbol{\delta}\|_2 \propto \lambda\|\boldsymbol{\delta}\|_2 \propto G^2\|\boldsymbol{\delta}\|_2, \tag{88}$$

$$\|\mathbb{E}_{p(\mathbf{z})}[r(\mathbf{z})\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{z})]\|_2 \propto \sqrt{n}G. \tag{89}$$

Therefore,

$$G^2\|\boldsymbol{\delta}\|_2 \propto \sqrt{n}G, \tag{90}$$

$$\|\boldsymbol{\delta}\|_2 \propto \frac{\sqrt{n}}{G}. \tag{91}$$

Assume $\partial^2\mathcal{L}(\boldsymbol{\theta};\mathcal{D})/(\partial\boldsymbol{\theta}_{(i)}\partial\boldsymbol{\theta}_{(j)}) = 0$ for $i \neq j$, namely $\boldsymbol{H} = \mathrm{diag}\{\boldsymbol{H}_{(1)}, \boldsymbol{H}_{(2)}, \cdots, \boldsymbol{H}_{(l)}\}$, where $\boldsymbol{H}_{(i)} = \mathbb{E}_{p(\mathbf{z})}[\nabla_{\boldsymbol{\theta}_{(i)}}\ell(\boldsymbol{\theta};\mathbf{z})\nabla_{\boldsymbol{\theta}_{(i)}}\ell(\boldsymbol{\theta};\mathbf{z})^{\mathrm{T}}]$, then

$$\boldsymbol{H}_{(i)}\boldsymbol{\delta}_{(i)} = -\mathbb{E}_{p(\mathbf{z})}[r(\mathbf{z})\nabla_{\boldsymbol{\theta}_{(i)}}\ell] + o(\|\boldsymbol{\delta}\|_2), \tag{92}$$

$$\|\boldsymbol{\delta}_{(i)}\|_2 \propto \frac{\sqrt{n_{(i)}}}{G_{(i)}}. \tag{93}$$

□

A.4 Details of Multi-step Implementation

Zhang et al. (2021) give the close-formed solution of $\mathbf{u}_{k+1}$ under the constraint $\|\mathbf{u}\|_p = \eta$,

$$\mathbf{u}_{k+1} = \arg\max_{\|\mathbf{u}\|_p=\eta} \mathbf{g}_k^{\mathrm{T}}\mathbf{u} \tag{94}$$

$$= \eta\left(\mathrm{sgn}(\mathbf{g_k}) \odot \frac{|\mathbf{g}_k|^{\frac{1}{p-1}}}{\||\mathbf{g}_k|^{\frac{1}{p-1}}\|_p}\right). \tag{95}$$

When the constraint is $S = \{\mathbf{u} : \|\boldsymbol{T}^{-1}\mathbf{u}\|_p = \eta\}$, $\mathbf{g}_k^{\mathrm{T}}\mathbf{u} = (\boldsymbol{T}\mathbf{g}_k^{\mathrm{T}})(\boldsymbol{T}^{-1}\mathbf{u})$, namely

$$\boldsymbol{T}^{-1}\mathbf{u}_{k+1} = \frac{\eta\big(\mathrm{sgn}(\boldsymbol{T}\mathbf{g}_k)\big) \odot |\boldsymbol{T}\mathbf{g}_k|^{\frac{1}{p-1}}}{\||\boldsymbol{T}\mathbf{g}_k|^{\frac{1}{p-1}}\|_p}. \tag{96}$$

Therefore,

$$\mathbf{u}_{k+1} = \frac{\eta\big(\boldsymbol{T}\mathrm{sgn}(\mathbf{g}_k)\big) \odot |\boldsymbol{T}\mathbf{g}_k|^{\frac{1}{p-1}}}{\||\boldsymbol{T}\mathbf{g}_k|^{\frac{1}{p-1}}\|_p}. \tag{97}$$

To get the k-th corruption $\mathbf{a}_k$, we project the updated corruption $\mathbf{a}_{k-1}+\mathbf{u}_k$ into the set S. Zhang et al. (2021) give the projecting functions,

$$\Pi_S(\mathbf{v}) = \min\{\|\mathbf{v}\|_2, \epsilon\}\frac{\mathbf{v}}{\|\mathbf{v}\|_2} \quad (p=2); \tag{98}$$

$$\Pi_S(\mathbf{v}) = \mathrm{clip}(\mathbf{v}, -\epsilon, \epsilon) \quad (p=+\infty); \tag{99}$$

similarly, when $S = \{\mathbf{u} : \|\boldsymbol{T}^{-1}\mathbf{u}\|_p \leq \epsilon\}$, we may assume $\boldsymbol{T}^{-1}\Pi_S(\mathbf{v}) = \Pi_S(\boldsymbol{T}^{-1}\mathbf{v})$. Suppose $\mathbf{x} = \boldsymbol{T}^{-1}\mathbf{v}$, we have,

$$\boldsymbol{T}^{-1}\Pi_{\|\mathbf{x}\|_2\leq\epsilon}(\mathbf{v}) = \min\{\|\mathbf{x}\|_2, \epsilon\}\frac{\mathbf{x}}{\|\mathbf{x}\|_2}; \tag{100}$$

$$\boldsymbol{T}^{-1}\Pi_{\|\mathbf{x}\|_{+\infty}\leq\epsilon}(\mathbf{v}) = \mathrm{clip}(\mathbf{x}, -\epsilon, \epsilon). \tag{101}$$

Therefore,

$$\Pi_{\|\boldsymbol{T}^{-1}\mathbf{v}\|_2\leq\epsilon}(\mathbf{v}) = \frac{\min\{\|\boldsymbol{T}^{-1}\mathbf{v}\|_2, \epsilon\}\mathbf{v}}{\|\boldsymbol{T}^{-1}\mathbf{v}\|_2}; \tag{102}$$

$$\Pi_{\|\boldsymbol{T}^{-1}\mathbf{v}\|_{+\infty}\leq\epsilon}(\mathbf{v}) = \boldsymbol{T}\mathrm{clip}(\boldsymbol{T}^{-1}\mathbf{v}, -\epsilon, \epsilon). \tag{103}$$

B Datasets and Baseline Implementations

In this section, we introduce the datasets and baseline implementations. The models enhanced with SAM algorithms adopt the same hyper-parameters to baselines except for the SAM hyper-parameter settings $(K, T_{(i)}, \epsilon, L_p)$. All experiments are conducted on NVIDIA TITAN RTX GPUs. We conduct every experiment for 4 runs and report the mean and validation results.

B.1 IMDB

We implement the base model TextCNN (Kim, 2014) on the IMDb movie reviews dataset (**IMDB**) (Maas et al., 2011). The reviews are classified into 3 classes, namely the positive reviews, negative reviews, and neutral reviews. The training size is 25000, the validation size is 25000, and the test size is 50000. In the preprocessing of the text, the vocab size is 30000, and the text length is 200.

The embedding size is 500. In TextCNN, the filter window sizes are 3, 4, and 5, with 500 feature maps each. The optimizer is Adam with a learning rate of 10^{-3} and a batch size of 64. We train models for 10 epochs and test the accuracy on the checkpoint with the best valid accuracy.

B.2 PTB-LM

We implement a 2-layer LSTM as a language model following Merity et al. (2017) on the word-level Penn TreeBank dataset (PTB)[4](Marcus et al., 1993). In the preprocessing of the text, the vocab size is 10000. We use the SGD optimizer with an initial learning rate of 50 and a gradient norm clip of 0.25. We adopt a learning rate decay of 0.5 after 10 epochs. The input and output embeddings are tied. The embedding size is 500, and the hidden size is 500. We train models for 20 epochs and test the perplexity on the checkpoint with the lowest valid perplexity.

B.3 En-Vi

We implement the transformer model following the fairseq Ott et al. (2019) "transformer_wmt_en_de" implementation as our baseline model on the En-Vi dataset, which is provided by the IWSLT 2015 Evaluation Campaign (Cettolo et al., 2015). The training size is 133K, the validation set is TED tst2012 with a size of 1553 sentences, and the test set is TED tst2013 with a size of 1268 sentences. In the preprocessing of the text, the English and Vietnamese vocabulary sizes are 17200 and 7800 respectively. We use the same hyper-parameters following fairseq (Ott et al., 2019). We train the model for 52 epochs and adopt a checkpoint average of 10. In testing, We adopt the BLEU metric and the beam size of the model inference is 5.

B.4 De-En

We implement the transformer model following the fairseq Ott et al. (2019) "transformer_wmt_en_de" implementation as our baseline model on the De-En dataset, which is provided by the IWSLT 2014 Evaluation Campaign (Cettolo et al., 2014). We use the same dataset splits and the same hyper-parameters following fairseq (Ott et al., 2019). The training, validation and test sizes are 153K, 7K, and 7K, respectively. In the preprocessing of the text, we adopt the BPE technique, and the German and English vocabulary sizes are 8848 and 6632 respectively. We train the model for 70 epochs and adopt a checkpoint average of 10. In testing, We adopt the BLEU metric and the beam size of the model inference is 5.

[4]Dataset is available at https://www.kaggle.com/nltkdata/penn-tree-bank?select=ptb

C Experimental Settings

In this section, we report experimental settings in the paper, including analytic trial details and hyper-parameters of different algorithms.

C.1 Analytic Trial Settings

We conduct a 3-layer Multi-Layer Perceptrons (MLP, sizes are $784, 100, 100, 10$) on the MNIST dataset (LeCun et al., 1998). The test minimum $\boldsymbol{\theta}^*$ is fine-tuned from the training minimum $\boldsymbol{\theta}$, and $\boldsymbol{\delta} = \boldsymbol{\theta}^* - \boldsymbol{\theta}$. We utilize linear interpolation to plot $\alpha\boldsymbol{\theta} + (1-\alpha)\boldsymbol{\theta}^*$ for different α, for visualizing the training loss, test loss and the shifted training loss. It can be concluded that the shifted training loss is similar to the test loss near the local minimum. To visualize the relation between $\|\boldsymbol{\delta}\|_2$ and the distribution shift. We conduct 100 trials with the same training set $\mathcal{D}$ and different mixed test sets $\mathcal{D}^{\text{mix}}$ (mixed with $(1-\eta)$ of the training data from $\mathcal{D}$ and η of the test data from $\mathcal{D}^*$). Here the test minimum is fine-tuned from the training minimum and $\boldsymbol{\delta} = \boldsymbol{\theta}^* - \boldsymbol{\theta}$. η can be utilized to measure the strength of the distribution shift between $\mathcal{D}$ and $\mathcal{D}^{\text{mix}}$. $\eta = 1\%, 2\%, 3\%, \cdots, 99\%, 100\%$. Results show that there exists an approximately linear relationship between $\|\boldsymbol{\delta}\|_2$ and distribution shift strengths.

C.2 Main Result Settings

We try both L_2 and $L_{+\infty}$. For ϵ, we first binary search proper order of magnitude, for example 10^{-5} to 10^{-4}, then we grid search ϵ, for example, $\{1\times10^{-5}, 2\times10^{-5}, 5\times10^{-5}, 8\times10^{-5}, 1\times10^{-4}, 2\times10^{-4}, 5\times10^{-4}, 8\times10^{-4}\}$.

SAM on Transformer Models. The Transformer models need larger K and more detailed hyper-parameter tuning since SAM learning on the Transformer is unstable. Besides, Zhang et al. (2021) propose that SAM learning in the early stage may harm the learning. In our GA-SAM, $\|\mathbf{g}\|_2$ is large and we can omit the SAM learning in the early stage. Therefore, we adopt $K = 2$ in multi-step implementations, and following Zhang et al. (2021), we adopt a start epoch of 30 for Transformer models.

C.2.1 IMDB

On the IMDB dataset, we adopt $K = 1$. On SAM, $L_{+\infty}, \epsilon = 2 \times 10^{-5}$. On ASAM, $L_{+\infty}, \epsilon = 5 \times 10^{-3}$. On Layer-SAM, $L_2, \epsilon = 1 \times 10^{-5}$. On Multi-step Defense, $L_{+\infty}, \epsilon = 2 \times 10^{-5}$. On GA-SAM, $L_{+\infty}, \epsilon = 1 \times 10^{-4}$.

C.2.2 PTB-LM

On the PTB-LM dataset, we adopt $K = 1$. On SAM, $L_2, \epsilon = 5 \times 10^{-3}$. On ASAM, $L_{+\infty}, \epsilon = 5 \times 10^{-4}$. On Layer-SAM, $L_2, \epsilon = 1 \times 10^{-6}$. On Multi-step Defense, $L_{+\infty}, \epsilon = 1 \times 10^{-3}$. On GA-SAM, $L_{+\infty}, \epsilon = 8 \times 10^{-6}$.

C.2.3 En-Vi

On the En-Vi dataset, we adopt $K = 2$ in multi-step implementations, and following Zhang et al. (2021), we adopt a start epoch of 30 and similar hyper-parameters. On SAM, $L_{+\infty}, \epsilon = 1.2 \times 10^{-3}$. On ASAM, $L_{+\infty}, \epsilon = 5 \times 10^{-4}$. On Layer-SAM, $L_{+\infty}, \epsilon = 1 \times 10^{-3}$. On Multi-step Defense, $L_{+\infty}, \epsilon = 5 \times 10^{-4}$. On GA-SAM, $L_{+\infty}, \epsilon = 0.9$.

C.2.4 De-En

On the De-En dataset, we adopt $K = 2$ in multi-step implementations, and following Zhang et al. (2021), we adopt a start epoch of 30 and similar hyper-parameters. On SAM, $L_{+\infty}, \epsilon = 5 \times 10^{-4}$. On ASAM, $L_{+\infty}, \epsilon = 2 \times 10^{-4}$. On Layer-SAM, $L_{+\infty}, \epsilon = 3 \times 10^{-3}$. On Multi-step Defense, $L_{+\infty}, \epsilon = 5 \times 10^{-4}$. On GA-SAM, $L_{+\infty}, \epsilon = 0.8$.

C.3 Ablation Study Settings

In this section, we report hyper-parameter settings in the ablation study.

C.3.1 IMDB

On the IMDB dataset. On single-step implementation, $L_{+\infty}, \epsilon = 5 \times 10^{-5}$. On element-wise implementation, $L_{+\infty}, \epsilon = 5 \times 10^{-6}$. On model-wise implementation, $L_{+\infty}, \epsilon = 5 \times 10^{-5}$. For $T_{(i)} = \|\mathbf{w}_{(i)}\|_2 / \|\mathbf{g}_{(i)}\|_2$, $L_{+\infty}, \epsilon = 2 \times 10^{-5}$. For $T_{(i)} = 1/\|\mathbf{g}_{(i)}\|_2$, $L_{+\infty}, \epsilon = 5 \times 10^{-5}$. For $T_{(i)} = \|\mathbf{w}_{(i)}\|_2 / \sqrt{n_{(i)}}$, $L_{+\infty}, \epsilon = 5 \times 10^{-3}$. For $T_{(i)} = \|\mathbf{w}_{(i)}\|_2$, $L_{+\infty}, \epsilon = 5 \times 10^{-6}$.

C.3.2 PTB-LM

On the PTB-LM dataset. On single-step implementation, $L_2, \epsilon = 5 \times 10^{-3}$. On element-wise implementation, $L_{+\infty}, \epsilon = 2 \times 10^{-6}$. On model-wise implementation, $L_{+\infty}, \epsilon = 5 \times 10^{-6}$. For $T_{(i)} = \|\mathbf{w}_{(i)}\|_2 / \|\mathbf{g}_{(i)}\|_2$, $L_2, \epsilon = 2 \times 10^{-5}$. For $T_{(i)} = 1/\|\mathbf{g}_{(i)}\|_2$, $L_{+\infty}, \epsilon = 5 \times 10^{-6}$. For $T_{(i)} = \|\mathbf{w}_{(i)}\|_2 / \sqrt{n_{(i)}}$, $L_{+\infty}, \epsilon = 1 \times 10^{-4}$. For $T_{(i)} = \|\mathbf{w}_{(i)}\|_2$, $L_{+\infty}, \epsilon = 1 \times 10^{-7}$.

C.4 Adversarial Training Settings

In this section, we report hyper-parameter settings in the study of adversarial training.

C.4.1 En-Vi

On the En-Vi dataset, we adopt $K = 1$ and adopt a start epoch of 30. We adopt the $L_{+\infty}$ constraint on virtual attacks on word embeddings and grid search ϵ in $\{$ 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5 $\}$. The best configuration is $\epsilon = 0.02$.

C.4.2 De-En

On the De-En dataset, we adopt $K = 3$ and adopt a start epoch of 30. We adopt the $L_{+\infty}$ constraint on virtual attacks on word embeddings and grid search ϵ in $\{$ 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5 $\}$. The best configuration is $\epsilon = 0.005$.