Improving Sharpness-Aware Minimization with Fisher Mask for Better Generalization on Language Models

Qihuang Zhong1, Liang Ding2, Li Shen2, Peng Mi3, Juhua Liu4, Bo Du1, Dacheng Tao2

1 National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence, School of Computer Science and Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, China
2 JD Explore Academy, China
3 School of Informatics, Xiamen University, China
4 Research Center for Graphic Communication, Printing and Packaging, and Institute of Artificial Intelligence, Wuhan University, China

Abstract

Fine-tuning large pretrained language models on a limited training corpus usually suffers from poor generalization. Prior works show that the recently-proposed sharpness-aware minimization (SAM) optimization method can improve the model generalization. However, SAM adds a perturbation to each model parameter equally (but not all parameters contribute equally to the optimization of training), which we argue is sub-optimal and will lead to excessive computation. In this paper, we propose a novel optimization procedure, namely FSAM, which introduces a Fisher mask to improve the efficiency and performance of SAM. In short, instead of adding perturbation to all parameters, FSAM uses the Fisher information to identify the important parameters and formulates a Fisher mask to obtain the sparse perturbation, i.e., making the optimizer focus on these important parameters. Experiments on various tasks in GLUE and SuperGLUE benchmarks show that FSAM consistently outperforms the vanilla SAM by 0.67~1.98 average score among four different pretrained models. We also empirically show that FSAM works well in other complex scenarios, e.g., fine-tuning on generation tasks or limited training data. Encouragingly, when training data is limited, FSAM improves the SAM by a large margin, i.e., up to 15.1.

1 Introduction

The “pretraining-finetuning” paradigm has become the de facto standard for the community of natural language processing (NLP) (Devlin et al., 2019; Liu et al., 2019; Clark et al., 2019b; Raffel et al., 2020; Brown et al., 2020; Lewis et al., 2020). Given a pretrained language model (PLM), the dominant fine-tuning manner is tuning the entire pretrained parameters for each downstream task (Radford et al., 2018; Devlin et al., 2019). While fine-tuning the entire PLM can improve performance on a wide range of NLP tasks, it usually suffers from overfitting and poorer generalization ability (Xu et al., 2021; Bahri et al., 2022), especially in the large-scale PLMs and limited training data scenarios.

Hence, some existing efforts attempt to provide more regularization in the fine-tuning stage (Zhang et al., 2018; Müller et al., 2019; Xu et al., 2021), among which the optimization of the training loss is an intuitive and effective method. Specifically, motivated by the finding (Keskar et al., 2016; Neyshabur et al., 2017) that the smoother loss landscape refers to the better model generalization, Foret et al. (2020) propose the “sharpness-aware minimization” (SAM) to simultaneously minimize loss value and loss sharpness, where the sharpness can be quantified as the maximized difference of loss when a perturbation is added to the current weights. In practice, SAM performs two forward-backward computations for each optimization step, where the first forward-backward is to obtain the perturbation for each model parameter and the second one is to update the parameters. Many prior works (Wu et al., 2020; Zheng et al., 2021) show the effectiveness of SAM in the vision domain, motivated by this, Bahri et al. (2022) first apply the SAM to the language domain, more recently.

Although Bahri et al. (2022) empirically show the remarkable performance of SAM on several language understanding tasks, SAM calculates perturbations indiscriminately for all parameters, which is time-consuming and hinders the application of SAM. Furthermore, inspired by the finding (Keskar et al., 2016) that only about 5% of parameters are sharp and rise steeply during optimization, we notice that not all parameters contribute equally to the optimization of training. Hence, this raises a question that whether we can calculate perturbations for only some individual parameters, and
Thus make the optimizer focus on these important parameters.

To this end, we propose a novel optimization approach, Fisher SAM (FSAM), which introduces a Fisher mask to improve the efficiency and effectiveness of SAM. In short, FSAM first uses the Fisher information (Fisher, 1922) as the metric to identify the sharper parameters and formulates a binary Fisher mask correspondingly. Then, the Fisher mask is multiplied with the perturbations to obtain the sparse perturbations, which are lastly used to perform regularization in the parameter update. In this way, only parts of sharper parameters will be added into the perturbations, and the optimizer can thus focus more on these important parameters. Also, the sparse perturbations could ensure the training acceleration via sparse back-propagation. Moreover, one may concern that the sparse Fisher mask would affect the convergence rate of FSAM (Lin et al., 2019). Hence, we theoretically provide the convergence analysis of FSAM, ensuring that the convergence of FSAM is irrelevant to the Fisher mask.

We conduct a large-scale and systematic study to evaluate the performance and effectiveness of FSAM. Firstly, we apply SAM and FSAM to fine-tune various PLMs on parts of GLUE and SuperGLUE benchmarks, where the results show that FSAM consistently outperforms the vanilla SAM by 0.67−1.98 average score among these PLMs, and surpasses the Adam (Kingma and Ba, 2015) optimizer by 1.41−1.91 points. Secondly, we conduct experiments on two popular generation tasks (i.e., XSUM and CoNLL2014) and prove that FSAM can deliver promising results against SAM. Lastly, quantitative analysis and in-depth discussion demonstrate the universality and effectiveness of FSAM in various complex scenarios, and prove that FSAM indeed brings better model generalization. Specifically, we show that our Fisher mask strategy not only works well in the SAM, but also can be applied to other SAM variants.

To summarize, our contributions are two-fold: (1) We propose a novel optimization approach (namely FSAM) with theoretical convergence guarantee for PLMs. Specifically, FSAM improves the performance and efficiency of recently-proposed SAM via a Fisher mask strategy, which can also be applied to more SAM variants. (2) Extensive experiments show that FSAM consistently outperforms the SAM by a large margin on both language understanding and generation tasks. The systematic study demonstrates the effectiveness and universality of FSAM on improving model generalization.

2 Related Work

SAM and its variants. Hochreiter and Schmidhuber (1994) first show the strong correlation between the flat minima and the generalization of a model, inspired by this, Foret et al. (2020) propose the SAM to find a flat minimum and thus improve model generalization. While many existing works prove the effectiveness of SAM on various computer vision tasks (Wu et al., 2020; Chen et al., 2021; Zheng et al., 2021), the double forward-propagation process of SAM brings more computational cost. To this end, Du et al. (2021) propose an Efficient SAM (ESAM) for reducing the computational cost of SAM. Additionally, there are also some efforts that focus on more efficient and effective SAM optimization (Zhuang et al., 2021; Kwon et al., 2021; Mi et al., 2022).

Improving Generalization. Recently, we have witnessed numerous PLMs that achieved tremendous success in the community of NLP (Yang et al., 2019; Devlin et al., 2019; Brown et al., 2020; Lewis et al., 2020; Raffel et al., 2020; Joshi et al., 2020; He et al., 2020; Qi et al., 2021; Zhong et al., 2022). The current dominant fine-tuning approach needs to tune all pretrained parameters for each downstream task, which makes the PLM easily memorize the training data and thus leads to overfitting. To tackle this issue, some works attempt to provide implicit and explicit regularization into the training of models, such as dropout (Srivastava et al., 2014), label smoothing (Müller et al., 2019), mixup (Zhang et al., 2018) and other data-augmentation methods (Sennrich et al., 2016; Wang et al., 2018b; Zhong et al., 2021; Wang et al., 2022; Ding et al., 2022). On the other hand, motivated by the successful applications of SAM in the vision domain, Bahri et al. (2022) involve applying SAM to optimize the T5 (Raffel et al., 2020) model on multiple language tasks and show that SAM can improve the generalization of PLMs effectively.

We depart from the prior work (Bahri et al.,
and ours as follows: 1) different motivations: instead of verifying the effect of vanilla SAM on several language understanding tasks, we aim to improve the efficiency and effectiveness of SAM. 2) different contributions: our main contribution is to propose a fisher mask strategy, which can be applied to both SAM and its variants. 3) more analysis: we provide more experimental results and analysis towards the effectiveness of our method in more complex scenarios.

3 Methodology

In this section, we first review the Sharpness-Aware Minimization, and then propose our Sharpness-Aware Minimization with Fisher mask, coined as FSAM. Finally, we theoretically analyze the convergence of FSAM with adaptive learning rate.

3.1 Sharpness-Aware Minimization

Preliminary. In this paper, we denote the weight of a neural network as \( w \in \mathbb{R}^d \). Suppose the training dataset \( S = \{(x_i, y_i)\}_{i=1}^n \) i.i.d. drawn from the distribution \( D \). The object function of the data \( x_i \) from \( S \) is denoted as \( f_S(x_i) \). Since the Adam (Kingma and Ba, 2015) and its variants are widely used in NLP tasks, the learning rate is estimated via RMSProp/Adam style.

Sharpness-Aware Minimization. For et al. (2020) propose the Sharpness-Aware Minimization (SAM) to improve the generalization, which is achieved by the following min-max problem:

\[
\min_w \max_{\|\epsilon\|_2 \leq \rho} f(w + \epsilon), \quad (1)
\]

where \( \rho \) is a predefined value to control the neighborhood size, and the \( \epsilon \) is the perturbation vector on model weight. The optimization is expected that the model loss will not significantly rise with a certain amount of weight change controlled by \( \rho \), which is intuitively consistent with the generalization capacity of model.

With the Taylor expansion, the perturbation vector \( \epsilon \) could be achieved approximately:

\[
\epsilon^* = \arg \max_{\|\epsilon\|_2 \leq \rho} f_S(w + \epsilon) \quad (2)
\]

\[
\approx \arg \max_{\|\epsilon\|_2 \leq \rho} f_S(w) + \epsilon \cdot \nabla_w f_S(w) \quad (3)
\]

\[
= \rho \cdot \nabla_w f_S(w) / \|\nabla_w f_S(w)\|_2, \quad (4)
\]

and the object function could be simplified as

\[
\min_w f(w + \rho \cdot \frac{\nabla_w f_S(w)}{\|\nabla_w f_S(w)\|_2}), \quad (5)
\]

The solution of the above function could be obtained by a two-step gradient descent. In the first gradient descent step, the perturbation vector \( \epsilon \) is calculated by Equation 2. The second gradient descent step is the actual weight update.

However, despite the improvement of SAM on many tasks, SAM requires a two-step gradient calculation which leads to the double overhead compared to the conventional optimizer, e.g., Stochastic Gradient Descent (SGD) and Adam.

3.2 Sharpness-Aware Minimization with Fisher Mask

In this subsection, we propose the Sharpness-Aware Minimization with Fisher Mask (FSAM) in detail, which reduces the computation of SAM by sparse calculation.

To be specific, we compute only a fraction of the elements in the perturbation vector \( \epsilon \), which would be multiplied by a sparse binary mask \( m \in \{0, 1\}^d \). To control the amount of perturbation, the sparse mask \( m \) satisfies \( 1^T m = (1-s) \cdot d \), where the \( s \) is the predefined sparse ratio and empirically set to 0.9. The objective function of FSAM is denoted as

\[
\min_w f_S(w + \rho \cdot \frac{\nabla_w f_S(w) \odot m}{\|\nabla_w f_S(w)\|_2}), \quad (6)
\]

where \( \odot \) is the Hadamard product, i.e., the element-wise multiplication. For the stability of optimization, we update the mask \( m \) with a fixed interval (denoted as \( F_i \)) during training. The algorithm of FSAM is shown in Algorithm 1.

To find the optimal mask during training, we apply the Fisher information to achieve sparse perturbation. The Fisher information is proposed by (Fisher, 1922) to measures the information carried by an observable random variable about the unknown parameters of the distribution. The Fisher information is defined by

\[
F = \mathbb{E}_x \left[ \mathbb{E}_y \nabla \log p(y|x) \nabla \log p(y|x)^T \right], \quad (7)
\]

where the \( p(y|x) \) is the output of model in machine learning. However, due to the over-parameterized model in deep learning, the computation of Fisher information is unacceptable, i.e., \( F \in \mathbb{R}^{[w] \times [w]} \). To save the computational effort, we approximate Fisher information as the diagonal matrix, i.e., \( F \in \mathbb{R}^{[w]} \). Consider the expectation in Equation 7, the first one is the data distribution \( x \sim p(x) \), which is
Algorithm 1 Fisher SAM (FSAM)

Input: sparse ratio $s$, dense model $w$, binary mask $m$, update interval $T_m$, base learning rate $\gamma_0 \in \mathbb{R}^d$, training set $S$.
1: Initialize $w$ and $m$ randomly.
2: for epoch $t = 1, \ldots, T$ do
3: for each training iteration do
4: Sample a batch from $S$: $B$.
5: Compute perturbation $\epsilon$ by Eq. 2.
6: if $t \mod T_m = 0$ then
7: Sample $N_{Fisher}$ data from distribution $S$.
8: Compute Empirical Fisher by Equation 9.
9: $m_0 \leftarrow \text{ArgTopK}(\hat{F}, (1-s) \cdot |w|)$
10: $m_t \leftarrow \text{ArgTopK}(\hat{F}, s \cdot |w|)$
11: Update mask $m$ by merging: $m = m_0 \cup m_t$.
12: end if
13: $\epsilon \leftarrow \epsilon \odot m$
14: end for
15: Compute SAM gradient $g_t = \nabla f_B(w + \epsilon)$.
16: $v_t = \beta_2 v_{t-1} + (1 - \beta_2)|g_t|^2$
17: $\hat{v}_t = \max(\beta_0 v_{t-1}, 0)$.
18: $w \leftarrow w - \gamma \nabla g_t \odot \frac{1}{\sqrt{\hat{v}_t}}$
19: end for
20: return Final weight of model $w$.

not available in most tasks. We approximate it by sampling $N_{Fisher}$ data from $p(x)$:

$$F = \frac{1}{N_{Fisher}} \mathbb{E}_{y \sim p} \nabla \log p(y | x_i)^2. \quad (8)$$

The second expectation is over $p(y | x)$, which can be achieved by the label $y_i$ for data $x_i$ in supervised learning. Finally, we calculate the Fisher information as "Empirical Fisher":

$$\hat{F} = \frac{1}{N_{Fisher}} \nabla \log p(y_i | x_i)^2. \quad (9)$$

Since the empirical Fisher is the same size as the weight, i.e., $\hat{F} \in \mathbb{R}^{|w|}$, the value of the element in Fisher $\hat{F}$ represents the importance of the corresponding element in weight $w$. Thus, we sort the elements of $\hat{F}$ in descending, and the weights with top $k$ Fisher values will be perturbed, i.e., the corresponding element in mask will be set to 1:

$$m_1 \leftarrow \text{ArgTopK}(\hat{F}, (1-s) \cdot |w|), \quad (10)$$

where $m_1$ is the set whose elements in the mask $m$ are 1, i.e., $m = \{m_1 = 1 | m_i \in m\}$, and ArgTopK $(x, k)$ returns the top $k$ largest values among $x$. On the other hand, the other weights with small Fisher values will not be perturbed, i.e., the corresponding element in mask will be set to 0:

$$m_0 \leftarrow \text{ArgTopK}(\hat{F}, s \cdot |w|). \quad (11)$$

3.3 Theoretical Analysis

In this subsection, we theoretically analyze the convergence and generalization of FSAM. Due to the space limitation, we only show the convergence analysis here, and the generalization analysis and whole proof are presented in Appendix A.1.

Assumption 1. (L-smooth.) Consider $f$ is differentiable with gradient Lipschitz property: It exists $L > 0$ s.t.

$$||\nabla f(w) - \nabla f(v)|| \leq L||w - v||, \forall w, v \in \mathbb{R}^d.$$  

Assumption 2. (Bounded stochastic gradients.) The variance of stochastic gradient is bounded:

$$\mathbb{E}[||\nabla f_i(x) - \nabla f(x)||^2] \leq \sigma^2$$

Assumption 3. (Bounded gradient.) The stochastic gradient is bounded: It exists $G \geq 0$ s.t.

$$||\nabla f_i(w)||_{\infty} \leq G$$

Theorem 1. Consider the function $f$ under the assumption 1,2,3, and a fixed base learning rate $\gamma_t$ satisfies that $\gamma_t \leq \frac{\delta}{8L}$, we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[||\nabla f(x_t)||^2] \leq 2Gf(x_0) - f^* + \frac{20GL^2 \rho^2}{\delta} + \frac{2G^3}{T^2}d\left(\frac{1}{\delta} - \frac{1}{G}\right) + \frac{4G\gamma_t L \rho^2}{\delta} + \frac{4G\gamma_t L^2}{b\delta}d(G^2 - \delta^2) + \frac{4\gamma_t LG^3}{T}d(G^2 - \delta^2)$$

Theorem 1 shows that when $T$ is large, FSAM could achieve the linear speedup convergence rate with respect to mini-batch size $b$ under the setting of $\gamma_t = O(\sqrt{\frac{b}{T}})$ and $\rho = O(\sqrt{\frac{T}{b^2}})$, i.e.,

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[||\nabla f(x_t)||^2] = O\left(\sqrt{\frac{1}{bT}}\right)$$

4 Experimental Setup

4.1 Tasks and Datasets

To investigate the effectiveness and universality of our FSAM method, we conduct extensive experiments on various NLP tasks. Specifically, different from Bahri et al. (2022) that only verify the method on several language understanding tasks, we evaluate our method on both language understanding and generation tasks. 4067
Table 1: Experimental results (dev scores) on various language understanding benchmarks. Comparison between vanilla SAM and our proposed FSAM applied to four widely used large-scale PLMs. The best results for each setting are in **bold**. “A VG.” denotes the average scores on all tasks, which are underlined. Results show that our FSAM brings consistent improvements across all understanding tasks among different PLMs.

| Method        | CoLA | MRPC | STS-B | RTE | CB | BoolQ | WSC | WiC | AVG. |
|---------------|------|------|-------|-----|----|-------|-----|-----|------|
|               | Mcc. | Acc. | F1    | Pear.| Spea.| Acc. | Acc. | Acc. | Acc. | Acc. |
| BERT-large    |      |      |       |     |    |       |     |     |      |
| Adam          | 62.8 | 87.3 | 91.1  | 89.5| 89.3| 70.7  | 87.5| 74.3 | 68.3 | 72.7 | 79.35|
| Adam+SAM      | 62.1 | 87.9 | 91.4  | 89.8| 89.4| 71.5  | 91.1| 72.9 | 68.3 | 74.1 | 79.85|
| Adam+FSAM     | **63.4** | **89.0** | **92.0** | **90.4** | **89.9** | **74.4** | **94.6** | **75.3** | **68.5** | **74.4** | **81.19**|
|                | **A VG.** |      |       |     |    |       |     |     |      |      |      |
| ELECTRA-large |      |      |       |     |    |       |     |     |      |
| Adam          | 69.0 | 90.2 | 92.4  | 92.1| 92.1| 87.3  | 91.1| 85.6 | 83.6 | 74.4 | **85.68**|
| Adam+SAM      | 63.9 | 90.9 | 94.2  | 92.4| 92.4| **89.2** | 92.9 | 82.2 | 84.6 | 72.4 | **85.61**|
| Adam+FSAM     | **69.6** | **92.4** | **94.5** | **92.3** | **92.5** | 88.8 | **96.4** | **85.9** | **89.4** | 74.1 | **87.59**|
|                | **A VG.** |      |       |     |    |       |     |     |      |      |      |
| ALBERT-xxlarge |      |      |       |     |    |       |     |     |      |
| Adam          | 71.1 | 90.7 | 93.3  | 92.9| 92.7| 87.0  | 89.3| 86.8 | 85.6 | 75.5 | 86.49|
| Adam+SAM      | 69.9 | 90.7 | 93.2  | 92.6| 92.4| 88.1  | **91.1** | **87.7** | **82.7** | **76.6** | 86.50|
| Adam+FSAM     | **72.3** | **91.9** | **94.2** | **93.0** | **92.8** | **88.8** | **91.1** | **87.9** | **86.5** | **76.6** | **87.51**|
|                | **A VG.** |      |       |     |    |       |     |     |      |      |      |
| RoBERTa-large |      |      |       |     |    |       |     |     |      |
| Adam          | 66.7 | 90.4 | 93.1  | 92.1| 92.0| 87.0  | 92.8| 86.0 | 78.1 | 73.3 | 85.15|
| Adam+SAM      | 68.5 | **90.7** | **93.3** | 91.5| 91.3| **87.7** | **96.4** | 84.2 | 81.3 | 74.0 | 85.89|
| Adam+FSAM     | **69.5** | **90.7** | **93.2** | 91.9| 91.6| **87.7** | **98.2** | **86.8** | **81.5** | **74.5** | **86.56**|

**Language Understanding Tasks.** Following many previous works (Vu et al., 2022; Bahri et al., 2022; Zhong et al., 2022), we conduct experiments on a combination of tasks from GLUE (Wang et al., 2018a) and SuperGLUE (Wang et al., 2019) benchmarks, including linguistic acceptability (CoLA), natural language inference (RTE, CB), paraphrase and similarity (MRPC and STS-B), question answering (BoolQ), word sense disambiguation (WiC) and coreference resolution (WSC). In practice, we evaluate the performance with Accuracy (“Acc.”) metric for most tasks, except the additional F1 score for MRPC, the Pearson-Spearman correlations (“Pear./Spea.”) for STS-B and the Matthew correlation (“Mcc.”) for CoLA.

**Language Generation Tasks.** We also use two popular generation tasks following Liu et al. (2021); Zhang et al. (2022) as the benchmarks, i.e., abstractive summarization (XSUM) and grammatical error correction (CoNLL2014). For the XSUM, we report results in terms of standard ROUGE metrics (Lin, 2004), i.e., Rouge-1, Rouge-2 and Rouge-L, respectively. For the CoNLL2014, MaxMatch scores (Dahlmeier and Ng, 2012) are used for evaluation with Precision, Recall, and $F_{0.5}$ values.

---

4 Due to the space limitation, we present the details of all used tasks and datasets in Appendix A.2

4068

4.2 Implementations

In practice, we use the pretrained models and code in HuggingFace\(^5\) (Wolf et al., 2019). Specifically, for the understanding tasks, we employ 4 widely used PLMs in our study, i.e., BERT (Devlin et al., 2019), ELECTRA (Clark et al., 2019b), ALBERT (Lan et al., 2019) and RoBERTa (Liu et al., 2019). Furthermore, an representative sequence-to-sequence model, BART (Lewis et al., 2020), is used for the generation tasks.

We compare our proposed FSAM method with the base optimizer (without using any SAM approach) and vanilla SAM method. Specifically, the Adam (Kingma and Ba, 2015) is used as the base optimizer to tune our models. The $\beta_2$ and weight decay of Adam are set as 0.999 and 0.01. SAM and FSAM use the same settings as above. More specially, we grid search for the neighborhood size of SAM and FSAM on \{1e-2, 5e-3, 1e-3\}. Additionally, for each downstream task, we follow the same hyper-parameter settings from the prior works (Lewis et al., 2020; Xu et al., 2021). The detailed hyper-parameters of fine-tuning on these downstream tasks can be seen in Appendix A.3. We report the averaged results over 5 random seeds for NLU tasks, while for NLG tasks, we follow ex-

---

\(^{5}\)https://github.com/huggingface/transformers
Table 2: Experimental results on two popular generation tasks. We use the representative sequence-to-sequence PLM (BART) in this study. It shows that our FSAM works well on the language generation tasks as well. “‡” indicates that FSAM is significantly better than baselines at significance level $p<0.05$.

| Method     | XSUM  | CoNLL2014 | AVG. |
|------------|-------|-----------|------|
|            | Rouge_1 | Rouge_2 | Rouge_L | Precision | Recall | F_0.5 |       |
| Adam       | 44.35  | 21.66    | 36.62  | 52.94     | 41.18  | 50.08 | 41.14 |
| Adam+SAM   | 44.81  | 21.95    | 36.97  | 53.52     | 41.76  | 50.70 | 41.62 |
| Adam+FSAM  | 45.03‡ | 22.15‡   | 37.14‡ | 54.33‡    | 42.15‡ | 51.36‡| 42.03 |
| BART-base  | 39.38  | 17.21    | 31.93  | 43.27     | 34.11  | 41.06 | 34.49 |
| Adam+SAM   | 40.38  | 18.00    | 33.00  | 50.39     | 33.51  | 45.78 | 36.84 |
| Adam+FSAM  | 40.60‡ | 18.31‡   | 33.28‡ | 51.77‡    | 34.04‡ | 46.89‡| 37.48 |

Table 3: Results of smaller PLMs with different optimizers on parts of understanding tasks. BERT-base and RoBERTa-base are used in this experiment.

| Method     | CoLA  | MRPC  | STS-B | RTE  | Mcc. | Acc. | F1. | Pear. | Spea. | Acc. |
|------------|-------|-------|-------|------|------|------|-----|-------|-------|------|
| BERT-base  |       |       |       |      |      |      |     |       |       |      |
| Adam       | 54.3  | 85.8  | 90.0  | 89.2 | 88.9 | 68.4 |
| -w SAM     | 53.0  | 87.3  | 90.9  | 89.3 | 89.1 | 66.4 |
| -w FSAM    | 53.8  | 87.7  | 91.3  | 89.5 | 89.2 | 70.0 |
| RoBERTa-base|      |       |       |      |      |      |     |       |       |      |
| Adam       | 61.3  | 87.5  | 90.6  | 90.6 | 90.4 | 78.3 |
| -w SAM     | 60.6  | 89.2  | 92.1  | 90.5 | 90.4 | 78.7 |
| -w FSAM    | 61.4  | 89.5  | 92.5  | 90.7 | 90.4 | 80.1 |

6.1 Parameter Analysis
There are two important hyper-parameters ($i.e.$, $s$ and $F_i$) in our FSAM, where $s$ refers to the

FSAM improves performance on various model sizes. To investigate whether our FSAM is helpful for various scales of PLMs, we evaluate the performance on smaller PLMs, i.e., BERT-base, RoBERTa-base and BART-base. The results are showed in Table 3 and Table 2, respectively. We can see that FSAM consistently outperforms the vanilla SAM on multiple smaller PLMs, to be specific, the relative improvements of BERT-base and BART-base are up to 0.92 and 0.64 average scores. These results prove that FSAM works well on various model sizes.
Table 4: Average performance (CoLA, MRPC, STS-B and RTE) of FSAM with different $Fi$, which denotes the fixed interval for updating Fisher mask.

| Method       | 10  | 50  | 100 | 200 | 500 |
|--------------|-----|-----|-----|-----|-----|
| FSAM-BERT    | 83.9| 84.0| 84.3| 84.2| 84.1|
| FSAM-RoBERTa | 80.0| 80.1| 80.3| 80.2| 80.2|

sparse ratio and $Fi$ is used to control the update frequency of Fisher mask. Here, we evaluate the performance of FSAM with different $s$ and $Fi$ on several downstream tasks to analyze their effects.

Firstly, Figure 1 shows the results based on different $s$. We can observe FSAM outperforms the vanilla SAM and base Adam in most settings, indicating the robustness of FSAM. Specifically, when the sparse ratio is 0.9, FSAM consistently achieves the best performance on both tasks. Secondly, for $Fi$, we show the performance of FSAM on different $Fi$ in Table 4. Too small $Fi$ (e.g., 10) may lead to the Fisher mask updating too fast, thus affecting the stability of model optimization. Recall that we set $s = 0.9$ and $Fi = 100$ as the default setting.

6.2 Complementarity with Other Optimizers

As aforementioned, we show the effectiveness of our Fisher mask strategy on SAM optimization. To further prove the universality of our proposed strategy, we examine whether the strategy is complementary with i) more base optimizers and ii) other efficient SAM variants.

To verify i), we use the additional AMSGrad and Adagrad as the base optimizers and evaluate the performance with different strategies, respectively. Table 5 lists the results of RoBERTa-large. It can be seen that FSAM consistently achieves the best performance upon these base optimizers, showing our strategy is not sensitive to the base optimizers.

For ii), we apply our strategy to another two cutting-edge SAM-variant optimizers, i.e., ESAM (Du et al., 2021) and GSAM (Zhuang et al., 2021). Table 6 shows the results, where F_ESAM and F_GSAM refer to the optimizations using our strategy. When evaluating RoBERTa-large on these tasks, compared to the vanilla ESAM and GSAM, our method can bring a 0.70 average score improvement. This indicates that our Fisher mask strategy is not only beneficial to the vanilla SAM, but also can be applied to other efficient SAM variants.

6.3 Results in Low-resource Scenarios

Prior works (Chen et al., 2021; Bahri et al., 2022) show that SAM helps more when there is less training data. Here, we verify how our Fisher mask strategy affects the effectiveness of SAM in low-resource scenarios. In practice, we follow Bahri et al. (2022) and sub-sample the training splits for several GLUE datasets at rates ranging from 10% to 90%. Notably, due to the space limitation, we only report parts of results on BERT-large and RoBERTa-large in Figure 2.

We can observe consistent gains from both SAM and our FSAM across all sizes of sub-sampled training sets, which confirms the statement in prior work (Bahri et al., 2022). Moreover, it can also be seen that our FSAM improves the vanilla SAM by a large margin in low-resource scenarios, especially when there is only 20% training data.
specifically, when fine-tuning the RoBERTa-large on the STS-B dataset, the relative improvements of FSAM are up to 15.0 and 15.1 in terms of accuracy and F1 score, respectively. These results show that our method is more helpful in low-resource scenarios.

6.4 Does FSAM Bring Better Generalization?

We prove the effectiveness of our FSAM by large-scale experiments as above. Here, to examine whether FSAM indeed brings better generalization, we i) measure the generalization properties (i.e., task generalization) of different optimizations, and ii) visualize the generalization of models via the training loss landscapes.

**Task Generalization.** The common wisdom is that models with better generalization would perform better on out-of-domain data (Xu et al., 2021). Thus, to measure the generalization ability of the model quantitatively, we follow the experiments from Xu et al. (2021) and evaluate the performance of various fine-tuned models on out-of-domain data. In practice, we first fine-tune RoBERTa-large on the QNLI task (one of GLUE tasks) and then transfer it to other tasks, i.e., CoLA, MRPC, STS-B and RTE. The results of different optimization strategies are illustrated in Figure 3.

We can observe that FSAM consistently outperforms the base Adam and vanilla SAM on different transferred tasks. To be more specific, compared with vanilla SAM, our FSAM brings a 2.37 relative average improvement score on these tasks, indicating that our method helps more in improving the generalization of model.

**Visualization of Landscape.** Here, we visualize the loss landscapes of RoBERTa-base model fine-tuned on CoLA with different optimizers. In practice, we follow Li et al. (2018); Zan et al. (2022) and show the 3D loss surface results in Figure 4 by sampling 25×25 points in the range of [-1, 1] from random “filter normalized” directions (Li et al., 2018). Additionally, following Hao et al. (2019); He et al. (2021), we also plot the 1D loss curve in Figure 5 by linear interpolation between the pretrained model weights before (denoted as $\theta_0$) and after (denoted as $\theta_1$) fine-tuning, i.e., “$\theta_1 + \alpha \cdot (\theta_1 - \theta_0)$”, where $\alpha$ is a scalar parameter that is ranged from -1 to 1. We can find that...
Figure 4: The loss surface of RoBERTa-base fine-tuned on CoLA with different optimizers. It can be seen that FSAM smooths the loss surface effectively, i.e., improving the model generalization.

Figure 5: 1D visualization of loss landscapes of RoBERTa-base model fine-tuned on different tasks. The landscape of FSAM is much flatter than both base Adam and SAM, especially in the area of low loss. These results prove that FSAM can smooth the loss landscape and improve the generalization of PLMs effectively.

7 Conclusion

In this paper, we improve the recently-proposed SAM optimization method with a novel Fisher mask strategy, and propose a new approach FSAM. Different from the vanilla SAM that adds a constant perturbation to all parameters, FSAM uses the Fisher information to calculate the Fisher mask and further obtains the sparse perturbation. Such a method can not only reduce the computation cost of optimization potentially, but also make the optimizer focus on the optimization of the important sharper parameters. Extensive experiments on five PLMs and various language understanding and generation tasks show that our FSAM consistently improves the performance of SAM by a large margin across all PLMs and tasks. Additionally, in-depth analysis and discussion demonstrate the robustness and universality of FSAM on improving the generalization of language models.

Limitations

Indeed, our work has some potential limitations, and we will discuss them in this section. Firstly, we only evaluate the BART on two generation tasks with different optimizers, and prove the effectiveness of our FSAM optimization method. It would be more valuable to consider other sequence-to-sequence PLMs and more generation tasks, e.g., fine-tuning T5 (Raffel et al., 2020) on CNN-DM (Hermann et al., 2015).

Additionally, as aforementioned in Section 1, we do not achieve the actual sparse training in this work, due to the limitation of the hardware. Specifically, to actually accelerate the unstructured sparsity (fine-grained sparsity), we need to implement the relevant sparse matrix calculation using the CUDA API on the recent NVIDIA Ampere A100 GPUs (Choquette et al., 2021) equipped with Sparse Tensor Cores (Pool, 2020) (Notably, although there is python API (i.e., ASP) provided by NVIDIA for accelerating the unstructured sparsity, it is only applicable to accelerate the model parameter sparsity, but not to the gradient-level sparse acceleration in our FSAM scenario). Unfortunately, it is relatively impracticable for us to do that. However, we still believe that FSAM has great potential to achieve true training acceleration in the future, with the development of hardware for fine-grained sparse operation.

Acknowledgements

We are grateful to the anonymous reviewers and the area chair for their insightful comments and suggestions. This work was supported in part by the National Natural Science Foundation of China under Grants 62141112, 62076186 and 62225113, and in part by the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170. The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.
References

Dara Bahri, Hossein Mobahi, and Yi Tay. 2022. Sharpness-aware minimization improves language model generalization. In ACL.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in nlp. In EMNLP.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In NeurIPS.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In Proceedings of SemEval-2017.

Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. 2021. When vision transformers outperform resnets without pre-training or strong data augmentations. arXiv.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multi-layer convolutional encoder-decoder neural network for grammatical error correction. In AAAI.

Jack Choquette, Wishwesh Gandhi, Olivier Giroux, Nick Stam, and Ronny Krashinsky. 2021. Nvidia a100 tensor core gpu: Performance and innovation. IEEE Micro.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019a. Boolq: Exploring the surprising difficulty of natural yes/no questions. In NAACL.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019b. Electra: Pre-training text encoders as discriminators rather than generators. In ICLR.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In ACL.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In NAACL.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In proceedings of Sinn und Bedeutung.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In NAACL.

Liang Ding, Longyue Wang, Yuebo Liu, Derek F Wong, Dacheng Tao, and Zhaopeng Tu. 2021. Progressive multi-granularity training for non-autoregressive translation. In Findings of the ACL.

Liang Ding, Longyue Wang, Shuming Shi, Dacheng Tao, and Zhaopeng Tu. 2022. Redistributing low-frequency words: Making the most of monolingual data in non-autoregressive translation. In ACL.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In Third International Workshop on Paraphrasing (IWP2005).

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.

Ronald A Fisher. 1922. On the mathematical foundations of theoretical statistics. Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2020. Sharpness-aware minimization for efficiently improving generalization. In ICLR.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In ACL.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness of bert. In EMNLP.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In ICLR.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. 2021. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In ACL.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In NeurIPS.

Sepp Hochreiter and Jürgen Schmidhuber. 1994. Simplifying neural nets by discovering flat minima. In NeurIPS.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. TACL.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2016. On large-batch training for deep learning: Generalization gap and sharp minima. arXiv.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In ICLR.
Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In ICLR.

Beatrice Laurent and Pascal Massart. 2000. Adaptive estimation of a quadratic functional by model selection. Annals of Statistics, pages 1302–1338.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In AAAI.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In ACL.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. In NeurIPS.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out.

Tao Lin, Sebastian U Stich, Luis Barba, Daniil Dmitriev, and Martin Jaggi. 2019. Dynamic model pruning with feedback. In ICLR.

Xuebo Liu, Longyue Wang, Derek F Wong, Liang Ding, Lidia S Chao, and Zhaopeng Tu. 2021. Understanding and improving encoder layer fusion in sequence-to-sequence learning. In ICLR.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv.

Peng Mi, Li Shen, Tianhe Ren, Yiyi Zhou, Xiaoshuai Sun, Rongrong Ji, and Dacheng Tao. 2022. Make sharpness-aware minimization stronger: A sparsified perturbation approach. arXiv preprint.

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

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In EMNLP.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. 2017. Exploring generalization in deep learning. In NeurIPS.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In CoNLL-2014.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In NAACL.

Jeff Pool. 2020. Accelerating sparsity in the nvidia ampere architecture. GTC 2020.

Weizhen Qi, Yeyun Gong, Yu Yan, Can Xu, Bolun Yao, Bartuer Zhou, Biao Cheng, Daxin Jiang, Jiusheng Chen, Ruofei Zhang, et al. 2021. Prophetnet-x: Large-scale pre-training models for english, chinese, multi-lingual, dialog, and code generation. In ACL.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In EMNLP.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In ACL.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. JMLR.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2022. SPoT: Better frozen model adaptation through soft prompt transfer. In ACL.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglu: A stickier benchmark for general-purpose language understanding systems. In NeurIPS.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018a. Glue: A multi-task benchmark and analysis platform for natural language understanding. In EMNLP.

Bing Wang, Liang Ding, Qihuang Zhong, Ximing Li, and Dacheng Tao. 2022. A contrastive cross-channel data augmentation framework for aspect-based sentiment analysis. arXiv.
Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018b. Switchout: an efficient data augmentation algorithm for neural machine translation. In *EMNLP*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *TACL*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv*.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. 2020. Adversarial weight perturbation helps robust generalization. In *NeurIPS*.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. 2021. Raise a child in large language model: Towards effective and generalizable fine-tuning. In *EMNLP*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Changtong Zan, Liang Ding, Li Shen, Yu Cao, Weifeng Liu, and Dacheng Tao. 2022. On the complementarity between pre-training and random-initialization for resource-rich machine translation. In *COLING*.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *ICLR*.

Zheng Zhang, Liang Ding, Dazhao Cheng, Xuebo Liu, Min Zhang, and Dacheng Tao. 2022. Bliss: Robust sequence-to-sequence learning via self-supervised input representation. *arXiv preprint*.

Yaowei Zheng, Richong Zhang, and Yongyi Mao. 2021. Regularizing neural networks via adversarial model perturbation. In *CVPR*.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2022. E2s2: Encoding-enhanced sequence-to-sequence pretraining for language understanding and generation. *arXiv*.

Qihuang Zhong, Fanzhou Zeng, Fei Liao, Juhua Liu, Bo Du, and Jedi S Shang. 2021. Joint image and feature adaptive attention-aware networks for cross-modality semantic segmentation. *Neural Computing and Applications*.

Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C Dvornek, James s Duncan, Ting Liu, et al. 2021. Surrogate gap minimization improves sharpness-aware training. In *ICLR*.
A Appendix

A.1 Missing Proof

Lemma 1. For any two vectors $x, y \in \mathbb{R}^d$, and any scalar $\alpha > 1$, we have the following inequality

$$
(x, y) \leq \frac{\alpha^2}{2} ||x||^2 + \frac{1}{2\alpha^2} ||y||^2
$$

Proof.

$$
RHS = \frac{\alpha^2}{2} \sum_{j=1}^{d} (x_j)^2 + \frac{1}{2\alpha^2} \sum_{j=1}^{d} (y_j)^2
\geq \sum_{j=1}^{d} 2\sqrt{\frac{\alpha^2}{2} \cdot (x_j)^2 \cdot \frac{1}{2\alpha^2} (y_j)^2} = LHS
$$

Lemma 2. We have the following inequality:

$$
\langle \nabla f(x_t), \frac{\gamma_t}{b} \sum_{i \in B} \nabla f_i(x_t) + \rho \sum_{i \in B} \nabla f_i(x_t) \circ m_t \bigg/ \|\sum_{i \in B} \nabla f_i(x_t)\| \rangle \\
\circ \bigg( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \bigg)
\leq \frac{\gamma_t}{b} \sum_{j=1}^{d} |\nabla f(x_t)(j)|
\times \|\sum_{i \in B} \nabla f_i(x_t) + \rho \sum_{i \in B} \nabla f_i(x_t) \circ m_t \bigg/ \|\sum_{i \in B} \nabla f_i(x_t)\| \|
\circ \bigg( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \bigg)(j)
\leq \frac{\gamma_t G}{b} \sum_{j=1}^{d} \sum_{i \in B} |\nabla f_i(x_t) + \rho \sum_{i \in B} \nabla f_i(x_t) \circ m_t \bigg/ \|\sum_{i \in B} \nabla f_i(x_t)\| |
\circ \bigg( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \bigg)(j)
\leq \frac{\gamma_t G^2}{b} \sum_{j=1}^{d} \sum_{i \in B} |\bigg( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \bigg)(j)|
\leq \frac{\gamma_t G^2}{b} \sum_{j=1}^{d} \sum_{i \in B} |\bigg( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \bigg)(j)|
\leq \gamma_t \sum_{i \in B} \nabla f_i(x_t) + \rho \sum_{i \in B} \nabla f_i(x_t) \circ m_t \bigg/ \|\sum_{i \in B} \nabla f_i(x_t)\| \|
\circ \bigg( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \bigg)
$$

\[ \text{LHS} = \langle \nabla f(x_t), \frac{\gamma_t}{b} \sum_{i \in B} \nabla f_i(x_t) + \rho \sum_{i \in B} \nabla f_i(x_t) \circ m_t \bigg/ \|\sum_{i \in B} \nabla f_i(x_t)\| \rangle \circ \bigg( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \bigg) \]

Proof.

Lemma 3. With the term defined before, we have the following inequality

$$
\langle \nabla f(x_t), \frac{\gamma_t}{b} \sum_{i \in B} \nabla f_i(x_t) + \rho \sum_{i \in B} \nabla f_i(x_t) \circ m_t \bigg/ \|\sum_{i \in B} \nabla f_i(x_t)\| \rangle \\
\circ \bigg( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \bigg) - \frac{\rho}{\|\sum_{i \in B} \nabla f_i(x_t)\|} \bigg( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \bigg) \bigg( \frac{1}{\sqrt{v_t}} \bigg)
\leq \frac{\gamma_t}{b} \sum_{i \in B} \nabla f_i(x_t) + \rho \sum_{i \in B} \nabla f_i(x_t) \circ m_t \bigg/ \|\sum_{i \in B} \nabla f_i(x_t)\| \|
\circ \bigg( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \bigg)
$$

where the $\mu > 1$ is a undetermined scalar.
By using Lemma 1, we have

\[
LHS \leq \frac{\mu^2 \gamma t}{2b^2} \left| \sum_{i \in B} \left( \nabla f_i(x_t + \rho \nabla f(x_t) \circ m_t \right) \right| \circ \sqrt{\frac{1}{v_t}}^2 \leq \frac{\mu^2 \gamma t}{2b} \sum_{i \in B} \left| \nabla f_i(x_t + \rho \nabla f(x_t) \circ m_t \right| \left| \nabla f(x_t) \right| \circ \sqrt{\frac{1}{v_t}}^2
\]

+ \frac{\gamma t}{2b} \left| \nabla f(x_t) \circ \sqrt{\frac{1}{v_t}}^2 \right| \leq \frac{\mu^2 \gamma t}{2b} \sum_{i \in B} \left| \nabla f_i(x_t + \rho \nabla f(x_t) \circ m_t \right| \left| \nabla f(x_t) \right| \circ \sqrt{\frac{1}{v_t}}^2
\]

+ \frac{\gamma t}{2b} \left| \nabla f(x_t) \circ \sqrt{\frac{1}{v_t}}^2 \right| \leq \frac{\mu^2 \gamma t}{2b} \sum_{i \in B} \left| \nabla f_i(x_t + \rho \nabla f(x_t) \circ m_t \right| \left| \nabla f(x_t) \right| \circ \sqrt{\frac{1}{v_t}}^2
\]

+ \frac{\gamma t}{b^2} \left| \nabla f(x_t) \circ \sqrt{\frac{1}{v_t}}^2 \right| \leq \frac{\mu^2 \gamma t}{2b} \sum_{i \in B} \left| \nabla f_i(x_t + \rho \nabla f(x_t) \circ m_t \right| \left| \nabla f(x_t) \right| \circ \sqrt{\frac{1}{v_t}}^2
\]

\[
\sum_{i \in B} \left| \nabla f_i(x_t) \circ m_t \right| \left| \nabla f(x_t) \right| \circ \sqrt{\frac{1}{v_t}}^2 \leq \frac{\gamma t}{2b} \sum_{i \in B} \left| \nabla f_i(x_t + \rho \nabla f(x_t) \circ m_t \right| \left| \nabla f(x_t) \right| \circ \sqrt{\frac{1}{v_t}}^2
\]

\[
\sum_{i \in B} \left| \nabla f_i(x_t) \right| \left| \nabla f(x_t) \right| \circ \sqrt{\frac{1}{v_t}}^2 \leq \frac{\gamma t}{2b} \sum_{i \in B} \left| \nabla f_i(x_t + \rho \nabla f(x_t) \circ m_t \right| \left| \nabla f(x_t) \right| \circ \sqrt{\frac{1}{v_t}}^2
\]

\[
\sum_{i \in B} \left| \nabla f_i(x_t) \right| \left| \nabla f(x_t) \right| \circ \sqrt{\frac{1}{v_t}}^2 \leq \frac{\gamma t}{2b} \sum_{i \in B} \left| \nabla f_i(x_t + \rho \nabla f(x_t) \circ m_t \right| \left| \nabla f(x_t) \right| \circ \sqrt{\frac{1}{v_t}}^2
\]

The \( \mu > 0 \) is the term to be determined.

---

**Lemma 4.** We have the following inequality:

\[
\mathbb{E}(\nabla f(x_t), -\frac{\gamma t}{b} \sum_{i \in B} \nabla f_i(x_t + \rho \nabla f(x_t) \circ m_t) \rangle \circ \frac{1}{v_t} \leq -\gamma t \left| \nabla f(x_t) \circ \sqrt{\frac{1}{v_t}}^2 \right| + \frac{\gamma t^2 L^2 \rho^2}{2b},
\]

where the \( \beta > 1 \) is a undetermined scalar.

**Proof:**

\[
LHS = -\gamma t \left| \nabla f(x_t) \circ \sqrt{\frac{1}{v_t}}^2 \right| + \mathbb{E}(\nabla f(x_t), -\frac{\gamma t}{b} \sum_{i \in B} (\nabla f_i(x_t) \circ m_t)) \circ \frac{1}{v_t} \leq -\gamma t \left| \nabla f(x_t) \circ \sqrt{\frac{1}{v_t}}^2 \right| + \frac{\gamma t^2 L^2 \rho^2}{2b}
\]

4077
By using the Lemma 1, we have

\[
\text{LHS} \leq -\gamma_t \|\nabla f(x_t)\| \sqrt{\frac{1}{\sqrt{v_{t-1}}}}^2 \\
+ \mathbb{E} \frac{\gamma_t}{\beta^2} \|\nabla f(x_t)\| \sqrt{\frac{1}{\sqrt{v_{t-1}}}}^2 \\
+ \mathbb{E} \frac{\gamma_t \beta^2}{2} \frac{1}{b} \sum_{i \in B} \|\nabla f_i(x_t)\| \\
- \nabla f_i(x_t + \rho \frac{\nabla f_i(x_t) \odot m_t}{\|\nabla f_i(x_t)\|}) \odot \sqrt{\frac{1}{\sqrt{v_{t-1}}}}^2 \\
\leq -\gamma_t \|\nabla f(x_t)\| \sqrt{\frac{1}{\sqrt{v_{t-1}}}}^2 \\
+ \mathbb{E} \frac{\gamma_t}{\beta^2} \|\nabla f(x_t)\| \sqrt{\frac{1}{\sqrt{v_{t-1}}}}^2 \\
+ \mathbb{E} \frac{\gamma_t \beta^2}{2\delta} \frac{1}{b} \sum_{i \in B} \|\nabla f_i(x_t)\| \\
- \nabla f_i(x_t + \rho \frac{\nabla f_i(x_t) \odot m_t}{\|\nabla f_i(x_t)\|}) \odot \sqrt{\frac{1}{\sqrt{v_{t-1}}}}^2 \\
\]

\text{Lemma 5. We have the following inequality:}

\[
\frac{L}{2} \mathbb{E} |x_{t+1} - x_t|^2 \leq \frac{\gamma_t^2 L}{2} \left( 3 \frac{1 + \alpha}{\alpha \delta} \right) \\
+ (\mathbb{E} \|\nabla f(x_t)\| \sqrt{\frac{1}{\sqrt{v_{t-1}}}}^2 + \frac{L \rho^2}{\delta}) \\
+ \frac{\sigma^2}{b \delta} + (1 + \alpha) G^2 \mathbb{E} \|\frac{1}{\sqrt{v_t}} - \frac{1}{\sqrt{v_{t-1}}})^2 \\
\]

\text{Proof:}

\[
\frac{L}{2} \mathbb{E} |x_{t+1} - x_t|^2 \\
= \frac{L}{2} \mathbb{E} \|\frac{\gamma_t}{b} \sum_{i \in B} (\nabla f_i(x_t + \rho \sum_{i \in B} \nabla f_i(x_t) \odot m_t) \odot \frac{1}{\sqrt{v_{t-1}}})^2 \\
+ (1 + \alpha) \mathbb{E} \|\sum_{i \in B} \nabla f_i(x_t) \odot m_t \| (\frac{1}{\sqrt{v_t}} - \frac{1}{\sqrt{v_{t-1}}})^2 \\
\]

By using the Lemma 1, we can obtain

\[
\frac{L}{2} \mathbb{E} |x_{t+1} - x_t|^2 \\
\leq \frac{L \gamma_t}{2} \left( 1 + \frac{1}{\alpha} \right) \mathbb{E} \|\frac{1}{b} \sum (\nabla f_i(x_t + \rho \sum_{i \in B} \nabla f_i(x_t) \odot m_t) \odot \frac{1}{\sqrt{v_{t-1}}})^2 \\
+ (1 + \alpha) \mathbb{E} \|\sum_{i \in B} \nabla f_i(x_t) \odot m_t \| (\frac{1}{\sqrt{v_t}} - \frac{1}{\sqrt{v_{t-1}}})^2 \\
\]

4078
Similar to the proof of Lemma 3, we have

\[
\begin{align*}
\frac{L}{2} \mathbb{E}[\|x_{t+1} - x_t\|^2] & \leq \frac{L^2 \gamma^2}{2} \left[ (1 + \frac{1}{\alpha}) \mathbb{E} \left[ \left( \frac{1}{\sqrt{v_t}} - \frac{1}{\sqrt{v_{t-1}}} \right) \right]^2 + \frac{\rho}{b} \left| \left| \frac{1}{\sqrt{v_t}} \right| \right| \mathbb{E} \left[ \frac{1}{\sqrt{v_t}} \right] \mathbb{E} \right[ \frac{1}{\sqrt{v_{t-1}}} \right] \right] \\
& \leq \frac{L^2 \gamma^2}{2} \left[ (1 + \frac{1}{\alpha}) \mathbb{E} \left[ \left( \frac{1}{\sqrt{v_t}} - \frac{1}{\sqrt{v_{t-1}}} \right) \right]^2 + \frac{\rho}{b} \left| \left| \frac{1}{\sqrt{v_t}} \right| \right| \mathbb{E} \left[ \frac{1}{\sqrt{v_t}} \right] \mathbb{E} \right[ \frac{1}{\sqrt{v_{t-1}}} \right] \right]
\end{align*}
\]

**Theorem 2.** By using the definition of \( L \)-smooth and the Lemma 2, 3, 4 and 5, we have

\[
\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(x_t)\|^2] \leq 2 G f(x_0) - f^* + \frac{2 G L^2 \rho^2}{\delta} + \frac{2 G^3}{T} d \left( \frac{1}{\delta} - \frac{1}{G} \right) + \frac{4 G L \rho^2}{\delta} d \frac{G L \sigma^2}{b \delta} + \frac{4 G^3}{T} d (G^2 - \delta^2)
\]

**Proof:**

\[
f(x_{t+1}) \leq f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{L}{2} \|x_{t+1} - x_t\|^2
\]

By splitting the term \( \mathbb{E} \left[ \frac{1}{b} \sum \nabla f_i(x_t) \right] \)
By re-range it, we obtain
$f(x_{t+1}) - f(x_t) \\ \leq \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
= \langle \nabla f(x_t), -\gamma_t b \sum_{i \in B} \nabla f_i(x_t) + \rho \sum_{i \in B} \nabla f_i(x_t) \odot m_t \rangle \odot \frac{1}{\sqrt{v_t}} + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
= \langle \nabla f(x_t), -\gamma_t b \sum_{i \in B} \nabla f_i(x_t) \rangle \odot \frac{1}{\sqrt{v_t}} + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
\leq \langle \nabla f(x_t), -\gamma_t b \sum_{i \in B} \nabla f_i(x_t) + \rho \sum_{i \in B} \nabla f_i(x_t) \odot m_t \rangle \odot \frac{1}{\sqrt{v_t}} + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
+ \rho \sum_{i \in B} \nabla f_i(x_t) \odot m_t \rangle \odot \frac{1}{\sqrt{v_t}} + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
+ \langle \nabla f(x_t), \gamma_t b \sum_{i \in B} \nabla f_i(x_t) + \rho \sum_{i \in B} \nabla f_i(x_t) \odot m_t \rangle \odot \frac{1}{\sqrt{v_t}} + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
+ \gamma_t G^2 \| \frac{1}{\sqrt{v_t}} - \frac{1}{\sqrt{v_t}} \|_1 + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
\leq \langle \nabla f(x_t), -\gamma_t b \sum_{i \in B} \nabla f_i(x_t) + \rho \nabla f(x_t) \odot m_t \rangle \odot \frac{1}{\sqrt{v_t}} + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
+ \gamma_t G^2 \| \frac{1}{\sqrt{v_t}} - \frac{1}{\sqrt{v_t}} \|_1 + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
$From the Lemma 3, we have the following inequality. Specifically,
$f(x_{t+1}) - f(x_t) \\ \leq \langle \nabla f(x_t), -\gamma_t b \sum_{i \in B} \nabla f_i(x_t) + \rho \nabla f(x_t) \odot m_t \rangle \odot \frac{1}{\sqrt{v_t}} + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
+ \gamma_t G^2 \| \frac{1}{\sqrt{v_t}} - \frac{1}{\sqrt{v_t}} \|_1 + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
By taking the expectation, we have the following inequality. Specifically,
$\mathbb{E} f(x_{t+1}) - f(x_t) \\ \leq \mathbb{E} \langle \nabla f(x_t), -\gamma_t b \sum_{i \in B} \nabla f_i(x_t) + \rho \nabla f(x_t) \odot m_t \rangle \odot \frac{1}{\sqrt{v_t}} + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
+ \gamma_t G^2 \| \frac{1}{\sqrt{v_t}} - \frac{1}{\sqrt{v_t}} \|_1 + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
$From the Lemma 4, we have the following inequality. Specifically,
$\mathbb{E} f(x_{t+1}) - f(x_t) \\ \leq \mathbb{E} \langle \nabla f(x_t), -\gamma_t b \sum_{i \in B} \nabla f_i(x_t) + \rho \nabla f(x_t) \odot m_t \rangle \odot \frac{1}{\sqrt{v_t}} + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
+ \gamma_t G^2 \| \frac{1}{\sqrt{v_t}} - \frac{1}{\sqrt{v_t}} \|_1 + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\
$
From the Lemma 5, we can obtain

\[
\begin{align*}
Ef(x_{t+1}) - f(x_t) & \leq -\gamma_t \|\nabla f(x_t)\| \sqrt{\frac{1}{\sqrt{v_{t-1}}}}^2 \\
+ \gamma_t G^2 E||\nabla f(x_t)\| \left( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \right) + \frac{2\gamma_t L^2 \rho^2}{\delta} + \frac{\gamma_t^2 L^2 (1 + \alpha)}{2} E||\nabla f(x_t)\| \left( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \right)^2
\end{align*}
\]

Set the $\mu^2 = \beta^2 = 4, \alpha = 3$ and set the $\frac{\gamma t L}{\alpha} \leq \frac{1}{8},$

we can simplify the inequality

\[
\begin{align*}
Ef(x_{t+1}) - f(x_t) & \leq -\gamma_t \|\nabla f(x_t)\| \sqrt{\frac{1}{\sqrt{v_{t-1}}}}^2 + \frac{8\gamma_t L^2 \rho^2}{\delta} \\
+ \gamma_t G^2 E||\nabla f(x_t)\| \left( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \right) + \frac{2\gamma_t L^2 \rho^2}{\delta} + \frac{\gamma_t^2 L^2 (1 + \alpha)}{2} E||\nabla f(x_t)\| \left( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \right)^2
\end{align*}
\]

Note that the $\frac{1}{\sqrt{v_t}}$ is bounded, we re-arrange the inequality and achieve

\[
\begin{align*}
\frac{\gamma t}{2G} \|\nabla f(x_t)\|^2 & \leq \frac{\gamma t}{2} \|\nabla f(x_t)\| \sqrt{\frac{1}{\sqrt{v_{t-1}}}}^2 \\
+ \gamma_t G^2 E||\nabla f(x_t)\| \left( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \right) + \frac{2\gamma_t L^2 \rho^2}{\delta} + \frac{\gamma_t^2 L^2 (1 + \alpha)}{2E\|\nabla f(x_t)\|} \frac{1}{\sqrt{v_{t-1}}} \|\nabla f(x_t)\| \left( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \right)^2
\end{align*}
\]

We sum up it and we obtain

\[
\begin{align*}
\frac{1}{T} \sum_{t=0}^{T-1} Ez\|\nabla f(x_t)\| & \leq \frac{2G}{\gamma_t T} \|\nabla f(x_0)\| - \|\nabla f(x_{t+1})\| \\
+ \frac{16GL^2 \rho^2}{\delta} + \frac{2G^3}{\gamma_t T} \|\nabla f(x_t)\| \left( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \right) + \frac{4G \gamma_t L \left( L \rho^2 + \frac{\sigma^2}{\beta}\right)}{\delta} \\
+ \frac{4\gamma_t L G^3}{T} \|\nabla f(x_t)\| \left( \frac{1}{\sqrt{v_{t-1}}} - \frac{1}{\sqrt{v_t}} \right)^2 \\
+ \frac{2G \gamma_t L \left( L \rho^2 + \frac{\sigma^2}{\beta}\right)}{\delta} \frac{\gamma_t T}{\delta} + \frac{20G \gamma_t L \left( L \rho^2 + \frac{\sigma^2}{\beta}\right)}{\delta}
\end{align*}
\]

Theorem 3. With probability $1 - \delta$ over the choice of training set $S \sim D$, we have the following inequality.

\[
L_D(w) \leq \max_{||\epsilon||_2 \leq \rho} L_S(w + \epsilon)
\]

\[
\left\lfloor k \log \left( 1 + \frac{||w||_2^2 \left( 1 + \sqrt{\log \frac{n}{k}} \right)^2 \right) + 4 \log \frac{n}{\delta} + O(1) \right\rfloor / n - 1
\]

The $n = |S|, k$ is the number of weight $w$, and assuming that $L_D(w) \leq L_{i\sim N(0,\rho)} |L_D(w + \epsilon)|$

Proof. The proof is mainly based on PAC-Bayesian Generalization Bound theorem. We start by any
prior $P$ over parameters with probability $1 - \delta$ for any posterior distribution $Q$, we have

\[ E_{w \sim W}[L_Q(w)] \leq E_{w \sim W}[L_S(w)] + \sqrt{KL(Q\|P) + \log \frac{n}{\delta}} \]

Assume $\epsilon \sim \mathcal{N}(0, \sigma)$, the $||\epsilon||^2_2$ follows the Chi-square distribution. From the Lemma 1 in (Laurent and Massart, 2000), we have the following inequality for any positive $t$:

\[ P(||\epsilon||^2_2 - k\sigma^2 \geq 2\sigma^2\sqrt{kt} + 2t\sigma^2) \leq \exp(-t) \]

Combine the above function and subtract the same constant $C$ on both side, following the assumption in (Foret et al., 2020), we finish the proof.

\[ \square \]

A.2 Details of Tasks and Datasets

As mention in Section 4, we conduct extensive experiments on parts of tasks from GLUE and SuperGLUE. In addition, two widely-used generation tasks are also used in this work. Here, we introduce the descriptions of the used tasks and datasets in detail. Firstly, we present the statistics of all datasets in Table 7. Then, each task is described as:

**CoLA.** Corpus of Linguistic Acceptability (Warstadt et al., 2019) is a binary single-sentence classification task to determine whether a given sentence is linguistically “acceptable”.

**MRPC.** Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005) is a task to predict whether two sentences are semantically equivalent.

**STS-B.** Semantic Textual Similarity (Cer et al., 2017) is a task to predict how similar two sentences are on a 1-5 scale in terms of semantic meaning.

**RTE.** Recognizing Textual Entailment (Giampiccolo et al., 2007), given a premise and a hypothesis, is a task to predict whether the premise entails the hypothesis.

**QNLI.** Question Natural Language Inference is a binary classification task constructed from SQuAD (Rajpurkar et al., 2016), which aims to predict whether a context sentence contains the answer to a question sentence.

**CB.** CommitmentBank (De Marneffe et al., 2019) is a task that can be framed as three-class textual entailment on a corpus of 1,200 naturally occurring discourses.

**BoolQ.** Boolean Question (Clark et al., 2019a) is a question answering task where each sample consists of a short passage and a yes/no question about the passage.

**WiC.** Word-in-Context (Pilehvar and Camacho-Collados, 2019) is a word sense disambiguation task that aims to predict whether the word is used with the same sense in sentence pairs.

**WSC.** Winograd Schema Challenge (Levesque et al., 2012) is a co-reference resolution task which aims to determine the correct referent of the pronoun from among the provided choices.

**XSUM.** The Extreme Summarization dataset (Narayan et al., 2018) is one of abstractive Summarization task that aims to convert the given document into a short and adequate summary in the same language.

**CoNLL2014.** CoNLL2014 (Ng et al., 2014) is a popular grammatical error correction task that aims to rewrite the input sentence with grammatical errors into the corresponding correct sentence, where the original and target sentences have the similar sentence lengths.

A.3 Hyper-parameters of Fine-tuning

In this paper, we fine-tune four different large-scale PLMs with our FSAM on the tasks of GLUE and SuperGLUE, including BERT-large (~340M) 6, ELECTRA-large (~340M) 7, ALBERT-xxlarge-v2 (~223M) 8 and RoBERTa-large (~355M) 9. Additionally, the BART-large (~406M) 10 is used for the generation tasks. The training epochs/steps, batch size, learning rate and warmup steps are listed in Table 8 and Table 10. Notably, the maximum sequence length of language understanding tasks is set as 128/256. For two generation tasks, we empirically set the minimum and maximum length of the XSUM dataset as 10 and 60, and closely follow Chollampatt and Ng (2018) to preprocess the data of CoNLL2014.

A.4 Training Curves

In this sub-section, we visualize the training curves of Adam, SAM and FSAM in detail. Specifically, Figure 6 shows the evaluation metrics v.s. training epochs. The metric curves prove that FSAM boosts the performance effectively during the training, which shows the effectiveness of FSAM.

---

6https://huggingface.co/bert-large-cased
7https://huggingface.co/google/electra-large-discriminator
8https://huggingface.co/albert-xxlarge-v2
9https://huggingface.co/roberta-large
10https://huggingface.co/facebook/bart-large
Table 7: Data statistics of all used tasks in this paper.

| Task   | Description                          | #Train | #Dev | #Class |
|--------|--------------------------------------|--------|------|--------|
| GLUE   | CoLA linguistic acceptability classification | 8.5K   | 1,042 | 2      |
|        | MRPC paraphrase classification        | 3.7K   | 409  | 2      |
|        | STS-B semantic textual similarity     | 5.7K   | 1,501 | -      |
|        | RTE natural language inference       | 2.5K   | 278  | 2      |
|        | QNLI natural language inference      | 104K   | 5,464| 2      |
| SuperGLUE | BoolQ question answering              | 9.4K   | 3,270 | 2      |
|         | CB natural language inference        | 250    | 57   | 2      |
|         | WiC word sense disambiguation        | 6K     | 638  | 2      |
|         | WSC coreference resolution           | 554    | 104  | 2      |
| Generation | XSUM abstractive summarization      | 204K   | 11,332 | -      |
|         | CoNLL2014 grammatical error correction | 1.3M   | 5,448| -      |

Table 8: Hyper-parameters settings for BART model on the generation tasks.

| Setting      | XSUM | CoNLL2014 |
|--------------|------|-----------|
| Learning Rate| 3e-5 | 2e-5      |
| Batch Size   | 56   | 800       |
| Training Steps| 15,000 | 15,000   |
| Warmup Steps | 500  | 500       |
| GPUs         | 8    | 4         |

A.5 More Results

In addition to the results in Table 1 with the momentum as 0.9, we also conduct the experiments with the momentum as 0 (i.e., $\beta_1 = 0$ in Algorithm 1) to evaluate the influence of the adaptive learning rate. In practice, we evaluate the performance on several downstream tasks upon two base optimizers (Adam and AMSGrad) and two PLMs (RoBERTa-large and RoBERTa-base). Table 9 shows the results. When there is no momentum term, FSAM also achieves better performance against the vanilla SAM and base optimizers. These results prove the universality of FSAM in various scenarios.
Figure 6: Accuracy on the dev set v.s. training epochs. We show some representative results on the four pretrained models. We can observe that our FSAM achieves the best performance among all tasks.

Table 9: Results of RoBERTa-large and RoBERTa-base fine-tuned with different optimizers ($\beta_1 = 0$). We can observe that FSAM consistently outperforms the vanilla SAM and base optimizer among all settings.

| Method     | RoBERTa-base, Adam | RoBERTa-large, Adam |
|------------|--------------------|---------------------|
|            | CoLA Mcc. Acc. Fl. Pear. Spea. Acc. AVG. | CoLA Mcc. Acc. Fl. Pear. Spea. Acc. AVG. |
| Adam       | 61.3 87.5 90.6 90.4 90.2 78.3 83.05 | 67.0 89.0 91.9 91.6 91.0 88.4 86.63 |
| Adam+SAM   | 61.6 88.2 91.6 90.5 90.2 79.4 83.58 | 68.6 90.4 93.1 90.7 90.4 88.1 86.88 |
| Adam+FSAM  | 63.2 88.5 92.0 90.6 90.3 82.7 84.55 | 69.0 90.4 93.2 91.6 91.3 88.8 87.38 |

| Method     | RoBERTa-base, AMSGrad | RoBERTa-large, AMSGrad |
|------------|-----------------------|-----------------------|
|            | CoLA Mcc. Acc. Fl. Pear. Spea. Acc. AVG. | CoLA Mcc. Acc. Fl. Pear. Spea. Acc. AVG. |
| AMSGrad    | 60.1 88.5 91.6 90.3 90.2 79.1 83.30 | 63.8 89.7 92.4 90.0 90.4 87.4 85.62 |
| AMSGrad+SAM| 60.7 89.2 92.2 90.2 89.9 80.2 83.73 | 68.5 90.0 92.8 91.6 91.2 87.0 86.85 |
| AMSGrad+FSAM| 62.0 88.7 91.9 90.5 90.4 79.8 83.88 | 68.5 90.2 92.9 91.6 91.0 88.1 87.05 |
Table 10: Hyper-parameters settings for different pretrained models on the language understanding tasks. We set the batch size to 16 for all settings. These settings are selected in best practice. Note that we apply these settings to fine-tune both large and small pretrained language models.

| Model   | Dataset | Learning Rate | Training Epochs/Steps | Warmup Ratio/Steps |
|---------|---------|---------------|-----------------------|--------------------|
| **BERT** | CoLA    | 2e-5          | 3 epochs              | 10%                |
|         | MRPC    | 2e-5          | 3 epochs              | 10%                |
|         | STS-B   | 4e-5          | 3 epochs              | 10%                |
|         | RTE     | 4e-5          | 3 epochs              | 10%                |
|         | BoolQ   | 2e-5          | 10 epochs             | 10%                |
|         | CB      | 3e-5          | 20 epochs             | 10%                |
|         | WiC     | 2e-5          | 10 epochs             | 10%                |
|         | WSC     | 1e-5          | 20 epochs             | 10%                |
| **ELECTRA** | CoLA    | 1e-5          | 3 epochs              | 10%                |
|         | MRPC    | 3e-5          | 3 epochs              | 10%                |
|         | STS-B   | 3e-5          | 10 epochs             | 10%                |
|         | RTE     | 2e-5          | 10 epochs             | 10%                |
|         | BoolQ   | 2e-5          | 10 epochs             | 10%                |
|         | CB      | 3e-5          | 20 epochs             | 10%                |
|         | WiC     | 2e-5          | 10 epochs             | 10%                |
|         | WSC     | 1e-5          | 20 epochs             | 10%                |
| **ALBERT** | CoLA    | 2e-5          | 3 epochs              | 10%                |
|         | MRPC    | 3e-5          | 3 epochs              | 10%                |
|         | STS-B   | 3e-5          | 3 epochs              | 10%                |
|         | RTE     | 5e-5          | 3 epochs              | 10%                |
|         | BoolQ   | 1e-5          | 10 epochs             | 10%                |
|         | CB      | 4e-5          | 20 epochs             | 10%                |
|         | WiC     | 3e-5          | 10 epochs             | 10%                |
|         | WSC     | 1e-5          | 20 epochs             | 10%                |
| **RoBERTa** | CoLA    | 1e-5          | 2668 steps            | 160 steps          |
|         | MRPC    | 1e-5          | 1148 steps            | 68 steps           |
|         | STS-B   | 2e-5          | 1799 steps            | 107 steps          |
|         | RTE     | 2e-5          | 1018 steps            | 61 steps           |
|         | BoolQ   | 1e-5          | 10 epochs             | 10%                |
|         | CB      | 2e-5          | 20 epochs             | 10%                |
|         | WiC     | 2e-5          | 10 epochs             | 10%                |
|         | WSC     | 1e-5          | 20 epochs             | 10%                |