ROBUSTIFYING LANGUAGE MODELS VIA ADVERSARIAL TRAINING WITH MASKED GRADIENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning pre-trained language models (LMs) has become the de-facto standard method for improving state-of-the-art performances on various NLP tasks. Although these models are usually evaluated with accuracy on fixed validation sets, it is insufficient for the reliable deployment of fine-tuned LMs in real-world settings, as there are known issues within existing model evaluations, such as adversarial robustness and model calibration. To address such issues, we propose a simple yet effective training algorithm, coined Robustifying LMs via Adversarial training with Masked gradient (RAM), to improve the robustness of fine-tuned LMs. In particular, we leverage adversarial training to robustify LMs for various types of perturbations. Simultaneously, to prevent the trained model from largely deviating from the initial pre-trained model, we selectively update the important model parameters using the masked gradients; their relative importance is obtained from the gradients calculated during training. Consequently, it enables the model to preserve the generalizability of the pre-trained model while improving its robustness. Additionally, we construct a new benchmark to evaluate the robustness of fine-tuned LMs in terms of four representative aspects of model robustness in a unified way. Under these benchmarks, we demonstrate the effectiveness of RAM compared to other state-of-the-art fine-tuning methods, and verify that RAM is successfully robustifying various types of LMs. Our work suggests a rethinking of the robustness aspect of LMs as an essential direction for their reliable deployment, along with a simple yet effective solution.

1 INTRODUCTION

Recent success of self-supervised learning (Jing & Tian, 2020; Brown et al., 2020) has enabled a new approach of learning language models (LMs) using a large amount of unlabeled text corpora (Kenton & Toutanova, 2019; Liu et al., 2019). Fine-tuning these pre-trained LMs now has become the de-facto standard, as it significantly boosts state-of-the-art performance on numerous natural language processing (NLP) tasks, including text classification (Wang et al., 2019) and machine translation (Raffel et al., 2020). Up to now, the common practice for evaluating fine-tuned LMs is measuring the task performance (*e.g.,* accuracy) on a separate validation set; various training methods have been proposed to improve the effectiveness of fine-tuning further, focusing on this evaluation methodology (Aghajanyan et al., 2021; Wu et al., 2021).

However, it is unclear whether or not *the performance on the fixed validation sets is sufficient* to determine if a model can be reliably deployed in real-world settings, as state-of-the-art LMs are still prone to having *robustness* issues in various aspects; for example, their predictions are known to be vulnerable to small perturbations (Jin et al., 2020) or superficial cues (Kavumba et al., 2019). Hence, an ideal model to be deployed in the real-world should exhibit the following characteristics in terms of model robustness: generalization on distribution-shifted data (Ng et al., 2020), resiliency to adversarial perturbation (Wang et al., 2021b), calibrated predictions (Desai & Durrett, 2020), and discriminability of anomaly inputs (Tack et al., 2020).

To this end, prior studies have investigated the robustness of fine-tuned language models in a single aspect separately (Wang et al., 2021a; Zhou et al., 2021; Nam et al., 2022; Hendrycks et al., 2020). But, it has been shown that optimizing for certain desired aspects frequently comes at the expense of others: adversarial training improves adversarial robustness but the performance often degrades on the clean test set (Zhang et al., 2022). Or the model could suffer to discriminate whether the given input is an anomaly or not, although it shows state-of-the-art accuracy in the validation set (Guo et al., 2017). These results indicate the necessity of a *unified evaluation* of model robustness in multiple aspects, and a *single effective method* that can improve robustness in these aspects simultaneously; however, despite its importance, this direction is yet under-explored in NLP tasks.

Contribution. In this paper, we propose a new effective fine-tuning method to improve the robustness of LMs, coined RAM: **R**obustifying LMs via **A**dversarial training with **M**asked gradient. Our idea is to simulate an artificially constructed noise during the training to make fine-tuned LM robust for various perturbations, via *adversarial training*; by directly minimizing the worst-case error, adversarial training has been shown the effectiveness for various aspects of model robustness (Madry et al., 2018; Volpi et al., 2018). To prevent the fine-tuned LM from deviating too much from the initial pre-trained LM during adversarial training, we further propose to selectively update the model parameters based on their relative importance. In this way, RAM can potentially preserve the generalizable knowledge of pre-trained LM and further enhance the model's robustness. To be specific, the relative importance of model parameters is measured using their gradients calculated during the backward pass of training. This relative importance is then converted into the gradient masks in a proposed way, and the model parameters are selectively updated via the masked gradients. The behavior of these masked gradients is demonstrated with a provided theoretical analysis.

To facilitate the unified evaluation of fine-tuned LMs across multiple aspects of model robustness, we construct new benchmarks on two popular NLP tasks: sentiment classification and entailment. Specifically, we consider four representative aspects of model robustness: distribution-shift generalization (*i.e.,* out-of-domain generalization), adversarial robustness, calibrated prediction, and anomaly detection. Although these aspects have been separately explored, we regard them together for evaluating fine-tuned LMs with a unified viewpoint of model robustness. Under these robustness benchmarks, we demonstrate the effectiveness of RAM for improving the robustness of fine-tuned LMs. For example, across 4 different robustness aspects along with a standard validation accuracy, RAM exhibited 18.39% and 7.63% average relative improvement compared to the vanilla fine-tuning method on sentiment classification and entailment tasks, respectively. We also found that RAM significantly improves the robustness of six state-of-the-art LMs, which further demonstrates its practical usefulness as a simple yet effective solution for model robustness.

Overall, our work highlights the importance of improving the robustness of fine-tuned LMs, which is often overlooked when we focus on the accuracy of the fixed validation sets. We believe that our work can inspire researchers to rethink the under-explored, yet important problem for the reliable deployment of LMs in real-world scenarios.

2 Related Works

Evaluation of model robustness in multiple aspects. For deployment in real-world settings, models should be robust in various aspects, not just be accurate on the given validation set, which is separately drawn from training distribution. Such robustness is crucial for safety-critical applications by preventing misuse of the classifiers, such as self-driving cars and medical diagnoses (Shen et al., 2017). Generalization on distribution-shifted (*i.e.,* out-of-domain) datasets considers how the models are well-generalized to various forms of distribution shift at test time (Ng et al., 2020; Liu et al., 2022); for example, the classifier trained on particular sentiment classification dataset is expected to also perform well on another sentiment classifation dataset, as both datasets are constructed to solve the same task. On the other hand, it is well known that the predictions of deep neural networks can be arbitrarily wrong, even with human-imperceptible perturbations (Goodfellow et al., 2015). Since this problem is not exceptional for the recent LMs with word- or sentence-level adversarial attacks (Jin et al., 2020; Li et al., 2020), the resiliency to the adversarial examples is also an important aspect of the robustness. Model calibration considers the alignment between the model's predicted probability and the true correctness likelihood (Guo et al., 2017; Desai & Durrett, 2020), and hence it is essential for the reliability and interpretability of the model's prediction. Anomaly detection performance measures the model's capability to distinguish whether the given input comes from out-of-distribution (OOD) to the trained one (Hendrycks et al., 2020; Zhou et al., 2021). In the vision domain, Hendrycks et al. (2022) recently conducted an investigation on the existing data augmentations to verify their robustness in multiple aspects; however, such investigation has not yet been conducted in NLP tasks.

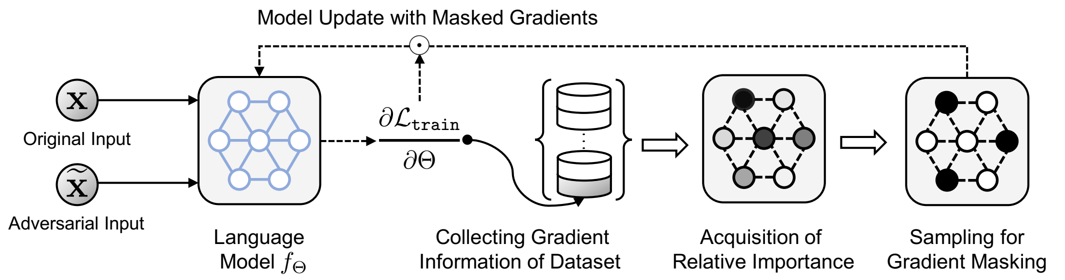

Model Update with Masked Gradients
$\frac{\partial \mathcal{L}_{\texttt{train}}}{\partial \Theta}$
Original Input
Adversarial Input
Language Model f_Θ
Collecting Gradient Information of Dataset
Acquisition of Relative Importance
Sampling for Gradient Masking

Figure 1: Illustration of proposed RAM: **R**obustifying via **A**dversarial training with **M**asked gradient.

Improved fine-tuning of language models. Various fine-tuning schemes have been explored to improve the generalization and robustness of the fine-tuned LMs upon a (limited) training data for the downstream tasks (Kim et al., 2022). One of the most representative ways is perturbation-based regularizations, which makes the model robust by adding the specific types of perturbation on the inputs during training. Wu et al. (2021) generates two different predictions of the same input using different Dropout masks, then make them similar with a consistency regularization. For the specific perturbation, Jiang et al. (2020) considers adversarial perturbation on the word embeddings to simulate the challenging view of inputs, and (Ng et al., 2020) substitutes the words using pre-trained masked LMs (*e.g.,* BERT), respectively. On the other hand, various training schemes have been explored to preserve the generalizable knowledge of pre-trained LMs during fine-tuning. Aghajanyan et al. (2021) identifies a risk of losing the generalizable knowledge from naive fine-tuning, and propose a noise-based regularization to prevent it. Chen et al. (2020) directly incorporates the distance between the parameters of fine-tuned and pre-trained models as a regularization, and Xu et al. (2021) proposes only to update the subset of the model's parameters. Recently, a two-step strategy of linear probing and then full fine-tuning is shown to be effective for the distribution-shift generalization (Kumar et al., 2022). However, despite its practical importance, an exploration for better fine-tuning of LMs to pursue the robustness in multiple aspects is under-explored; hence, we propose a *single effective method* that can improve model robustness considering these *multiple aspects simultaneously*.

3 Improving Robustness of Fine-tuned Language Models

3.1 Overview and Problem description

Overview. In this section, we present our technique, RAM, that can be applied to improve the model robustness during fine-tuning. We first impose an artificially constructed noise into training samples via adversarial training, to improve the robustness during fine-tuning. To further enhance the adversarial training by preserving the generalizable knowledge of pre-trained LM , we selectively update the model parameters by identifying the relatively important ones. Specifically, the relative importance of model parameters is calculated from their gradients and converted into the gradient masks. Then, the model is updated with these masked gradients. In Section 3.2, we elaborate the specific components of RAM. Figure 1 presents the overview of our method.

Problem description. We first describe the problem setups of our interest under a text classification scenario. Let $\mathcal{D}$ denote the given training dataset for target task consisting of tuples $(\mathbf{x}, y) \in \mathcal{D}$ where $\mathbf{x} = [x_1, \dots, x_L]$ and y are the sequence of input tokens x_i and the target label, respectively. Then, we train a classifier $f_\Theta(\mathbf{x})$ on $\mathcal{D}$, initialized with a pre-trained transformer-based language model (*e.g.,* BERT (Kenton & Toutanova, 2019)), by minimizing the task-specific training loss $\mathcal{L}_{\texttt{task}}(\mathbf{x}, y)$ such as a cross-entropy loss $\ell_{\texttt{xe}} := -y \log f_\Theta(\mathbf{x})_y$. Unlike the typical scenario, our goal is not only to improve the model's accuracy on the validation set from training distribution but also to make it robust in various aspects for reliable deployment in real-world settings.

3.2 Robustifying via adversarial training with masked gradient

Next, we present our specific techniques to improve the robustness of fine-tuned language models: (a) adversarial training, (b) measuring the relative importance of the model's parameters from the gradient, and (c) gradient masking from the relative importance for selective update of model parameters.

Algorithm 1 `GetMask`: Sampling Gradient Mask from Relelative Importance Scores

```
Input: Relative importance scores $\{s(\theta)|\theta \in \Theta\}$, masking ratio $\alpha$, smoothness factor $\beta$
for each parameter $\theta \in \Theta$ do
    $\widetilde{s}(\theta) \leftarrow \text{argsort}(s(\theta))/|\{\theta\}|$   // Normalization with relative order
    $p(\theta) \leftarrow 1/(1+\exp(2\beta(\widetilde{s}(\theta)-\alpha)))$   // Smooth approximation of thresholding
    $m(\theta) \sim \text{Bernoulli}\big(p(\theta)\big)$   // Sampling mask from Bernoulli distribution
end for
Return: $\{m(\theta)|\theta \in \Theta\}$
```

Adversarial training. To improve the robustness of the trained model f_Θ, we first incorporate adversarial training during fine-tuning. Specifically, for each training iteration, we construct adversarial example $\widetilde{\mathbf{x}}$ and then train the model using both of $\mathbf{x}$ and $\widetilde{\mathbf{x}}$:[1]

$$\mathcal{L}_{\texttt{train}} = \mathcal{L}_{\texttt{task}}(\mathbf{x}, y) + \mathcal{L}_{\texttt{task}}(\widetilde{\mathbf{x}}, y) + \lambda \mathcal{L}_{\texttt{cons}}(\mathbf{x}, \widetilde{\mathbf{x}}) \tag{1}$$

where $\mathcal{L}_{\texttt{cons}}(\mathbf{x}, \widetilde{\mathbf{x}}) := \mathcal{D}_{\text{KL}}(f_\Theta(\mathbf{x}), f_\Theta(\widetilde{\mathbf{x}})) + \mathcal{D}_{\text{KL}}(f_\Theta(\widetilde{\mathbf{x}}), f_\Theta(\mathbf{x}))$ is a bidirectional KL divergence (Wu et al., 2021) with a hyper-parameter λ. For constructing $\widetilde{\mathbf{x}}$, we use a single step gradient ascent (Jiang et al., 2020) with a step size δ under ℓ_∞ norm, *i.e.*, $\widetilde{\mathbf{x}} := \mathbf{x} + \delta \cdot (\partial \mathcal{L}_{\texttt{task}}/\partial \mathbf{x})/||\partial \mathcal{L}_{\texttt{task}}/\partial \mathbf{x}||_\infty$.

Measuring relative importance of model parameters. To measure the relative importance $s(\theta)$ of each model parameter $\theta \in \Theta$ with respect to the given task $\mathcal{D}$ and the training loss $\mathcal{L}_{\texttt{train}}$, we use a sum of the square of gradients as its measurement:

$$s(\theta) = \sum\nolimits_{(\mathbf{x},y)\in\mathcal{D}} |g(\theta)|^2, \;\; g(\theta) = \partial \mathcal{L}_{\texttt{train}}/\partial \theta \tag{2}$$

It is worth noting that this sum of the square of the gradients is highly connected to Fisher information matrix, which provides a point estimate of the task-relevant importance of the parameters (Kirkpatrick et al., 2017; Xuhong et al., 2018). However, additional calculation of the gradients for all samples significantly enlarges the training cost. To alleviate this, we instead use the gradients calculated during the backward pass for model training; although the different value of θ at different training steps is used to calculate the gradient of each sample, we empirically verify that such approximation is effective and remark that similar approach has been used in other domain (Berthelot et al., 2019).

Sampling of gradient mask. Using the obtained relative importance scores $\{s(\theta)|\theta \in \Theta\}$, we decide whether update each model parameter θ or not, as described in Algorithm 1. Specifically, we first obtained the normalized scores $\widetilde{s}(\theta)$ from their relative order in terms of magnitude, *e.g.,* $\widetilde{s}(\theta_{\texttt{min}}) = 0 \leq \widetilde{s}(\theta) \leq 1 = \widetilde{s}(\theta_{\texttt{max}})$ where $\theta_{\texttt{min}} := \arg\min_{\theta\in\Theta} s(\theta)$ and $\theta_{\texttt{max}} := \arg\max_{\theta\in\Theta} s(\theta)$. After that, we set a sampling probability $p(\theta)$ using a smooth approximation of the Heaviside step function with the logistic function (Davies, 2002):

$$p(\theta) = 1/\Big(1 + \exp\big(2\beta(\widetilde{s}(\theta) - \alpha)\big)\Big) \tag{3}$$

where α and β are hyper-parameters to control the masking ratio and smoothness, respectively. Remarkably, $p(\theta)$ becomes the previous hard thresholding (Zhang et al., 2021; Xu et al., 2021) as $\beta \to \infty$, while the smooth approximation uses the relative importance in a more calibrated way. Then, from Bernoulli distribution with a probability $p(\theta)$, gradient mask $m(\theta)$ is sampled as follow:

$$m(\theta) \sim \text{Bernoulli}\big(p(\theta)\big) \tag{4}$$

Finally, we selectively update the model parameters using the masked gradient $\widetilde{g}(\theta)$ from the obtained mask $m(\theta)$ and scaling term $1/p(\theta)$ with a learning rate η:

$$\theta \leftarrow \theta - \eta \cdot \widetilde{g}(\theta), \;\; \widetilde{g}(\theta) := \big(m(\theta)/p(\theta)\big) \odot g(\theta) \tag{5}$$

To further demonstrate the proposed training scheme with the obtained masked gradient $\widetilde{g}(\theta)$, we provide the following theoretical backup based on the results by Xu et al. (2021).

Corollary. *Under the mild assumptions, the obtained masked gradients $\widetilde{g}(\theta)$ becomes an unbiased estimator of the original gradient $g(\theta)$, i.e.,* $\mathbb{E}[\widetilde{g}(\theta)] = g(\theta)$.

[1]Instead of adding discrete token-level adversarial perturbation (Jin et al., 2020), we consider embedding-level continuous perturbation (Zhu et al., 2020) that add noise on the word embedding of each token.

Algorithm 2 `RAM`: Robustifying Fine-tuning via Adversarial Training with Masked Gradients

```
**Input:** Classifier from a pre-trained model $f_\Theta$, training data $\mathcal{D}$, learning rate $\eta$, mask update frequency $T_\mathcal{M}$, masking ratio $\alpha$, smoothness factor $\beta$, adversarial noise magnitude $\delta$, coefficient for consistency regularization $\lambda$
$\mathcal{G}(\Theta) \leftarrow \text{InitGrad}(f_\Theta, \mathcal{D})$   // Obtaining initial gradient information from pre-trained model
for each iteration t do
    if t % $T_\mathcal{M}$ = 0 then
        $\{s(\theta)|\theta \in \Theta\} \leftarrow \mathcal{G}(\Theta),\ \mathcal{G}(\Theta) \leftarrow \emptyset$ // Get relative importance
        $\{m(\theta)|\theta \in \Theta\} \leftarrow \texttt{GetMask}(\{s(\theta)\}, \alpha, \beta)$ // Sample gradient mask from Algorithm 1
    end if
    $(\mathbf{x}, y) \sim \mathcal{D}$ // Draw a training data (e.g., mini-batch)
    $\widetilde{\mathbf{x}} \leftarrow \mathbf{x} + \delta \cdot (\partial \mathcal{L}_{\texttt{task}} / \partial \mathbf{x}) / ||\partial \mathcal{L}_{\texttt{task}} / \partial \mathbf{x}||_\infty$ // Construction of adversarially perturbed samples
    $g(\theta) \leftarrow \partial \mathcal{L}_{\texttt{train}} / \partial \theta,\ \mathcal{L}_{\texttt{train}} = \mathcal{L}_{\texttt{task}}(\mathbf{x}, y) + \mathcal{L}_{\texttt{task}}(\widetilde{\mathbf{x}}, y) + \lambda \mathcal{L}_{\texttt{cons}}(\mathbf{x}, \widetilde{\mathbf{x}})$ // Back-propagation
    $\theta \leftarrow \theta - \eta \cdot \big((m(\theta)/p(\theta)) \odot g(\theta)\big)$ // Update parameter with masked gradients
    $\mathcal{G}(\theta) \leftarrow \mathcal{G}(\theta) \cup g(\theta)$ // Obtaining gradient information from training gradients
end for
```

We present the formal derivation of Corollary in Appendix B. In addition, we empirically observe that the proposed method can work without scaling term and it sometimes outperforms the original. Hence, we consider the apply of scaling or not as the additional hyper-parameter. In summary, we accumulate the gradients during each training epoch to calculate relative importance, and then generate the gradient masks for the next epoch. Overall procedure of RAM is described in Algorithm 2. Also, more details of RAM are in Appendix A.3.

4 Experiments

In this section, we evaluate the effectiveness of our algorithm for the robustness of fine-tuned language models using two important NLP tasks: (1) sentiment classification and (2) entailment task. We first describe the experimental setups, including the details how the robustness benchmarks are constructed, in Section 4.1. In Section 4.2, we present empirical evaluations on RAM and other baseline algorithms with two constructed robustness benchmarks. Finally, in Section 4.3, we provide additional results on our method regarding (a) compatibility with various language models, (b) ablation of each component, (c) qualitative analysis.

4.1 Experimental setups

Tasks and metrics. With two popular NLP tasks, sentiment classification and entailment tasks, we develop benchmarks for measuring the robustness of trained model in a unified way. We use SST-2 (Socher et al., 2013) and MNLI (Williams et al., 2018) for training on each task, respectively.

• *In-distribution performance* ($\text{Acc}_{\texttt{in}}$): To measure the performance with respect to training distribution (*i.e.,* in-distribution), we measure the accuracy on the validation sets provided from both datasets, following a usual practice (Wang et al., 2019).

• *Distribution-shift generalization* ($\text{Acc}_{\texttt{shift}}$): To evaluate the model's capability on distribution-shift (*i.e.,* out-of-domain) generalization, we measure the average accuracy on multiple distribution shifted datasets. For entailment task, we follow the setups in (Liu et al., 2022); 7 different entailment datasets are used to evaluate the entailment classifier. For sentiment classification, we use 5 different sentiment datasets following the setups in (Potts et al., 2021).

• *Adversarial robustness* ($\text{Acc}_{\texttt{adv}}$): To measure the adversarial robustness of model, we first construct the text-level adversarial examples using TextFooler (Jin et al., 2020) on vanilla fine-tuned BERT and RoBERT models, following (Wang et al., 2021a). We also consider the datasets constructed via dynamic adversarial data collection with human-in-the-loop (Nie et al., 2020; Potts et al., 2021). In addition, we use the datasets from a recent benchmark for adversarial robustness, AdvGLUE (Wang et al., 2021b), which incorporate the various types of adversarial noises. At the end, we report the average performance on these multiple datasets.

• *Model calibration* (ECE): To measure the model's calibration performance, we report the average Expected Calibration Error (Guo et al., 2017), denoted ECE, calculated during the all evaluations on different datasets including in-distribution, distribution-shifted, and adversarial datasets. As a lower ECE indicates a better calibration unlike other considered robustness metrics, we denote it with (↓).

• *Anomaly detection* (AUROC): To measure the performance about the anomaly detection, we use the multiple external datasets as anomaly samples following the setups in the recent related works (Hendrycks et al., 2020; Zhou et al., 2021). We use the maximum softmax probability score function (Hendrycks et al., 2020) for the evaluation method, and AUROC as the evaluation metric, respectively.

To evaluate a given model by considering multiple robustness aspects in a unified way, we first report average relative improvement $\Delta_{\texttt{avg}}$ compared to vanilla algorithm across different evaluation metrics:

$$\Delta_{\texttt{avg}} := \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\Delta_s,\ \Delta_s = \frac{s - s_{\texttt{base}}}{s_{\texttt{max}} - s_{\texttt{base}}},\ 0 \leq \Delta_s \leq 1 \tag{6}$$

where $\mathcal{S} = \{\text{Acc}_{\texttt{in}}, \text{Acc}_{\texttt{shift}}, \text{Acc}_{\texttt{adv}}, \text{ECE}, \text{AUROC}\}$. s and $s_{\texttt{base}}$ are the performances of the given model and vanilla model, respectively. $s_{\texttt{max}}$ denotes the best score of each metric: 100 for accruacy metrics, 0 for ECE, and 1 for AUROC. In addition, we report the average rank, $\text{Rank}_{\texttt{avg}}$, among multiple baseline algorithms averaged across five different measurements in $\mathcal{S}$. More details about the datasets for each robustness measurement are presented in Appendix A.1.

Baselines. We compare our method with various baseline fine-tuning algorithms in NLP tasks. We first consider a naïve fine-tuning method, denoted by Vanilla. Then, we first consider a wide range of perturbation-based fine-tuning algorithms denoted by (*1a*) WordDrop (Guo et al., 2020): dropping input words; (*1b*) LayerDrop (Fan et al., 2020): dropping some blocks; (*1c*) DropHead (Zhou et al., 2020): dropping some heads within attention layer; (*1d*) HiddenCut (Chen et al., 2021a): dropping spanned hidden features; (*1e*) AdvWeight (Bahri et al., 2022): adding adversarial perturbation on the model parameters; (*1f*) AdvEmbed (Zhu et al., 2020): adding adversarial perturbation on the word embeddings. On the other hand, we also consider the recent state-of-the-art algorithms to preserve the generalizable knowledge of pre-trained language models during fine-tuning: (*2a*) R3F (Aghajanyan et al., 2021): noise-based consistency regularization to prevent representation collapse; (*2b*) Weight Consolidation (WConsol) (Chen et al., 2020): incorporation of ℓ_2 distance between trained and pre-trained models as a regularization; (*2c*) Child-tuning (C-Tune) (Xu et al., 2021): selective update of model parameters with a sampled child model; (*2d*) LP-FT (Kumar et al., 2022): two-step strategy of linear probing and then full fine-tuning. More details of baselines are presented in Appendix A.2.

Training details. All models are trained using AdamW (Loshchilov & Hutter, 2019) with its default parameters $(\beta_1, \beta_2, \epsilon)$=(0.9, 0.98, 1e-6) and a fixed weight decay of 0.01. We use a linear learning rate decay with fixed warmup ratio 0.06 and learning rate η=1e-5 (Liu et al., 2019). Then, the models are fine-tuned using the specified method with batch size 16 for 10 epochs. Except the experiments with Table 3, all the experiments are conducted with RoBERTa-large. Both of baselines and our method are optimized with its own hyper-parameters from a fixed set of candidates based on the validation set, with presented candidates in Appendix A.2. In the case of RAM, we use a fixed $\delta = 0.1$ with $\lambda \in \{0.01, 0.1, 0.5\}$ for its adversarial training. For the hyper-parameters of gradient masking, we use $\alpha \in [0.6, 0.95]$ and $\beta \in \{1, 5, 10\}$ along with the application of a scaling term. As described in Section 3.2, we accumulate the gradient information through each training epoch, then sample the gradient mask from them for the next epoch. More details can be found in Appendix A.3.

4.2 Experimental results on robustness benchmarks

To verify the effectiveness of the proposed RAM for improving the robustness of language models, we evaluate RAM with various baselines under the scenario of fine-tuning RoBERTa-large model (Liu et al., 2019). As described in Section 3.1, we use SST-2 and MNLI datasets for the training on both sentiment classification and entailment tasks, respectively; then, the robustness of fine-tuned model is evaluated with a unified viewpoint of model robustness using the constructed benchmarks. Table 1 and 2 summarizes the experimental results on sentiment classification and entailment task, respectively. First, it is worth noting that the common evaluation method using the accuracy on given validation set is actually not enough to fully capture the effectiveness of given model with the consideration of robustness. For example, all the baselines successfully improve the vanilla fine-tuning in the aspect of validation accuracy, but their robustness could be rather degraded sometimes as denoted in Table

Table 1: Robustness measurements of RoBERTa-large fine-tuned on SST-2 dataset for sentiment classification. All the values and error bars are mean and standard deviation across 3 random seeds. The best and the second best results are indicated in **bold** and underline, respectively.

Method	$\mathrm{Acc}_{\mathtt{in}}$	$\mathrm{Acc}_{\mathtt{shift}}$	$\mathrm{Acc}_{\mathtt{adv}}$	ECE (↓)	AUROC	$\Delta_{\mathtt{avg}}$	$\mathrm{Rank}_{\mathtt{avg}}$
Vanilla	96.29$_{\pm0.14}$	91.79$_{\pm0.13}$	66.30$_{\pm2.14}$	7.11$_{\pm0.82}$	86.72$_{\pm3.60}$	0.00	11.0
WordDrop	96.44$_{\pm0.03}$	89.95$_{\pm0.70}$	69.46$_{\pm0.69}$	7.33$_{\pm0.78}$	87.57$_{\pm1.91}$	-1.14	10.8
LayerDrop	96.60$_{\pm0.05}$	91.87$_{\pm0.12}$	69.65$_{\pm0.79}$	5.96$_{\pm0.28}$	89.19$_{\pm0.24}$	10.76	5.6
DropHead	96.48$_{\pm0.14}$	91.50$_{\pm0.23}$	69.53$_{\pm0.14}$	5.90$_{\pm0.10}$	88.81$_{\pm0.40}$	8.78	8.4
R-Drop	96.44$_{\pm0.19}$	91.75$_{\pm0.21}$	69.00$_{\pm1.52}$	6.05$_{\pm0.87}$	89.19$_{\pm1.20}$	9.02	8.6
HiddenCut	96.67$_{\pm0.34}$	91.14$_{\pm0.20}$	70.32$_{\pm0.78}$	5.47$_{\pm0.43}$	89.91$_{\pm0.38}$	12.26	5.2
AdvWeight	96.41$_{\pm0.29}$	91.92$_{\pm0.15}$	65.47$_{\pm0.77}$	7.36$_{\pm0.34}$	87.80$_{\pm0.98}$	1.33	10.0
AdvEmbed	96.48$_{\pm0.05}$	91.75$_{\pm0.32}$	69.90$_{\pm0.90}$	5.51$_{\pm0.16}$	**90.79**$_{\pm0.35}$	13.70	5.0
ChildPTune	96.56$_{\pm0.04}$	91.75$_{\pm0.23}$	69.54$_{\pm0.14}$	5.57$_{\pm0.15}$	87.00$_{\pm0.15}$	7.98	7.4
R3F	96.56$_{\pm0.09}$	91.79$_{\pm0.09}$	69.13$_{\pm1.55}$	5.83$_{\pm0.15}$	88.49$_{\pm0.49}$	9.38	7.4
WConsol	96.60$_{\pm0.22}$	92.15$_{\pm0.25}$	70.86$_{\pm0.12}$	5.01$_{\pm0.27}$	89.61$_{\pm1.03}$	15.47	2.8
LP-FT	96.33$_{\pm0.25}$	91.85$_{\pm0.17}$	72.48$_{\pm0.05}$	**4.05**$_{\pm0.20}$	89.46$_{\pm0.77}$	16.75	5.0
RAM (Ours)	**96.87**$_{\pm0.20}$	**92.38**$_{\pm0.12}$	**72.57**$_{\pm0.53}$	5.45$_{\pm0.40}$	90.37$_{\pm0.65}$	**18.39**	**1.6**

Table 2: Robustness measurements of RoBERTa-large fine-tuned on MNLI dataset for entailment task. All the values and error bars are mean and standard deviation across 3 random seeds. The best and the second best results are indicated in **bold** and underline, respectively.

Method	$\mathrm{Acc}_{\mathtt{in}}$	$\mathrm{Acc}_{\mathtt{shift}}$	$\mathrm{Acc}_{\mathtt{adv}}$	ECE (↓)	AUROC	$\Delta_{\mathtt{avg}}$	$\mathrm{Rank}_{\mathtt{avg}}$
Vanilla	89.97$_{\pm0.04}$	64.31$_{\pm0.58}$	48.60$_{\pm1.31}$	12.64$_{\pm0.79}$	92.09$_{\pm2.26}$	0.00	8.4
WordDrop	90.35$_{\pm0.17}$	63.20$_{\pm0.44}$	**52.45**$_{\pm0.78}$	12.17$_{\pm0.04}$	87.97$_{\pm1.40}$	-8.15	7.8
LayerDrop	90.55$_{\pm0.10}$	62.44$_{\pm0.13}$	49.50$_{\pm0.55}$	11.37$_{\pm0.13}$	89.89$_{\pm0.60}$	-3.20	8.0
DropHead	90.59$_{\pm0.09}$	63.03$_{\pm0.69}$	50.63$_{\pm0.45}$	11.09$_{\pm0.47}$	90.98$_{\pm2.07}$	0.88	6.0
R-Drop	**90.64**$_{\pm0.03}$	62.74$_{\pm0.13}$	51.24$_{\pm0.90}$	11.73$_{\pm0.32}$	91.89$_{\pm0.47}$	2.45	5.4
HiddenCut	90.48$_{\pm0.18}$	63.84$_{\pm0.26}$	50.43$_{\pm0.12}$	11.83$_{\pm0.26}$	91.64$_{\pm0.39}$	1.60	6.4
AdvWeight	90.19$_{\pm0.13}$	62.87$_{\pm0.57}$	48.00$_{\pm0.72}$	12.38$_{\pm0.71}$	91.01$_{\pm0.93}$	-2.97	10.4
AdvEmbed	90.24$_{\pm0.07}$	64.23$_{\pm0.38}$	50.20$_{\pm0.72}$	12.48$_{\pm0.71}$	**93.83**$_{\pm0.52}$	5.72	6.4
ChildPTune	90.08$_{\pm0.07}$	64.09$_{\pm0.84}$	46.48$_{\pm1.00}$	13.63$_{\pm3.29}$	86.60$_{\pm5.90}$	-16.14	11.2
R3F	90.41$_{\pm0.07}$	63.54$_{\pm0.31}$	50.29$_{\pm0.49}$	11.39$_{\pm1.25}$	91.81$_{\pm1.41}$	2.31	6.6
WConsol	90.54$_{\pm0.01}$	**65.04**$_{\pm0.55}$	49.00$_{\pm0.10}$	**10.84**$_{\pm0.90}$	91.49$_{\pm1.40}$	2.99	4.8
LP-FT	90.42$_{\pm0.14}$	64.70$_{\pm0.54}$	48.82$_{\pm0.90}$	12.64$_{\pm0.39}$	93.63$_{\pm0.88}$	5.11	6.4
RAM (Ours)	**90.64**$_{\pm0.11}$	63.95$_{\pm0.51}$	51.33$_{\pm0.34}$	11.02$_{\pm0.45}$	93.25$_{\pm0.15}$	**7.63**	**2.8**

2. These results support the necessity of consideration for model robustness with various aspects in a unified way, rather than only focusing on validation accuracy. Also, it is noticeable that there is no single best method when the multiple aspects of model robustness are considered, which would indicate the value of single unified measurement to facilitate the evaluation.

On the other hand, it is observed that RAM consistently outperforms the baseline fine-tuning methods. To be specific, across 4 different robustness metrics along with a validation accuracy, RAM exhibits 18.39 % and 7.63% average relative improvement compared to the vanilla fine-tuning on both sentiment classification and entailment tasks, respectively. Furthermore, compared to the previous best fine-tuning methods in the aspect of unified robustness, RAM outperforms them with the margins of 1.64% and 1.89% for both tasks, respectively. Consequently, our method achieves the average ranking of 2.2 while the previous best method achieves 3.8. These results demonstrate that the proposed RAM could serve as a simple yet strong method for robustifying language models. The results on each dataset are presented in Appendix D.

4.3 Additional Analyses

Effectiveness on different language models. To further demonstrate the practical usefulness of our RAM, we verify its effectiveness across the different types of pre-trained language models.

Table 3: Robustness with different language models. Average relative improvements ($\Delta_{\texttt{avg}}$) compared to vanilla fine-tuned BERT-large are reported for each model. All the values is mean across 3 random seeds. The best result of each language model is indicated in **bold**. More detailed experimental results, such as the performances on individual robustness metrics, are presented in Appendix C.

	Entailment				Sentiment Classification			
Model	Vanilla	AdvEmbed	WConsol	RAM	Vanilla	AdvEmbed	WConsol	RAM
BERT-large	00.00	22.85	20.23	**25.34**	00.00	18.24	15.85	**22.76**
RoBERTa-large	37.98	39.35	40.75	**41.31**	38.40	44.94	47.35	**48.33**
ALBERT-xxlarge	42.86	42.74	41.43	**44.29**	24.75	45.34	42.36	**51.13**
XLNet-large	32.79	34.48	33.19	**36.04**	37.24	37.03	34.38	**41.46**
ELECTRA-large	39.47	41.36	38.49	**42.96**	47.85	48.24	46.21	**49.76**
DeBERTa-large	36.18	39.73	37.74	**44.48**	40.21	44.84	42.33	**52.23**

Table 4: Ablation study with each component of RAM on the sentiment classification with RoBERTa-large. All the values and error bars are mean and standard deviation across 3 random seeds. The best and the second best results are indicated in **bold** and underline, respectively.

Method	AT	$s(\theta)$	$p(\theta)$	$m(\theta)$	$\text{Acc}_{\texttt{in}}$	$\text{Acc}_{\texttt{shift}}$	$\text{Acc}_{\texttt{adv}}$	ECE (↓)	AUROC	$\Delta_{\texttt{avg}}$
Vanilla	-	-	-	-	$96.29_{\pm 0.14}$	$91.79_{\pm 0.13}$	$66.30_{\pm 2.14}$	$7.11_{\pm 0.82}$	$86.72_{\pm 3.60}$	0.00
AdvEmbed	✓	-	-	-	$96.48_{\pm 0.05}$	$91.75_{\pm 0.32}$	$69.90_{\pm 0.90}$	$5.51_{\pm 0.16}$	$\underline{90.79}_{\pm 0.35}$	13.70
RAM (Ours)	-	✓	-	-	$96.67_{\pm 0.19}$	$\underline{91.99}_{\pm 0.03}$	$70.25_{\pm 0.12}$	$5.65_{\pm 0.48}$	$89.92_{\pm 0.83}$	13.80
	✓	✓	-	-	$\underline{96.71}_{\pm 0.24}$	$91.88_{\pm 0.26}$	$\underline{72.01}_{\pm 0.73}$	$\mathbf{5.14}_{\pm 0.31}$	$89.66_{\pm 0.73}$	$\underline{15.83}$
	✓	✓	✓	-	$96.44_{\pm 0.25}$	$91.65_{\pm 0.26}$	$70.48_{\pm 1.21}$	$6.17_{\pm 0.71}$	$\mathbf{91.23}_{\pm 1.94}$	12.28
	✓	✓	✓	✓	$\mathbf{96.87}_{\pm 0.20}$	$\mathbf{92.38}_{\pm 0.12}$	$\mathbf{72.57}_{\pm 0.53}$	$\underline{5.45}_{\pm 0.40}$	$90.37_{\pm 0.65}$	**18.39**

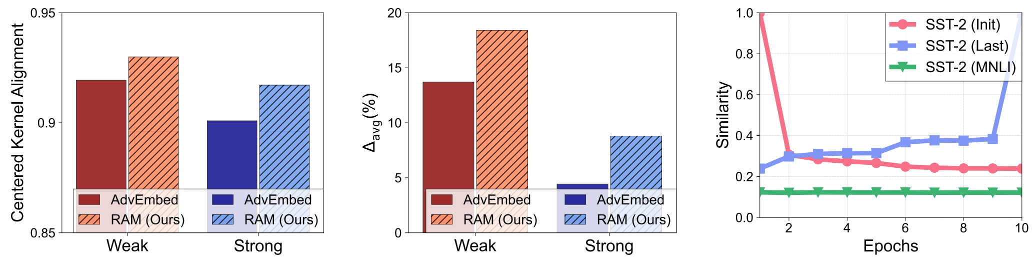
Specifically, in addition to RoBERTa-large used in Section 4.2, we conduct the additional experiments with five recent state-of-the-art language models with similar number of model parameters: BERT-large (Kenton & Toutanova, 2019), ALBERT-xxlarge (Lan et al., 2020), XLNet-large (Yang et al., 2019), ELECTRA-large (Clark et al., 2020), and DeBERTa-large (He et al., 2021). In Table 3, we present the experimental results by comparing the average relative improvement from RAM compared to two representative baselines, AdvEmbed and WConsol, that show the large improvements in Table 1 and 2. We note that the best hyper-parameters found in Section 4.2 are inherited without additional cost from the separate search. Also, to facilitate the comparison between different language models, we calculate the average relative improvement using the vanilla fine-tuned BERT as the common baseline for $s_{\texttt{base}}$ in Eq. 6. Here, one can verify that RAM significantly improves the robustness of fine-tuned models regardless of types of language models. More interestingly, RAM could be useful to reveal the true potential of language models; DeBERTa-large becomes the most robust one with our method while it was originally far from that one.

Ablation study. Here, we conduct an ablation study to understand further how RAM works. Specifically, we fine-tune RoBERTa-large on SST-2 dataset by varying the specific components of RAM: (a) adversarial training (Eq. 1) and selective update of the model parameters with (b) thresholding using collected relative importance (Eq. 2), (c) scaling with smooth approximation (Eq. 3), and (d) sampling from Bernoulli distribution (Eq. 4). Then, the robustness of each method is evaluated using the constructed robustness benchmark for sentiment classification with the same way in Table 1, Table 4 summarizes the results. First, it is observed that the adversarial training is solely effective for improving the robustness by learning to be robust for peturbation. Also, one can verify the effectiveness of collected relative importance $s(\theta)$, as it significantly improves the robustness under a simple hard thresholding; remarkably, the larger improvement is observed when it is combined with adversarial training. However, the proposed smooth approximation of step function is not solely enough when it is used as continuous scaling term rather than discrete mask, as it becomes inefficient to keep the generalizable knowledge within pre-trained language models. By sampling discrete mask from Bernoulli distribution, its true potential is revealed along with the large improvements.

Additionally, we conduct experiments to verify the effect of different policies of gradient masking. Instead of focusing on the relatively important parameters as RAM has done (Eq. 3), *Min* gives more weights on the unimportant ones by considering the reversed order to obtain the normalized score, *i.e.,* $\widetilde{s}(\theta_{\texttt{max}}) = 0 \leq \widetilde{s}(\theta) \leq 1 = \widetilde{s}(\theta_{\texttt{min}})$. *Rand* randomly assigns $\widetilde{s}(\theta)$ regardless of consideration of the relative importance score $s(\theta)$. From Table 5, one can verify that the effectiveness of *Min*

Table 5: Robustness of fine-tuned RoBERTa-large with RAM under different masking policies. All the values are mean across 3 random seeds and results with variance are presented in Appendix D.

	Entailment						Sentiment Classification					
Method	$\text{Acc}_{\texttt{in}}$	$\text{Acc}_{\texttt{shift}}$	$\text{Acc}_{\texttt{adv}}$	ECE (↓)	AUROC	$\Delta_{\texttt{avg}}$	$\text{Acc}_{\texttt{in}}$	$\text{Acc}_{\texttt{shift}}$	$\text{Acc}_{\texttt{adv}}$	ECE (↓)	AUROC	$\Delta_{\texttt{avg}}$
Vanilla	89.97	64.31	48.60	12.64	92.09	0.00	96.29	91.79	66.30	7.11	86.72	0.00
Min	90.68	64.37	50.87	11.89	90.89	0.44	96.22	91.97	71.45	5.61	86.70	7.26
Rand	90.65	64.13	50.98	11.39	91.13	1.67	96.22	92.25	72.68	5.19	90.03	14.88
RAM	90.64	63.95	51.33	11.02	93.25	7.63	96.87	92.38	72.57	5.45	90.37	18.39



(a) Distance to pre-trained LM (b) Effect on robustness (c) Similarity of sampled masks

Figure 2: Qualitative analysis of RAM. (a) Similarity between fine-tuned and initial pre-trained models. (b) Different robustness with the strengh of adversarial training and improvement from RAM. (c) Dynamics of the sampled gradient masks during the training.

and *Rand* is largely degraded compared to RAM. Remarkably, *Min* shows worsen results than *Rand*, which further demonstrate the importance of focusing on the task-relavant model parameters.

Qualitative analysis. We further present qualitative analysis on RAM. Here, we consider RoBERTa-large on SST-2 dataset with two different strength of adversarial training to facilitate the analysis: *Weak* ($\lambda = 0.01$) and *Strong* ($\lambda = 0.5$). Then, we measure the similarity between fine-tuned model and initial pre-trained model using centered kernel alignment (CKA) (Kornblith et al., 2019); specifically, average CKA between each layer's hidden features across training epochs is measured. In Figure 2(a), it is observed that the stronger adversarial training incurs the larger deviation from the pre-trained model, and it could be effectively prevented by RAM. In this way, RAM helps fine-tuned model to preserve the generalizable knowledge of pre-trained one and hence improves the model robustness as shown in Figure 2(b). In addition, we further investigate the dynamics of gradient mask (Eq. 4) by measuring the intersect over union between (1) mask at the first epoch and other epochs (*SST-2 (Init)*), (2) mask at the last epoch and other epochs (*SST-2 (Last)*), and (3) masks from SST-2 and MNLI at each epoch (*SST-2 (MNLI)*). As the model is trained by focusing on the few task-relevant parameters with RAM, the sampled mask becomes task-specific and far from the initial one as training goes on (Figure 2(c)). Adaptability of RAM to the task is further observable from a low similarity between different tasks' masks.

5 Conclusion

In this paper, we propose to consider various aspects of model robustness for the reliable deployment of language models in real-world settings, rather than focusing only on their performance evaluated on a given validation set. To improve the robustness of fine-tuned language models, we present a simple yet effective training method (RAM) by leveraging the advantages of adversarial training while preventing its potential risk with an efficient gradient masking method. Through the extensive experiments with the constructed benchmarks for unified evaluation of model robustness, we demonstrate the effectiveness of our fine-tuning method and its generalizability to several recent state-of-the-art language models. As the investigation of multiple aspects of model robustness in a unified viewpoint is under-explored in the literature despite its practical importance, we expect our work to contribute to this research direction to enable us to use the well-performing pre-trained language models with more reliability. Furthermore, since our proposed method of robust fine-tuning is task- and domain-agnostic, we believe that RAM can benefit other NLP tasks (e.g., question answering) and domains (e.g., vision and graph) as well, as interesting future work directions.

Ethics statement

In this paper, the robustness issue of language models has been extensively investigated and a simple yet effective fine-tuning algorithm, RAM, has been proposed to address this issue. Similar to other fine-tuning methods, the behavior of trained language models with the proposed method might highly rely on the training dataset. Therefore, they can suffer from the inherent problems of the given dataset, *e.g.,* gender bias (Bordia & Bowman, 2019) or annotation artifacts (Gururangan et al., 2018). However, as the proposed idea of RAM is general and can be easily extended to other training objective, we believe that the concerns on such problems could be alleviated by incorporating the recently proposed methods for those problems (Sagawa et al., 2020; Moon et al., 2021); for example, one could add a new training objective in Eq.1 to address each problem while keeping the idea of gradient masking from gradients of the overall objective (Eq.2).

Reproducibility statement

We describe the implementation details of the method in Section 4.1 and Appendix A.3. Also, we provide the details of the datasets and baselines in Appendix A and A.2, respectively. We also have a plan to publicly release the code upon the acceptance, and all the used packages are provided along with the code. In our experiments, we mainly use NVIDIA A100 GPUs.

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations (ICLR)*, 2021.

Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pp. 131–198, 2016.

Shikha Bordia and Samuel R Bowman. Identifying and reducing gender bias in word-level language models. *arXiv preprint arXiv:1904.03035*, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jiaao Chen, Dinghan Shen, Weizhu Chen, and Diyi Yang. Hiddencut: Simple data augmentation for natural language understanding with better generalizability. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021a.

Jifan Chen, Eunsol Choi, and Greg Durrett. Can nli models verify qa systems' predictions? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3841–3854, 2021b.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations (ICLR)*, 2020.

Brian Davies. *Integral transforms and their applications*, volume 41. Springer Science & Business Media, 2002.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pp. 70–74, 2016.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations (ICLR)*, 2020.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations (ICLR)*, 2021.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Demi Guo, Yoon Kim, and Alexander M Rush. Sequence-level mixed sample data augmentation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations (ICLR)*, 2021.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song, and Jacob Steinhardt. Pixmix: Dreamlike pictures comprehensively improve safety measures. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, 2020.

Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations (ICLR)*, 2020.

Pride Kavumba, Naoya Inoue, Benjamin Heinzerling, Keshav Singh, Paul Reisert, and Kentaro Inui. When choosing plausible alternatives, clever hans can be clever. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pp. 33–42, 2019.

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

Jaehyung Kim, Dongyeop Kang, Sungsoo Ahn, and Jinwoo Shin. What makes better augmentation strategies? augment difficult but not too different. In *International Conference on Learning Representations (ICLR)*, 2022.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, 2019.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations (ICLR)*, 2022.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations (ICLR)*, 2020.

Ken Lang. Newsweeder: Learning to filter netnews. In *International Conference on Machine Learning (ICML)*, 1995.

Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. Bert-attack: Adversarial attack against bert using bert. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Alisa Liu, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. Wanli: Worker and ai collaboration for natural language inference dataset creation. *arXiv preprint arXiv:2201.05955*, 2022.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165–172, 2013.

Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Seung Jun Moon, Sangwoo Mo, Kimin Lee, Jaeho Lee, and Jinwoo Shin. Masker: Masked keyword regularization for reliable text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Junhyun Nam, Jaehyung Kim, Jaeho Lee, and Jinwoo Shin. Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. In *International Conference on Learning Representations (ICLR)*, 2022.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. Ssmba: Self-supervised manifold based data augmentation for improving out-of-domain robustness. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. Dynasent: A dynamic benchmark for sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. Natural language understanding with the quora question pairs dataset. *arXiv preprint arXiv:1907.01041*, 2019.

Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221, 2017.

Emily Sheng and David C Uthus. Investigating societal biases in a poetry composition system. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pp. 93–106, 2020.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*, 2018.

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2019.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. Infobert: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations (ICLR)*, 2021a.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021b.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, 2018.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. R-drop: Regularized dropout for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. Raise a child in large language model: Towards effective and generalizable fine-tuning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning (ICML)*, 2018.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Cenyuan Zhang, Xiang Zhou, Yixin Wan, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. Improving the adversarial robustness of nlp models by information bottleneck. In *Findings of Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Dinghuai Zhang, Kartik Ahuja, Yilun Xu, Yisen Wang, and Aaron Courville. Can subnetwork structure be the key to out-of-distribution generalization? In *International Conference on Machine Learning (ICML)*, 2021.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Wangchunshu Zhou, Tao Ge, Furu Wei, Ming Zhou, and Ke Xu. Scheduled drophead: A regularization method for transformer models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

Wenxuan Zhou, Fangyu Liu, and Muhao Chen. Contrastive out-of-distribution detection for pretrained transformers. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2020.

Appendix

Robustifying Language Models via Adversarial Training with Masked Gradient

A Details on Experimental Setups

A.1 Datasets for robustness benchmarks

As described in Section 4.1, we consider two popular NLP tasks, sentiment classification and entailment tasks, to verify the multiple aspects of model robustness with fine-tuned LMs. To fine-tune LMs for both tasks, we use SST-2 (Socher et al., 2013) and MNLI (Williams et al., 2018) datasets.

- **SST-2**: a binary single sentence classification task about movie reviews with human labels of their sentiment (*positive* or *negative*). It is composed of 67k training and 872 validation samples.
- **MNLI**: a ternary entailment task which is composed of 393k training and 20k development samples. Given a pair of sentences (premise and hypothesis), the given task is to predict whether the hypothesis is an *entailment, contradiction,* or *neutral* with respect to the premise.

Both datasets are available at `https://gluebenchmark.com/tasks`. After that we measure the following aspects using the described datasets.

1. *In-distribution accuracy* ($\text{Acc}_{\texttt{in}}$): To measure the performance with respect to training distribution (*i.e.,* in-distribution), we measure the accuracy on the validation sets provided from both datasets, following a usual practice (Wang et al., 2019).

2. *Distribution-shift generalization* ($\text{Acc}_{\texttt{shift}}$): To evaluate the model's capability on distribution-shift (*i.e.,* out-of-domain) generalization, we measure the accuracy on multiple distribution shifted datasets. For sentiment classification, we use the following five different sentiment datasets based on the setups in (Potts et al., 2021).

- **Yelp** (Zhang et al., 2015): The Yelp reviews dataset consists of reviews from Yelp. It is extracted from the Yelp Dataset Challenge 2015 data. As Yelp has five star-rating categories, we bin these ratings by taking the lowest two ratings to be negative and the highest two ratings to be positive, following (Potts et al., 2021). This dataset is available at `https://huggingface.co/datasets/yelp_review_full`.
- **Amazon** (McAuley & Leskovec, 2013): The Amazon reviews dataset consists of reviews from amazon with a rating from 1 to 5. Hence, similar to Yelp, we take review score 1 and 2 as negative, and 4 and 5 as positive. This binarized dataset is available at `https://huggingface.co/datasets/amazon_polarity`.
- **IMDB** (Maas et al., 2011): IMDB is a dataset for binary sentiment classification on movie reviews. It is composed of 25,000 labeled training samples and 25,000 test samples. We use IMDB dataset provided at `https://huggingface.co/datasets/imdb`.
- **cIMDB** (Kaushik et al., 2020): Given documents and their initial labels, Kaushik et al. (2020) asked people to change each one so that it matched a counterfactual target label, as long as they avoided making any unnecessary changes to facts that were semantically unrelated to the label's applicability and produced revisions that resulted in internally consistent documents. The constructed cIMDB dataset is publicly available at `https://github.com/acmi-lab/counterfactually-augmented-data`.
- **Poem** (Sheng & Uthus, 2020): Poem is a binary single classification task about the sentiment of poem verses from Project Gutenberg. There are 892 training, 105 validation, 104 test samples, respectively. The dataset is available at `https://huggingface.co/datasets/poem_sentiment`

For entailment task, we follow the setups in the recent work (Liu et al., 2022); 7 different entailment datasets are used to evaluate the distribution-shift generalization of entailment classifier; here, some of the datasets are binary classification rather than ternary (denoted by *). Hence, following (Liu et al., 2022), the MNLI classifier is treated as binary classifier by merging the predicts as *neutral* or *contradiction* into *not entailment*. All the datasets are available at https://github.com/alisawuffles/wanli.

- **Diag** (Wang et al., 2019): NLI Diagnostics uses naturally-occurring sentences from several domains to evaluate the variety of linguistic phenomena.
- **HANS** (McCoy et al., 2019): Based on the lexical overlap between the premise and the hypothesis, HANS seeks faulty syntactic heuristics.
- **QNLI** (Wang et al., 2019): QNLI is a binary classification task that decides whether the given (question, sentence) pair contains the correct answer (entailment) or not.
- **WNLI** (Levesque et al., 2012): Winograd NIL (WNLI) is from the Winograd Schema Challenge (Levesque et al., 2012), which checks the correct coreference through common sense. By replacing the proper referent, an entailed hypothesis is created, and by replacing the incorrect referent, a non-entailed hypothesis is created.
- **NQ-NLI** (Chen et al., 2021b): Using Natural Questions QA dataset (Kwiatkowski et al., 2019), NQ-NLI creates a decontextualized sentence from the original context for the premise and a hypothesis from a question-and-answer candidate converted into a declarative form.
- **FEVER-NLI** (Thorne et al., 2018): It is adapted from the FEVER dataset (Thorne et al., 2018). In each case, the hypothesis is a statement that is either supported (implied), refuted (contradicted), or neither (neutral), and the premise is a brief context from Wikipedia.
- **WANLI** (Liu et al., 2022): Starting with an existing dataset such as MNLI, the new samples are automatically generated with GPT-3 focusing on the ambiguous samples. To further improve the quality of constructed dataset, automatic filtering is applied and then each sample is annotated by human labelers.

3. *Adversarial robustness* ($\text{Acc}_{\texttt{adv}}$): To measure the adversarial robustness of model, we first construct the text-level adversarial examples using TextFooler (Jin et al., 2020) on vanilla fine-tuned BERT and RoBERT models, following (Wang et al., 2021a). We also consider the datasets constructed via dynamic adversarial data collection with human-in-the-loop (Nie et al., 2020; Potts et al., 2021). In addition, we use the datasets from a recent benchmark for adversarial robustness, AdvGLUE (Wang et al., 2021b), which incorporate the various types of adversarial noises. Overall, for the sentiment classification, the following five different adversarially constructed datasets are used to measure the adversarial robustness of fine-tuned LMs.

- **TextFooler** (Jin et al., 2020): We denote the dataset with the adversarial texts from vanilla fine-tuned BERT as (1) *TF-B*. Similarly, the dataset with the adversarial text from vanilla fine-tuned RoBERTa is denoted as (2) *TF-R*. The official code of TextFooler is available at https://github.com/jind11/TextFooler.
- **DynaSent** (Potts et al., 2021): DynaSent is dynamically constructed through multiple iterations of training a classifier model and finding its adversarial samples by involving a human annotator in the loop. In our experiments, we use the dataset from the first round, (3) *DynaSent-R1*, and the dataset from the second round, (4) *DynaSent-R2*. As DynaSent is a ternary sentiment classification (*Positive, Neutral, and Negative*), we remove the samples with Neutral. Also, we use both validation and test sets for the evaluation. The datasets are publicly released at https://huggingface.co/datasets/dynabench/dynasent.
- **AdvGLUE** (Wang et al., 2021b): To construct principled and comprehensive benchmark for adversarial robustness in NLP tasks, Wang et al. (2021b) systematically apply 14 textual adversarial attack methods to GLUE tasks to construct AdvGLUE, which is further validated by humans for reliable annotations. Hence, we measure the robustness of sentiment classifier using the dataset for SST-2 in AdvGLUE and denote it as (5) *AdvGLUE (SST-2)*. AdvGLUE dataset is available at https://adversarialglue.github.io.

Similarly, for entailment task, we use the following nine different adversarially constructed datasets to evaluate the adversarial robustness of entailment classifier. Here, *-m* indicates that the dataset is constructed from MNLI's matched validation set and *-mm* indicates from mismatched validation set.

- **TextFooler** (Wang et al., 2019): TF-B and TF-R are defined in the same way with the case of sentiment classification. Hence, we consider (1) *TF-B-m*, (2) *TF-B-mm*, (3) *TF-R-m*, and (4) *TF-R-mm*. We remark that the corresponding datasets constructed from other researchers are available at https://drive.google.com/file/d/1xWwABFkzJ6fEnR1f3xr-vkMesxdO7IZm/view.
- **ANLI** (Nie et al., 2020): Similar to the case of DynaSent, ANLI is dynamically constructed through multiple iterations of training a classifier model and finding its adversarial samples by involving a human annotator in the loop. As there are three rounds in overall, we utilize all of these datasets: (5) *ANLI-R1*, (6) *ANLI-R2*, and (7) *ANLI-R3*. ANLI dataset is available at https://huggingface.co/datasets/anli.
- **AdvGLUE** (Wang et al., 2021b): We use the datasets for MNLI in AdvGLUE and denote it as (8) *AdvGLUE-m* and (9) *AdvGLUE-mm*.

4. *Model calibration* (ECE): To measure the model's calibration performance, we report the average Expected Calibration Error (Guo et al., 2017), denoted ECE, calculated during the all evaluations on different datasets including in-distribution, distribution-shifted, and adversarial datasets introduced in above. Namely, we report the average ECE across 11 datasets for sentiment classification and 18 datasets for entailment.

5. *Anomaly detection* (AUROC): To measure the performance about the anomaly detection, we use the following four external datasets as anomaly samples based on the setups in the recent related works (Hendrycks et al., 2020; Zhou et al., 2021).

- ☐ **WMT16** (Bojar et al., 2016): WMT is a translation dataset based on the data from statmt.org and versions exist for different years using a combination of data sources. We use the English source side of English source side of English-German WMT 16, following the previous works. WMT dataset could be downloaded from https://huggingface.co/datasets/wmt16.
- ☐ **Multi30K** (Elliott et al., 2016): Multi30K is a translation datasets which extends the Flickr30K dataset with German translations created by professional translators over a subset of the English descriptions, and independently crowd-sourced descriptions of the original English descriptions. The dataset is available at https://github.com/multi30k/dataset.
- ☐ **20 NG** (Lang, 1995): 20 Newsgroup is a dataset for topic classification consists of 20 classes. 20 NG dataset is publicly available at http://qwone.com/~jason/20Newsgroups/.
- ☐ **QQP** (Sharma et al., 2019): QQP is a binary classification datasets for entailment task, where the goal is to determine if two questions in a given pair are semantically equivalent or not. As a part of GLUE benchmark, it is available at https://huggingface.co/datasets/glue.

In addition, we consider that the one dataset becomes anomaly samples to the other, *i.e.*, SST-2 become anomaly dataset with respect to MNLI. Hence, we use total six anomaly datasets in case of sentiment classification and five datasets in case of entailment, respectively.

A.2 Baselines

In this paper, we consider various baseline fine-tuning algorithms in NLP tasks. Specifically, we first consider a wide range of perturbation-based fine-tuning algorithms and their training loss $\mathcal{L}_{\texttt{train}}$ can be described as follow:

$$\mathcal{L}_{\texttt{train}} = \mathcal{L}_{\texttt{task}}\big(f_\Theta(\mathbf{x}), y\big) + \mathcal{L}_{\texttt{task}}\big(\widetilde{f}_\Theta(\mathbf{x}), y\big) + \lambda \mathcal{L}_{\texttt{cons}}\big(f_\Theta(\mathbf{x}), \widetilde{f}_\Theta(\mathbf{x})\big) \tag{7}$$

where $\widetilde{f}_\Theta(\mathbf{x})$ indicates the perturbed prediction of model f_Θ for input $\mathbf{x}$. Also, $\mathcal{L}_{\texttt{cons}}$ is a bidirectional KL divergence introduced in Eq.1. Here, for better explanation, we slightly abuse the notations of inputs for $\mathcal{L}_{\texttt{task}}$ and $\mathcal{L}_{\texttt{cons}}$, compared to Eq.1. With Eq.7, the baselines in this categories only have a difference in how they impose the perturbation for the prediction:

- ○ **WordDrop** (Guo et al., 2020) impose the perturbation by randomly dropping the input tokens with a probability $p_{\texttt{Wd}}$ similar to Dropout. We select $p_{\texttt{WD}} \in \{0.05, 0.10, 0.15\}$.
- ○ **LayerDrop** (Fan et al., 2020) randomly drops the self-attention block of Transformer model with a probability $p_{\texttt{LD}}$. We tune $p_{\texttt{LD}} \in \{0.1, 0.2, 0.3\}$.

- ◦ **DropHead** (Zhou et al., 2020) randomly drops each head in multi-head attention module of Transformer with a probability $p_{\texttt{DH}}$. We tune $p_{\texttt{DH}} \in \{0.1, 0.2, 0.3\}$.
- ◦ **HiddenCut** (Chen et al., 2021a) drops the contiguous spans within the hidden features of Transformer model during fine-tuning. As the attention-based strategy for sampling the spans shows the best results in (Chen et al., 2021a), we adopt it and tune the HiddenCut ratio $p_{\texttt{HC}} \in \{0.1, 0.2, 0.3\}$ with the fixed selection ratio of 0.4. The official code is available at `https://github.com/SALT-NLP/HiddenCut`.
- ◦ **AdvWeight** (Bahri et al., 2022) adds adversarial noise on the model parameters rather than input. It is noteworthy that this method is also called as *Sharpness-Aware Minimization (SAM)* (Foret et al., 2021). We tune the step size ρ for gradient ascent step among $\{0.01, 0.03, 0.05\}$. We adopt the codes from `https://github.com/davda54/sam`.
- ◦ **AdvEmbed** (Zhu et al., 2020; Jiang et al., 2020) imposes adversarial perturbation on the word embeddings of input tokens. We set the same values between the magnitude of perturbation and step size for gradient ascent step; a step size δ is tuned among {1e-5, 1e-3, 1e-1} under ℓ_∞ norm.

Also, we commonly tune the hyper-parameter λ among $\{0.01, 0.1, 0.5\}$ in addition to the specific hyper-parameter of each method.

On the other hand, we also consider the recent methods that prevents the model from deviating too much from the initial pre-trained model to preserve the generalizable knowledge of pre-trained language models during fine-tuning:

- **R3F** (Aghajanyan et al., 2021) introduces a noise-based consistency regularization to prevent representation collapse. Hence, we use the same candidate for $\lambda \in \{0.01, 0.1, 0.5\}$. In addition, we consider the fixed variance of noise σ=1e-5 with the two noise distributions as additional hyper-parameter $[\mathcal{U}, \mathcal{N}]$ following the original paper (Aghajanyan et al., 2021). Also, the official code is available at `https://github.com/pytorch/fairseq`.
- **Weight Consolidation** (Chen et al., 2020) incorporates ℓ_2 distance between trained and pre-trained models as a regularization during fine-tuning. To gradually control the strength of such regularization, the authors considers the sigmoid annealing function $\lambda(t) = 1/\big(1 + \exp(-k \cdot (t - t_0))\big)$ where $t \in [0, 1]$. Here, we tune the hyper-parameters k and t_0 among $\{0.1, 0.5, 1.0\}$ and $\{0.1, 0.3, 0.5\}$, respectively. We denote it as *WConsol*. We use the official code from `https://github.com/Sanyuan-Chen/RecAdam`.
- **Child-tuning** (Xu et al., 2021) selectively update the subset of model parameters (called child network) with a fixed child model. As the task-driven approach shows the better performance compared to task-free variant in (Xu et al., 2021), we adopt the task-driven one as baseline. We tune the child network's sparsity p_D among $\{0.1, 0.2, 0.3\}$ following the original paper (Xu et al., 2021). We denote this method as *ChildTune* in our paper. Official code by the authors is publicly released at `https://github.com/PKUnlp-icler/ChildTuning`.
- **LP-FT** (Kumar et al., 2022) uses a two-step strategy of linear probing and then full fine-tuning. For a linear probing, we train the linear classifier on the frozen backbone using Adam optimizer with a fixed learning rate η = 1e-3 and 5 epochs. Then, we tune the learning rate $\eta_{\texttt{ft}}$ during the full fine-tuning among {1e-6, 3e-5, 1e-5}.

A.3 RAM

As described in Section 4.1, we use a fixed step size $\delta = 0.1$ for the gradient ascent step to construct the adversarial perturbation. Also, similar to the case of perturbation-based regularization methods, we tune the coefficient of consistency regularization $\mathcal{L}_{\texttt{cons}}$ with $\lambda \in \{0.01, 0.1, 0.5\}$ (Eq. 1). For the hyper-parameters of gradient masking, we use $\alpha \in [0.6, 0.95]$ and $\beta \in \{1, 5, 10\}$ along with the application of a scaling term. We remark that relatively higher masking ratio α has been effective for sentiment classification, while the smaller α has been effective for entailment during our experiments. Based on such observation, we tune α among $\{0.95, 0.9, 0.8\}$ for sentiment classification, $\{0.6, 0.65, 0.7\}$ for entailment task along with $\beta \in \{1, 5, 10\}$. As described in Section 3.2, we accumulate the gradient information through each training epoch, then sample the gradient mask from them for the next epoch. In case of InitGrad in Algorithm 2, similar to the setups in (Xu et al., 2021), we gather the sum of the square of gradients with respect to vanilla cross-entropy loss, since the classifier is not trained at that time.

B Proof of Corollary

In this section, we present the formal proof of Corollary in Section 3.2. To this end, we first present the theoretical results by Xu et al. (2021):

Theorem. *(Xu et al., 2021) Suppose $\mathcal{L}$ denotes the loss function on the parameter θ, for multiple data instances in the training set $\mathbf{x} \sim \mathcal{D}$, the gradients obey a Gaussian distribution $\mathcal{N}(\frac{\partial \mathcal{L}}{\partial \theta}, \sigma_g^2 \mathbb{1}_k)$. For a randomly sampled batch $\mathcal{B} \sim \mathcal{S}$, when the learning algorithm is SGD with learning rate η, the probability of the gradient mask from Bernoulli distribution is p, then the mean and covariance of the update $\Delta\theta := -\eta\big(\frac{\partial \mathcal{L}}{\partial \theta} \odot m(\theta)\big)$ are,*

$$\mathbb{E}[\Delta\theta] = -\eta \frac{\partial \mathcal{L}}{\partial \theta}, \quad \Sigma[\Delta\theta] = \frac{\eta^2 \sigma_g^2 \mathbb{1}_k}{p|\mathcal{B}|} + \frac{(1-p)\eta^2 diag\{\frac{\partial \mathcal{L}}{\partial \theta}\}^2}{p}$$

where Σ is the covariance matrix and $diag(X)$ is the diagonal matrix of the vector X.

Here, the key difference with the above problem setup (Xu et al., 2021) and our case is the probability for masking: Xu et al. (2021) assumes the identical Bernoulli distribution, while we assume the element-wise Bernoulli distribution with the different probability for each model parameter. However, in the below Corollay, we show that our problem can be also proved in the almost same way.

Corollary. *We consider the same assumption in Theorem by Xu et al. (2021), expect the probability of the gradient mask follows different Bernoulli distribution for each parameter $p(\theta)$ and $m(\theta) \sim Ber\big(p(\theta)\big)$. Then, the mean of the update $\Delta\theta$ is,*

$$\mathbb{E}[\Delta\theta] = -\eta \frac{\partial \mathcal{L}}{\partial \theta}$$

where $p_{\mathtt{min}} := \min_\theta p(\theta)$.

Proof. Let $g^{(i)}$ is the gradient of sample $\mathbf{x}_i$, $\;1 \le i \le |\mathcal{B}|$, then $g^{(i)} \sim \mathcal{N}(\frac{\partial \mathcal{L}}{\partial \theta}, \sigma_g^2 \mathbb{1}_k)$ by the assumption. Let $g = \sum_{i=1}^{|\mathcal{B}|} \frac{g_i}{|\mathcal{B}|}$, then we have

$$\Delta\theta = -\eta\Big(\sum_{i=1}^{|\mathcal{B}|} \frac{g^{(i)}}{|\mathcal{B}|}\Big) \odot m(\theta) = -\eta g \odot m(\theta)$$

Consider g, we have

$$\mathbb{E}[g] = \frac{\partial \mathcal{L}}{\partial \theta}, \Sigma[g] = \frac{\sigma_g^2 \mathbb{1}_k}{|\mathcal{B}|}$$

Suppose $\widetilde{g} := \big(m(\theta)/p(\theta)\big) \odot g$, then we have:

$$\mathbb{E}[\widetilde{g}] = \frac{p(\theta)}{p(\theta)} \times \frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \theta}$$

Table 6: Robustness measurements of six different LMs fine-tuned on SST-2 dataset for sentiment classification. All the values and error bars are mean and standard deviation across 3 random seeds.

Type of LM	Method	$\text{Acc}_{\texttt{in}}$	$\text{Acc}_{\texttt{shift}}$	$\text{Acc}_{\texttt{adv}}$	ECE (↓)	AUROC
BERT	Vanilla	$94.04_{\pm 0.28}$	$89.30_{\pm 0.54}$	$57.10_{\pm 1.65}$	$9.21_{\pm 2.01}$	$83.68_{\pm 0.69}$
	AdvEmbed	$94.42_{\pm 0.14}$	$88.75_{\pm 0.75}$	$61.16_{\pm 0.26}$	$9.46_{\pm 1.67}$	$83.40_{\pm 1.13}$
	WConsol	$93.81_{\pm 0.16}$	$89.46_{\pm 0.61}$	$56.34_{\pm 1.63}$	$9.35_{\pm 0.65}$	$85.09_{\pm 0.35}$
	RAM (Ours)	$94.07_{\pm 0.20}$	$88.87_{\pm 0.31}$	$60.31_{\pm 0.53}$	$7.00_{\pm 0.49}$	$85.89_{\pm 0.29}$
RoBERTa	Vanilla	$96.29_{\pm 0.14}$	$91.79_{\pm 0.13}$	$66.30_{\pm 2.14}$	$7.11_{\pm 0.82}$	$86.72_{\pm 3.60}$
	AdvEmbed	$96.48_{\pm 0.05}$	$91.75_{\pm 0.32}$	$69.90_{\pm 0.90}$	$5.51_{\pm 0.16}$	$90.79_{\pm 0.35}$
	WConsol	$96.60_{\pm 0.22}$	$92.15_{\pm 0.25}$	$70.86_{\pm 0.12}$	$5.01_{\pm 0.27}$	$89.61_{\pm 1.03}$
	RAM (Ours)	$96.87_{\pm 0.20}$	$92.38_{\pm 0.12}$	$72.57_{\pm 0.53}$	$5.45_{\pm 0.40}$	$90.37_{\pm 0.65}$
ALBERT	Vanilla	$95.24_{\pm 0.52}$	$88.63_{\pm 0.59}$	$64.13_{\pm 0.31}$	$8.71_{\pm 2.34}$	$88.15_{\pm 1.30}$
	AdvEmbed	$96.52_{\pm 0.39}$	$91.78_{\pm 0.17}$	$73.63_{\pm 1.69}$	$5.88_{\pm 0.30}$	$87.34_{\pm 0.54}$
	WConsol	$96.25_{\pm 0.66}$	$90.92_{\pm 1.20}$	$71.95_{\pm 4.64}$	$5.75_{\pm 1.07}$	$87.44_{\pm 2.27}$
	RAM (Ours)	$96.62_{\pm 0.06}$	$92.33_{\pm 0.24}$	$78.20_{\pm 0.64}$	$5.00_{\pm 0.10}$	$89.32_{\pm 0.20}$
XLNet	Vanilla	$95.87_{\pm 0.34}$	$90.80_{\pm 0.31}$	$68.33_{\pm 0.83}$	$6.57_{\pm 0.46}$	$86.76_{\pm 2.47}$
	AdvEmbed	$96.25_{\pm 0.29}$	$91.22_{\pm 0.61}$	$67.39_{\pm 1.42}$	$7.56_{\pm 1.59}$	$88.26_{\pm 2.82}$
	WConsol	$95.26_{\pm 0.35}$	$90.60_{\pm 0.51}$	$67.77_{\pm 1.70}$	$6.82_{\pm 0.84}$	$88.58_{\pm 3.53}$
	RAM (Ours)	$96.02_{\pm 0.30}$	$90.61_{\pm 0.46}$	$69.61_{\pm 1.29}$	$5.48_{\pm 0.62}$	$92.25_{\pm 0.30}$
ELECTRA	Vanilla	$96.96_{\pm 0.06}$	$91.62_{\pm 0.10}$	$74.15_{\pm 0.27}$	$4.92_{\pm 0.07}$	$82.43_{\pm 1.31}$
	AdvEmbed	$97.13_{\pm 0.08}$	$90.97_{\pm 0.46}$	$74.81_{\pm 0.16}$	$5.20_{\pm 1.78}$	$88.98_{\pm 1.07}$
	WConsol	$97.08_{\pm 0.06}$	$91.51_{\pm 0.24}$	$73.16_{\pm 0.69}$	$5.55_{\pm 0.08}$	$82.40_{\pm 1.82}$
	RAM (Ours)	$97.02_{\pm 0.19}$	$91.75_{\pm 0.08}$	$74.74_{\pm 3.41}$	$4.97_{\pm 0.19}$	$88.87_{\pm 0.67}$
DeBERTa	Vanilla	$96.48_{\pm 0.20}$	$90.95_{\pm 0.74}$	$69.96_{\pm 1.25}$	$6.63_{\pm 0.69}$	$86.72_{\pm 4.27}$
	AdvEmbed	$96.64_{\pm 0.05}$	$91.83_{\pm 0.44}$	$71.18_{\pm 0.50}$	$6.08_{\pm 5.72}$	$90.21_{\pm 0.86}$
	WConsol	$96.60_{\pm 0.14}$	$91.34_{\pm 1.16}$	$69.74_{\pm 0.77}$	$6.23_{\pm 0.63}$	$88.01_{\pm 2.52}$
	RAM (Ours)	$97.13_{\pm 0.09}$	$92.31_{\pm 0.24}$	$74.25_{\pm 0.11}$	$4.46_{\pm 0.03}$	$89.74_{\pm 0.25}$

C DETAILED EXPERIMENTAL RESULTS WITH VARIOUS LANGUAGE MODELS

In this section, we present the detailed experimental results on the individual robustness metrics like Table 1 and 2. In Table 6 and 7, the results on sentiment classification and entailment are presented, respectively.

Table 7: Robustness measurements of six different LMs fine-tuned on MNLI dataset for entailment task. All the values and error bars are mean and standard deviation across 3 random seeds.

Type of LM	Method	$\text{Acc}_{\texttt{in}}$	$\text{Acc}_{\texttt{shift}}$	$\text{Acc}_{\texttt{adv}}$	ECE (↓)	AUROC
BERT	Vanilla	86.25±0.11	60.20±0.59	39.25±0.30	23.03±0.58	70.88±1.64
	AdvEmbed	86.99±0.09	60.77±0.25	42.90±0.37	18.50±2.31	81.98±3.37
	WConsol	86.27±0.26	60.52±0.34	39.34±0.51	17.38±4.21	75.71±5.03
	RAM (Ours)	86.89±0.06	60.38±0.61	42.58±0.47	14.97±0.38	82.41±0.80
RoBERTa	Vanilla	89.97±0.04	64.31±0.58	48.60±1.31	12.64±0.79	92.09±2.26
	AdvEmbed	90.24±0.07	64.23±0.38	50.20±0.72	12.48±0.71	93.83±0.52
	WConsol	90.54±0.01	65.04±0.55	49.00±0.10	10.84±0.90	91.49±1.40
	RAM (Ours)	90.64±0.11	63.95±0.51	51.33±0.34	11.02±0.45	93.25±0.15
ALBERT	Vanilla	90.62±0.18	64.94±0.33	58.79±0.21	10.25±0.92	83.14±0.80
	AdvEmbed	90.60±0.15	64.19±0.14	58.82±0.25	10.92±0.68	86.63±4.49
	WConsol	90.43±0.20	64.37±0.35	56.14±1.95	9.67±0.08	80.65±3.98
	RAM (Ours)	90.72±0.10	66.57±0.03	59.24±0.20	11.83±4.61	91.47±2.63
XLNet	Vanilla	89.29±0.10	63.74±0.23	49.33±0.36	14.04±1.66	77.53±8.00
	AdvEmbed	89.46±0.09	63.56±0.94	50.52±0.90	12.69±0.54	77.31±1.67
	WConsol	89.33±0.04	63.55±0.08	49.91±0.30	14.64±1.37	81.35±7.83
	RAM (Ours)	90.03±0.04	63.69±0.52	50.11±0.14	10.28±0.37	78.53±3.07
ELECTRA	Vanilla	90.68±0.05	66.20±0.90	56.06±1.16	13.87±5.47	82.75±2.79
	AdvEmbed	90.64±0.01	65.66±0.57	56.95±0.91	12.96±2.51	88.43±0.11
	WConsol	90.51±0.13	67.21±0.41	55.43±0.84	15.36±5.44	84.05±4.02
	RAM (Ours)	91.07±0.07	64.80±0.46	58.38±0.68	10.17±0.40	81.01±2.16
DeBERTa	Vanilla	90.09±0.15	65.01±0.80	52.61±1.39	15.87±7.81	87.89±2.40
	AdvEmbed	90.43±0.04	65.73±1.02	54.91±0.30	13.29±1.94	86.37±2.27
	WConsol	90.02±0.18	64.83±1.38	53.67±0.08	14.51±7.15	89.03±0.83
	RAM (Ours)	90.97±0.09	66.07±0.31	56.53±0.66	9.94±0.34	88.15±1.73

D More experimental results

Here, we present the results on each dataset for each robustness metric. First, we present the results from sentiment classification. Specifically, we report the accuracy and ECE on distribution shifted datasets in Table 8 and 9, respectively. Also, we report the accuracy and ECE on adversarially constructed datasets in Table 10 and 11, respectively. The results of anomaly detection are shown in Table 12. Next, we present the results from entailment task; we report the accuracy and ECE on distribution shifted datasets in Table 13 and 14, respectively. Then, we report the accuracy and ECE on adversarially constructed datasets in Table 15 and 16, respectively. Finally, we report the results of anomaly detection in Table 17.

Table 8: Accuracy of RoBERTa-large fine-tuned using SST-2 dataset for sentiment classifcation task. One in-distribution validation set and five distribution shifted datasets are evaluated. All the values with larger font are mean across 3 random seeds. The values with smaller font and plus-minus sign ($\pm$) are corresponding variance. Numbers in bracket means the number of samples.

		Distribution Shifted Datasets				
Method	SST-2 (872)	Yelp (20K)	IMDB (25K)	c-IMDB (2440)	Poem (359)	Amazon (100K)
Vanilla	96.29 ±0.14	96.43 ±0.22	88.82 ±0.40	91.79 ±0.45	88.02 ±0.60	93.91 ±0.11
WordDrop	96.44 ±0.03	96.31 ±0.08	88.50 ±0.33	89.06 ±0.70	82.37 ±2.51	93.52 ±0.14
LayerDrop	96.60 ±0.05	96.37 ±0.01	89.04 ±0.05	91.54 ±0.20	88.67 ±0.57	93.70 ±0.07
DropHead	96.48 ±0.14	96.24 ±0.08	88.87 ±0.20	91.60 ±0.34	87.09 ±1.46	93.71 ±0.18
R-Drop	96.44 ±0.19	96.18 ±0.11	88.78 ±0.49	91.80 ±0.21	88.21 ±1.25	93.78 ±0.07
HiddenCut	96.67 ±0.34	96.17 ±0.18	88.89 ±0.17	91.39 ±0.45	85.79 ±0.82	93.47 ±0.20
AdvWeight	96.41 ±0.29	96.34 ±0.11	88.33 ±0.14	91.78 ±0.20	89.88 ±0.73	93.25 ±0.04
AdvEmbed	96.48 ±0.05	96.34 ±0.14	89.21 ±0.10	91.42 ±0.34	87.93 ±1.25	93.87 ±0.05
ChildPTune	96.56 ±0.04	96.69 ±0.15	89.53 ±0.11	91.93 ±0.36	86.26 ±0.95	94.33 ±0.02
R3F	96.56 ±0.09	96.24 ±0.22	88.83 ±0.16	90.97 ±0.57	89.23 ±0.66	93.67 ±0.12
WConsol	96.60 ±0.22	97.02 ±0.07	89.75 ±0.30	91.93 ±0.23	87.56 ±1.93	94.49 ±0.14
LP-FT	96.33 ±0.25	97.54 ±0.03	89.84 ±0.37	90.04 ±0.53	87.28 ±1.51	94.57 ±0.13
RAM (Ours)	96.87 ±0.20	96.90 ±0.28	89.52 ±0.07	92.10 ±0.50	89.04 ±0.57	94.32 ±0.15

Table 9: Expected Calibration Error (ECE) of RoBERTa-large fine-tuned using SST-2 dataset for sentiment classifcation task. One in-distribution validation set and five distribution shifted datasets are evaluated. All the values with larger font are mean across 3 random seeds. The values with smaller font and plus-minus sign (±) are corresponding variance. Numbers in bracket means the number of samples. Lower ECE value indicates the better calibration.

		Distribution Shifted Datasets				
Method	SST-2 (872)	Yelp (20K)	IMDB (25K)	c-IMDB (2440)	Poem (359)	Amazon (100K)
Vanilla	3.80 ±0.37	5.30 ±1.66	3.70 ±1.77	5.13 ±1.33	5.87 ±0.79	5.03 ±1.27
WordDrop	5.93 ±1.23	6.97 ±1.58	9.77 ±1.54	10.13 ±1.76	8.57 ±1.28	7.80 ±3.32
LayerDrop	3.70 ±0.54	4.60 ±0.08	7.10 ±0.22	6.33 ±0.54	7.20 ±0.42	6.53 ±1.80
DropHead	3.83 ±0.21	4.37 ±0.56	6.40 ±1.22	6.37 ±0.42	5.63 ±0.73	5.90 ±1.69
R-Drop	4.90 ±1.45	5.27 ±0.17	3.80 ±1.56	6.17 ±2.36	5.53 ±0.99	5.30 ±2.69
HiddenCut	3.10 ±0.62	4.00 ±0.85	6.70 ±1.31	6.50 ±0.16	7.23 ±0.57	6.37 ±1.36
AdvWeight	5.30 ±0.59	7.20 ±1.00	9.60 ±0.99	7.87 ±0.71	7.27 ±0.09	6.30 ±1.51
AdvEmbed	3.27 ±0.19	2.77 ±0.09	4.07 ±0.40	5.43 ±0.62	5.73 ±0.66	5.33 ±0.09
ChildPTune	3.33 ±0.34	4.43 ±0.76	2.40 ±0.67	3.87 ±1.16	5.33 ±0.26	4.20 ±1.22
R3F	4.60 ±0.86	4.30 ±1.24	6.90 ±1.70	6.60 ±0.50	5.70 ±0.86	6.13 ±1.21
WConsol	2.50 ±0.24	3.00 ±0.36	4.20 ±0.73	5.00 ±1.02	6.13 ±1.48	5.57 ±1.41
LP-FT	3.83 ±0.17	2.27 ±0.12	1.83 ±0.24	2.60 ±0.24	6.43 ±1.01	4.00 ±0.22
RAM (Ours)	3.50 ±0.49	4.03 ±0.80	7.17 ±1.32	7.07 ±0.26	6.90 ±1.61	4.80 ±1.63

Table 10: Accuracy of RoBERTa-large fine-tuned using SST-2 dataset for sentiment classification task. Five adversarially constructed datasets are evaluated. All the values with larger font are mean across 3 random seeds. The values with smaller font and plus-minus sign (±) are corresponding variance. Numbers in bracket means the number of samples.

	Adversarially Constructed Datasets				
Method	TF-B (694)	TF-R (686)	Dynasent-R1 (4800)	Dynasent-R2 (960)	AdvGLUE (148)
Vanilla	73.05 ±0.62	45.87 ±1.82	78.19 ±0.58	77.64 ±0.30	56.76 ±1.66
WordDrop	77.38 ±0.47	60.50 ±1.09	76.34 ±0.37	76.35 ±0.08	56.76 ±2.40
LayerDrop	77.62 ±0.56	61.52 ±1.76	76.54 ±0.24	77.64 ±0.26	54.95 ±1.39
DropHead	76.03 ±1.00	59.23 ±0.56	76.87 ±0.08	76.94 ±0.62	58.56 ±0.64
R-Drop	75.50 ±1.71	58.16 ±2.08	77.85 ±0.65	77.43 ±0.05	56.08 ±4.52
HiddenCut	78.15 ±0.95	60.88 ±1.07	76.33 ±0.57	77.01 ±0.26	59.23 ±1.77
AdvWeight	73.01 ±0.36	53.26 ±0.84	75.30 ±0.23	76.25 ±0.47	49.55 ±2.09
AdvEmbed	75.36 ±0.62	61.03 ±0.97	77.23 ±0.36	77.57 ±0.91	58.33 ±2.23
ChildPTune	75.55 ±1.19	54.86 ±1.92	79.57 ±0.83	78.92 ±0.18	58.78 ±2.53
R3F	76.03 ±1.01	58.94 ±2.00	77.04 ±0.44	77.12 ±0.21	56.53 ±4.59
WConsol	76.70 ±0.58	55.64 ±0.66	80.75 ±0.31	79.96 ±0.50	61.26 ±2.49
LP-FT	76.32 ±0.77	57.82 ±0.72	81.92 ±0.37	81.04 ±0.44	65.31 ±0.64
RAM (Ours)	77.09 ±0.24	61.18 ±0.86	79.79 ±0.41	79.27 ±0.47	65.54 ±2.40

Table 11: Expected Calibration Error (ECE) of RoBERTa-large fine-tuned using SST-2 dataset for sentiment classification task. Five adversarially constructed datasets are evaluated. All the values with larger font are mean across 3 random seeds. The values with smaller font and plus-minus sign (±) are corresponding variance. Numbers in bracket means the number of samples. Lower ECE value indicates the better calibration.

	Adversarially Constructed Datasets				
Method	TF-B (694)	TF-R (686)	Dynasent-R1 (4800)	Dynasent-R2 (960)	AdvGLUE (148)
Vanilla	5.57 ±1.93	19.73 ±4.63	5.53 ±1.41	4.90 ±0.45	13.60 ±5.06
WordDrop	8.53 ±1.84	3.40 ±0.14	6.97 ±2.38	7.57 ±1.50	4.97 ±0.86
LayerDrop	5.80 ±0.36	5.80 ±0.86	4.27 ±0.17	3.80 ±0.14	10.47 ±2.23
DropHead	5.67 ±1.20	8.10 ±0.94	3.03 ±0.42	3.20 ±0.16	12.40 ±2.55
R-Drop	5.30 ±1.59	8.43 ±3.04	5.40 ±2.81	5.80 ±0.50	10.63 ±3.60
HiddenCut	5.90 ±1.13	4.43 ±0.74	2.70 ±0.96	2.87 ±0.09	10.40 ±1.66
AdvWeight	5.43 ±0.52	8.30 ±1.28	4.17 ±0.62	3.90 ±1.27	15.63 ±4.15
AdvEmbed	3.97 ±0.78	9.10 ±1.77	3.23 ±0.12	4.50 ±1.07	13.17 ±2.10
ChildPTune	4.13 ±0.87	12.93 ±0.57	2.77 ±0.19	3.73 ±0.70	14.17 ±3.27
R3F	5.07 ±1.60	6.77 ±2.16	2.43 ±0.61	3.53 ±0.17	12.10 ±2.14
WConsol	6.50 ±1.00	8.10 ±0.96	3.50 ±0.36	3.27 ±0.66	7.33 ±0.53
LP-FT	5.13 ±1.17	6.00 ±0.41	3.33 ±0.12	3.27 ±0.34	5.80 ±0.43
RAM (Ours)	6.00 ±0.94	5.13 ±0.90	4.37 ±0.81	4.70 ±0.24	6.30 ±0.59

Table 12: Anomaly detection performance (AUROC) of RoBERTa-large fine-tuned using SST-2 dataset for sentiment classification task. Six anomaly datasets are evaluated. All the values with larger font are mean across 3 random seeds. The values with smaller font and plus-minus sign (±) are corresponding variance. Numbers in bracket means the number of samples.

	Anomaly Datasets					
Method	WMT16 (2999)	Multi30K (3071)	20News (592)	QQP (40K)	MNLI-m (9815)	MNLI-mm (9832)
Vanilla	83.47 ±4.41	84.87 ±7.05	96.03 ±0.45	87.33 ±3.65	84.17 ±3.86	84.47 ±4.23
WordDrop	85.40 ±3.19	88.03 ±2.66	93.90 ±0.54	86.47 ±1.39	86.20 ±2.87	85.40 ±2.86
LayerDrop	87.20 ±0.29	91.70 ±0.22	94.13 ±0.41	89.77 ±0.85	86.67 ±0.25	85.70 ±0.14
DropHead	86.30 ±0.41	92.00 ±0.93	94.87 ±0.76	88.20 ±0.16	86.17 ±0.45	85.30 ±0.41
R-Drop	86.07 ±1.03	90.87 ±1.86	96.33 ±0.17	89.97 ±1.78	86.40 ±1.21	85.53 ±1,69
HiddenCut	87.90 ±0.08	92.53 ±0.37	95.40 ±0.99	89.23 ±1.26	87.47 ±0.25	86.90 ±0.16
AdvWeight	84.60 ±1.31	90.77 ±0.87	92.93 ±0.29	88.67 ±0.94	85.07 ±1.30	84.77 ±1.65
AdvEmbed	88.30 ±0.80	91.90 ±1.18	97.13 ±0.12	89.47 ±1.38	89.30 ±0.64	88.63 ±0.82
ChildPTune	84.64 ±1.92	82.03 ±3.84	96.17 ±0.75	88.40 ±1.41	85.23 ±1.58	85.53 ±1.76
R3F	84.90 ±0.79	91.33 ±0.50	95.57 ±0.74	87.80 ±1.24	86.13 ±0.45	85.23 ±0.68
WConsol	87.50 ±1.31	90.27 ±1.39	96.30 ±0.37	90.30 ±0.91	86.90 ±1.18	86.37 ±1.34
LP-FT	86.23 ±1.02	91.57 ±0.73	95.20 ±0.45	90.60 ±0.86	87.10 ±0.83	86.03 ±1.23
RAM (Ours)	88.60 ±0.54	91.30 ±1.58	95.43 ±0.47	90.87 ±0.94	88.17 ±0.65	87.83 ±0.94

Table 13: Accuracy of RoBERTa-large fine-tuned using MNLI dataset for entailment task. Two in-distribution validation sets (MNLI-m and MNLI-mm) and seven distribution shifted datasets are evaluated. All the values with larger font are mean across 3 random seeds. The values with smaller font and plus-minus sign (±) are corresponding variance. Numbers in bracket means the number of samples.

			Distribution Shifted Datasets						
Method	MNLI-m (9815)	MNLI-mm (9832)	Diag (1104)	HANS* (30K)	QNLI* (5266)	WNLI* (635)	NQ-NLI* (4855)	FEVER-NLI (20K)	WANLI (5000)
Vanilla	90.15 ±0.06	89.80 ±0.12	66.09 ±0.55	75.70 ±0.71	60.35 ±2.76	53.91 ±1.45	61.94 ±0.74	69.62 ±0.40	62.55 ±0.77
WordDrop	90.44 ±0.15	90.26 ±0.24	64.98 ±0.64	76.70 ±1.63	54.84 ±0.53	52.86 ±0.20	61.28 ±0.31	68.82 ±0.44	62.93 ±0.79
LayerDrop	90.66 ±0.11	90.44 ±0.16	64.79 ±0.76	73.38 ±0.21	55.96 ±0.37	50.34 ±0.20	60.89 ±0.32	69.34 ±0.15	62.40 ±0.16
DropHead	90.84 ±0.12	90.35 ±0.07	65.40 ±0.74	75.46 ±3.23	55.63 ±1.88	51.60 ±1.30	60.84 ±0.86	69.65 ±0.38	62.67 ±0.69
R-Drop	90.61 ±0.19	90.67 ±0.23	65.46 ±0.56	74.82 ±0.69	54.88 ±0.92	51.34 ±0.72	60.56 ±0.05	68.92 ±0.35	63.20 ±0.19
HiddenCut	90.62 ±0.23	90.35 ±0.18	66.76 ±0.39	76.75 ±0.70	54.55 ±0.69	53.75 ±0.91	62.11 ±0.23	69.14 ±0.23	63.79 ±0.32
AdvWeight	90.39 ±0.13	90.00 ±0.15	65.94 ±0.97	74.83 ±1.56	54.91 ±1.20	51.71 ±0.83	61.29 ±1.02	69.19 ±0.39	62.21 ±0.71
AdvEmbed	90.51 ±0.03	89.97 ±0.12	67.75 ±0.94	77.77 ±0.56	55.77 ±2.64	53.23 ±0.45	61.81 ±0.26	69.29 ±0.03	63.99 ±0.60
ChildPTune	90.14 ±0.06	90.02 ±0.07	65.94 ±0.63	75.19 ±2.45	57.83 ±1.59	53.86 ±1.29	62.38 ±0.88	69.46 ±0.47	63.96 ±0.58
R3F	90.57 ±0.12	90.25 ±0.02	65.55 ±1.04	76.75 ±1.90	56.54 ±4.32	52.55 ±0.30	61.53 ±0.78	69.10 ±0.24	62.75 ±0.80
WConsol	90.71 ±0.08	90.37 ±0.07	67.09 ±0.50	76.52 ±1.65	61.64 ±2.43	55.91 ±0.68	61.28 ±0.53	70.15 ±0.25	62.72 ±0.42
LP-FT	90.57 ±0.15	90.28 ±0.13	67.60 ±0.30	77.12 ±1.40	56.85 ±1.37	55.59 ±1.70	62.28 ±0.97	69.86 ±0.15	63.63 ±0.58
RAM (Ours)	90.78 ±0.08	90.50 ±0.21	67.18 ±0.86	75.82 ±2.06	58.26 ±3.44	53.75 ±0.45	60.24 ±0.98	69.03 ±0.61	63.34 ±0.88

Table 14: Expected Calibration Error (ECE) of RoBERTa-large fine-tuned using MNLI dataset for entailment task. Two in-distribution validation sets (MNLI-m and MNLI-mm) and seven distribution shifted datasets are evaluated. All the values with larger font are mean across 3 random seeds. The values with smaller font and plus-minus sign (±) are corresponding variance. Numbers in bracket means the number of samples. Lower ECE value indicates the better calibration.

			Distribution Shifted Datasets						
Method	MNLI-m (9815)	MNLI-mm (9832)	Diag (1104)	HANS* (30K)	QNLI* (5266)	WNLI* (635)	NQ-NLI* (4855)	FEVER-NLI (20K)	WANLI (5000)
Vanilla	10.10 ±0.96	11.03 ±2.29	4.73 ±0.90	8.67 ±1.40	27.07 ±2.44	9.17 ±2.21	34.23 ±2.45	4.70 ±0.62	3.03 ±1.80
WordDrop	12.80 ±0.94	12.33 ±1.11	6.97 ±0.77	13.17 ±1.18	24.43 ±0.76	3.47 ±0.31	24.70 ±0.51	8.73 ±1.11	5.80 ±0.36
LayerDrop	9.30 ±0.33	9.00 ±0.28	6.37 ±0.69	8.67 ±0.45	20.13 ±1.45	8.60 ±0.57	26.93 ±0.37	4.40 ±0.14	4.60 ±0.22
DropHead	9.40 ±1.77	8.27 ±2.28	6.27 ±0.12	6.40 ±1.58	25.30 ±0.70	7.17 ±0.45	28.70 ±0.59	6.07 ±1.27	4.27 ±1.60
R-Drop	8.73 ±0.25	8.87 ±0.45	7.87 ±1.19	6.97 ±0.74	24.83 ±2.89	7.40 ±0.51	26.30 ±0.65	5.13 ±0.41	4.97 ±0.12
HiddenCut	11.70 ±1.28	12.83 ±0.33	6.70 ±1.12	9.17 ±0.68	26.40 ±0.92	7.13 ±0.77	30.20 ±1.02	5.67 ±0.29	1.70 ±0.08
AdvWeight	11.40 ±1.92	11.33 ±1.36	4.83 ±1.54	8.97 ±1.03	28.03 ±3.64	7.87 ±1.28	29.73 ±3.27	6.13 ±1.10	4.40 ±1.10
AdvEmbed	8.87 ±2.00	9.23 ±2.61	4.37 ±0.60	9.13 ±0.31	31.00 ±4.41	10.33 ±3.23	34.13 ±4.71	3.37 ±1.52	4.30 ±1.47
ChildPTune	8.03 ±3.96	7.87 ±3.88	4.83 ±0.92	8.70 ±2.62	29.40 ±8.36	10.37 ±3.62	35.13 ±6.26	5.67 ±0.62	4.30 ±2.07
R3F	9.93 ±1.62	10.73 ±1.48	4.90 ±1.75	8.43 ±1.33	23.33 ±10.62	7.80 ±1.18	29.00 ±4.00	5.03 ±1.35	3.33 ±0.93
WConsol	8.77 ±1.64	8.57 ±1.70	5.93 ±0.48	7.23 ±2.19	19.90 ±3.15	5.87 ±0.25	28.27 ±2.32	5.00 ±1.44	3.20 ±1.28
LP-FT	11.03 ±0.70	10.87 ±0.71	5.40 ±1.63	11.13 ±0.77	26.57 ±3.35	6.90 ±0.83	32.67 ±1.62	4.03 ±1.20	2.57 ±0.61
RAM (Ours)	7.40 ±0.29	7.43 ±0.56	6.47 ±0.49	10.30 ±2.13	22.00 ±6.84	7.27 ±0.74	26.83 ±3.38	5.73 ±1.70	3.97 ±0.60

Table 15: Accuracy of RoBERTa-large fine-tuned using MNLI dataset for entailment task. Nine adversarially constructed datasets are evaluated. All the values with larger font are mean across 3 random seeds. The values with smaller font and plus-minus sign (±) are corresponding variance. Numbers in bracket means the number of samples.

	Adversarially Constructed Datasets								
	TF-B-m	TF-B-mm	TF-R-m	TF-R-mm	ANLI-R1	ANLI-R2	ANLI-R3	AdvGLUE-m	AdvGLUE-mm
Method	(772)	(746)	(775)	(775)	(1000)	(1000)	(1200)	(121)	(162)
Vanilla	71.11 ±0.42	68.36 ±1.26	50.84 ±2.76	52.52 ±1.90	42 ±2.79	28.70 ±0.51	27.56 ±0.51	55.37 ±2.34	40.95 ±2.38
WordDrop	74.61 ±1.04	73.28 ±0.17	58.06 ±0.32	61.63 ±0.16	42.77 ±0.79	30.33 ±1.27	26.61 ±0.86	62.81 ±1.79	41.98 ±3.07
LayerDrop	74.09 ±0.69	72.30 ±0.46	60.00 ±0.80	62.15 ±0.70	39.83 ±0.99	28.07 ±0.45	26.53 ±0.14	49.04 ±2.81	33.54 ±1.05
DropHead	72.15 ±1.14	71.49 ±1.70	57.89 ±2.77	57.94 ±1.97	40.70 ±1.02	29.57 ±1.81	27.97 ±0.71	57.85 ±3.09	40.12 ±2.31
R-Drop	73.36 ±0.24	74.04 ±1.35	59.57 ±0.80	63.48 ±1.70	42.47 ±1.51	29.37 ±0.98	27.39 ±0.34	56.75 ±2.55	34.77 ±1.05
HiddenCut	73.06 ±1.50	70.46 ±0.71	55.74 ±0.18	57.98 ±1.72	43.07 ±0.86	29.67 ±0.79	26.86 ±0.75	58.13 ±1.95	38.89 ±0.87
AdvWeight	70.03 ±1.66	68.72 ±1.66	52.47 ±2.48	53.59 ±2.26	42.30 ±0.71	27.60 ±0.29	27.78 ±1.19	52.62 ±0.39	36.83 ±1.54
AdvEmbed	71.76 ±0.56	71.13 ±0.46	54.02 ±2.25	57.03 ±0.46	44.43 ±0.61	29.60 ±0.50	28.14 ±0.39	59.23 ±2.81	36.42 ±4.31
ChildPTune	67.75 ±1.70	66.67 ±1.97	47.01 ±2.41	48.17 ±4.30	44.10 ±1.02	25.87 ±0.98	26.22 ±0.98	54.27 ±1.56	38.27 ±0.50
R3F	72.93 ±0.92	71.49 ±0.17	56.64 ±1.95	57.98 ±1.23	41.80 ±1.66	27.97 ±0.54	26.67 ±0.65	56.20 ±2.94	40.95 ±0.58
WConsol	72.06 ±1.25	71.36 ±0.96	51.87 ±1.66	50.97 ±1.53	46.90 ±1.51	26.20 ±0.42	24.86 ±0.67	56.47 ±1.03	40.33 ±2.27
LP-FT	70.90 ±1.33	70.24 ±1.52	49.76 ±1.78	51.23 ±4.10	45.90 ±1.49	27.07 ±0.21	23.69 ±0.79	57.58 ±1.40	43.00 ±0.77
RAM (Ours)	75.60 ±1.44	72.74 ±0.84	56.30 ±1.33	56.60 ±2.26	45.23 ±1.93	28.00 ±0.14	25.92 ±0.42	58.95 ±0.78	42.59 ±2.02

Table 16: Expected Calibration Error (ECE) of RoBERTa-large fine-tuned using MNLI dataset for entailment task. Nine adversarially constructed datasets are evaluated. All the values with larger font are mean across 3 random seeds. The values with smaller font and plus-minus sign (±) are corresponding variance. Numbers in bracket means the number of samples. Lower ECE value indicates the better calibration.

	Adversarially Constructed Datasets								
	TF-B-m	TF-B-mm	TF-R-m	TF-R-mm	ANLI-R1	ANLI-R2	ANLI-R3	AdvGLUE-m	AdvGLUE-mm
Method	(772)	(746)	(775)	(775)	(1000)	(1000)	(1200)	(121)	(162)
Vanilla	6.80 ±1.06	6.93 ±1.58	6.80 ±0.00	5.43 ±0.17	12.80 ±4.70	25.93 ±2.31	26.17 ±2.99	7.97 ±2.17	16.00 ±1.88
WordDrop	13.43 ±0.70	12.83 ±0.56	7.57 ±0.39	8.17 ±0.88	6.13 ±0.54	17.47 ±1.55	20.77 ±1.46	12.57 ±1.50	7.70 ±1.02
LayerDrop	13.13 ±0.42	11.23 ±0.61	7.57 ±0.39	8.23 ±0.33	6.33 ±0.12	18.10 ±0.73	19.30 ±0.00	11.33 ±4.37	11.50 ±1.53
DropHead	11.10 ±0.45	9.80 ±2.15	6.63 ±1.83	5.83 ±1.90	7.20 ±1.31	19.30 ±1.04	19.93 ±1.83	9.53 ±3.47	8.50 ±0.67
R-Drop	11.40 ±0.88	12.50 ±0.86	6.73 ±1.18	7.67 ±1.70	8.90 ±1.16	19.40 ±1.27	20.37 ±0.24	9.73 ±2.81	13.30 ±0.90
HiddenCut	7.93 ±0.76	6.23 ±1.30	3.17 ±0.60	4.83 ±0.21	10.53 ±1.04	23.17 ±0.82	24.90 ±0.59	8.20 ±0.86	12.43 ±0.71
AdvWeight	8.97 ±3.43	9.30 ±2.67	6.80 ±1.34	5.67 ±1.48	10.63 ±2.43	23.03 ±3.51	22.03 ±2.45	8.83 ±0.62	14.90 ±3.28
AdvEmbed	5.83 ±1.48	6.03 ±2.32	4.87 ±1.56	5.20 ±1.28	11.00 ±3.63	25.50 ±3.92	26.47 ±4.03	7.37 ±1.22	18.40 ±1.28
ChildPTune	7.20 ±1.87	6.97 ±1.93	10.13 ±6.32	11.10 ±6.37	11.43 ±7.38	28.90 ±8.81	28.00 ±8.67	9.73 ±1.03	17.57 ±7.33
R3F	8.27 ±0.92	8.20 ±2.30	4.77 ±1.35	5.27 ±1.13	9.47 ±2.27	22.80 ±2.91	23.47 ±4.00	10.03 ±1.01	10.17 ±3.30
WConsol	10.63 ±2.255	11.10 ±1.43	6.27 ±0.65	4.83 ±0.93	5.43 ±1.77	22.27 ±3.21	22.73 ±2.26	10.63 ±1.31	8.43 ±0.66
LP-FT	7.17 ±1.66	6.53 ±0.45	6.57 ±0.61	6.60 ±1.31	9.00 ±1.23	27.30 ±2.15	29.43 ±1.76	11.10 ±0.54	12.63 ±2.17
RAM (Ours)	11.47 ±1.26	9.87 ±1.22	6.53 ±1.13	7.17 ±1.35	6.10 ±0.28	20.43 ±1.60	21.63 ±1.84	10.23 ±1.92	7.47 ±0.62

Table 17: Anomaly detection performance (AUROC) of RoBERTa-large fine-tuned using MNLI dataset for entailment task. Five anomaly datasets are evaluated. All the values with larger font are mean across 3 random seeds. The values with smaller font and plus-minus sign (±) are corresponding variance. Numbers in bracket means the number of samples.

	Anomaly Datasets				
Method	WMT16 (2999)	Multi30K (3071)	SST-2 (872)	20News (592)	QQP (40K)
Vanilla	94.93 ±2.65	98.23 ±0.87	97.50 ±1.06	85.67 ±3.41	84.10 ±4.11
WordDrop	92.00 ±1.21	99.47 ±0.21	97.97 ±0.62	74.97 ±4.39	75.43 ±0.88
LayerDrop	94.07 ±1.40	98.43 ±0.05	94.60 ±0.78	84.90 ±1.36	77.47 ±0.94
DropHead	95.63 ±0.79	97.27 ±2.78	96.03 ±3.30	87.07 ±0.98	78.90 ±2.84
R-Drop	96.03 ±1.06	98.50 ±1.26	98.30 ±0.57	84.67 ±1.55	81.97 ±1.59
HiddenCut	95.57 ±1.28	97.87 ±1.68	97.50 ±1.04	85.23 ±2.17	82.03 ±1.17
AdvWeight	96.70 ±0.92	99.43 ±0.38	98.70 ±0.45	78.37 ±5.00	81.87 ±0.82
AdvEmbed	96.93 ±0.53	99.00 ±0.49	98.23 ±0.09	88.60 ±2.50	86.40 ±1.04
ChildPTune	82.87 ±14.91	95.90 ±3.41	87.43 ±7.23	83.57 ±1.51	83.27 ±3.81
R3F	93.63 ±2.10	99.07 ±0.38	95.73 ±2.88	87.20 ±1.37	83.40 ±2.38
WConsol	90.67 ±3.56	98.80 ±0.29	94.40 ±1.36	90.17 ±0.47	84.43 ±1.84
LP-FT	93.33 ±1.68	98.63 ±0.40	94.40 ±0.08	91.03 ±1.65	90.73 ±2.08
RAM (Ours)	94.90 ±1.14	98.37 ±0.61	96.67 ±1.17	90.83 ±1.88	85.47 ±0.62