ROAST: Robustifying Language Models via Adversarial Perturbation with Selective Training

Jaehyung Kim[†*] **Yuning Mao**[‡] **Rui Hou**[‡] **Hanchao Yu**[‡] **Davis Liang**[‡]
Pascale Fung[◇] **Qifan Wang**[‡] **Fuli Feng**[△] **Lifu Huang**[□] **Madian Khabsa**[‡]
[†]KAIST, [‡]Meta AI, [◇]HKUST, [△]USTC, [□]Virginia Tech
jaehyungkim@kaist.ac.kr

Abstract

Fine-tuning pre-trained language models (LMs) has become the *de facto* standard in many NLP tasks. Nevertheless, fine-tuned LMs are still prone to robustness issues, such as adversarial robustness and model calibration. Several perspectives of robustness for LMs have been studied independently, but lacking a unified consideration in multiple perspectives. In this paper, we propose Robustifying LMs via Adversarial perturbation with Selective Training (ROAST), a simple yet effective fine-tuning technique to enhance the multi-perspective robustness of LMs in a unified way. ROAST effectively incorporates two important sources for the model robustness, robustness on the perturbed inputs and generalizable knowledge in pre-trained LMs. To be specific, ROAST introduces adversarial perturbation during fine-tuning while the model parameters are selectively updated upon their relative importance to minimize unnecessary deviation. Under a unified evaluation of fine-tuned LMs by incorporating four representative perspectives of model robustness, we demonstrate the effectiveness of ROAST compared to state-of-the-art fine-tuning methods on six different types of LMs, which indicates its usefulness in practice.

1 Introduction

Fine-tuning pre-trained language models (Jing and Tian, 2020; Brown et al., 2020) has now become the *de facto* standard in many NLP tasks (Kenton and Toutanova, 2019; Liu et al., 2019; Wang et al., 2019). The typical practice for evaluating fine-tuned LMs is to measure the task performance (*e.g.,* accuracy) on a fixed (validation) set of labeled samples. However, this evaluation approach may fall short in ensuring the reliability of fine-tuned LMs, as they are still prone to *robustness* issues in real-world usage. For example, their predictions are known to be still vulnerable to small word-level perturbations (Jin et al., 2020; Li et al., 2020), and also can be biased by the superficial cues, *e.g.*, keyword or negation (McCoy et al., 2019; Niven and Kao, 2019). As such, it is critical to additionally account for robustness during fine-tuning and evaluation of LMs.

The robustness of fine-tuned LMs has been investigated in various perspectives, such as adversarial robustness or distribution-shift generalization (Wang et al., 2021a; Zhou et al., 2021; Nam et al., 2022; Hendrycks et al., 2020). However, existing research exhibits certain limitations as these studies primarily focus on a single perspective of model robustness, rather than considering multiple perspectives simultaneously. Since real-world applications necessitate models to simultaneously exhibit robustness across multiple dimensions, a unified framework is crucial to build a reliable system. Moreover, as diverse and distinct methods are available for enhancing the robustness of each perspective, it is hard for practitioners to find an efficient approach for fine-tuning LMs to ensure such comprehensive reliability. Consequently, these limitations inspire us to explore *a single effective way for fine-tuning LMs to improve its robustness in multiple perspectives*.

In this paper, we propose a simple yet effective fine-tuning technique (Figure 1) that aims to improve the multi-perspective robustness of LMs, coined **Ro**bustifying LMs with **A**dversarial perturbation with **S**elective **T**raining (**ROAST**). The high-level idea of ROAST is effectively incorporating two important sources for the model robustness during the fine-tuning: robustness on the perturbed input and generalizable knowledge learned during pre-training of LMs. While both factors have been separately demonstrated to enhance the various facets of robustness, the collaborative framework pursuing unified robustness has been less explored yet. Specifically, ROAST first generates adversarial perturbation and adds it to the training in-

[*]Work done during a Meta AI internship.

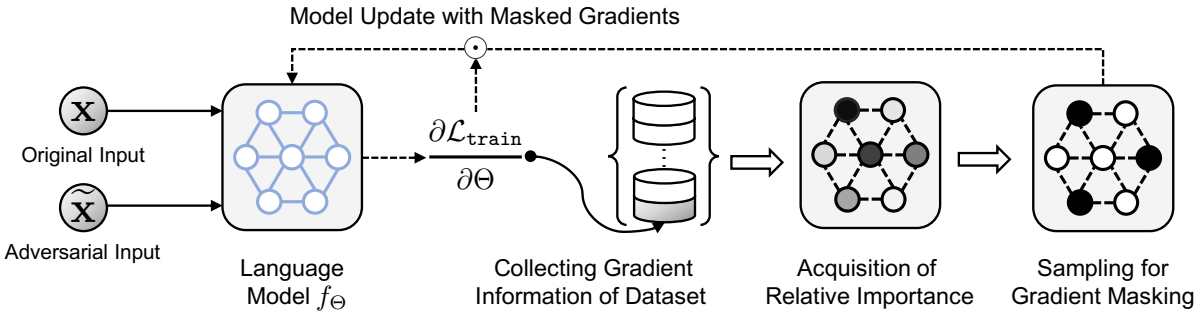


Figure 1: Illustration of ROAST: **Ro**bustifying LMs via **A**dversarial perturbation with **S**elective **T**raining.

put. Then, to prevent a large deviation from the pre-trained model while learning new task-relevant knowledge, ROAST updates model parameters *selectively*; after measuring their relative importance for solving a given task on the fly, only key responsible parameters are updated. We further justify this technique with a theoretical analysis, providing insight into the proposed selective training method.

To perform a unified evaluation of the multi-perspective robustness of LMs, we construct a new evaluation benchmark using two popular NLP tasks, *sentiment classification* and *entailment* tasks; alongside the performance on the validation set, we integrate *four distinct aspects of model robustness*: distribution-shift generalization (Ng et al., 2020), adversarial robustness (Wang et al., 2021b), model calibration (Desai and Durrett, 2020), and anomaly detection (Tack et al., 2020). Under this robustness benchmark, we demonstrate the effectiveness of ROAST for enhancing the robustness of fine-tuned LMs. Notably, across the four robustness aspects along with a standard validation accuracy, ROAST yields 18.39% and 7.63% average relative improvement compared to the traditional fine-tuning methodology on sentiment classification and entailment tasks, respectively. Furthermore, we discover that ROAST significantly improves the robustness of six state-of-the-art LMs, which further indicates its practical usefulness as a simple yet effective solution for robustiying LMs.

2 Background

Multi-perspective of model robustness. For reliable deployment in real-world scenarios (Shen et al., 2017; Gupta et al., 2021), the models should be robust in various perspectives, not just be accurate on a given validation set, drawn from the trained distribution. To this end, various perspectives of model robustness have been investigated. *Distribution-shift (i.e., out-of-domain) generalization* evaluates how well the models can generalize to various forms of distribution shift from trained one at inference time (Ng et al., 2020; Liu et al., 2022). For example, a classifier trained on a sentiment classification dataset is expected to also perform well on another sentiment classification 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 adversarial perturbations (Goodfellow et al., 2015; Zhang et al., 2019; Wu et al., 2020). Since this problem is not exceptional for LMs with word- or sentence-level adversarial perturbation (Jin et al., 2020; Li et al., 2020), *resiliency to the adversarial examples* is also an important perspective of model robustness. In addition, *model calibration* considers the alignment between the model's predicted probability and the true correctness likelihood (Guo et al., 2017; Desai and Durrett, 2020), and hence it is essential for the reliability and interpretability of model prediction. Lastly, *anomaly detection* performance measures the model's capability to distinguish whether a given input is drawn out-of-distribution (OOD) of the training set (Hendrycks et al., 2020; Zhou et al., 2021) or not. In the vision domain, Hendrycks et al. (2022) recently investigates the effectiveness of existing data augmentation methods to enhance robustness in multiple perspectives; however, such investigation has not yet been explored in NLP.

Enhancing robustness of language models. To enhance the robustness of fine-tuned LMs upon (limited) training in the downstream tasks (Kim et al., 2022), various fine-tuning techniques have been explored. One line of work has explored *perturbation-based regularization*, which enhances model robustness by simulating the specific types

of perturbation to the inputs during fine-tuning. For example, Wu et al. (2021); Chen et al. (2021a) utilize Dropout to impose a stochastic perturbation to consider a different view of input, and Ng et al. (2020) substitutes the words using pre-trained masked LMs (*e.g.,* BERT (Kenton and Toutanova, 2019)). In addition, Zhu et al. (2020); Li et al. (2021) impose adversarial perturbation on word embeddings to generate a challenging view of inputs. More interestingly, Kireev et al. (2022) reveals that training with adversarial perturbation can be effective for improving model calibration. On the other hand, another line of work has focused on preserving the *generalizable knowledge* learned during pre-training of LMs; Aghajanyan et al. (2021) identifies a risk of losing the generalizable knowledge during fine-tuning and proposes a noise-based regularization to prevent it. Chen et al. (2020) directly minimizes the distance between the parameters of fine-tuned and pre-trained models as regularization, and Xu et al. (2021) proposes to only update a fixed subset of model parameters during the entire fine-tuning. A two-step strategy of linear probing and then full fine-tuning has recently been shown to be effective for distribution-shift generalization by reducing the deviation from the pre-trained model (Kumar et al., 2022). In addition, the recent result (Uppaal et al., 2023) indicates the importance of generalizable knowledge in pre-trained LMs for better anomaly detection. As both approaches enhance the different perspectives of model robustness, the effective framework for their collaborative utilization is expected to can serve as a unified way for *robusifying* LMs. However, such direction is under-explored from now on, and we try to fill this gap in this work.

3 Robustifying LMs via Adversarial Perturbation and Selective Training

Overview. In this section, we present our method, ROAST, that aims to enhance the multi-perspective robustness of LMs in a unified way. Our main idea is collaboratively incorporating two important sources of model robustness during fine-tuning; we improve the generalization of LMs on the perturbed input using *adversarial perturbation* and preserve the generalizable knowledge within pre-trained LMs via *selective training with gradient masking*. Figure 1 shows an overview of ROAST. Next, we describe in detail our techniques to improve the robustness of fine-tuned LMs.

Adversarial training. To improve the robustness of LM f_Θ, we first incorporate adversarial perturbation during fine-tuning. Specifically, at each training iteration, we construct an adversarial example $\widetilde{\mathbf{x}}$ for training example $\mathbf{x}$. Instead of discrete token-level adversarial perturbation (Jin et al., 2020), we consider embedding-level continuous perturbation (Zhu et al., 2020) which adds noise to the word embedding of each token. Specifically, to construct $\widetilde{\mathbf{x}}$, we use a single-step gradient ascent (Jiang et al., 2020) with a step size δ under ℓ_∞ norm: $\widetilde{\mathbf{x}} := \mathbf{x} + \delta \cdot (\partial \mathcal{L}_{\texttt{task}}/\partial \mathbf{x})/||\partial \mathcal{L}_{\texttt{task}}/\partial \mathbf{x}||_\infty$. We then train f_Θ with a following training loss:

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

where $\mathcal{L}_{\texttt{task}}$ is a task-specific loss (*e.g.*, cross-entropy) with given label y and $\mathcal{L}_{\texttt{cons}}(\mathbf{x}, \widetilde{\mathbf{x}}) := \mathcal{D}_{\texttt{KL}}(f_\Theta(\mathbf{x}), f_\Theta(\widetilde{\mathbf{x}})) + \mathcal{D}_{\texttt{KL}}(f_\Theta(\widetilde{\mathbf{x}}), f_\Theta(\mathbf{x}))$ is bidirectional KL divergence (Wu et al., 2021).

Selective model training via gradient masking. However, fine-tuning with adversarial perturbation could be suboptimal in some perspectives of model robustness, as it incurs a relatively high training loss compared to naive fine-tuning (Dong et al., 2021) and hence may lose useful pre-trained knowledge due to large updates. Hence, to reduce such a potential risk, we reduce an unnecessary deviation by explicitly constraining the update. Specifically, during each iteration of the model update, we selectively update the model parameters by masking out the gradient of less important ones to solve a given task. To measure the relative importance score $s(\theta)$ of each model parameter $\theta \in \Theta$, we use the sum of the square of gradients[1]:

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

Note that $s(\theta)$ is highly connected to Fisher's information matrix, which provides a point estimate of the task-relevant importance of the parameters (Kirkpatrick et al., 2017; Xuhong et al., 2018). However, the calculation of gradients for all samples in $\mathcal{D}$ significantly increases training cost. To alleviate this, we approximate them by using the gradients calculated during the backward pass of model training, *i.e.*, which can be obtained *for free*. Although different values of θ are used to calculate the gradients at each training step, we empirically

[1]For solving the given task with $\mathcal{D}$ and training loss $\mathcal{L}_{\texttt{train}}$

verify that such an approximation is effective and remark that similar approaches have been adopted in other domains (Berthelot et al., 2019).

Using the importance scores $\{s(\theta)|\theta \in \Theta\}$, RoAST decides whether to update each model parameter θ or not. Specifically, we first obtain a normalized score $\widetilde{s}(\theta)$ from the relative order between $|s(\theta)|$ such that $\widetilde{s}(\theta_{\texttt{min}}) = 0 \leq \widetilde{s}(\theta) \leq 1 = \widetilde{s}(\theta_{\texttt{max}})$ where $\theta_{\texttt{min}} := \arg\min_{\theta\in\Theta} s(\theta)$ and $\theta_{\texttt{max}} := \arg\max_{\theta\in\Theta} s(\theta)$. Then, we set a sampling probability $p(\theta)$ using a smooth approximation of the Heaviside step function with the logistic function (Davies, 2002):

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

where α and β are hyper-parameters that control the masking ratio and smoothness, respectively. Remarkably, $p(\theta)$ becomes the hard thresholding that has been utilized in prior studies (Zhang et al., 2021; Xu et al., 2021) as $\beta \to \infty$. Compared to this, our smooth approximation enables a more calibrated use of the importance score. Then, gradient mask $m(\theta)$ is sampled from Bernoulli distribution with a probability $p(\theta)$, and RoAST selectively updates model parameters by masking the gradient using $m(\theta)$ with a scaling term $1/p(\theta)$:

$$\begin{aligned} &\theta \leftarrow \theta - \eta \cdot \widetilde{g}(\theta), \\ &\widetilde{g}(\theta) := \big(m(\theta)/p(\theta)\big) \odot g(\theta), \end{aligned} \quad (4)$$

where η denotes a learning rate. To demonstrate the soundness of proposed selective training with a masked gradient $\widetilde{g}(\theta)$, we derive a theoretical corollary based on previous result (Xu et al., 2021).

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

We present a formal derivation of Corollary in Appendix B. The corollary establishes that, under certain conditions, the masked gradient $\widetilde{g}(\theta)$ is an unbiased estimator of the original gradient $g(\theta)$, which indicates that the masked gradient correctly identifies the true gradient on average; therefore, the variance introduced by masking doesn't introduce a systematic bias that could deviate the model from converging to its optimal parameters. In addition, the upper bound on the norm of covariance provides confidence in the stability of this estimator. In practical terms, it assures us that the masked gradient won't be too volatile or far off from the true gradient, thus safeguarding the convergence properties of the optimization process.

In practice, we accumulate the gradients during each training epoch to calculate the importance scores, and then generate the gradient masks for the next epoch. In addition, we empirically observe that RoAST can work without a scaling term and it sometimes outperforms the original. Hence, we consider whether to apply scaling as an additional hyper-parameter. The overall procedure of RoAST is described in Algorithm 1. More details are presented in Appendix A.3.

4 Experiments

In this section, we evaluate the effectiveness of RoAST to enhance the robustness of fine-tuned LMs in multi-perspectives. We first describe the experimental setups, including how the robustness evaluation benchmark is constructed, in Section 4.1. In Section 4.2, we present experimental results of RoAST and other baselines on the constructed benchmark. In Section 4.3, we provide more analysis of RoAST including (a) ablation study, (b) comparison with different methods sharing similar intuition, and (c) qualitative analysis.

4.1 Experimental Setup

Tasks and metrics. We select two popular NLP tasks, sentiment classification and entailment, to measure the robustness of fine-tuned LMs in a unified way. We take training sets of SST-2 (Socher et al., 2013) and MNLI (Williams et al., 2018) as training data for each task, respectively.

• *In-distribution performance* ($\text{Acc}_{\texttt{in}}$): We first evaluate model performance with respect to training distribution: we measure the accuracy on the validation sets following the common practice.

• *Distribution-shift generalization* ($\text{Acc}_{\texttt{shift}}$): To evaluate model capability on out-of-domain generalization, we measure the average accuracy on multiple distribution shifted datasets, *i.e.*, different datasets with the same task. For entailment, 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 datasets following Potts et al. (2021).

• *Adversarial robustness* ($\text{Acc}_{\texttt{adv}}$): To measure the adversarial robustness of LMs, we first construct text-level adversarial examples using TextFooler (Jin et al., 2020) on vanilla fine-tuned BERT and RoBERTa models following Wang et al. (2021a).

We also consider the datasets constructed via human-in-the-loop dynamic adversarial data collection (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 various types of adversarial noises. We report the average performance on these datasets.

• *Model calibration* (ECE): To measure the model's calibration performance, we report the average Expected Calibration Error (Guo et al., 2017), calculated during the 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 model performance on anomaly detection, we use multiple external datasets as anomaly samples following the setups in recent studies (Hendrycks et al., 2020; Zhou et al., 2021). We use the maximum softmax probability score (Hendrycks et al., 2020) and AUROC for the evaluation method and metric.

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

$$\Delta_{\texttt{avg}} := \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\Delta_s,\ \Delta_s = \frac{s - s_{\texttt{base}}}{s_{\texttt{max}} - s_{\texttt{base}}}, \quad (5)$$

$\mathcal{S} = \{\text{Acc}_{\texttt{in}}, \text{Acc}_{\texttt{shift}}, \text{Acc}_{\texttt{adv}}, \text{ECE}, \text{AUROC}\}$ and $0 \leq \Delta_s \leq 1$. s and $s_{\texttt{base}}$ are the performances of a given method and vanilla, respectively. $s_{\texttt{max}}$ denotes the best score of each metric: 100 for accuracy metrics, 0 for ECE, and 1 for AUROC. Here, we note that we present the AUROC values as 100 × AUROC in our tables to maintain a consistent scale with other metrics. Also, we use a relative average instead of a direct average for measuring multi-perspective robustness, as it is more appropriate for the problem at hand; to evaluate LMs' multi-perspective robustness, it is crucial to equally incorporate multiple metrics from various perspectives. However, these metrics often have different scales, and the direct average is limited in preventing a single metric from dominating the aggregate results due to its numerical magnitude. 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 datasets are presented in Appendix A.1.

Baselines. We compare RoAST with various fine-tuning algorithms; we first consider a naïve fine-tuning method, denoted by Vanilla. We then consider a wide range of perturbation-based fine-tuning algorithms: (*1a*) WordDrop (Guo et al., 2020): dropping input words; (*1b*) HiddenCut (Chen et al., 2021a): dropping spanned hidden features; (*1c*) AdvWeight (Bahri et al., 2022): adding adversarial perturbation on the model parameters; (*1d*) AdvEmbed (Madry et al., 2018): adding adversarial perturbation on the word embeddings. (*1e*) FreeLB (Zhu et al., 2020): introducing efficient adversarial training with gradient accumulation. (*1f*) SMART (Jiang et al., 2020): in addition to adversarial perturbation, introducing EMA-based Bregman proximal point regularization. (*1g*) RIFT (Dong et al., 2021): introducing adversarial fine-tuning method from an information-theoretical perspective. Furthermore, we consider recent state-of-the-art algorithms that 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 (WCons) (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 and Hutter, 2019) with its default parameters $(\beta_1, \beta_2, \epsilon)$=(0.9, 0.98, 1e-6) and a weight decay of 0.01. We use linear learning rate decay with warmup ratio 0.06 and learning rate η=1e-5 (Liu et al., 2019). The models are fine-tuned with batch size 16 for 10 epochs. All the experiments are conducted with RoBERTa-large (Liu et al., 2019) except for the experiments in Table 3. Both baselines and our method are optimized with their own hyper-parameters from a set of candidates (described in Appendix A.2) based on the validation set. For RoAST, we use $\delta = 0.1$ with $\lambda \in \{0.01, 0.1, 0.5\}$ for adversarial training. For the hyper-parameters of gradient masking, we use $\alpha \in [0.6, 0.95]$, $\beta \in \{1, 5, 10\}$ along with a scaling term. More details are in Appendix A.3.

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 in bold and underline, respectively.

Method	$\text{Acc}_{\texttt{in}}$	$\text{Acc}_{\texttt{shift}}$	$\text{Acc}_{\texttt{adv}}$	ECE (↓)	AUROC	$\Delta_{\texttt{avg}}$	$\text{Rank}_{\texttt{avg}}$
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	11.8
WordDrop	96.44$_{\pm 0.03}$	89.95$_{\pm 0.70}$	69.46$_{\pm 0.69}$	7.33$_{\pm 0.78}$	87.57$_{\pm 1.91}$	-1.14	11.2
R-Drop	96.44$_{\pm 0.19}$	91.75$_{\pm 0.21}$	69.00$_{\pm 1.52}$	6.05$_{\pm 0.87}$	89.19$_{\pm 1.20}$	9.02	9.0
HiddenCut	96.67$_{\pm 0.34}$	91.14$_{\pm 0.20}$	70.32$_{\pm 0.78}$	5.47$_{\pm 0.43}$	89.91$_{\pm 0.38}$	12.26	5.8
AdvWeight	96.41$_{\pm 0.29}$	91.92$_{\pm 0.15}$	65.47$_{\pm 0.77}$	7.36$_{\pm 0.34}$	87.80$_{\pm 0.98}$	1.33	10.6
AdvEmbed	96.48$_{\pm 0.05}$	91.75$_{\pm 0.32}$	69.90$_{\pm 0.90}$	5.51$_{\pm 0.16}$	90.79$_{\pm 0.35}$	13.70	6.0
FreeLB	96.33$_{\pm 0.34}$	91.94$_{\pm 0.38}$	70.06$_{\pm 0.27}$	6.49$_{\pm 0.51}$	89.82$_{\pm 0.40}$	9.21	7.6
SMART	96.86$_{\pm 0.05}$	91.49$_{\pm 0.33}$	70.09$_{\pm 0.04}$	6.32$_{\pm 0.34}$	**90.89**$_{\pm 0.44}$	13.10	5.8
RIFT	96.41$_{\pm 0.05}$	89.55$_{\pm 0.25}$	70.67$_{\pm 1.08}$	6.85$_{\pm 0.39}$	89.93$_{\pm 1.07}$	3.31	8.6
ChildPTune	96.56$_{\pm 0.04}$	91.75$_{\pm 0.23}$	69.54$_{\pm 0.14}$	5.57$_{\pm 0.15}$	87.00$_{\pm 0.15}$	7.98	8.2
R3F	96.56$_{\pm 0.09}$	91.79$_{\pm 0.09}$	69.13$_{\pm 1.55}$	5.83$_{\pm 0.15}$	88.49$_{\pm 0.49}$	9.38	7.8
WCons	96.60$_{\pm 0.22}$	92.15$_{\pm 0.25}$	70.86$_{\pm 0.12}$	5.01$_{\pm 0.27}$	89.61$_{\pm 1.03}$	15.47	3.6
LP-FT	96.33$_{\pm 0.25}$	91.85$_{\pm 0.17}$	72.48$_{\pm 0.05}$	**4.05**$_{\pm 0.20}$	89.46$_{\pm 0.77}$	16.75	5.6
RoAST (Ours)	**96.87**$_{\pm 0.20}$	**92.38**$_{\pm 0.12}$	**72.57**$_{\pm 0.53}$	5.45$_{\pm 0.40}$	90.37$_{\pm 0.65}$	**18.39**	**1.8**

4.2 Main results

To evaluate the effectiveness of RoAST for improving the robustness of LMs, we compare it with various baselines by fine-tuning RoBERTa-large model (Liu et al., 2019). Tables 1 and 2 summarize the experimental results on sentiment classification and entailment task, respectively. First, it is worth noting that the common evaluation method using the accuracy on the validation set is not enough to capture the robustness of a given model. For example, all the baselines successfully improve the vanilla fine-tuning on the validation accuracy ($\text{Acc}_{\texttt{in}}$), but their robustness sometimes degrades severely as listed in Table 2. Such results support the necessity of considering multi-perspective model robustness in a unified manner, rather than naively focusing on validation accuracy. Also, we note that there is no single best method when considering multiple perspectives of model robustness; this result indicates the value of a unified measurement to facilitate robustness evaluation.

In this sense, one can observe that RoAST consistently outperforms the baseline fine-tuning methods. To be specific, across 4 different robustness metrics along with a validation accuracy, RoAST exhibits 18.39 % and 7.63% average relative improvement compared to vanilla fine-tuning on sentiment classification and entailment, respectively. As a result, RoAST achieves an average ranking of 2.7 while the previous best method achieves 4.4. These results demonstrate that RoAST could serve as a simple yet strong method for robustifying LMs. Interestingly, advanced fine-tuning methods are more effective in the sentiment classification task than in the entailment task. One possible reason is the difference in the intrinsic difficulty of the task and training dataset size; while the size of the training data is much smaller for SST-2 (67k vs MNLI: 393k), the accuracy is much higher in the sentiment classification task. This indicates that LMs could be more vulnerable to overfitting, and hence regularization of training could be more effective.

To further demonstrate the effectiveness of RoAST, we verify its compatibility across different types of pre-trained LMs. Specifically, in addition to RoBERTa-large, we conduct additional experiments with five recent state-of-the-art LMs which have the similar number of model parameters: BERT-large (Kenton and 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). We note that the best hyper-parameters found in Tables 1 and 2 are inherited without additional cost from a separate search. In Table 3, we present the experimental results by comparing the average relative improvement of RoAST compared to two representative baselines, AdvEmbed and WConsol,

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. ROAST again achieves the best overall performance across different evaluation aspects.

Method	$\text{Acc}_{\texttt{in}}$	$\text{Acc}_{\texttt{shift}}$	$\text{Acc}_{\texttt{adv}}$	ECE (↓)	AUROC	$\Delta_{\texttt{avg}}$	$\text{Rank}_{\texttt{avg}}$
Vanilla	89.97±0.04	64.31±0.58	48.60±1.31	12.64±0.79	92.09±2.26	0.00	9.2
WordDrop	90.35±0.17	63.20±0.44	52.45±0.78	12.17±0.04	87.97±1.40	-8.15	8.2
R-Drop	90.64±0.03	62.74±0.13	51.24±0.90	11.73±0.32	91.89±0.47	2.45	6.4
HiddenCut	90.48±0.18	63.84±0.26	50.43±0.12	11.83±0.26	91.64±0.39	1.60	6.8
AdvWeight	90.19±0.13	62.87±0.57	48.00±0.72	12.38±0.71	91.01±0.93	-2.97	11.4
AdvEmbed	90.24±0.07	64.23±0.38	50.20±0.72	12.48±0.71	**93.83**±0.52	5.72	6.8
FreeLB	90.27±0.10	64.94±0.14	49.48±0.43	12.26±1.43	93.24±0.92	4.74	6.4
SMART	**90.74**±0.04	63.27±0.11	52.27±0.10	11.41±0.23	91.42±0.02	2.56	5.8
RIFT	89.73±0.17	62.33±0.41	**53.26**±0.97	13.46±0.35	92.88±1.54	0.85	9.4
ChildPTune	90.08±0.07	64.09±0.84	46.48±1.00	13.63±3.29	86.60±5.90	-16.14	12.0
R3F	90.41±0.07	63.54±0.31	50.29±0.49	11.39±1.25	91.81±1.41	2.31	6.8
WCons	90.54±0.01	**65.04**±0.55	49.00±0.10	**10.84**±0.90	91.49±1.40	2.99	5.2
LP-FT	90.42±0.14	64.70±0.54	48.82±0.90	12.64±0.39	93.63±0.88	5.11	6.6
ROAST (Ours)	90.64±0.11	63.95±0.51	51.33±0.34	11.02±0.45	93.25±0.15	**7.63**	**3.6**

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 are mean across 3 random seeds. More detailed experimental results, such as the performances on individual robustness metrics, are presented in Appendix D.

	Entailment				Sentiment Classification			
Model	Vanilla	AdvEmbed	WCons	ROAST	Vanilla	AdvEmbed	WCons	ROAST
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**

which show competitive performance in Tables 1 and 2. Here, to facilitate the comparison between different LMs, we calculate the average relative improvement using the vanilla fine-tuned BERT-large as the common baseline for $s_{\texttt{base}}$ in Eq. 5. We observe that ROAST significantly improves the robustness of fine-tuned models regardless of the type of LMs. More interestingly, ROAST could be useful to reveal the true potential of LMs; DeBERTa-large becomes the most robust LM with ROAST while it was originally far from the best. Detailed results on each dataset and LM are presented in Appendix D and E, respectively.

4.3 More analysis with ROAST

Ablation study. To validate the proposed components of ROAST, we conduct an ablation study; specifically, we fine-tune RoBERTa-large on SST-2 by varying each component of ROAST: (a) adversarial perturbation (*Advp*) in Eq.1 and selective update of the model parameters with (b) thresholding (*Thre*) using relative importance $s(\theta)$ in Eq.2, (c) scaling (*Scal*) with smooth approximation using $p(\theta)$ in Eq.3, and (d) sampling (*Samp*) from Bernoulli distribution with $m(\theta)$ in Eq.4. Then, we evaluate the robustness of each method using the constructed robustness benchmark as same as Table 1. We summarize the results in Table 4. As shown in Table 4, the incorporation of the proposed importance score via re-scaling, denoted by (d), performs slightly worse than naive adversarial training, denoted by (a). However, the incorporation of the importance score via a selective update with hard thresholding, denoted by (c), outperforms both.

Table 4: Ablation study with each component of ROAST on the sentiment classification with RoBERTa-large. All the values and error bars are mean and standard deviation across 3 random seeds.

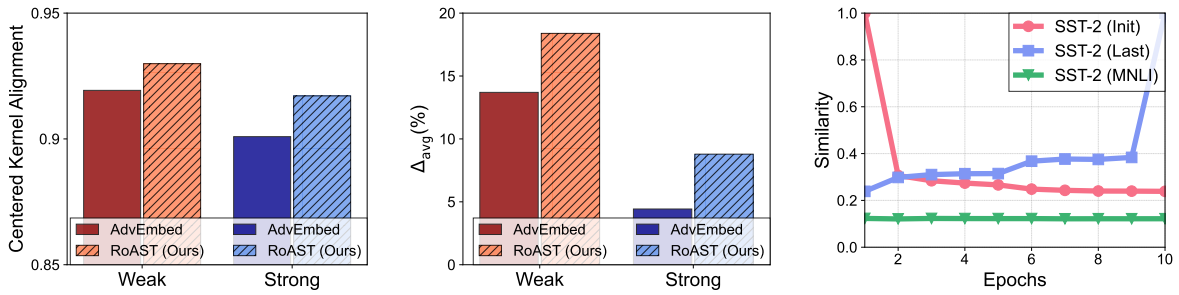
Method	AdvP	Thre	Scal	Samp	$\text{Acc}_{\texttt{in}}$	$\text{Acc}_{\texttt{shift}}$	$\text{Acc}_{\texttt{adv}}$	ECE (↓)	AUROC	$\Delta_{\texttt{avg}}$
Vanilla	-	-	-	-	96.29$_{\pm 0.14}$	91.79$_{\pm 0.13}$	66.30$_{\pm 2.14}$	7.11$_{\pm 0.82}$	86.72$_{\pm 3.60}$	0.00
(a)	✓	-	-	-	96.48$_{\pm 0.05}$	91.75$_{\pm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

Table 5: Robustness of fine-tuned RoBERTa-large with different ways to enhance the robustness by training model under perturbation while preserving pre-trained knowledge. Rand and Min are ROAST with different masking strategies. All the values are mean across 3 random seeds and results with variance are presented in Appendix E.

	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
WCons*	90.62	65.20	50.96	12.01	93.23	6.52	96.71	91.89	71.46	4.99	89.13	15.16
LP-FT*	90.55	64.88	48.95	11.46	92.83	5.29	96.10	92.14	70.69	5.18	90.97	14.24
ROAST	90.64	63.95	51.33	11.02	93.25	7.63	96.87	92.38	72.57	5.45	90.37	18.39
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
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

This result indicates that the better robustness of $s(\theta)$ in Table 4 is not solely due to incorporating the importance score into the model update, but rather to the careful design of using them via selective training with sparse gradients. Moreover, ROAST explicitly constrains the model update with $m(\theta)$, significantly enhancing the robustness from its two distinct advantages for training the model. Firstly, it preserves the parameters by selectively training them using masked (*i.e.*, sparse) gradients. Secondly, it reduces distortion from the original gradients by sampling the mask instead of deterministic selection. Here, the gain from the first advantage is not only obtainable from $m(\theta)$. Hard thresholding via $s(\theta)$ also shares the same advantages of selective training and it can be verified with the results in Table 4, *i.e.*, (c) > (a), (b), (d). However, it is important to note that its effectiveness can be limited since thresholding can distort the gradients from the original direction through deterministic updates of important parameters. Therefore, the second advantage of $m(\theta)$ is crucial, as it improves selective training by reducing the risk of distortion. By sampling the mask for updates from the distribution $p(\theta)$, which prioritizes important parameters, $m(\theta)$ continuously benefits from selective training while also covering overlooked parameters through stochastic sampling. This essential advantage of integrating $m(\theta)$ is demonstrated by improvements over (c,d) in Table 4.

Comparison to methods with similar intuition. The key intuition of ROAST is effectively learning a given task under challenging perturbation while preserving the generalizable knowledge within pre-trained LM. ROAST achieves this goal via selective model training, but one can consider different ways for the same purpose. For example, SMART (Jiang et al., 2020) and RIFT (Dong et al., 2021) introduce additional regularization to the preservation of pre-trained knowledge upon the adversarial training. To demonstrate the superiority of ROAST, we additionally consider the extra baselines with similar intuition, *WCons** and *LP-FT**, by incorporating adversarial perturbation to the methods for the preservation. model robust under perturbations as well. As shown in Table 1 and 2, ROAST significantly outperforms SMART and RIFT; since both approaches additionally introduce the regularization loss, it can induce the learning difficulty to find Pareto optimal between learning and preservation. Also, in Table 5, one can further verify that ROAST significantly outperforms both *WCons** and *LP-FT**. This result clearly demonstrates the effectiveness of the proposed selective



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

Figure 2: Qualitative analysis of RoAST. (a) Similarity between the initial pre-trained model and fine-tuned one. (b) Robustness under the different strengths of adversarial perturbation and improvement from RoAST. (c) Dynamics of the sampled gradient masks for selective training.

training scheme compared to naive combination with existing works.

Additionally, we conduct experiments to verify the effect of different strategies of selective training. Instead of focusing on the relatively important parameters in RoAST (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 the relative importance score $s(\theta)$. From Table 5, we find that the effectiveness of *Min* and *Rand* is largely degraded compared to RoAST. Remarkably, *Min* shows worse results than *Rand*, which reveals the importance of focusing on the task-relevant, important model parameters for selective training.

Qualitative analysis. We further present qualitative analysis on RoAST. To this end, we consider RoBERTa-large on SST-2 with two different strengths of adversarial perturbation to facilitate the analysis: *Weak* ($\lambda = 0.01$) and *Strong* ($\lambda = 0.5$). Then, we measure the similarity between the initial pre-trained model and fine-tuned one using centered kernel alignment (CKA) (Kornblith et al., 2019). Specifically, we measure the average CKA between each layer's hidden features across training epochs. In Figure 2(a), we observe that stronger adversarial training incurs larger deviation from the pre-trained model, and it could be effectively prevented by RoAST. This result indicates that RoAST helps fine-tuned model to preserve the generalizable knowledge of the pre-trained LM while learning a new task under adversarial perturbation, and hence can effectively improve the model robustness (Figure 2(b)). In addition, we investigate the dynamics of gradient masking by measuring the intersection over union between (1) masks at the first epoch and other epochs (*SST-2 (Init)*), (2) masks 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 under RoAST, the sampled masks become task-specific and far from the initial one as training goes on (Figure 2(c)). The adaptability of RoAST for each task is further observed from a low similarity between the masks of different tasks.

5 Conclusion

In this paper, we propose to consider multiple aspects of model robustness for the reliable deployment of LMs in real-world settings. To improve the robustness of fine-tuned LMs, we present a new fine-tuning method (RoAST) by leveraging adversarial perturbation while preventing its potential risk with efficient selective training. Through extensive experiments under constructed benchmark, we demonstrate the effectiveness of RoAST and its generalizability to multiple state-of-the-art LMs. As investing multiple aspects of model robustness in a unified viewpoint is under-explored in the literature, 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 RoAST can benefit other NLP tasks (e.g., question answering) and domains (e.g., vision and graph) as well, as interesting future work directions.

Limitations

Although we have conducted comprehensive experiments on two representative NLP tasks with a wide range of LMs and multiple representative robustness aspects, results and analysis on more datasets, tasks, and domains (e.g., computer vision) would likely draw a more decisive conclusion. In addition to empirical evidence, theoretical analysis of why gradient masking improves language model robustness in aspects like model calibration and anomaly detection is also worth further exploration.

Ethics Statement

The robustness issue of language models has been extensively investigated and a simple yet effective fine-tuning algorithm, RoAST, 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 and Bowman, 2019) or annotation artifacts (Gururangan et al., 2018). However, as the proposed idea of RoAST is general and can be easily extended to other training objectives, we believe that the concerns on such problems could be alleviated by incorporating recently proposed solutions 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 preserving the idea of gradient masking of the overall objective (Eq.2).

References

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

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

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

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

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

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

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

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

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

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

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

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

Xinshuai Dong, Anh Tuan Luu, Min Lin, Shuicheng Yan, and Hanwang Zhang. 2021. How should pre-trained language models be fine-tuned towards adversarial robustness? In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

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

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

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

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

Abhishek Gupta, Alagan Anpalagan, Ling Guan, and Ahmed Shaharyar Khwaja. 2021. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array*, 10:100057.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. 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)*.

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

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

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

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. 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)*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. 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*.

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

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

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

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

Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. 2022. On the effectiveness of adversarial training against common corruptions. In *Uncertainty in Artificial Intelligence (UAI)*.

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

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

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

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

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

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

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

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

Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021. Searching for an effective defender: Benchmarking defense against adversarial word substitution. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3137–3147.

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

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

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

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

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

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

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

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

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

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

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

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jungsoo Park, Gyuwan Kim, and Jaewoo Kang. 2022. Consistency training with virtual adversarial discrete perturbation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

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

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

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

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

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

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Rheeya Uppaal, Junjie Hu, and Yixuan Li. 2023. Is fine-tuning needed? pre-trained language models are near perfect for out-of-domain detection. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

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

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021b. 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)*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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)*, pages 1112–1122.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. 2020. Adversarial weight perturbation helps robust generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

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

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

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

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

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*.

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

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

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

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 and 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 and 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

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:

$$\mathcal{L}_{\texttt{train}} = \mathcal{L}_{\texttt{task}}\big(f_\Theta(\mathbf{x}), y\big) + \mathcal{L}_{\texttt{task}}\big(\widetilde{f}_\Theta(\mathbf{x}), y\big) + \lambda \mathcal{L}_{\texttt{cons}}(f_\Theta(\mathbf{x}), \widetilde{f}_\Theta(\mathbf{x})), \quad (6)$$

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.6, 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 $p_{\texttt{WD}} \in \{0.05, 0.10, 0.15\}$.

- **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** (Madry et al., 2018) imposes adversarial perturbation on the word embeddings of input tokens. We set the same values between the magnitude of perturbation and step size for the gradient ascent step; a step size δ is tuned among {1e-5, 1e-3, 1e-1} under ℓ_∞ norm.

- **FreeLB** (Zhu et al., 2020) proposes an efficient way to construct multi-step adversarial perturbation via gradient accumulation. We follow the best hyper-parameters provided by the authors after careful tuning. The official code is available at https://github.com/zhuchen03/FreeLB.

- **SMART** (Jiang et al., 2020) additionally incorporates the regularization based on Breg proximal point method. Specifically, they use EMA model (Tarvainen and Valpola, 2017) and consistency loss between fine-tuned model. We set the same hyper-parameters in the paper, except the coefficient between each loss; for such coefficient, we tune among {0.01, 0.1, 1.0}. The official code is available at https://github.com/namisan/mt-dnn.

- **RIFT** (Dong et al., 2021) introduce regularization loss which is derived from information-theoretical perspective. RIFT encourages an objective model to retain the features learned from the pre-trained model throughout the entire fine-tuning process. We tune hyper-parameters α among {0.1, 0.3, 0.7}, following the official code by authors: https://github.com/dongxinshuai/RIFT-NeurIPS2021.

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 RoAST

As described in Section 4.1, we use a fixed step size $\delta = 0.1$ for the gradient ascent step to construct

Algorithm 1 ROAST: Robustifying LMs via Adversarial Perturbation with Selective Training

Input: Pre-trained LM f_Θ, training data $\mathcal{D}$, learning rate η, update frequency T, masking ratio α, smoothness factor β, adversarial noise magnitude δ, coefficient of regularization λ

```
/* Obtaining initial gradient */
```
$\mathcal{G}(\Theta) \leftarrow \text{InitGrad}(f_\Theta, \mathcal{D})$
for each iteration t **do**
  if t % T = 0 **then**
    `/* Get relative importance */`
    $\{s(\theta)|\theta \in \Theta\} \leftarrow \mathcal{G}(\Theta),\ \mathcal{G}(\Theta) \leftarrow \emptyset$
    `/* Sample gradient mask */`
    $\{m(\theta)|\theta \in \Theta\} \leftarrow \texttt{Mask}(\{s(\theta)\}, \alpha, \beta)$
  end if
  `/* Sampling training data */`
  $(\mathbf{x}, y) \sim \mathcal{D}$
  `/* Add adversarially perturbation */`
  $\widetilde{\mathbf{x}} \leftarrow \mathbf{x} + \delta \cdot (\partial \mathcal{L}_{\texttt{task}}/\partial \mathbf{x})/||\partial \mathcal{L}_{\texttt{task}}/\partial \mathbf{x}||_\infty$
  `/* Get gradient during backward */`
  $g(\theta) \leftarrow \partial \mathcal{L}_{\texttt{train}}/\partial\theta,\ \mathcal{L}_{\texttt{train}}$
  $= \mathcal{L}_{\texttt{task}}(\mathbf{x}, y) + \mathcal{L}_{\texttt{task}}(\widetilde{\mathbf{x}}, y) + \lambda \mathcal{L}_{\texttt{cons}}(\mathbf{x}, \widetilde{\mathbf{x}})$
  `/* Update with masked gradients */`
  $\theta \leftarrow \theta - \eta \cdot \big((m(\theta)/p(\theta)) \odot g(\theta)\big)$
  `/* Accumulate training gradients */`
  $\mathcal{G}(\theta) \leftarrow \mathcal{G}(\theta) \cup g(\theta)$
end for

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 a 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 $\beta \in \{1, 5, 10\}$. We accumulate the gradient information through each training epoch, then sample the gradient mask from them for the next epoch. In the case of InitGrad in Algorithm 1, 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. We use NVIDIA A100 GPUs in our experiments.

B Proof of Corollary

In this section, we present a formal proof of the Corollary. 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*

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

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

Here, the key difference between 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 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,*

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

and its covariance is bounded as,

$$|\Sigma[\Delta\theta]||_F \leq d \left\| \frac{\eta^2 \sigma_g^2 \mathbb{1}_k}{\hat{p}|\mathcal{B}|} + \frac{(1-\hat{p})\eta^2 diag\{\frac{\partial \mathcal{L}}{\partial \theta}\}^2}{\hat{p}} \right\|_F,$$

where $\hat{p} := \min p(\theta)$.

Proof. Let $g^{(i)}$ is the gradient of sample $\mathbf{x}_i,\ 1 \leq i \leq |\mathcal{B}|$, 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

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

When we consider g, the followings are obtained:

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

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

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

Let $\widetilde{g}_i, g_i, p_i$ are the i-th dimension of $\widetilde{g}, g, p$. Then,

$$\begin{aligned} Var[\widetilde{g}_i] &= \mathbb{E}[\widetilde{g}_i^2] - (\mathbb{E}[\widetilde{g}_i])^2 \\ &= p_i \mathbb{E}[(\frac{g_i}{p_i})^2] - (\mathbb{E}[\widetilde{g}_i])^2 \\ &= \frac{\mathbb{E}[g_i^2]}{p_i} - (\mathbb{E}[\widetilde{g}_i])^2 \\ &= \frac{(\mathbb{E}[g_i])^2 + Var[g_i]}{p_i} - (\mathbb{E}[\widetilde{g}_i])^2 \\ &= \frac{Var[g_i]}{p_i} + \frac{(1-p_i)(\mathbb{E}[\widetilde{g}_i])^2}{p_i} \end{aligned}$$

Since $\frac{1}{p_i}$ is a decreasing function regarding p_i, one can derive following bound with $\hat{p} := \min_i p_i$,

$$||\Sigma[\widetilde{g}]||_F \leq \left\| \frac{\sigma_g^2 \mathbb{1}_k}{\hat{p}|\mathcal{B}|} + \frac{(1-\hat{p})\text{diag}\{\frac{\partial \mathcal{L}}{\partial \theta}\}^2}{\hat{p}} \right\|_F,$$

C More Quantitative Results with RoAST

C.1 Generalization beyond embedding-level perturbation

For RoAST, we chose embedding-level perturbation (Zhu et al., 2020; Jiang et al., 2020) over token-level perturbation (Jin et al., 2020; Li et al., 2020), as it is more computationally efficient and better suited for enhancing multi-perspective robustness. Specifically, the construction of token-level adversarial perturbations requires more computation to solve the discrete optimization problem, and additional regularization is often introduced to prevent degenerate cases such as significant changes in semantic or lexical violation (Jin et al., 2020; Li et al., 2020; Park et al., 2022). For example, a relatively simple construction of token-level perturbation with Park et al. (2022) requires 15% more times per iteration, compared to embedding-level perturbation. In addition, while the token-level adversarial perturbation is effective for adversarial robustness, it often comes at the cost of a decrease in clean accuracy (Dong et al., 2021) due to the relatively large perturbation on a discrete space. In contrast, the embedding-level perturbation can be constructed by adding small noise to continuous space, making it more feasible to train the model without the loss of accuracy (Zhu et al., 2020; Jiang et al., 2020)

To further validate the generalization of RoAST beyond the embedding-level perturbation, we conduct additional experiments by adapting RoAST with discrete token-level adversarial perturbations by fine-tuning RoBERTa-large using VAT-D (Park et al., 2022). We used the same hyper-parameters for discrete token-level perturbation as in Park et al. (2022), and for RoAST, we used the same values previously found with embedding-level perturbation. The results are shown in the table below.

Here, we observe that the discrete token-level adversarial perturbation can improve the multi-perspective robustness, especially for adversarial robustness ($\text{Acc}_{\texttt{adv}}$). However, it comes at the cost of degradation in clean accuracy ($\text{Acc}_{\texttt{in}}$). With RoAST, such degradation can be mitigated and the overall robustness of the model could be improved.

C.2 Absolute average improvement of RoAST

During the experiments, we used the average of relative improvement, instead of absolute improvement, as it is more appropriate to measure the multi-perspective robustness than the average of absolute improvement, not to scale up the values. However, we also recognize its weak points, such as the risk of amplification, which is the reason why we additionally report $\text{Rank}_{\texttt{avg}}$, which does not have similar issues. We emphasize that RoAST exhibits the lowest rank among the state-of-the-art fine-tuning methods on both sentiment classification and entailment tasks. Nevertheless, to address the concerns about this, we additionally calculate the absolute average improvement ($\text{Abs}_{\texttt{avg}}$) of {$\text{Acc}_{\texttt{in}}$, $\text{Acc}_{\texttt{shift}}$, $\text{Acc}_{\texttt{adv}}$, 100 - ECE, 100 * AUROC}, on the sentiment classification task. The results are presented in Table 7. Here, one can observe that our method still outperforms the state-of-the-art fine-tuning method with a large gap; RoAST exhibits 46.72% relative improvement on $\text{Abs}_{\texttt{avg}}$, compared to SMART (1.16% vs 0.79%). This result further demonstrates the effectiveness of our method, and we do believe that our empirical results clearly show the merit of the proposed framework.

Table 6: Generalization of RoAST with discrete token-level adversarial perturbation. Here, for such perturbation, we use VAT-D (Park et al., 2022). All the values are mean across 3 random seeds.

Method	$\text{Acc}_{\texttt{in}}$	$\text{Acc}_{\texttt{shift}}$	$\text{Acc}_{\texttt{adv}}$	ECE (↓)	AUROC	$\Delta_{\texttt{avg}}$
Vanilla	96.29	91.79	66.30	7.11	86.72	0.00
VAT-D	95.83	90.86	70.37	6.33	90.39	5.37
VAT-D + RoAST	96.27	90.45	72.12	6.94	92.16	8.73

Table 7: Average absolute improvements with different baselines. All the values are mean across 3 random seeds.

Method	Vanilla	WordDrop	R-Drop	HiddenCut	AdvWeight	AdvEmbed	FreeLB	SMART	RIFT	ChildPTune	R3F	WCons	LP-FT	RoAST (Ours)
$\text{Abs}_{\texttt{avg}}$	76.47	76.36	76.96	76.91	76.94	77.20	77.13	77.26	76.95	74.72	76.93	77.05	76.99	**77.63**

C.3 Additional comparison with FreeLB++

Here, we present additional experimental results with FreeLB++ (Li et al., 2021) on the sentiment classification task, which runs adversarial perturbation for 10 steps without constraints of norm-bounded projection. We also tuned the adversarial step size appropriately as the number of steps varies. The results are summarized in Table 8. Here, one can observe that FreeLB++ outperforms FreeLB, especially in $\text{ACC}_{\texttt{adv}}$ (70.07 → 72.45). Consequently, FreeLB++ is better than FreeLB for multi-perspective robustness as well ($\Delta_{\texttt{avg}}$: 9.21 → 11.02). However, RoAST still outperforms FreeLB++ with a large gap, which further demonstrates the effectiveness of our method. On the other hand, these results indicate a potential room for further improvement in our method at the additional cost of the increased number of steps, since RoAST also uses a single step for constructing adversarial perturbation.

C.4 Relationship between overfitting and robustness

To investigate the relationship between overfitting and robustness of LMs, we performed additional experiments by training RoBERTa-large on SST-2, using the Vanilla method with a constant learning rate for 100 epochs. As shown in Table 9, the model's robustness significantly decreases as the training progresses. This outcome confirms that preserving the parameters is critical for maintaining model robustness, which is a fundamental principle of selective training with $m(\theta)$ in RoAST.

D Additional Results with Various LMs

First, we present the detailed results on the individual robustness metrics like Tables 1 and 2. In Table 10 and 11, the results on sentiment classification and entailment are presented, respectively.

Next, we provide the additional results with different vanilla algorithm to calculate $\Delta_{\texttt{avg}}$; In Table 3, we use BERT-large as a universal vanilla algorithm to calculate relative improvement across different LMs to facilitate comparison in terms of multi-perspective robustness. This choice was made to provide additional insight into the question of *"which LM is the most robust"*. While answering this question is important, we also acknowledge that using the corresponding LM's score as the baseline could provide additional insights into how RoAST performs with each specific LM. Hence, we recalculate Table 3 with the corresponding LM's score and present it in Table 12. One can observe that RoAST mostly outperforms the baselines except in only 1 case, and achieves large improvements compared to baselines in both tasks. This result demonstrates the robustness and effectiveness of RoAST with respect to different LMs.

Lastly, we conducted additional experiments on the sentiment classification task with GPT2-large (Radford et al., 2019), a popular decoder-only LM, to validate the applicability of our approach. Following Radford et al. (2019), we added a linear classifier head on the last token's embedding output for fine-tuning. The results are presented in Table 13. Here, we first observe that the improvements with baseline methods are largely limited. We speculate that this ineffectiveness may occur due to the different nature of decoder-only LMs compared to encoder-only ones, such as BERT, as it could result in different effectiveness of the baseline algorithms and tuned hyper-parameters, which were originally developed and demonstrated only using encoder-only models; for instance, most of the baselines (Jiang et al., 2020; Zhu et al., 2020; Chen et al.,

Table 8: Comparison with additional baseline, FreeLB++ (Li et al., 2021) on the sentiment classification task. All the values are mean across 3 random seeds.

Method	$\text{Acc}_{\texttt{in}}$	$\text{Acc}_{\texttt{shift}}$	$\text{Acc}_{\texttt{adv}}$	ECE (↓)	AUROC	$\Delta_{\texttt{avg}}$
Vanilla	96.29	91.79	66.30	7.11	86.72	0.00
FreeLB	96.33	91.94	70.07	6.49	89.82	9.21
FreeLB++	96.14	91.79	72.45	**5.44**	**90.79**	11.02
RoAST (Ours)	**96.87**	**92.38**	**72.57**	5.45	90.37	**18.39**

Table 9: Tradeoff between robustness and training epochs. All the values are mean across 3 random seeds.

Method	$\text{Acc}_{\texttt{in}}$	$\text{Acc}_{\texttt{shift}}$	$\text{Acc}_{\texttt{adv}}$	ECE (↓)	AUROC	$\Delta_{\texttt{avg}}$
Vanilla (Orig)	96.29	91.79	66.30	7.11	86.72	0.00
Epoch 20	94.88	90.66	64.82	7.75	83.83	-17.45
Epoch 40	93.92	88.52	60.54	9.81	77.04	-46.38
Epoch 60	93.42	88.11	59.63	10.16	72.28	-58.74
Epoch 80	93.39	87.43	59.03	10.68	73.83	-60.08
Epoch 100	93.31	87.52	58.28	10.62	67.61	-69.94

2020; Xu et al., 2021) have been demonstrated under encoder-only models and not shown the results of decoder-only ones. Nevertheless, our RoAST approach continues to enhance the multi-perspective robustness of GPT2-large, with an average relative improvement of 5.07% compared to the Vanilla method. These results indicate that the effectiveness of our approach is not limited to BERT-based LMs with Transformer-encoder architecture.

E Individual Experimental Results

Next, 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 14 and 15, respectively. Also, we report the accuracy and ECE on adversarially constructed datasets in Table 16 and 17, respectively. The results of anomaly detection are shown in Table 18. Next, we present the results from entailment task; we report the accuracy and ECE on distribution shifted datasets in Table 19 and 20, respectively. Then, we report the accuracy and ECE on adversarially constructed datasets in Table 21 and 22, respectively. Finally, we report the results of anomaly detection in Table 23.

Table 10: 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	$93.81_{\pm 0.16}$	$89.46_{\pm 0.61}$	$56.34_{\pm 1.63}$	$9.35_{\pm 0.65}$	$85.09_{\pm 0.35}$
	RoAST (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	$96.60_{\pm 0.22}$	$92.15_{\pm 0.25}$	$70.86_{\pm 0.12}$	$5.01_{\pm 0.27}$	$89.61_{\pm 1.03}$
	RoAST (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	$96.25_{\pm 0.66}$	$90.92_{\pm 1.20}$	$71.95_{\pm 4.64}$	$5.75_{\pm 1.07}$	$87.44_{\pm 2.27}$
	RoAST (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	$95.26_{\pm 0.35}$	$90.60_{\pm 0.51}$	$67.77_{\pm 1.70}$	$6.82_{\pm 0.84}$	$88.58_{\pm 3.53}$
	RoAST (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	$97.08_{\pm 0.06}$	$91.51_{\pm 0.24}$	$73.16_{\pm 0.69}$	$5.55_{\pm 0.08}$	$82.40_{\pm 1.82}$
	RoAST (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	$96.60_{\pm 0.14}$	$91.34_{\pm 1.16}$	$69.74_{\pm 0.77}$	$6.23_{\pm 0.63}$	$88.01_{\pm 2.52}$
	RoAST (Ours)	$97.13_{\pm 0.09}$	$92.31_{\pm 0.24}$	$74.25_{\pm 0.11}$	$4.46_{\pm 0.03}$	$89.74_{\pm 0.25}$

Table 11: 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$_{\pm 0.11}$	60.20$_{\pm 0.59}$	39.25$_{\pm 0.30}$	23.03$_{\pm 0.58}$	70.88$_{\pm 1.64}$
	AdvEmbed	86.99$_{\pm 0.09}$	60.77$_{\pm 0.25}$	42.90$_{\pm 0.37}$	18.50$_{\pm 2.31}$	81.98$_{\pm 3.37}$
	WConsol	86.27$_{\pm 0.26}$	60.52$_{\pm 0.34}$	39.34$_{\pm 0.51}$	17.38$_{\pm 4.21}$	75.71$_{\pm 5.03}$
	ROAST (Ours)	86.89$_{\pm 0.06}$	60.38$_{\pm 0.61}$	42.58$_{\pm 0.47}$	14.97$_{\pm 0.38}$	82.41$_{\pm 0.80}$
RoBERTa	Vanilla	89.97$_{\pm 0.04}$	64.31$_{\pm 0.58}$	48.60$_{\pm 1.31}$	12.64$_{\pm 0.79}$	92.09$_{\pm 2.26}$
	AdvEmbed	90.24$_{\pm 0.07}$	64.23$_{\pm 0.38}$	50.20$_{\pm 0.72}$	12.48$_{\pm 0.71}$	93.83$_{\pm 0.52}$
	WConsol	90.54$_{\pm 0.01}$	65.04$_{\pm 0.55}$	49.00$_{\pm 0.10}$	10.84$_{\pm 0.90}$	91.49$_{\pm 1.40}$
	ROAST (Ours)	90.64$_{\pm 0.11}$	63.95$_{\pm 0.51}$	51.33$_{\pm 0.34}$	11.02$_{\pm 0.45}$	93.25$_{\pm 0.15}$
ALBERT	Vanilla	90.62$_{\pm 0.18}$	64.94$_{\pm 0.33}$	58.79$_{\pm 0.21}$	10.25$_{\pm 0.92}$	83.14$_{\pm 0.80}$
	AdvEmbed	90.60$_{\pm 0.15}$	64.19$_{\pm 0.14}$	58.82$_{\pm 0.25}$	10.92$_{\pm 0.68}$	86.63$_{\pm 4.49}$
	WConsol	90.43$_{\pm 0.20}$	64.37$_{\pm 0.35}$	56.14$_{\pm 1.95}$	9.67$_{\pm 0.08}$	80.65$_{\pm 3.98}$
	ROAST (Ours)	90.72$_{\pm 0.10}$	66.57$_{\pm 0.03}$	59.24$_{\pm 0.20}$	11.83$_{\pm 4.61}$	91.47$_{\pm 2.63}$
XLNet	Vanilla	89.29$_{\pm 0.10}$	63.74$_{\pm 0.23}$	49.33$_{\pm 0.36}$	14.04$_{\pm 1.66}$	77.53$_{\pm 8.00}$
	AdvEmbed	89.46$_{\pm 0.09}$	63.56$_{\pm 0.94}$	50.52$_{\pm 0.90}$	12.69$_{\pm 0.54}$	77.31$_{\pm 1.67}$
	WConsol	89.33$_{\pm 0.04}$	63.55$_{\pm 0.08}$	49.91$_{\pm 0.30}$	14.64$_{\pm 1.37}$	81.35$_{\pm 7.83}$
	ROAST (Ours)	90.03$_{\pm 0.04}$	63.69$_{\pm 0.52}$	50.11$_{\pm 0.14}$	10.28$_{\pm 0.37}$	78.53$_{\pm 3.07}$
ELECTRA	Vanilla	90.68$_{\pm 0.05}$	66.20$_{\pm 0.90}$	56.06$_{\pm 1.16}$	13.87$_{\pm 5.47}$	82.75$_{\pm 2.79}$
	AdvEmbed	90.64$_{\pm 0.01}$	65.66$_{\pm 0.57}$	56.95$_{\pm 0.91}$	12.96$_{\pm 2.51}$	88.43$_{\pm 0.11}$
	WConsol	90.51$_{\pm 0.13}$	67.21$_{\pm 0.41}$	55.43$_{\pm 0.84}$	15.36$_{\pm 5.44}$	84.05$_{\pm 4.02}$
	ROAST (Ours)	91.07$_{\pm 0.07}$	64.80$_{\pm 0.46}$	58.38$_{\pm 0.68}$	10.17$_{\pm 0.40}$	81.01$_{\pm 2.16}$
DeBERTa	Vanilla	90.09$_{\pm 0.15}$	65.01$_{\pm 0.80}$	52.61$_{\pm 1.39}$	15.87$_{\pm 7.81}$	87.89$_{\pm 2.40}$
	AdvEmbed	90.43$_{\pm 0.04}$	65.73$_{\pm 1.02}$	54.91$_{\pm 0.30}$	13.29$_{\pm 1.94}$	86.37$_{\pm 2.27}$
	WConsol	90.02$_{\pm 0.18}$	64.83$_{\pm 1.38}$	53.67$_{\pm 0.08}$	14.51$_{\pm 7.15}$	89.03$_{\pm 0.83}$
	ROAST (Ours)	90.97$_{\pm 0.09}$	66.07$_{\pm 0.31}$	56.53$_{\pm 0.66}$	9.94$_{\pm 0.34}$	88.15$_{\pm 1.73}$

Table 12: Robustness with different language models. Average relative improvements ($\Delta_{\texttt{avg}}$) compared to each vanilla fine-tuned LM are reported for each LM. All the values are mean across 3 random seeds.

	Entailment				Sentiment Classification			
Model	Vanilla	AdvEmbed	WCons	ROAST	Vanilla	AdvEmbed	WCons	ROAST
BERT-large	0.00	<u>22.85</u>	20.23	**25.34**	0.00	<u>18.24</u>	15.85	**22.76**
RoBERTa-large	0.00	<u>5.72</u>	2.99	**7.63**	0.00	13.70	<u>15.47</u>	**18.39**
ALBERT-xxlarge	0.00	<u>2.60</u>	-3.83	**8.19**	0.00	<u>21.34</u>	18.25	**30.62**
XLNet-large	0.00	2.44	<u>2.75</u>	**3.30**	0.00	<u>1.44</u>	-1.76	**12.74**
ELECTRA-large	0.00	**7.89**	-0.70	<u>4.39</u>	0.00	<u>6.44</u>	-2.83	**8.29**
DeBERTa-large	0.00	2.81	<u>3.79</u>	**11.93**	0.00	<u>11.95</u>	8.25	**18.90**

Table 13: Robustness with GPT, LM with Transformer-decoder architecture. Average relative improvements ($\Delta_{\texttt{avg}}$) compared to each vanilla fine-tuned GPT. All the values are mean across 3 random seeds.

Method	$\text{Acc}_{\texttt{in}}$	$\text{Acc}_{\texttt{shift}}$	$\text{Acc}_{\texttt{adv}}$	ECE (↓)	AUROC	$\Delta_{\texttt{avg}}$
Vanilla	94.99	84.33	62.38	8.01	95.30	0.00
AdvEmbed	94.92	84.19	63.58	7.90	95.81	2.60
WCons	94.88	80.72	61.80	9.06	93.57	-15.37
ROAST (Ours)	95.03	81.91	64.96	8.52	97.16	5.07

Table 14: 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 (±) 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
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
FreeLB	96.33 ±0.34	96.34 ±0.49	89.20 ±0.23	91.27 ±0.64	88.95 ±2.28	93.96 ±0.05
SMART	96.86 ±0.05	96.23 ±0.04	88.49 ±0.16	90.50 ±0.16	89.04 ±1.60	93.18 ±0.07
RIFT	96.41 ±0.05	95.63 ±0.03	88.14 ±0.21	89.25 ±0.89	81.80 ±0.47	92.93 ±0.04
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
ROAST (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 15: 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
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
FreeLB	3.63 ±1.43	5.47 ±2.15	6.17 ±2.88	5.97 ±0.61	4.37 ±0.59	4.50 ±1.51
SMART	4.30 ±0.49	7.73 ±0.56	7.50 ±1.55	6.60 ±0.45	5.93 ±0.66	5.73 ±0.33
RIFT	4.40 ±0.70	4.97 ±2.58	7.97 ±1.60	7.20 ±0.36	8.63 ±0.82	5.57 ±1.11
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
RoAST (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 16: 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
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
FreeLB	74.54 ±0.59	57.92 ±2.48	77.94 ±0.26	77.53 ±0.79	62.39 ±2.83
SMART	77.95 ±0.62	65.31 ±1.37	75.82 ±0.33	76.22 ±0.30	55.18 ±1.59
RIFT	77.14 ±0.41	67.35 ±2.75	74.35 ±0.62	77.33 ±1.01	57.21 ±1.77
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
RoAST (Ours)	77.09 ±0.24	61.18 ±0.86	79.79 ±0.41	79.27 ±0.47	65.54 ±2.40

Table 17: 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
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
FreeLB	6.43 ±0.17	13.70 ±6.25	3.93 ±2.07	5.47 ±0.98	11.77 ±5.00
SMART	7.53 ±0.33	6.53 ±0.78	3.70 ±0.16	4.37 ±0.90	9.57 ±3.80
RIFT	11.43 ±0.46	11.13 ±2.41	2.73 ±0.74	3.17 ±0.24	8.10 ±1.14
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
RoAST (Ours)	6.00 ±0.94	5.13 ±0.90	4.37 ±0.81	4.70 ±0.24	6.30 ±0.59

Table 18: 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
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
FreeLB	86.97 ±0.17	89.63 ±1.55	96.37 ±2.09	91.03 ±0.82	87.43 ±0.44	87.50 ±0.78
SMART	88.63 ±0.70	94.33 ±0.31	95.80 ±0.22	89.70 ±0.22	88.53 ±0.60	88.33 ±0.66
RIFT	85.97 ±1.46	91.00 ±1.14	96.83 ±1.00	88.93 ±1.53	88.80 ±1.67	88.03 ±2.33
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
RoAST (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 19: 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
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
FreeLB	90.38 ±0.08	90.16 ±0.13	67.39 ±0.09	79.47 ±0.24	55.22 ±0.65	56.30 ±0.08	62.62 ±0.08	69.44 ±0.12	64.12 ±0.14
SMART	90.78 ±0.05	90.69 ±0.04	65.53 ±0.23	77.73 ±0.53	53.95 ±0.03	51.42 ±0.24	60.87 ±0.12	69.82 ±0.11	63.57 ±0.11
RIFT	89.75 ±0.25	89.69 ±0.13	65.90 ±0.51	67.22 ±0.83	57.85 ±1.80	52.17 ±1.13	60.79 ±0.84	69.01 ±0.42	63.00 ±0.29
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
RoAST (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 20: 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
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
FreeLB	9.60 ±1.50	8.70 ±0.50	4.40 ±1.00	10.35 ±4.65	27.80 ±7.40	7.75 ±1.45	32.75 ±3.95	5.45 ±0.65	3.45 ±0.25
SMART	10.30 ±0.23	10.20 ±0.18	7.05 ±0.05	5.05 ±0.25	25.55 ±0.35	7.00 ±3.23	26.65 ±0.15	5.00 ±1.23	6.30 ±0.53
RIFT	12.50 ±0.86	12.48 ±0.99	4.20 ±0.25	12.20 ±1.59	29.83 ±3.11	7.48 ±1.72	30.93 ±2.39	5.53 ±0.60	2.00 ±1.04
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
RoAST (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 21: 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								
Method	TF-B-m (772)	TF-B-mm (746)	TF-R-m (775)	TF-R-mm (775)	ANLI-R1 (1000)	ANLI-R2 (1000)	ANLI-R3 (1200)	AdvGLUE-m (121)	AdvGLUE-mm (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
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
FreeLB	70.27 ±1.10	69.37 ±1.27	52.06 ±2.77	50.19 ±3.10	45.45 ±3.15	27.70 ±0.40	28.00 ±0.83	61.57 ±0.41	40.74 ±1.23
SMART	73.32 ±0.13	74.80 ±0.67	58.97 ±1.42	64.71 ±0.45	43.85 ±0.75	31.20 ±0.30	28.46 ±0.13	57.44 ±0.41	37.65 ±1.23
RIFT	77.08 ±2.11	75.74 ±1.68	65.13 ±4.10	66.13 ±3.59	43.45 ±0.51	28.15 ±1.06	26.83 ±0.19	59.50 ±2.26	39.97 ±1.41
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
RoAST (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 22: 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								
Method	TF-B-m (772)	TF-B-mm (746)	TF-R-m (775)	TF-R-mm (775)	ANLI-R1 (1000)	ANLI-R2 (1000)	ANLI-R3 (1200)	AdvGLUE-m (121)	AdvGLUE-mm (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
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
FreeLB	6.70 ±1.80	6.05 ±1.85	7.10 ±1.20	6.55 ±3.35	8.15 ±0.95	26.30 ±2.80	26.65 ±2.85	9.55 ±2.95	13.30 ±1.80
SMART	12.15 ±0.55	13.35 ±1.05	8.25 ±0.85	10.40 ±0.70	5.35 ±0.15	16.50 ±0.50	17.55 ±0.15	10.35 ±0.85	8.45 ±1.15
RIFT	11.40 ±1.86	9.95 ±2.31	8.68 ±1.38	8.05 ±0.87	11.15 ±0.78	26.68 ±1.37	26.43 ±0.59	9.53 ±1.56	14.13 ±0.95
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
RoAST (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 23: 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
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
FreeLB	98.15 ±0.05	99.35 ±0.25	98.50 ±0.10	84.45 ±2.35	85.75 ±2.35
SMART	94.55 ±0.95	99.30 ±0.10	96.10 ±0.50	87.20 ±1.30	79.95 ±0.85
RIFT	93.18 ±3.08	98.35 ±1.35	97.28 ±0.97	91.80 ±1.52	82.00 ±1.40
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
ROAST (Ours)	94.90 ±1.14	98.37 ±0.61	96.67 ±1.17	90.83 ±1.88	85.47 ±0.62