Adaptive Adversarial Training for Balancing Model Robustness and Standard Performance

Bin Zhu, Siyu Liu, Yuyin Lu, Jielin Song, Yanghui Rao†

School of Computer Science and Engineering, Sun Yat-sen University, Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, China

Abstract

Adversarial training (AT) is widely used to boost model robustness against adversarial attacks, i.e., adding minor perturbations on the clean input to fool the target model. However, AT can also lead to degraded clean accuracy since it changes the distribution of the training set. Using the Taylor expansion, we find that commonly used adversarial loss functions inherently include clean loss, making it challenging for previous methods to effectively balance the standard performance and robustness. Based on this, we establish a flexible AT framework that can explicitly balance the model robustness and clean accuracy by assigning learnable weights to the clean and adversarial loss components. Comprehensive experimental results indicate that our method boosts model robustness while maintaining comparable standard performance.

Keywords: Language models; adversarial training; adversarial robustness

1. Introduction

Adversarial training (AT) attempts to boost the robustness of classifiers against adversarial examples by augmenting the training set with perturbed samples. While this approach effectively reduces adversarial errors or boosts the generalization accuracy on adversarial test examples, it has been observed to impair the standard accuracy on clean test data [1, 2, 3, 4]. Recent discussions [3, 4, 5] suggest a trade-off in AT, implying the challenge of simultaneously minimizing standard and adversarial risks.

This paper focuses on AT for natural language processing (NLP) tasks, especially for text classification. The overarching concept of AT involves a two-level optimization process to enhance the model robustness. On the inner level, gradient ascent is employed to optimize small perturbations of the input data, aiming to maximize the model's loss function. On the outer level, gradient descent is utilized to adjust the model parameters to minimize the classification loss of these adversarial examples.

We note that in textual AT, the default iteration number k is often quite small, e.g., 3 for FreeLB [6], TAVAT [7], and InfoBERT [8], resulting in small perturbation sizes for these methods. Their empirical results indicate that a relatively small perturbation size helps boost both model robustness and performance. Nevertheless, we doubt whether a small perturbation size is

†Corresponding author: Yanghui Rao (Email: raoyangh@mail.sysu.edu.cn)

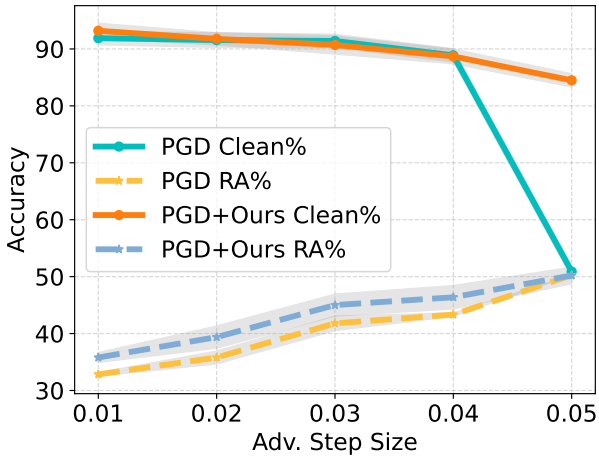


Figure 1: The robust accuracy (**RA**) and the clean accuracy (**Clean**) of the original PGD method [11] and ours on the SST-2 dataset [13]. The backbone model is BERT-base [12]. As the step size increases, a trade-off exists between **RA** and **Clean**. It is hard to achieve optimal robustness and clean accuracy simultaneously.

really helpful in improving robustness. Because the inner maximization greatly affects the effectiveness of AT [9]. A small perturbation size usually generates lower-quality adversarial data, which makes AT useless. For example, the fast gradient sign method (FGSM) [10] can quickly generate adversarial data using one step, but contributing little to robustness. Thus, it is reasonable to vary the iteration number and the adversarial step size to study how inner maximization affects AT. To this end, we choose two widely adopted AT methods, i.e., the projected gradient descent (PGD) method [11] and the FreeLB method [6] as our baselines to conduct preliminary experiments on the BERT-base [12] model.

We report clean and robust accuracies[1] equipped with PGD and FreeLB in Figures 1 and 2. We find that the existing AT method can hardly improve robustness without hurting clean accuracy, which contradicts the results in previous works [6, 7]. As the perturbation size increases in AT, the robustness increases while the accuracy decreases. Additionally, AT will easily collapse and fail to converge when the perturbation size becomes too large.

This preliminary result indicates that we must rethink the trade-off between robustness and accuracy for NLP models. It also motivates us to investigate whether there exists an optimal perturbation size for the sake of balance, and how to make AT converge in a large perturbation size to achieve strong robustness.

To this end, we theoretically analyze the impact of the perturbation size on the learning objective of AT. In particular, we perform Taylor expansion on the adversarial loss and decompose it into a clean data loss and an adversarial one, in which the adversarial one is the weighted sum of squares of all perturbations. The clean loss corresponds to the model's accuracy, while the adversarial one corresponds to the model's robustness. By assigning trainable weights to all the perturbations, we can explicitly balance the two losses to achieve comparable model robustness and standard performance.

[1]In this paper, we use clean accuracy to refer to the standard accuracy on clean text data and use robust accuracy to refer to the generalization accuracy on adversarial test examples.

We further provide extensive experimental results. Compared with existing state-of-the-art AT methods, our method demonstrates a remarkable improvement in robustness without sacrificing the clean accuracy. Our main contributions are:

- We demonstrate that existing AT methods for NLP models either fail to improve robustness or compromise clean accuracy.
- We conduct theoretical analysis on a series of gradient-based AT methods. We decompose their learning objectives into distinct adversarial and clean loss components, allowing us to explicitly balance model robustness and accuracy on clean data.
- We establish a flexible AT framework where one can balance adversarial loss and clean loss by assigning learnable weights to adversarial perturbations. Empirical evaluations show that our method can improve model robustness without sacrificing clean accuracy.

2. Related Work

2.1. Adversarial Training

AT is widely used to improve robustness against malicious adversarial attacks. Let Θ be the model parameters, X be the input feature set and Y be the corresponding label set, with each input data $x \in X$ and label $y \in Y$. In practice, AT is developed to solve the following max-min optimization problem:

$$\min_{\Theta} \max_{\delta} \mathcal{L}(\Theta, x + \delta, y), \tag{1}$$

where δ denotes the minor perturbation term added to the input.

While the outer minimization is often solved by stochastic gradient descent, how to tackle the inner maximization objective function is still under continuous study. Goodfellow et al. [10] proposed FGSM to generate perturbations in one gradient ascent step as follows:

$$\delta = sign(\nabla_x \mathcal{L}(\Theta, x, y)), \tag{2}$$

where $sign(\cdot)$ is the sign function.

However, this approximation can hardly find high-quality adversarial data that can maximize the loss function. To seek more precise solutions, Madry et al. [11] proposed the Projected Gradient Descent (PGD) method to generate perturbations using multi-step gradient ascent steps. The perturbation δ_t and input x_t corresponding to time step t are calculated as follows:

$$\begin{aligned} \delta_t &= \alpha \cdot \nabla_{x_{t-1}} \mathcal{L}(\Theta, x_{t-1}, y), \\ x_t &= x_{t-1} + \delta_t, \end{aligned} \tag{3}$$

where α is the adversarial step size to control the size of perturbations.

Moreover, PGD initializes the search for adversarial data at random starting points within the allowed norm ball, improving the diversity of adversarial data. Empirically, PGD and its variants are still considered the most effective AT methods.

For NLP tasks, AT was first used to improve the generalization of models. Miyato et al. [14] proposed virtual AT to enhance text classification in a semi-supervised manner. To further improve language understanding for pre-trained language models, Zhu et al. [6] proposed FreeLB to provide a large virtual batch size in AT.

In another line of work, AT was adopted to boost the robustness of NLP models. By adversarially perturbing their embedding layer, NLP models were trained to predict consistently on both clean and adversarial data, thereby achieving better adversarial robustness. For example, Li and Qiu [7] proposed TAVAT to generate token-level perturbations accounting for the importance of tokens. Li et al. [15] increased the iteration numbers of AT and found it useful for boosting robustness. Wu et al. [16] introduced adversarial self-attention (ASA), a theoretical framework that restructures transformer attention maps using adversarial perturbations. ASA learns input-dependent prior biases automatically through gradient reversal layers, mitigating overfitting and enhancing robustness. Sinha et al. [17] developed a theoretical framework for generating human-like adversarial examples to bridge the gap between synthetic and real-world attacks. Their work proves that traditional metrics (e.g., semantic similarity) fail to capture real attack patterns. Theoretically, they formalize attack generation as imitating human attack distributions via a generator-discriminator setup, ensuring synthesized samples reflect realistic perturbations. This approach reduces robustness gaps in tasks like natural language inference and hate speech detection. Gao et al. [18] proposed to minimize the distribution shift risk between clean and adversarial data. Formento et al. [19] learned robust word embeddings to defend against adversarial attacks. An et al. [20] formulated a theoretical framework for counterfactual data augmentation using non-likelihood AT. By perturbing causal features via gradient saliency analysis, the above framework generates label-flipped samples that disrupt spurious correlations (e.g., polysemy in Chinese).

With the advancement of large language models (LLMs), research in adversarial attacks against LLMs has revealed that they also exhibit adversarial vulnerability. Geisler et al. [21] proposed REINFORCE to extract sensitive information or training data from the model's memory. Maloyan et al. [22] investigated the vulnerability of LLM-as-a-Judge architectures to prompt-injection. Maloyan and Namiot [23] manipulated the outputs of LLM-as-a-Judge systems. To enhance the robustness of LLMs, researchers have improved the pre-training process by controlling data provenance through methods like data cleansing, while also incorporating safety considerations during the alignment phase [24].

2.2. The Trade-off between Robustness and Accuracy

In computer vision, while AT helps improve robustness, a vast amount of empirical evidence exists that the clean accuracy can be hurt [11, 25]. Zhang et al. [26] theoretically identified the trade-off between robustness and accuracy by decomposing the prediction error for adversarial examples (robust error) as the sum of the natural error and boundary error. Nevertheless, Yang et al. [27] proved that robustness and accuracy should both be achievable for benchmark datasets through locally Lipschitz functions.

For NLP models, early research generally holds that AT improves both robustness and accuracy [14, 28, 6]. However, few studies have focused on the trade-off between robustness and accuracy in AT of NLP models.

It is worth noting that several adversarial data augmentation (ADA) methods [28, 29, 30, 31] expand the original training set with crafted adversarial examples. ADA methods introduce larger perturbations than gradient-based AT methods, leading to relatively low clean accuracy. It demonstrates that there is also a trade-off between robustness and accuracy in AT of NLP models.

In this work, we first demonstrate that with a large perturbation size, robustness trades off clean accuracy in gradient-based AT of NLP models. Further, by decomposing the learning objective of AT into a clean classification loss and an adversarial one, we can explicitly balance clean accuracy and robustness.

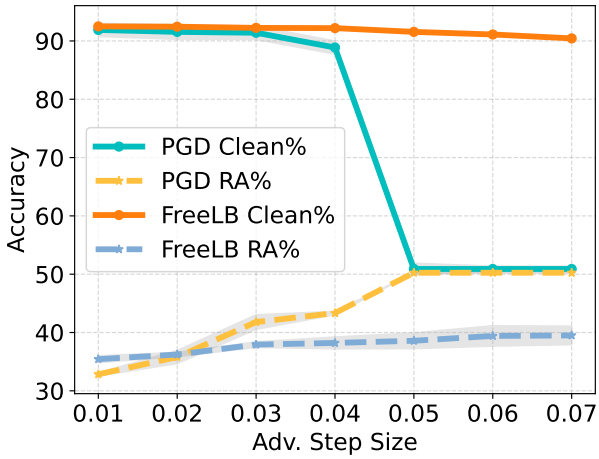


Figure 2: The robust accuracy (**RA**) and clean accuracy (**Clean**) of PGD and FreeLB under different adversarial step sizes on the SST-2 dataset. The backbone model is BERT-base. Although PGD can achieve higher robustness than FreeLB, the clean accuracy of the model is greatly damaged. When the perturbation is too large, the training cannot converge.

3. On the Convergence of Adversarial Training

It is widely pointed out that AT is more difficult than standard training for both computer vision and NLP models [11, 32]. The main reason is that a distribution difference exists between adversarial data and clean data, which makes one model unable to converge well on two widely different data distributions. According to [18], one can model the difference via Wasserstein distance. The authors proved that the distribution shift is bounded by the adversarial perturbation δ. Therefore, δ is crucial in the convergence of AT. We then vary δ to show its effect[2].

Figure 2 shows that as δ increases, the clean accuracy drops significantly, which implies that AT cannot converge well with large perturbations. Furthermore, robust accuracy gradually increases, demonstrating a trade-off between clean accuracy and robust accuracy.

It is also intriguing to see that FreeLB can converge under larger perturbations than PGD. We theoretically analyse the differences among different AT methods to understand this phenomenon. Recall the following learning objective in AT:

$$\min_{\Theta} \mathbb{E}_{(x,y)\sim D}\left[\max_{\|\delta\|\leq\epsilon} \mathcal{L}(\Theta, x+\delta, y)\right], \tag{4}$$

where Θ is the model parameters, D is the data distribution, and ϵ is the allowed perturbation size. In the min-max process, multi-step gradient ascent methods often solve the inner maximization. Take PGD as an example. By initializing x_0 to x and denoting the iteration number as k and the adversarial step size as α, we have:

$$x_t = proj_{\epsilon}(x_{t-1} + \alpha \cdot norm(g(x_{t-1}))), \tag{5}$$

[2]In practice, we vary the step size α in AT to control the perturbation size.

where t is the time step within the range $[1, k]$, $g(x_{t-1})$ is the gradient of x_{t-1}, and $norm(\cdot)$ can be L_2 normalization. The initial value x_0 can also be randomly sampled within the ϵ-neighborhood of x. In that case, we have $x_0 = x + \delta_0$, where δ_0 is randomly sampled.

For simplicity, we omit the projection function and the normalization function. The main reason is that in [15], the authors have demonstrated that removing the norm-bounded limitation helps achieve better model robustness. Thus, we have:

$$x_t = x_{t-1} + \delta_t = x_0 + \sum_{i=1}^{t} \delta_i. \tag{6}$$

In this way, the inner maximization can be reformulated as follows:

$$\max_{\|\delta\| \leq \epsilon} \mathcal{L}(\Theta, x_0 + \sum_{t=1}^{k} \delta_t, y). \tag{7}$$

The magnitude of perturbation at each time step t is determined by the step size α—which typically has a very small value. Therefore, we treat δ_t as an infinitesimal quantity relative to x, and perform the first-order Taylor expansion on the loss function. By omitting the high-order terms and deriving $\delta_k = \alpha \cdot \nabla_{x_{k-1}} \mathcal{L}(\Theta, x_0 + \sum_{t=1}^{k-1} \delta_t, y)$ from Eq. (3), we have:

$$\begin{aligned}
\mathcal{L}(\Theta, x_0 + \sum_{t=1}^{k} \delta_t, y) &= \mathcal{L}(\Theta, x_0 + \sum_{t=1}^{k-1} \delta_t + \delta_k, y) \\
&\approx \mathcal{L}(\Theta, x_0 + \sum_{t=1}^{k-1} \delta_t, y) + \delta_k \cdot \nabla_{x_{k-1}} \mathcal{L}(\Theta, x_0 + \sum_{t=1}^{k-1} \delta_t, y) \\
&= \mathcal{L}(\Theta, x_0 + \sum_{t=1}^{k-1} \delta_t, y) + \frac{1}{\alpha}\delta_k^2 \\
&= \mathcal{L}(\Theta, x_0 + \sum_{t=1}^{k-2} \delta_t + \delta_{k-1}, y) + \frac{1}{\alpha}\delta_k^2 \\
&\approx \mathcal{L}(\Theta, x_0 + \sum_{t=1}^{k-2} \delta_t, y) + \frac{1}{\alpha}\delta_{k-1}^2 + \frac{1}{\alpha}\delta_k^2 \\
&\cdots \\
&\approx \mathcal{L}(\Theta, x_0, y) + \frac{1}{\alpha}\sum_{t=1}^{k} \delta_t^2.
\end{aligned} \tag{8}$$

Eq. (8) indicates that one can decompose the loss of adversarial data during PGD training into the corresponding loss of clean data and the sum of squares of all perturbations δ_t.

Therefore, it is reasonable that as the perturbation size increases, the adversarial loss becomes larger and begins to dominate the training, leading to higher robustness. For clean accuracy, as the perturbation size enlarges, the model gets harder to converge on the original training set, resulting in lower clean accuracy.

Based on Eq. (8), we further study how δ affects the convergence of AT. We firstly extend Eq. (8) to FreeLB. It can also be easily extended to other PGD-like methods such as FreeLB++.

According to the FreeLB method, the number of iterations is k and the step size is α. The loss of each iteration will be divided by k and accumulated. The model parameters will be updated at the end (for comparison, PGD only uses the loss of the last iteration to update the model parameters). Similarly, the inner maximization of FreeLB can be formulated as follows:

$$\max_{\|\delta\|\leq\epsilon} \frac{1}{k}\sum_{t=1}^{k}\mathcal{L}(\Theta, x_0 + r^t, y), \tag{9}$$

where $r^t = \sum_{i=1}^{t}\delta_i$. By performing the first-order Taylor expansion on Eq. (9), similar to Eq. (8), we have:

$$\begin{aligned}\frac{1}{k}\sum_{t=1}^{k}\mathcal{L}(\Theta, x_0 + r^t, y) &\approx \frac{1}{k}\sum_{t=1}^{k}(\mathcal{L}(\Theta, x_0, y) + \frac{1}{\alpha}\sum_{i=1}^{t}\delta_i^2) \\ &= \mathcal{L}(\Theta, x_0, y) + \frac{1}{\alpha}\sum_{i=1}^{k}\frac{k-i+1}{k}\delta_i^2.\end{aligned} \tag{10}$$

Eq. (10) indicates that the learning objective of FreeLB can also be decomposed into the clean data loss and the weighted sum of squares of all perturbations δ_i, where the weight of δ_i is $\frac{k-i+1}{k}$.

At this point, we can explain more clearly in Figure 2. Since the PGD method inherently has a greater weight for adversarial loss, it can achieve higher robustness than FreeLB, but the training of PGD is more difficult to converge.

4. Adaptive Adversarial Training

4.1. A Unifying Framework for Adversarial Training

Comparing the two learning objectives, one can find an implicit set of weights weighing the perturbation δ_i produced at each iteration i. Further, the weights of the clean classification loss and the adversarial one are also implicitly given. For the PGD method with an iteration number of k, the weights of clean loss and the adversarial loss are 1 and k, respectively. The FreeLB method's weights are 1 and $(k+1)/2$, respectively.

Therefore, we summarize the learning objective $\mathcal{J}$ of the two methods into the following formula:

$$\begin{aligned}\mathcal{J}(\Theta, x_0, y, \mathbf{w}, \beta) &= \mathcal{L}(\Theta, x_0, y) + \beta\frac{1}{\alpha}\sum_{i=1}^{k} w_i\delta_i^2, \\ s.t. &\sum_{i=1}^{k} w_i = 1, w_i \geq 0,\end{aligned} \tag{11}$$

where β balances the clean loss and the adversarial loss, and w_i balances all the perturbations.

However, since the derivative of the sum of squared perturbations involves computing the second-order derivative, we further manipulate the above formula to avoid the second-order derivative. Specifically, we introduce the following equation:

$$\hat{\mathcal{J}}(\Theta, x_0, y, \mathbf{w}, \beta) = \mathcal{L}(\Theta, x_0 + \beta\cdot\sum_{i=1}^{k} w_i\delta_i, y). \tag{12}$$

Algorithm 1 Adaptive Adversarial Training

Require: Model parameters Θ, loss function $\mathcal{L}$, training set $D = \{x_i, y_i\}_{i=1}^{n}$, number of epochs T, batch size m, number of iterations k, number of batches M, perturbation weights $\mathbf{w}$, weighting factor β

Ensure: robust model parameters Θ

```
for epoch = 1 to T do
    for batch = 1 to M do
        Sample a mini-batch b = {(x_i, y_i)}_{i=1}^m
        Generate adversarial perturbations δ via Eq. (3)
        Compute the overall loss Ĵ(Θ, x, y, w, β) via Eq. (12)
        Update w via ∇_w Ĵ(Θ, x, y, w, β)
        Update Θ via ∇_Θ Ĵ(Θ, x, y, w, β)
    end for
end for
```

By performing Taylor expansion on Eq. (12) and leveraging $\delta_k = \alpha \cdot \nabla_{x_{k-1}} \mathcal{L}(\Theta, x_0 + \sum_{t=1}^{k-1} \delta_t, y)$, we can easily verify that each term corresponds one-to-one with Eq. (11), i.e.,

$$
\begin{aligned}
\hat{\mathcal{J}}(\Theta, x_0, y, \mathbf{w}, \beta) &= \mathcal{L}(\Theta, x_0 + \beta \cdot \sum_{i=1}^{k} w_i \delta_i, y) \\
&= \mathcal{L}(\Theta, x_0 + \beta \cdot \sum_{i=1}^{k-1} w_i \delta_i + \beta w_k \delta_k, y) \\
&\approx \mathcal{L}(\Theta, x_0 + \beta \cdot \sum_{i=1}^{k-1} w_i \delta_i, y) + \beta w_k \delta_k \nabla_{x_{k-1}} \mathcal{L}(\Theta, x_0 + \beta \cdot \sum_{i=1}^{k-1} w_i \delta_i, y) \\
&= \mathcal{L}(\Theta, x_0 + \beta \cdot \sum_{i=1}^{k-1} w_i \delta_i, y) + \beta \frac{1}{\alpha} w_k \delta_k^2 \\
&\approx \mathcal{L}(\Theta, x_0 + \beta \cdot \sum_{i=1}^{k-2} w_i \delta_i, y) + \beta \frac{1}{\alpha} w_{k-1} \delta_{k-1}^2 + \beta \frac{1}{\alpha} w_k \delta_k^2 \\
&\cdots \\
&= \mathcal{L}(\Theta, x_0, y) + \beta \frac{1}{\alpha} \sum_{i=1}^{k} w_i \delta_i^2 \\
&= \mathcal{J}(\Theta, x_0, y, \mathbf{w}, \beta)
\end{aligned}
\tag{13}
$$

Therefore, $\hat{\mathcal{J}}$ is equal to $\mathcal{J}$. In our experiments, we initialize $\mathbf{w}$ to a vector of ones and update it automatically using its gradient.

4.2. The Rationale behind Our Framework

Next, we explain the rationale behind introducing β and $\mathbf{w}$. As deduced above, in the PGD method, the weight of the clean loss is naturally set to 1, while the weight of the adversarial loss is set to k. In the FreeLB method, the weight of the clean loss is also 1, but the weight of the

adversarial loss is $(k + 1)/2$. To ensure the extensibility of our AT framework, we introduce the parameter β to balance the clean loss and adversarial loss. Specifically, PGD and FreeLB are two special cases of the proposed framework.

Eq. (10) shows that while maintaining the original ratio between clean loss and adversarial loss, the perturbations at each time step t are assigned different weights. Therefore, we introduce a set of parameters $\mathbf{w}$, ensuring that the sum of w_i equals 1, and utilize gradients to solve for the worst-case scenario. The weights $\mathbf{w}$ are continuously updated throughout the training process, in order to find the best values across the entire training set rather than achieve a local solution based on a single batch of data. This weighting strategy allows us to identify the worst-case scenarios that maximize the model's loss. Subsequently, we update the model parameters using these adversarially weighted examples to enhance robustness. Although gradient descent does not guarantee convergence to the global solution, our extensive empirical evidence demonstrates that the resulting weight assignment achieved by this method is superior to using fixed weights. For example, in the PGD method, even though different time-step perturbations are not explicitly weighted, one can assume that their weights are uniformly set to 1.

Following the min-max optimization widely used in AT, the final training objective can be defined as:

$$\min_{\Theta} \max_{\mathbf{w}} \hat{\mathcal{J}}(\Theta, x_0, y, \mathbf{w}, \beta). \tag{14}$$

In this way, we build our novel framework of adaptive AT in a constrained manner, where both the PGD and the FreeLB methods can be considered special cases of our framework.

Notably, our framework can encompass a wider range of PGD-based AT algorithms, not limited to FreeLB. We show our proposed adaptive AT method in Algorithm 1.

5. Experimental Setup

5.1. Tasks and Datasets

Following previous important works [18, 15, 7], we compare our adaptive AT method with baselines on two tasks, i.e., text classification and natural language inference. In the main experiments, we choose the SST-2 [13][3] and the QNLI [33][4] datasets to perform text classification and natural language inference tasks, respectively. For completeness, we also test the applicability of our method on the IMDB dataset [34] and the AGNEWS dataset [35], both used for text classification. Detailed characteristics and examples of the four datasets are presented below.

- For the SST-2 dataset, an example of x and y is "*On the worst revenge-of-the-nerds clichés the filmmakers could dredge up*" and "*Negative*".
- For the IMDB dataset, an example of x and y is "*Fred "The Hammer" Williamson delivers another cheaply made movie. He might have set a new standard for himself. Look for the painfully obvious special effects mortar cannon that is visible in the street during a chase scene. You don't see it just once, you see it several times. Look for the out of focus shot in one scene and the camera operator try to fix it as the scene rolls on. Watch this with a group of people and make your own Mystery Science Theater!*" and "*Negative*".

[3] `https://dl.fbaipublicfiles.com/glue/data/SST-2.zip`
[4] `https://huggingface.co/datasets/nyu-mll/glue`

- For the AGNEWS dataset, an example of x and y is "*Wall St. Bears Claw Back Into the Black (Reuters) Reuters - Short-sellers, Wall Street's dwindling band of ultra-cynics, are seeing green again.*" and "*Business*".
- For the QNLI dataset, an example of x and y is "*When did the third Digimon series begin? Unlike the two seasons before it and most of the seasons that followed, Digimon Tamers takes a darker and more realistic approach to its story featuring Digimon who do not reincarnate after their deaths and more complex character development in the original Japanese*" and "*Not entailment*".

We list the characteristics of the four datasets below.

Dataset	**# train**	**# dev / test**	**# words**
SST-2	67,349	872	17
QNLI	105,000	5,460	37
IMDB	25,000	25,000	201
AG news	120,000	7,600	40

Table 1: Summary of the four datasets.

5.2. Baseline Methods

5.2.1. Defence Methods

We apply our framework to various AT-based defence methods, including PGD [11], FreeLB [6], and TA-VAT [7]. To comprehensively benchmark existing defence methods, we report the results of InfoBERT [8], Flooding-X [36], and SMART [37] which enhance AT by an information bottleneck, "flooding", and smoothness-inducing regularization, respectively. The performance of TRADES [26], which is the most relevant method from the computer vision domain to ours, is also presented.

SemRoDe [19] and ADPMIXUP [38] are not chosen for the main experiment. Because these two methods incorporate valid adversarial examples into the training process (a process also known as adversarial augmentation training), and they leak information about the attacking methods, making the results less convincing. Considering such methods have attracted the attention of some researchers, in section 7.2, we further investigate our method's effectiveness compared with state-of-the-art adversarial augmentation training methods in a fair setup.

5.2.2. Attacking Methods

Following previous works [15, 19], we use TextFooler [30], TextBugger [29], and BAE [39] as our attacking methods to dynamically generate adversarial examples during test time.

We also consider assessing AT methods against high-quality adversarial examples pre-crafted by human annotators. Therefore, we report the robust accuracy of all the models on the adversarial GLUE dataset [40].

5.3. Implementation Details

We implement PGD [11], FreeLB [6], TA-VAT [7], and InfoBERT [8] based on TextDefender [15]. We implement Flooding-X [36], SMART [37], and TRADES [26] following the original paper. The weighting factor α in TRADES is set to 0.5 to achieve the optimal performance. The

SST-2	Clean %	TextFooler	TextBugger	BAE	AdvGLUE
		RA %	RA %	RA %	RA %
BERT-base [12]	92.32	8.14	26.83	33.72	31.32
InfoBERT [8]	91.74	10.89	32.68	37.96	32.17
Flooding-X [36]	92.32	12.60	32.45	35.44	27.00
SMART [37]	91.78	10.45	30.15	33.26	23.54
TRADES [26]	87.19	9.46	29.53	35.41	30.99
PGD [11]	89.11	12.96	32.22	35.21	39.13
+Ours	88.99 (-0.12)	16.06 (+3.10)	35.68 (+3.46)	40.02 (+4.81)	43.44 (+4.31)
FreeLB [6]	92.20	9.98	34.06	37.73	30.13
+Ours	91.63 (-0.57)	15.69 (+5.71)	38.73 (+4.67)	41.22 (+3.49)	38.53 (+8.40)
TA-VAT [7]	91.40	11.93	35.89	37.61	32.00
+Ours	91.51 (+0.11)	18.46 (+6.53)	39.60 (+3.71)	40.94 (+3.33)	39.42 (+7.42)

Table 2: The clean accuracy ("Clean %") and the robust accuracy ("RA %") on the SST-2 dataset against TextFooler, TextBugger, and BAE attacks. We report the robust accuracy of the models on the adversarial GLUE dataset (i.e., AdvGLUE) to evaluate AT methods against pre-crafted adversarial examples. The backbone model is BERT-base.

three adversarial attacks are conducted using TextAttack[5] [41]. All experiments are conducted using GeForce RTX 3090 GPUs. All the settings of adversarial hyper-parameters settings are consistent to provide a fair comparison.

Unless otherwise mentioned, the adversarial step size is set to 0.04; the batch size is 128; the epoch number is 10. To align with the weighting factor of the original method, β is set to k for PGD and TA-VAT and $(k+1)/2$ for FreeLB.

For the natural language inference task, we adhere to prior research [30] by allowing the attacking methods to modify the premise while keeping the hypothesis unchanged.

6. Main Results

Our proposed method can be easily extended to PGD-like AT methods. In this part, we advance PGD, FreeLB and TA-VAT with adaptive perturbations to assess the effectiveness of our method. We conduct the main experiments on the BERT-base model to provide comprehensive comparisons with other AT methods.

Note that the value of β is related to the methods being extended. For example, when extending the PGD method using our framework, the value of β is set to k (i.e., the number of iterations) according to Eq. (8). We leave the exploration of the effects of different β values for future work.

Table 2 reports the defence results against different types of adversarial attacks on the SST-2 dataset, including two word-level attacks (TextFooler and BAE), one multi-level attack (TextBugger), and an adversarial test dataset (Adversarial GLUE). The main findings are:

- For clean accuracy, all the baseline AT methods maintain a similar level, since the adversarial strength is moderate. The PGD method has the lowest clean accuracy, which is consistent with the conclusions of previous work.

[5] https://github.com/QData/TextAttack

QNLI	Clean %	TextFooler	TextBugger	BAE	AdvGLUE
		RA %	RA %	RA %	RA %
BERT-base [12]	90.60	8.80	9.50	27.90	42.75
InfoBERT [8]	89.10	5.30	6.80	30.90	44.00
Flooding-X [36]	91.50	12.00	16.60	40.30	47.00
SMART [37]	91.77	8.50	13.22	33.46	39.02
TRADES [26]	86.22	9.45	12.14	35.44	43.50
PGD [11]	87.00	11.30	16.80	43.60	41.50
+Ours	87.90 (+0.90)	16.80 (+5.50)	17.20 (+0.40)	41.20 (-2.40)	48.89 (+7.39)
FreeLB [6]	89.60	14.40	14.10	40.50	44.75
+Ours	89.70 (+0.10)	16.60 (+2.20)	17.70 (+3.60)	43.10 (+2.60)	51.75 (+7.00)
TA-VAT [7]	91.51	12.60	14.30	40.94	43.00
+Ours	91.00 (-0.51)	18.46 (+5.86)	20.30 (+6.00)	44.20 (+3.26)	51.00 (+8.00)

Table 3: The clean accuracy ("Clean %") and the robust accuracy ("RA %") on the QNLI dataset against TextFooler, TextBugger, and BAE attacks. We report the robust accuracy of the models on the adversarial GLUE dataset (i.e., AdvGLUE) to evaluate AT methods against pre-crafted adversarial examples. The backbone model is BERT-base.

- For robust accuracy against dynamic adversarial attacks and human-crafted adversarial examples, our method can boost the performance of three AT methods. Compared with InfoBERT and Flooding-X, our method also maintains higher robustness.

- Our method can boost the robust accuracy of PGD, FreeLB and TA-VAT methods while achieving comparable clean accuracy, which is consistent with our motivations.

We also conduct experiments on the QNLI dataset. The main results are consistent with that on the SST-2 dataset. Our method consistently enhances robust accuracy across various adversarial attacks and test sets. Thanks to the adaptive strength of perturbations, the clean accuracy remains at a comparable level compared to other AT methods.

We note that the PGD method still has the lowest clean accuracy. According to Eq. (8), the PGD method implicitly places a greater weight on the adversarial loss than FreeLB, which sacrifices the clean accuracy. Since it is directly adopted from the visual domain, no adjustments have been made to the trade-off between robustness and clean accuracy. As a consequence, this method exhibits lower clean accuracy on NLP tasks.

The results on the IMDB and the AGNEWS datasets are reported in Tables 4 and 5. In terms of clean accuracy, our method maintains a performance level comparable to the baseline. In terms of robustness accuracy, our method improves the robustness of the baseline in most scenarios.

It is noteworthy that the improvement in robustness is relatively small on these two datasets. This may be related to the sentence length in the datasets. Existing adversarial attack algorithms typically set the maximum number of word replacements based on a percentage of the sentence's token count, such as 20%. As the length increases, the number of words to be replaced also increases, which may result in less significant improvements in robustness.

IMDB	Clean %	TextFooler	TextBugger	BAE
		RA %	RA %	RA %
BERT [12]	91.21	24.48	47.26	20.31
InfoBERT [8]	91.90	23.00	37.30	22.40
Flooding-X [36]	92.30	34.50	32.30	35.42
SMART [37]	91.90	24.50	45.40	22.32
TRADES [26]	88.34	25.50	47.60	18.34
PGD [11]	90.43	26.31	52.37	21.44
+Ours	90.56 (+0.13)	27.12 (+0.81)	53.50 (+1.13)	21.55 (+0.11)
FreeLB [6]	92.14	27.50	50.60	31.34
+Ours	91.80 (-0.34)	26.82 (-0.68)	52.74 (+2.14)	33.10 (+1.76)
TA-VAT [7]	91.50	27.40	51.70	23.12
+Ours	92.08 (+0.58)	25.70 (-1.70)	51.66 (-0.04)	24.30 (+1.18)

Table 4: The clean accuracy ("Clean %") and the robust accuracy ("RA %") on the IMDB dataset against TextFooler, TextBugger, and BAE attacks. The backbone model is BERT-base. The IMDB dataset does not have a corresponding AdvGLUE version. Therefore, the robustness accuracy for this dataset is not reported.

7. Discussions

In this section, we discuss the relationship between our method and existing AT methods. We compare our adaptive AT method with adversarial augmentation training methods to further demonstrate its effectiveness. We highlight the importance of conducting AT on small language models like BERT, rather than solely focusing on LLMs. We also provide an error analysis of the approximate loss and demonstrate the PGD loss and approximate loss in AT in practice.

7.1. Relation to Existing Work

We list a series of loss functions of AT methods in Table 6 and discuss the difference between our proposed adaptive AT and conventional AT methods, including fast gradient method (FGM) [14], PGD [11], TRADES [26], and FreeLB [6].

Specifically, the standard method is designed to minimize the clean data loss, i.e., the cross-entropy on the clean data. The FGM [14] method generates adversarial examples in one gradient ascent step, minimising both clean and adversarial data loss. The PGD method [11] generates adversarial examples using multi-step gradient ascent and only minimizes the adversarial data loss in the last step. Similarly, the FreeLB method [6] generates adversarial examples using multi-step gradient ascent. Different from PGD, FreeLB minimize the average of the adversarial loss at each step.

It is important to point out that all these methods implicitly include the clean data loss in the adversarial loss. In particular, as revealed by Eq. (8) and Eq. (10), the conventional adversarial loss can be decomposed into a clean data loss and an adversarial loss. Therefore, although we can introduce hyperparameters to balance clean loss and adversarial loss in these methods, we cannot precisely balance the two losses.

TRADES [26] is theoretically designed to achieve a good trade-off between accuracy and robustness in the computer vision domain, which is the most relevant AT method with our adaptive AT. TRADES decomposes the adversarial error into a natural error and a boundary error.

AGNEWS	Clean %	TextFooler	TextBugger	BAE
		RA %	RA %	RA %
BERT [12]	91.90	20.50	42.71	16.21
InfoBERT [8]	92.00	19.20	31.41	12.70
Flooding-X [36]	91.39	33.40	55.60	29.40
SMART [37]	92.20	22.45	37.80	15.60
TRADES [26]	89.42	33.90	48.65	27.61
PGD [11]	90.82	37.20	58.20	32.83
+Ours	91.10 (+0.28)	38.70 (+1.50)	57.92 (-0.28)	35.20 (+2.37)
FreeLB [6]	91.20	32.33	48.50	22.65
+Ours	91.07 (-0.13)	32.10 (-0.23)	50.10 (+1.60)	24.12 (+1.47)
TA-VAT [7]	92.17	39.70	55.81	23.66
+Ours	91.66 (-0.51)	37.26 (-2.44)	57.36 (+1.55)	23.77 (+0.11)

Table 5: The clean accuracy ("Clean %") and the robust accuracy ("RA %") on the AGNEWS dataset against TextFooler, TextBugger, and BAE attacks. The backbone model is BERT-base. The AGNEWS dataset does not have a corresponding AdvGLUE version. Therefore, the robustness accuracy for this dataset is not reported.

However, the boundary error cannot be effectively computed. In practice, the authors introduce a surrogate loss (i.e., the KL divergence between the model output of clean data and adversarial data) to approximate the boundary error. In this way, TRADES cannot precisely balance the standard performance and robustness.

Our proposed adaptive AT addresses this issue by decomposing the conventional adversarial loss using Taylor expansion. In our learning objective, clean loss and adversarial loss only affect standard performance and model robustness, respectively.

7.2. Comparisons with Adversarial Augmentation Training

Adversarial augmentation training methods have attracted the attention of some researchers [38, 19] by expanding training sets with valid adversarial examples, albeit at the cost of potentially leaking information about the attacking methods. To further demonstrate the effectiveness of our framework, we here apply the same extra data in our approach for a fair comparison with adversarial augmentation training.

Specifically, we employ the TextFooler algorithm [30] to generate adversarial perturbations for the SST-2 training set, thereby constructing an augmented dataset. Leveraging this enhanced training corpus, we conduct a systematic comparison between our adaptive AT framework and two state-of-the-art adversarial augmentation training methods, ADPMIXUP [38] and SemRoDe [19], quantitatively evaluating their effectiveness in improving model robustness.

As can be seen in Table 7, compared to the two state-of-the-art adversarial augmentation training methods, our AT framework can achieve comparable clean and robust accuracy. In particular, we find that before and after the introduction of additional data, although our method does not construct a specific learning objective for the extra data like ADPMIXUP and SemRoDe, it still achieves a huge improvement in robust accuracy (36.5% on the SST2 dataset and 28.5% on the QNLI dataset). We give a possible explanation that the existing adversarial attack algorithms share knowledge bases, such as replacement word sets. Expanding the training set after generating additional data will leak this information. Even if different attack algorithms are

Methods	**Loss Function**	**Flexibility**
Standard	$\mathcal{L}(\Theta, x, y)$	-
FGM [14]	$\mathcal{L}(\Theta, x, y) + \mathcal{L}(\Theta, x + \delta, y)$	✗
PGD [11]	$\mathcal{L}(\Theta, x + \delta_k, y)$	✗
TRADES [26]	$\mathcal{L}(\Theta, x, y) + \lambda KL(p(\Theta, x) \| p(\Theta, x + \delta))$	✓✗
FreeLB [6]	$\frac{1}{k}\sum_{i=1}^{k} \mathcal{L}(\Theta, x + \delta_i, y)$	✗
Ours	$\mathcal{L}(\Theta, x, y) + \beta\frac{1}{\alpha}\sum_{i=1}^{k} w_i\delta_i^2$	✓

Table 6: Comparisons of different loss functions in AT. The adversarial perturbations in TRADES are generated by maximizing its regularization term (KL-divergence). The **Flexibility** indicates whether the method can explicitly control the weighting between clean loss and adversarial loss. ✗indicates that the method cannot balance clean and adversarial losses. ✓✗ indicates that the method introduces a hyperparameter to balance the two types of loss, but lacks flexibility because the adversarial loss still contains the clean loss. ✓indicates that it can explicitly balance clean loss and adversarial loss.

Dataset	**Method**	**RoBERTa**		**BERT**	
		Clean %	**RA %**	**Clean %**	**RA %**
SST2	ADPMIXUP [38]	**96.3**	58.9	92.3	**67.9**
	SemRoDe [19]	94.2	46.6	**94.2**	40.2
	PGD+Ours	**96.3**	**59.1**	92.7	52.5
QNLI	ADPMIXUP [38]	94.4	**66.7**	87.2	44.2
	SemRoDe [19]	91.2	35.2	90.6	39.7
	PGD+Ours	**95.2**	58.3	**92.4**	**45.3**

Table 7: The clean and robust accuracy on the SST-2 and QNLI datasets with two adversarial augmentation methods, ADPMIXUP and SemRoDe. Our method uses the same extra data to provide fair comparisons. The attacking method is TextFooler. Our method performs comparable to the two state-of-the-art adversarial augmentation training methods when sharing the same extra data.

replaced during testing, the leakage of this shared information can result in a huge improvement in robustness.

7.3. Beyond Model Parameters

Recently, LLMs have achieved remarkable results across many NLP tasks [42, 43]. There is also a body of work investigating adversarial vulnerabilities specifically for LLMs [21, 22, 23, 24]. Therefore, it is necessary to explain why this work focuses on AT for pre-trained language models. We select a more practical task, namely spam detection, and report the standard performance of models with varying parameter sizes in Table 8, including naïve Bayes (NB), support vector machine (SVM), BERT, and LLMs. We adopt the SMS Spam Collection dataset [44], which contains 747 spam messages and 4,825 non-spam messages. The long-tail distribution of the data makes it more realistic and challenging.

As can be seen, even the state-of-the-art DeepSeek-r1 model [43] performs poorly on this dataset, which may be related to the data distribution. However, small models generalize well on this dataset.

Method	Acc.	Pre.	Recall	F1
SVM (linear)	97.56	97.01	84.82	90.50
Multinomia NB	98.21	98.26	88.48	93.11
BERT-base	99.48	94.44	91.15	92.61
DeepSeek-r1-zeroshot	87.74	52.71	95.77	68.00
DeepSeek-r1-fewshot	95.45	79.75	91.30	85.14

Table 8: The performance of models with varying parameter sizes on the spam detection task. We use deepseek-r1 [43] to demonstrate the performance of LLMs on this dataset in zero-shot and few-shot manners.

Given the constraints of computational resources and training efficiency, this study proposes to investigate AT for BERT-based architectures to mitigate vulnerabilities against adversarial perturbations, rather than focusing on LLMs.

The details on the usage of DeepSeek are presented below. We employ the DeepSeek-r1 model [43] for spam detection and evaluate its performance under zero-shot and few-shot settings. In the zero-shot setting, the model receives no examples or labels and is prompted to classify the message based solely on its inherent reasoning ability. The prompt provided is: "You are a professional spam classifier. Please analyze the following message and determine whether it is spam. Just reply 'spam' or 'ham', no explanation is needed." This setup tests the model's ability to classify messages without prior examples or labels.

In the few-shot setting, we supply the model with two examples and their corresponding labels. The first example is a spam message: "*URGENT! This is the 2nd attempt to contact U!U have WON å£1000CALL 09071512432 b4 300603t&csBCM4235WC1N3XX.callcost150 ppmmobilesvary. maxå£7.50*", labeled as spam. The second example is a non-spam message: "*Why don't you go tell your friend you're not sure you want to live with him because he smokes too much then spend hours begging him to come smoke*", labeled as ham. This setting aims to examine how the model leverages the provided examples to classify messages.

Through these two setups, we assess the model's generalization ability and performance when there are no explicit labels or examples available.

7.4. Impact of Adversarial Step Size

We aim to investigate the impact of the perturbation size in AT. In AT, the maximum perturbation size is typically specified. However, what effectively determines the perturbation magnitude are the number of iterations and the adversarial step size. Therefore, given a perturbation size, we vary the number of iterations and step size to investigate their impact on robustness. In other words, we want to find out whether increasing the perturbation strength of AT always helps robustness. To achieve this, we conduct PGD on the BERT-base model. Based on our previous experiments, the product of iteration numbers k and adversarial step size α is empirically set to 10 and 0.4.

The main result is reported in Figure 3. It can be seen that when the number of iterations is moderate (5 and 6), the model achieves the best robustness. We suggest that it is unnecessary to set a huge number of iterations during AT. As suggested in [45], robust overfitting hinders the AT of NLP models. Too many iterations may lead to robust overfitting of the model and reduce its robustness accuracy on the test set.

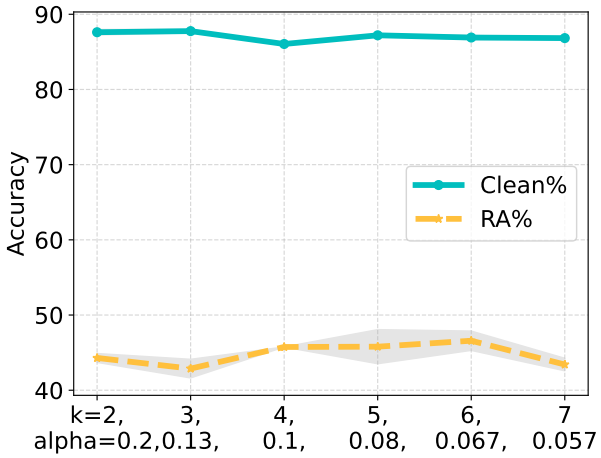


Figure 3: The robust accuracy and clean accuracy under different k and α, while the maximal perturbation size is set to $k\alpha$ following [15].

SST2	**Clean %**	**TextFooler**	**AdvGLUE**
		RA %	**RA %**
RoBERTa-base	95.07	6.19	39.50
+PGD	94.27	11.47	44.59
+Ours	94.95	11.82	45.22
DeBERTa-v3-base	95.99	12.60	55.41
+PGD	95.18	13.99	57.14
+Ours	95.76	14.50	67.34

Table 9: The clean and robust accuracy on RoBERTa [47] and DeBERTa-v3-base [46].

7.5. Performance on Other Models

We choose DeBERTa-v3-base [46] and RoBERTa [47], two improved versions of BERT, as our backbone models to investigate whether our method can boost the robustness of more complex and larger language models. The clean and robust accuracy of DeBERTa-v3-base and RoBERTa-base models are reported in Table 9. These two models can bear a larger perturbation size than the BERT-base model to explore the impact of a larger perturbation range on AT. The empirical results indicate that our adaptive AT framework can generalize well on larger, more complex models.

7.6. Time Consumption

To further substantiate the comparative advantages of our method, a systematic benchmarking analysis was conducted to evaluate GPU training durations between our proposed approach and established AT methods, with the quantitative comparisons meticulously documented in Table 10. Our method incurs approximately a 10% increase in computational overhead. This empirical investigation demonstrates our method's computational efficiency while maintaining equivalent adversarial robustness metrics.

Method	SST-2	QNLI
PGD [11]	902	4123
+Ours	912	4237
FreeLB [6]	781	3122
+Ours	920	3745
TA-VAT [7]	853	3455
+Ours	1013	4123

Table 10: The GPU time consumption (seconds) of training one epoch on the SST-2 and QNLI datasets. The backbone model is BERT-base. The iteration number is set to 5 for all the methods.

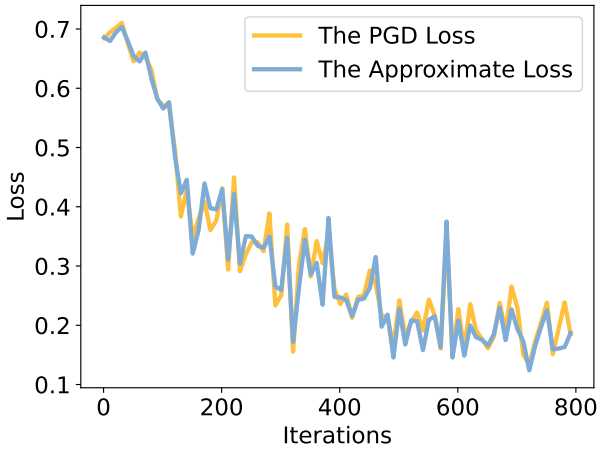


Figure 4: The error between the approximate loss and the original PGD loss on the SST2 dataset over the BERT model. This indicates that our approximation of the experiments is quite practicable.

7.7. Error Analysis

It is necessary to analyze the error of our method since we have ignored the higher-order terms in the Taylor expansion. Taking the PGD method as an example, we show the error between the approximate loss and the original PGD loss. The original PGD loss is computed by Eq. (7). The approximate is computed by Eq. (8).

In Figure 4, we observe that the approximate loss can well match the loss curve of the PGD method, which indicates that the impact of ignoring higher-order terms is negligible. This also demonstrates that our approximation is accurate in the experiments and it can be used to develop AT with an adaptive perturbation.

8. Conclusions

This work seeks to balance model robustness and accuracy. To this end, we demonstrate that existing AT methods contribute little to model robustness with a small perturbation size. Through theoretical analysis of existing AT paradigms, we decompose the learning objective of AT into a pure adversarial loss and clean loss, which correspond to model robustness and

clean accuracy, respectively. This way, we can explicitly assign learnable weights to the two losses to balance model robustness and clean accuracy. Experimental results on four datasets over BERT, RoBERTa and DeBERTa models show that our method can boost model robustness without sacrificing clean accuracy. We also provide extensive discussions about the parameter sensitivity, time consumption, and the relation to existing work. In the future, we plan to integrate our adaptive AT framework with a dynamic weight allocation strategy based on sample difficulty, which is expected to mitigate the issue of robust overfitting [48]. To strengthen the theoretical foundation and improve the solution quality, we also plan to theoretically analyze the effect of learning weights dynamically.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (62372483).

Author Contributions

Bin Zhu, Conceptualization; Bin Zhu and Siyu Liu, Methodology; Bin Zhu, Software; Bin Zhu, Writing original draft; Bin Zhu, Data curation; Bin Zhu, and Yuyin Lu, Formal analysis; Yanghui Rao, Funding acquisition; Bin Zhu, Investigation; Bin Zhu, Siyu Liu, Yuyin Lu, and Jielin Song, Writing-review and editing; Yanghui Rao, Project administration; Yanghui Rao, Supervision; Bin Zhu, Visualization.

References

[1] F. Tramer, D. Boneh, Adversarial training and robustness for multiple perturbations, in: Proc. of NeurIPS, 2019, pp. 5858–5868.
[2] A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, P. Liang, Understanding and mitigating the tradeoff between robustness and accuracy, in: Proc. of ICML, 2020, pp. 7909–7919.
[3] X. Yuan, P. He, Q. Zhu, X. Li, Adversarial examples: Attacks and defenses for deep learning, IEEE Transactions on Neural Networks and Learning Systems 30 (9) (2019) 2805–2824.
[4] J. Zhang, C. Li, Adversarial examples: Opportunities and challenges, IEEE Transactions on Neural Networks and Learning Systems 31 (7) (2020) 2578–2593.
[5] J. Y. Yoo, Y. Qi, Towards improving adversarial training of NLP models, in: Proc. of EMNLP Findings, 2021, pp. 945–956.
[6] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, J. Liu, Freelb: Enhanced adversarial training for natural language understanding, in: Proc. of ICLR, 2020.
[7] L. Li, X. Qiu, Token-aware virtual adversarial training in natural language understanding, in: Proc. of AAAI, 2021, pp. 8410–8418.
[8] B. Wang, S. Wang, Y. Cheng, Z. Gan, R. Jia, B. Li, J. Liu, Info{bert}: Improving robustness of language models from an information theoretic perspective, in: Proc. of ICLR, 2021.
[9] Y. Wang, X. Ma, J. Bailey, J. Yi, B. Zhou, Q. Gu, On the convergence and robustness of adversarial training, in: Proc. of ICML, 2019, pp. 6586–6595.
[10] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: Proc. of ICLR, 2015.
[11] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: Proc. of ICLR, 2018.
[12] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proc. of NAACL-HLT, 2019, pp. 4171–4186.
[13] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proc. of ACL, 2013, pp. 1631–1642.
[14] T. Miyato, S. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: A regularization method for supervised and semi-supervised learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (8) (2019) 1979–1993.

[15] Z. Li, J. Xu, J. Zeng, L. Li, X. Zheng, Q. Zhang, K. Chang, C. Hsieh, Searching for an effective defender: Benchmarking defense against adversarial word substitution, in: Proc. of EMNLP, 2021, pp. 3137–3147.
[16] H. Wu, R. Ding, H. Zhao, P. Xie, F. Huang, M. Zhang, Adversarial self-attention for language understanding, in: Proc. of AAAI, 2023, pp. 13727–13735.
[17] A. Sinha, A. Balashankar, A. Beirami, T. Avrahami, J. Chen, A. Beutel, Break it, imitate it, fix it: Robustness by generating human-like attacks, Transactions on Machine Learning Research 2024.
[18] S. Gao, S. Dou, Y. Liu, X. Wang, Q. Zhang, Z. Wei, J. Ma, Y. Shan, DSRM: boost textual adversarial training with distribution shift risk minimization, in: Proc. of ACL, 2023, pp. 12177–12189.
[19] B. Formento, W. Feng, C. Foo, A. T. Luu, S. Ng, Semrode: Macro adversarial training to learn representations that are robust to word-level attacks, in: Proc. of NAACL-HLT, 2024, pp. 8005–8028.
[20] D. An, F. Wang, S. Zhang, Y. Li, Exploiting non-likelihood adversarial training for chinese counterfactual data augmentation, Engineering Applications of Artificial Intelligence 156 (2025) 111149.
[21] S. Geisler, T. Wollschläger, M. H. I. Abdalla, V. Cohen-Addad, J. Gasteiger, S. Günnemann, REINFORCE adversarial attacks on large language models: An adaptive, distributional, and semantic objective, CoRR abs/2502.17254, 2025. arXiv:2502.17254.
[22] N. Maloyan, B. Ashinov, D. Namiot, Investigating the vulnerability of llm-as-a-judge architectures to prompt-injection attacks, CoRR abs/2505.13348, 2025. arXiv:2505.13348.
[23] N. Maloyan, D. Namiot, Adversarial attacks on llm-as-a-judge systems: Insights from prompt injections, CoRR abs/2504.18333, 2025. arXiv:2504.18333.
[24] X. Qi, A. Panda, K. Lyu, X. Ma, S. Roy, A. Beirami, P. Mittal, P. Henderson, Safety alignment should be made more than just a few tokens deep, in: Proc. of ICLR, 2025.
[25] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, Q. Gu, Improving adversarial robustness requires revisiting misclassified examples, in: Proc. of ICLR, 2020.
[26] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. E. Ghaoui, M. Jordan, Theoretically principled trade-off between robustness and accuracy, in: Proc. of ICML, 2019, pp. 7472–7482.
[27] Y. Yang, C. Rashtchian, H. Zhang, R. Salakhutdinov, K. Chaudhuri, A closer look at accuracy vs. robustness, in: Proc. of NeurIPS, 2020.
[28] S. Ren, Y. Deng, K. He, W. Che, Generating natural language adversarial examples through probability weighted word saliency, in: Proc. of ACL, 2019, pp. 1085–1097.
[29] J. Li, S. Ji, T. Du, B. Li, T. Wang, Textbugger: Generating adversarial text against real-world applications, in: Proc. of NDSS, 2019.
[30] D. Jin, Z. Jin, J. T. Zhou, P. Szolovits, Is BERT really robust? A strong baseline for natural language attack on text classification and entailment, in: Proc. of AAAI, 2020, pp. 8018–8025.
[31] L. Li, R. Ma, Q. Guo, X. Xue, X. Qiu, BERT-ATTACK: adversarial attack against BERT using BERT, in: Proc. of EMNLP, 2020, pp. 6193–6202.
[32] A. Kurakin, I. J. Goodfellow, S. Bengio, Adversarial machine learning at scale, in: Proc. of ICLR, 2017.
[33] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. R. Bowman, GLUE: A multi-task benchmark and analysis platform for natural language understanding, in: Proc. of ICLR, 2019.
[34] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proc. of ACL, 2011, pp. 142–150.
[35] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Proc. of NeurIPS, 2015, pp. 649–657.
[36] Q. Liu, R. Zheng, B. Rong, J. Liu, Z. Liu, Z. Cheng, L. Qiao, T. Gui, Q. Zhang, X. Huang, Flooding-x: Improving bert's resistance to adversarial attacks via loss-restricted fine-tuning, in: Proc. of ACL, 2022, pp. 5634–5644.
[37] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, T. Zhao, SMART: robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization, in: Proc. of ACL, 2020, pp. 2177–2190.
[38] T. V. Nguyen, T. Le, Adapters mixup: Mixing parameter-efficient adapters to enhance the adversarial robustness of fine-tuned pre-trained text classifiers, in: Proc. of EMNLP, 2024, pp. 21183–21203.
[39] S. Garg, G. Ramakrishnan, BAE: bert-based adversarial examples for text classification, in: Proc. of EMNLP, 2020, pp. 6174–6181.
[40] B. Wang, C. Xu, S. Wang, Z. Gan, Y. Cheng, J. Gao, A. H. Awadallah, B. Li, Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models, in: Proc. of NeurIPS, 2021.
[41] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, Y. Qi, Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP, in: Proc. of EMNLP, 2020, pp. 119–126.
[42] OpenAI, GPT-4 technical report, CoRR abs/2303.08774, 2023. arXiv:2303.08774.
[43] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al., Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, CoRR abs/2501.12948, 2025. arXiv:2501.12948.
[44] T. A. Almeida, J. M. G. Hidalgo, A. Yamakami, Contributions to the study of SMS spam filtering: new collection and results, in: Proc. of DocEng, 2011, pp. 259–262.

[45] B. Zhu, Y. Rao, Exploring robust overfitting for pre-trained language models, in: Proc. of ACL Findings, 2023, pp. 5506–5522.
[46] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, in: Proc. of ICLR, 2023.
[47] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692, 2019. arXiv:1907.11692.
[48] C. Yu, B. Han, L. Shen, J. Yu, C. Gong, M. Gong, T. Liu, Understanding robust overfitting of adversarial training and beyond, in: Proc. of ICML, 2022, pp. 25595–25610.

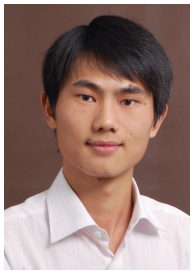
Author Biography

Bin Zhu is currently pursuing his Ph.D. degree at the School of Computer Science and Engineering, Sun Yat-sen University, China. His research interests include Natural Language Processing and Adversarial Robustness.

Siyu Liu received her Bachelor's degree in Computer Science from the School of Artificial Intelligence at Sun Yat-sen University in 2024. She is currently pursuing a Master's degree at the School of Computer Science, Sun Yat-sen University. Her research focuses on large language models.

Yuyin Lu is currently pursuing her Ph.D. degree at the School of Computer Science and Engineering, Sun Yat-sen University. Her research interests include knowledge graph reasoning, structured knowledge discovery, and uncertainty reasoning.

Jielin Song received the bachelor's degree in E-commerce from Beijing University of Posts and Telecommunications, Beijing, China, in 2023. He is currently working towards the master's degree with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. His research interests include large language models and natural language processing.



Yanghui Rao received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2014. He is an Associate Professor with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. He has published over 50 refereed journals and conference papers, including TKDE, TOIS, TNNLS, TCYB, TKDD, TACL, IJCAI, ACL, ECML, EMNLP, NAACL, and COLING. His current research interests include language model, topic discovery, and natural language processing.