Abstract

Recent studies have found that removing the norm-bounded projection and increasing search steps in adversarial training can significantly improve robustness. However, we observe that a too large number of search steps can hurt accuracy. We aim to obtain strong robustness efficiently using fewer steps. Through a toy experiment, we find that perturbing the clean data to the decision boundary but not crossing it does not degrade the test accuracy. Inspired by this, we propose friendly adversarial data augmentation (FADA) to generate “friendly” adversarial data. On top of FADA, we propose geometry-aware adversarial training (GAT) to perform adversarial training (e.g., FGM) on friendly adversarial data so that we can save a large number of search steps. Comprehensive experiments across two widely used datasets and three pre-trained language models demonstrate that GAT can obtain stronger robustness via less steps. In addition, we provide extensive empirical results and in-depth analyses on robustness to facilitate future studies.

1 Introduction

Deep neural networks (DNNs) outperform humans on many natural language processing (NLP) tasks (Kim, 2014; Vaswani et al., 2017; Devlin et al., 2019). However, recent studies have shown that DNNs are vulnerable to crafted adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2014). For instance, an attacker can mislead an online sentiment analysis system by making minor changes to the input sentences (Papernot et al., 2016; Liang et al., 2017). It has raised concerns among researchers about the security of DNN-based NLP systems. As a result, a growing number of studies are focusing on enhancing robustness to defend against textual adversarial attacks (Jia et al., 2019; Ye et al., 2020; Jones et al., 2020; Zhu et al., 2020).

Existing adversarial defense methods fall into two categories: empirical and certified defenses. Empirical defenses include gradient-based adversarial training (AT) and discrete adversarial data augmentation (ADA). Certified defenses provide a provable guaranteed robustness boundary for NLP models. This work focuses on empirical defenses.

There was a common belief that gradient-based AT methods in NLP was ineffective compared with ADA in defending against textual adversarial attacks (Li and Qiu, 2021; Si et al., 2021). Li et al. (2021) find that removing the norm-bounded projection and increasing the number of search steps in adversarial training can significantly improve robustness. Nonetheless, we observe that increasing the number of search steps further does not significantly improve robustness but hurts accuracy.

We give a possible explanation from a geometry-aware perspective. Removing the norm-bounded projection enlarge the search space. Appropriately increasing the number of search steps brings the adversarial data closer to the decision boundary. In
this case, the model learns a robust decision boundary. Further increasing the number of search steps can make the adversarial data cross the decision boundary too far, hindering the training of natural data and hurting natural accuracy.

To verify our hypothesis, we train a base model using adversarial data, which are generated by adversarial word substitution (AWS) on the SST-2 (Socher et al., 2013) dataset. We report its training accuracy (“ada training acc”) on adversarial data and test accuracy (“ada test acc”) on the clean test set in Figure 1. Although achieving nearly 100% training accuracy, its test accuracy is only about 15%, which implicates the adversarial data make the test performance degraded. Then we train another base model, whose training data is more “friendly”. We just recover their last modified words to return to the correct class, namely friendly adversarial data augmentation (FADA). It means that only one word is different in each sentence. Surprisingly, it achieves a high test accuracy of $\sim$93%.

This preliminary inspired us to address two existing problems:

- **The number of search steps is always large, which is computationally inefficient.**

- **A too large number of steps leads to degraded test performance.**

Geometrically speaking, the friendly adversarial data are close to the ideal decision boundary. We can address the above two issues in one fell swoop if we perform gradient-based adversarial training on these friendly adversarial data. It is like we start one step before the end, allowing us to obtain strong robustness through a tiny number of search steps. We name it geometry-aware adversarial training (GAT). Figure 2 illustrates our proposed GAT.

In addition, the friendly adversarial data only need to be generated once per dataset. It can be reused, so it is computationally efficient. It can also be updated for every iteration or epoch but computationally expensive.

Our contributions are summarized as follows:

1) We propose FADA to generate friendly adversarial data which are close to the decision boundary (but not crossing it).

2) We propose GAT, a geometry-aware adversarial training method that adds FADA to the training set and performs gradient-based adversarial training.

3) GAT is computationally efficient, and it outperforms state-of-the-art baselines even if using the simplest FGM. We further provide extensive ablation studies and in-depth analyses on GAT, contributing to a better understanding of robustness.

### 2 Related Work

#### 2.1 Standard Adversarial Training

Let $f_\theta(x)$ be our neural network, $\mathcal{L}(f_\theta(x), y)$ be the loss function (e.g., cross entropy), where $x \in X$ is the input data and $y \in Y$ is the true label. The learning objective of standard adversarial training is

$$
\min_\theta \mathbb{E}_{(X,Y) \sim D} \max_{\|\delta\| \leq \epsilon} \mathcal{L}(f_\theta(X + \delta), y),
$$

where $D$ is the data distribution, $\delta$ is the minor perturbation, $\epsilon$ is the allowed perturbation size. To optimize the intractable min-max problem, we search for the optimal $\delta$ to maximize the inner loss and then minimize the outer loss w.r.t the parameters $\theta$, step by step.

The gradient $g$ of the inner loss w.r.t the input $x$ is used to find the optimal perturbation $\delta$. Goodfellow et al. (2014) proposed fast gradient sign method (FGSM) to obtain $\delta$ by one step:

$$
\delta = \epsilon \cdot \text{sgn}(g),
$$

where $\text{sgn}(\cdot)$ is the signum function. Madry et al. (2018) proposed projected gradient descent (PGD)
to solve the inner maximization as follows:
\[
\delta^{(t+1)} = \Pi \alpha \cdot g^{(t)} / \|g^{(t)}\|, \forall t \geq 0, \tag{3}
\]
where \(\alpha > 0\) is the step size (i.e., adversarial learning rate), \(\Pi\) is the projection function that projects the perturbation onto the \(\epsilon\)-norm ball. Conventionally, PGD stops after a predefined number of search steps \(K\), namely PGD-\(K\). In addition, TRADES (Zhang et al., 2019), MART (Wang et al., 2020) and FAT (Zhang et al., 2020) are also effective adversarial training methods for boosting model robustness.

Regarding FAT, the authors propose to stop adversarial training in a predefined number of steps after crossing the decision boundary, which is a little different from our definition of “friendly”.

### 2.2 Adversarial Training in NLP

Gradient-based adversarial training has significantly improved model robustness in vision, while researchers find it helps generalization in NLP. Miyato et al. (2016) find that adversarial and virtual adversarial training have good regularization performance. Sato et al. (2018) propose an interpretable adversarial training method that generates reasonable adversarial texts in the embedding space and enhance models’ performance. Zhu et al. (2020) develop FreeLB to improve natural language understanding.

There is also a lot of work focused on robustness. Wang et al. (2021) improve model robustness from an information theoretic perspective. Dong et al. (2021) use a convex hull to capture and defense against adversarial word substitutions. Zhou et al. (2021) train robust models by augmenting training data using Dirichlet Neighborhood Ensemble (DNE).

Besides, adversarial data augmentation is another effective approach to improve robustness (Ebrahimi et al., 2017; Li et al., 2018; Ren et al., 2019; Jin et al., 2019; Zang et al., 2020; Li et al., 2020; Garg and Ramakrishnan, 2020; Si et al., 2021). However, it only works when the augmentation happens to be generated by the same attacking method and often hurts accuracy.

It is worth noting that recent empirical results have shown that previous gradient-based adversarial training methods have little effect on defending against textual adversarial attacks (Li et al., 2021; Si et al., 2021). The authors benchmark existing defense methods and conclude that gradient-based

#### Algorithm I: Friendly Adversarial Data Augmentation (FADA)

**Input:** The original text \(x\), ground truth label \(y_{\text{true}}\), base model \(f_{\theta}\), adversarial word substitution function \(AWS(\cdot)\)

**Output:** The friendly adversarial example \(x_f\)

1. **Initialization:**
   2. \(x_f \leftarrow x\)
   3. the last modified word \(w^* \leftarrow \text{None}\)
   4. the last modified index \(i^* \leftarrow 0\)
   5. \(x_{adv}, w^*, i^* = AWS(x, y_{true}, f_{\theta})\)
   6. if \(w^* = \text{None}\) then
      7. **return** \(x_f\)
   8. **end if**
   9. Replace \(w_{i^*}\) in \(x_{adv}\) with \(w^*\)
   10. \(x_f \leftarrow x_{adv}\)
   11. **return** \(x_f\)

AT can achieve the strongest robustness by removing the norm bounded projection and increasing the search steps.

### 3 Methodology

#### 3.1 Friendly Adversarial Data Augmentation

For a sentence \(x \in X\) with a length of \(n\), it can be denoted as \(x = w_1w_2...w_{i-1}w_{i}w_{i+1}...w_n\), where \(w_i\) is the \(i\)-th word in \(x\). Its adversarial counterpart \(x_{adv}\) can be denoted as \(w'_1w'_2...w'_iw_{i+1}'...w'_{n-1}'w'_n\). In this work, \(x_{adv}\) is generated by adversarial word substitution, so \(x_{adv}\) has the same length with \(x\). Conventional adversarial data augmentation generates adversarial data fooling the victim model and mixes them with the original training set. As we claim in section 1, these adversarial data can hurt test performance. An interesting and critical question is when it becomes detrimental to test accuracy.

One straightforward idea is to recover all the \(x_{adv}\) to \(x\) word by word and evaluate their impact on test accuracy. We train models only with these adversarial data and test models with the original test set. We are excited that the test accuracy immediately returns to the normal level when we recover the last modified word. We denote these data with only one word recovered as \(x_f\). Geometrically, the only difference between \(x_{adv}\) and \(x_f\) is whether they have crossed the decision boundary.

To conclude, when the adversarial data cross the decision boundary, they become incredibly harmful to the test performance. We name all the \(x_f\)
Algorithm 2 Ideal Geometry-aware Adversarial Training (GAT)

Input: Our base network $f_0$, cross entropy loss $\mathcal{L}_{CE}$, training set $D = \{(x_i, y_i)\}_{i=1}^n$, number of epochs $T$, batch size $m$, number of batches $M$

Output: robust network $f_0$

1: for epoch = 1 to $T$ do
2:   for batch = 1 to $M$ do
3:     Sample a mini-batch $b = \{(x_i, y_i)\}_{i=1}^m$ in $b$
4:     for all $x_i$, in $b$ do
5:       Generate friendly adversarial example $x_i'$ via Algorithm 1
6:     end for
7:     Apply an adversarial training method (e.g., FreeLB++) on both $x_i$ and $x_i'$ to obtain their adversarial counterpart $\bar{x}_i$ and $\bar{x}_i'$
8:   end for
9:   Update $f_0$ via $\nabla_{\theta} \mathcal{L}_{CE}(f_0(\bar{x}_i), y_i)$ and $\nabla_{\theta} \mathcal{L}_{CE}(f_0(\bar{x}_i'), y_i)$
10: end for

end for

as friendly adversarial examples (FAEs) because they improve model robustness without hurting accuracy. Similarly, we name the generation of FAEs as friendly adversarial data augmentation (FADA).

We show our proposed FADA in Algorithm 1.

3.2 Geometry-aware Adversarial Training

3.2.1 Seeking for the optimal $\delta$

Recall the inner maximization issue of the learning objective in Eq. (1). Take PGD-$K$ for instance. It divides the search for the optimal perturbation $\delta$ into $K$ search steps, and each step requires a backpropagation (BP), which is computationally expensive.

We notice that random initialization of $\delta^0$ is widely used in adversarial training, where $\delta^0$ is always confined to a $\epsilon$-ball centered at $x$. However, we initialize the clean data via discrete adversarial word substitution in NLP. It is similar to data augmentation (DA), with the difference that we perturb clean data in the direction towards the decision boundary, whereas the direction of data augmentation is random.

By doing so, we decompose the $\delta$ into two parts, which can be obtained by word substitution and gradient-based adversarial training, respectively. We denote them as $\delta_t$ and $\delta_s$. Therefore, the inner maximization can be reformulated as

$$\max_{\|\delta_t + \delta_s\| \leq \epsilon} \mathcal{L}(f_0(X + \delta_t + \delta_s), y).$$ (4)

We aim to find the maximum $\delta_t$ that helps improve robustness without hurting accuracy. As we claim in Section 3.1, FADA generates friendly adversarial data which are close to the decision boundary. Furthermore, the model trained with these friendly adversarial data keeps the same test accuracy as the original training set (Figure 1). Therefore we find the maximum $\delta_t$ which is harmless to the test accuracy through FADA.

Denote $X_f$ as the friendly adversarial data generated by FADA, Eq. (4) can be reformulated as

$$\max_{\|\delta_t\| \leq \epsilon} \mathcal{L}(f_0(X_f + \delta_t), y).$$ (5)

The tiny $\delta_s$ can be obtained by some gradient-based adversarial training methods (e.g., FreeLB++ (Li et al., 2021)) in few search steps. As a result, a large number of search steps are saved to accelerate adversarial training. We show our proposed geometry-aware adversarial training in Algorithm 2.

3.2.2 Final Learning Objective

It is computationally expensive to update friendly adversarial data for every mini-batch. In practice, we generate static augmentation $(X_f, Y)$ for the training dataset $(X, Y)$ and find it works well with GAT. The static augmentation $(X_f, Y)$ is reusable. Therefore, GAT is computationally efficient.

Through such a tradeoff, our final objective function can be formulated as

$$\mathcal{L} = \mathcal{L}_{CE}(X, Y, \theta) + \mathcal{L}_{CE}(\bar{X}, Y, \theta) + \mathcal{L}_{CE}(X_f, Y, \theta),$$ (6)

where $\mathcal{L}_{CE}$ is the cross entropy loss. $\bar{X}$ and $X_f$ are generated from $X$ and $X_f$ using gradient-based adversarial training methods, respectively.

4 Experiments

4.1 Datasets

We conduct experiments on the SST-2 (Socher et al., 2013) and IMDb (Maas et al., 2011) datasets which are widely used for textual adversarial learning. Statistical details are shown in Table 1. We use the GLUE (Wang et al., 2019) version of the SST-2 dataset whose test labels are unavailable. So we report its accuracy on the develop set in our experiments.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># train</th>
<th># dev / test</th>
<th>avg. length</th>
</tr>
</thead>
<tbody>
<tr>
<td>SST-2</td>
<td>67349</td>
<td>872</td>
<td>17</td>
</tr>
<tr>
<td>IMDb</td>
<td>25000</td>
<td>25000</td>
<td>201</td>
</tr>
</tbody>
</table>

Table 1: Summary of the two datasets.
### Table 2: Main defense results on the SST-2 dataset, including the test accuracy on the clean test set (Clean %), the robust accuracy under adversarial attacks (RA %), the attack success rate (ASR %), and the average number of queries requiring by the attacker (# Query).

<table>
<thead>
<tr>
<th>Model</th>
<th>Clean %</th>
<th>TextFooler</th>
<th>TextBugger</th>
<th>BAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT&lt;sub&gt;base&lt;/sub&gt;</td>
<td>92.4</td>
<td>32.8</td>
<td>64.1</td>
<td>72.8</td>
</tr>
<tr>
<td>ADA</td>
<td>92.2</td>
<td>46.7</td>
<td>48.7</td>
<td>79.4</td>
</tr>
<tr>
<td>ASCC</td>
<td>87.2</td>
<td>32.0</td>
<td>63.3</td>
<td>71.6</td>
</tr>
<tr>
<td>DNE</td>
<td>86.6</td>
<td>26.5</td>
<td>69.6</td>
<td>69.0</td>
</tr>
<tr>
<td>InfoBERT</td>
<td>92.2</td>
<td>41.7</td>
<td>54.8</td>
<td>74.9</td>
</tr>
<tr>
<td>TA V AT</td>
<td>92.2</td>
<td>40.4</td>
<td>56.3</td>
<td>74.3</td>
</tr>
<tr>
<td>FreeLB</td>
<td>93.1</td>
<td>42.7</td>
<td>53.7</td>
<td>75.9</td>
</tr>
<tr>
<td>FreeLB++10</td>
<td>93.3</td>
<td>41.9</td>
<td>54.8</td>
<td>75.8</td>
</tr>
<tr>
<td>FreeLB++30</td>
<td>93.4</td>
<td>45.6</td>
<td>50.6</td>
<td>78.1</td>
</tr>
<tr>
<td>FreeLB++50</td>
<td>92.0</td>
<td>45.5</td>
<td>50.4</td>
<td>77.2</td>
</tr>
<tr>
<td>GAT&lt;sub&gt;FGM&lt;/sub&gt; (ours)</td>
<td>92.8</td>
<td>45.8</td>
<td>49.8</td>
<td>78.5</td>
</tr>
<tr>
<td>GAT&lt;sub&gt;FreeLB++10&lt;/sub&gt; (ours)</td>
<td>93.2</td>
<td>49.5</td>
<td>46.3</td>
<td>80.6</td>
</tr>
<tr>
<td>GAT&lt;sub&gt;FreeLB++30&lt;/sub&gt; (ours)</td>
<td>92.7</td>
<td>52.5</td>
<td>42.2</td>
<td>82.3</td>
</tr>
</tbody>
</table>

#### 4.2 Attacking Methods

Follow Li et al. (2021), we adopt TextFooler (Jin et al., 2019), TextBugger (Li et al., 2018) and BAE (Garg and Ramakrishnan, 2020) as attackers. TextFooler and BAE are word-level attacks and TextBugger is a multi-level attacking method. We also impose restrictions on these attacks for a fair comparison, including:

1. The maximum percentage of perturbed words \( p_{max} \)
2. The minimum semantic similarity \( \varepsilon_{min} \) between the original input and the generated adversarial example
3. The maximum size \( K_{syn} \) of one word’s synonym set

Since the average sentence length of IMDb and SST-2 are different, \( p_{max} \) is set to 0.1 and 0.15, respectively; \( \varepsilon_{min} \) is set to 0.84; and \( K_{syn} \) is set to 50. All settings are referenced from previous work.

#### 4.3 Adversarial Training Baselines

We use BERT<sub>base</sub> (Devlin et al., 2019) as the base model to evaluate the impact of the following variants of adversarial training on accuracy and robustness and provide a comprehensive comparison with our proposed GAT.

- Adversarial Data Augmentation
- ASCC (Dong et al., 2021)
- DNE (Zhou et al., 2021)
- InfoBERT (Wang et al., 2021)
- TAVAT (Li and Qiu, 2021)
- FreeLB (Zhu et al., 2020)
- FreeLB++ (Li et al., 2021)

ASCC and DNE adopt a convex hull during training. InfoBERT improves robustness using mutual information. TAVAT establishes a token-aware robust training framework. FreeLB++ removes the norm bounded projection and increases search steps.

We only compare GAT with adversarial training-based defense methods and leave comparisons with other defense methods (e.g., certified defenses) for future work.

#### 4.4 Implementation Details

We implement ASCC, DNE, InfoBERT, and TAVAT models based on TextDefender (Li et al., 2021). We implement FGM, FreeLB, FreeLB++, and our GAT based on HuggingFace Transformers<sup>1</sup>. We implement ADA and FADA based on TextAttack<sup>2</sup>. All the adversarial hyper-parameters settings are following their original papers. All the models are trained on two GeForce RTX 2080 GPUs and eight Tesla T4 GPUs.

Regarding the training settings and hyper-parameters, the optimizer is AdamW (Lovschilov and Hutter, 2019); the learning rate is \( 2e^{-5} \); the number of epochs is 10; the batch size is 64 for SST-2 and 24 for IMDb.

<sup>1</sup>https://huggingface.co/transformers

<sup>2</sup>https://github.com/QData/TextAttack
4.5 Main Results

Our proposed GAT can easily combine with other adversarial training methods. In our experiments, we combine GAT with FGM (GAT\textsubscript{FGM}) and FreeLB++ (GAT\textsubscript{FreeLB++}), respectively. We aim to evaluate if GAT can bring improvements to the simplest (FGM) and the most effective (FreeLB++) AT methods.

We summarize the main defense results on the SST-2 dataset in Table 2. When GAT works with the simplest adversarial training method, FGM, the resulting robustness improvement exceeds FreeLB++50. The effectiveness and efficiency of GAT allow us to obtain strong robustness while saving many search steps. Further combining FreeLB++ on GAT can obtain stronger robustness and outperform all other methods.

Regarding the accuracy, FreeLB++30 obtains the highest 93.4%. GAT also significantly improves accuracy.

In addition, ADA is effective in improving robustness but hurts accuracy. It is not surprising that ASCC and DNE suffer from significant performance losses. However, there is no improvement in robustness and even weaker robustness under TextFooler and TextBugger attacks than the other methods.

Table 3 shows the defense results on the IMDb dataset. The defense performances are generally consistent with that on the SST-2 dataset. It is worth noting that GAT\textsubscript{FGM} achieved an extremely high RA % with a medium #Query, which needs further exploration.

5 Discussions

We further explore other factors that affect robustness and provide comprehensive empirical results.

5.1 Ablation Studies

We conduct ablation studies on the SST-2 dataset to assess the impact of each component of GAT.

As shown in Table 4, “FADA” consistently outperforms “ADA” and “None” with different adversarial training methods. Furthermore, “FADA&FGM” achieve a higher RA % than “None&FreeLB++30”, which implies that “FADA” can obtain strong robustness in one adversarial search step. “ADA” also helps improve robustness. However, as the number of search steps increases, so does the hurt it does to Clean %. On the contrary, “FADA” does not harm Clean % but improves it, implying its friendliness.
We have shown that GAT brings significant improvements against TextFooler and TextBugger attacks. We investigate whether GAT is effective under combinatorial optimization attacks, such as PSO (Zang et al., 2020) and FastGA (Jia et al., 2019).

We can see from Table 5 that GAT\textsubscript{FreeLB++30} obtains the highest RA % against the two attacks and GAT\textsubscript{FreeLB++10} has the highest clean accuracy. The results demonstrate that our proposed GAT consistently outperforms other defenses against combinatorial optimization attacks.

5.3 Results with More Steps

As we claim in Section 1, the accuracy should degrade with a large number of search steps. But what happens for robustness?

We aim to see if RA % can be further improved. Figure 3(a) shows that the RA % gradually increases against TextFooler and TextBugger attacks. However, RA % decreases against BAE with steps more than 30, which needs more investigation. As the steps increase, the growth rate of RA % decreases, and the Clean % decreases. We conclude that a reasonable number of steps will be good for both RA % and Clean %. It is unnecessary to search for too many steps since robustness grows very slowly in the late adversarial training period while accuracy drops.

5.4 Impact of Step Size

A large step size (i.e., adversarial learning rate) will cause performance degradation for conventional adversarial training. Nevertheless, what impact does it have on robustness? We explore the impact of different step sizes on robustness and accuracy. As shown in Figure 3(b), the clean test accuracy slightly drops as the step size increases. The robust accuracy under TextFooler attack increases, while the robust accuracy under Textbugger and BAE attacks decrease. Overall, the impact of step size on robustness needs further study.

5.5 Impact of Training Epochs

Ishida et al. (2020) have shown that preventing further reduction of the training loss when reaching a small value and keeping training can help generalization. In adversarial training, it is naturally hard to achieve zero training loss due to the insufficient capacity of the model (Zhang et al., 2021). Therefore, we investigate whether more training iterations result in stronger robustness in adversarial training. We report the RA % achieved by GAT\textsubscript{FreeLB++30} at each epoch in Figure 3(c). We observe that the RA % tends to improve slowly, implying that more training iterations result in stronger model robustness using GAT.
5.6 Results with Other Models

We show that GAT can work on more advanced models. We choose RoBERTa_{base} (Liu et al., 2019) and DeBERTa_{base} (He et al., 2021), two improved versions of BERT, as the base models. As shown in Table 6 and Table 7, GAT slightly improve robustness of RoBERTa and DeBERTa models. We can also conclude that DeBERTa is significantly more robust than RoBERTa.

5.7 Limitations

We discuss the limitations of this work as follows.

• As we clarify in Section 3.2.2, instead of dynamically generating friendly adversarial data in training, we choose to pre-generate static augmentation. We do this for efficiency, as dynamically generating discrete sentences in training is computationally expensive. Although it still significantly improves robustness in our experiments, such a tradeoff may lead to failure because the decision boundary changes continuously during training.

• GAT performs adversarial training on friendly adversarial data. It may help if we consider the decision boundaries when performing gradient-based adversarial training—for example, stopping early when the adversarial data crosses the decision boundary. We consider this as one of the directions for future work.

6 Conclusion

In this paper, we study how to improve robustness from a geometry-aware perspective. We first propose FADA to generate friendly adversarial data that are close to the decision boundary. Then we combine gradient-based adversarial training methods on FADA to save a large number of search steps, termed geometry-aware adversarial training (GAT). GAT can efficiently achieve state-of-the-art defense performance without hurting test accuracy.

We conduct extensive experiments to give in-depth analysis, and we hope this work can provide helpful insights on robustness in NLP.

References


Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.


