Flooding-X: Improving BERT's Resistance to Adversarial Attacks via Loss-Restricted Fine-Tuning

Anonymous ACL submission

Abstract

Adversarial robustness has attracted much attention recently, and the mainstream solution is adversarial training. However, the tradition of generating adversarial perturbations for each input embedding (in the settings of NLP) scales up the training computational complexity by the number of gradient steps it takes to obtain the adversarial samples. To address this problem, we leverage Flooding method which primarily aims at better generalization and we find promising in defending adversarial attacks. We further propose an effective criterion to bring hyper-parameter-dependent flooding into effect with a narrowed-down search space by measuring how the gradient steps taken within one epoch affect the loss of each batch. Our approach requires zero adversarial sample for training, and its time consumption is equivalent to fine-tuning, which can be 2-15 times faster than standard adversarial training. We experimentally show that our method improves Bert's resistance to textual adversarial attacks by a large margin, and achieves state-of-the-art robust accuracy on various text classification and GLUE tasks.

1 Introduction

Despite their impressive performances on various NLP tasks, deep neural networks such as BERT (Devlin et al., 2019) suffer a sharp decline facing deliberately constructed adversarial attacks (Zeng et al., 2021; Nie et al., 2020; Zang et al., 2020; Ren et al., 2019; Zhang et al., 2019). A line of works attempt to alleviate this problem by creating adversarially robust models via defense methods, including adversarial data augmentation (Chen et al., 2021; Si et al., 2021), regularizing (Wang et al., 2020a), and adversarial training (Wang et al., 2020b; Zhu et al., 2019; Madry et al., 2018). Data augmentation and adversarial training rely on extra adversarial examples generated either by hand-crafting or conducting gradient ascent on the clean data for virtual adversarial samples.

However, generating adversarial examples scales up the cost of training computationally, which makes vanilla adversarial training almost impractical on large-scale NLP tasks like QNLI (Question-answering NLI). Increasing researches express their concern of the time-consuming property of standard adversarial training and offer cheaper but competitive alternatives by (i) replacing the perturbation generation with an extra generator network (Baluja and Fischer, 2017; Xiao et al., 2018), or by (ii) combining the gradient computation of clean data and perturbations into one backward pass (Shafahi et al., 2019). These approaches still rely on extra adversarial examples generated either by the model itself or by an extra module.

In this work, we propose a novel method, Flooding-X, to largely improve adversarial robustness without any adversarial examples, maintaining the same computational cost as conventional BERT fine-tuning. The vanilla Flooding (Ishida et al., 2020) method is a practical regularization technique to boost model generalization by preventing further reduction of the training loss when it reaches a *reasonably small value*. It results in a model performing normal gradient descent when training loss is above the decided value but gradient ascent when below. By continuing to "random walk" with the same non-zero value as a "virtual loss", the model drifts into an area with a flat loss landscape that is claimed to lead to better generalization (Ishida et al., 2020). Interestingly, we find that Flooding method is also promising in increasing models' resistance to adversarial attacks. Despite the significant rise in robust accuracy, the so-called reasonably small value, which is a hyper-parameter, takes effort to be found and varies for each dataset, which requires an overly extensive search among the numerous candidates.

In an attempt to narrow down the candidates of hyper-parameter, we propose *gradient accordance* as an informative criterion for optimal values that bring Flooding into effect, which is used as a building-block in Flooding-X. We measure how *accordant* the gradients of the batches are by analyzing how the gradient descent steps based on part of an epoch affect the loss of each batch. Gradient accordance is computationally friendly and is tractable during training process. Experiments on various tasks show a close relation between gradient accordance and overfitting. As a result, we propose gradient accordance as a reliable flooding criterion to make the training loss flood around the level when the model has nearly overfitted. That is to say, we leverage the training loss of the model right before overfitting as the value of flood level.

Flooding-X is especially useful and shows great advantage over adversarial training in terms of computational cost when the training dataset is relatively large. Experimental results demonstrate that our method achieves stated-of-the-art robust accuracy with BERT on various tasks and improves its robust accuracy by 100 to 400% without using any adversarial example, consuming any extra training time, or conducting overly extensive search for hyper-parameter. Our main contributions are as follows.

1) We propose a novel method, Flooding-X, that achieves state-of-the-art robust accuracy for BERT on various tasks, which is adversarial-example-free and takes no more training time than fine-tuning.

2) We propose a promising indicator, i.e. gradient accordance, to alleviate Flooding method from tedious search of the hyper-parameter.

3) We conduct comprehensive experiments on NLP tasks to illustrate the potential of Flooding for improving BERT's adversarial robustness.

2 Why Does Flooding Boost Adversarial Robustness?

2.1 Vanilla Flooding

We first describe the vanilla Flooding regularization method (Ishida et al., 2020) for alleviating overfitting via keeping training loss from reducing to zero. Under the main assumption that learning until zero loss is harmful, Ishida et al. (2020) propose Flooding to intentionally prevent further reduction of the training loss when it reaches a reasonably small value, which is called the *flood level*. Intuitively, this approach makes the training loss float around the pre-defined flood level and alter from normal mini-batch gradient descent to gradient ascent if the loss is below the flood level.

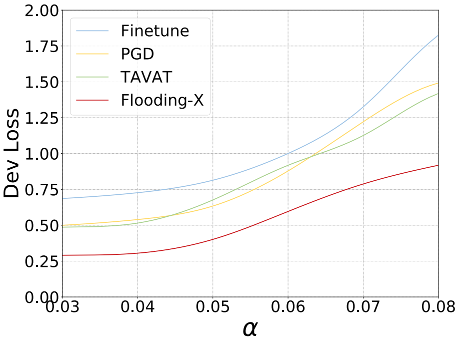

Figure 1: Input loss landscape of vanilla BERT and different adversarial training algorithms under Gaussian random noise of standard deviation α on SST-2 dataset.

With the constraint of flood level, the model will continue to "random walk" around the non-zero training loss, which is expected to reach a flat loss landscape.

The algorithm of Flooding is defined as follow:

$$\widetilde{J}(\boldsymbol{\theta}) = |J(\boldsymbol{\theta}) - b| + b, \tag{1}$$

where J denotes the original learning objective, and $\widetilde{J}$ represents the modified learning objective with flooding. The positive value b is the flood level specified by user, and $\boldsymbol{\theta}$ is the model parameter. Accordingly, the flooded empirical risk is then defined as

$$\widetilde{R}(\boldsymbol{f}) = |\widehat{R}(\boldsymbol{f}) - b| + b, \tag{2}$$

within which $\widehat{R}(\boldsymbol{f})$ / $\widetilde{R}(\boldsymbol{f})$ denotes the original / flooded empirical risk respectively, and $\boldsymbol{f}$ refers to the score function to be learned by the model. During the back propagation process, the gradient of $\widehat{R}(\boldsymbol{f})$ w.r.t. model parameters and $\widetilde{R}(\boldsymbol{f})$ point to the same direction when $\widetilde{R}(\boldsymbol{f})$ is above b but to the opposite direction when it is below b. As a result, model performs normal gradient descent when the learning objective is above the flood level, and gradient ascent when below.

2.2 Smooth Parameter Landscape Leads to Better Robustness

According to the definition described in the previous section, Flooding does not make any difference to the training process when the loss is beyond the flood level. When the training loss approaches the flood level, on closer inspection, gradient descent and gradient ascent begin to alternate. Assume that the model with learning rate ε performs gradient descent for the n-th batch and then gradient ascent for batch $n+1$, which results in:

$$\begin{aligned}\boldsymbol{\theta}_n &= \boldsymbol{\theta}_{n-1} - \varepsilon \boldsymbol{g}(\boldsymbol{\theta}_{n-1}),\\ \boldsymbol{\theta}_{n+1} &= \boldsymbol{\theta}_n + \varepsilon \boldsymbol{g}(\boldsymbol{\theta}_n).\end{aligned} \tag{3}$$

In the equations above, $\boldsymbol{g}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ is the gradient of $J(\boldsymbol{\theta})$ w.r.t. model parameters. We can then get

$$\begin{aligned}\boldsymbol{\theta}_{n+1} = &\boldsymbol{\theta}_{n-1} - \varepsilon \boldsymbol{g}(\boldsymbol{\theta}_{n-1}) + \varepsilon \boldsymbol{g}\big(\boldsymbol{\theta}_{n-1}\\ &- \varepsilon \boldsymbol{g}(\boldsymbol{\theta}_{n-1})\big),\end{aligned} \tag{4}$$

which is, by Taylor expansion, approximately equivalent to

$$\begin{aligned}&\approx \boldsymbol{\theta}_{n-1} - \varepsilon \boldsymbol{g}(\boldsymbol{\theta}_{n-1}) + \varepsilon\big(\boldsymbol{g}(\boldsymbol{\theta}_{n-1})\\ &\quad - \varepsilon \nabla_{\boldsymbol{\theta}} \boldsymbol{g}(\boldsymbol{\theta}_{n-1}) \boldsymbol{g}(\boldsymbol{\theta}_{n-1})\big)\\ &= \boldsymbol{\theta}_{n-1} - \frac{\varepsilon^2}{2} \nabla_{\boldsymbol{\theta}} \|\boldsymbol{g}(\boldsymbol{\theta}_{n-1})\|^2.\end{aligned} \tag{5}$$

Thus, theoretically, when the training loss is relatively low, the model alters into a new learning mode where the learning rate is $\varepsilon^2/2$ and the objective is to minimize $\|\boldsymbol{g}(\boldsymbol{\theta})\|^2$. Generally, the flooded model is guided into an area with a smooth parameter landscape that leads to better adversarial robustness (Prabhu et al., 2019; Yu et al., 2018; Li et al., 2018a). As is demonstrated in Figure 1, adversarial training brings about a smoother loss change to the model when the input embedding is perturbed by Gaussian random noise.

2.3 Achilles' Heel of Flooding

Despite its potential in boosting model's resistance to adversarial attacks, the optimal flood level has to be searched by performing exhaustive search within a wide range at tiny steps, which is not easily at hand. A relatively large value of flood level lengthens the gradient steps and keeps the model from convergence, while a tiny value causes hardly any difference to the training process. The effect of Flooding deeply relies on the flood level, which, at the same time, is also sensitive to the subtle change of this hyper-parameter. Figure 2 reveals that even a slight change on the value of flood level can make a huge difference on the adversarial robustness of the so-trained model. In an attempt to ease the effort of searching and make the best of Flooding, we propose a promising and reliable criterion to narrow down the search space, which is described in detail in the next section.

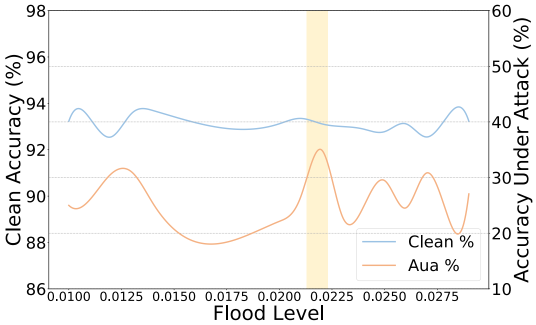

Figure 2: Influence of different flood levels on performance of the trained BERT on SST-2. The range marked in yellow is lined out by our proposed criterion , i.e., gradient accordance. The optimal value of flood level is guaranteed within the narrowed-down space.

3 Gradient Accordance as a Criterion for Flooding

Since Flooding is proposed as an attempt to avoid overfitting, we intuitively suppose that the optimal flood level would be found at the stage when the model is about to overfit. That is, we leverage the training loss before overfitting as the flood level. Inspired by influence function (Koh and Liang, 2017), we propose *gradient accordance* as a criterion for flooding, which is empirically proved to be reliable and indicative. We consider the effect of the model updated w.r.t. one epoch on each of its batches as a signal of overfitting. As is indicated by its name, this criterion measures the relation among the gradients of each batch on epoch level, evaluating whether the model updated on an epoch has the same positive effect on the batches on average. Now we provide the formal definition of gradient accordance.

3.1 Preliminaries

We denote a model as a functional approximation f which is parameterized by $\boldsymbol{\theta}$. Consider a training data point x with the ground truth label y, which results in a loss $\mathcal{L}(f(\boldsymbol{\theta}, x), y)$. The gradient of the loss w.r.t. the parameters is thus

$$\boldsymbol{g} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(f(\boldsymbol{\theta}, x), y), \tag{6}$$

whose negation denotes the direction in which the parameters $\boldsymbol{\theta}$ are updated to better correspond to the desired outputs on the training data (Fort et al., 2019). Now let's consider two data points x_1 and x_2 with their corresponding labels y_1 and y_2. According to the definition above, the gradient of sample 1 is $\boldsymbol{g_1} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(f(\boldsymbol{\theta}, x_1), y_1)$. We try to inspect how the small change of $\boldsymbol{\theta}$ in the direction $-\boldsymbol{g_1}$ influences the loss on sample x_1 or x_2:

$$\begin{aligned}\Delta\mathcal{L}_1 =& \mathcal{L}(f(\boldsymbol{\theta} - \varepsilon\boldsymbol{g_1}, x_1), y_1) \\ &- \mathcal{L}(f(\boldsymbol{\theta}, x_1), y_1),\end{aligned} \tag{7}$$

where $f(\boldsymbol{\theta}, x_1)$ can be expanded by Taylor expansion to be:

$$f(\boldsymbol{\theta}, x_1) = f(\boldsymbol{\theta} - \varepsilon\boldsymbol{g_1}, x_1) + \varepsilon\boldsymbol{g_1}\frac{\partial f}{\partial \boldsymbol{\theta}} + \mathcal{O}(\varepsilon^2). \tag{8}$$

Here, we refer to $(\varepsilon\boldsymbol{g_1}\frac{\partial f}{\partial \boldsymbol{\theta}} + \mathcal{O}(\varepsilon^2))$ as $T(x_1)$; and by repeating the similar expansion we can get

$$\begin{aligned}&\mathcal{L}(f(\boldsymbol{\theta}, x_1), y_1) \\ &\quad = \mathcal{L}(f(\boldsymbol{\theta} - \varepsilon\boldsymbol{g_1}, x_1) + T(x_1), y_1) \\ &\quad = \mathcal{L}(f(\boldsymbol{\theta} - \varepsilon\boldsymbol{g_1}, x_1), y_1) \\ &\qquad + \frac{\partial \mathcal{L}}{\partial f}T(x_1) + \mathcal{O}(T^2(x_1)).\end{aligned} \tag{9}$$

Equation (7) is thus equal to

$$\begin{aligned}\Delta\mathcal{L}_1 &= -\frac{\partial \mathcal{L}}{\partial f}T(x_1) - \mathcal{O}(T^2(x_1)) \\ &= -\frac{\partial \mathcal{L}}{\partial f}(\varepsilon\boldsymbol{g_1}\frac{\partial f}{\partial \boldsymbol{\theta}} + \mathcal{O}(\varepsilon^2)) \\ &= -\varepsilon\boldsymbol{g_1} \cdot \boldsymbol{g_1} - \mathcal{O}(\varepsilon^2).\end{aligned} \tag{10}$$

Similarly, the change of the loss on x_2 caused by the gradient update by x_1 is $\Delta\mathcal{L}_2 = -\varepsilon\boldsymbol{g_1} \cdot \boldsymbol{g_2} - \mathcal{O}(\varepsilon^2)$. Notably, $\Delta\mathcal{L}_1$ is negative by definition since the model is updated with respect to x_1 and naturally leads to a decrease on its loss. The model updated on x_1 is considered to have a positive effect on x_2 if $\Delta\mathcal{L}_2$ is also negative while an opposite effect if positive. The equations above demonstrate that this co-relation is equivalent to the overlap between the gradients of the two data points $\boldsymbol{g_1} \cdot \boldsymbol{g_2}$, which we hereafter refer to as *gradient accordance*.

3.2 Coarse-Grained Gradient Accordance

Data-point-level gradient accordance is too fine-grained to be tractable in practice. Thus, we attempt to scale it up and result in coarse-grain gradient accordance at batch level, which is computationally tractable and still reliable as a criterion for overfitting.

Consider a training batch B_0 with n samples $\boldsymbol{X} = \{x_1, x_2, \ldots, x_n\}$ and labels $\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\}$ of k classes $\{c_1, c_2, \ldots, c_k\}$. These samples can be divided into k groups according to their labels $\boldsymbol{X} = \boldsymbol{X_1} \cup \boldsymbol{X_2} \cup \cdots \cup \boldsymbol{X_k}$, and so are the labels $\boldsymbol{y} = \bigcup_{i=1}^{k} \boldsymbol{y_i}$, where all the samples in $\boldsymbol{X_m}$ belong to class c_m. Thus, we have the sub-batch $B_0^1 = \{\boldsymbol{X_1}, \boldsymbol{y_1}\}$. We then define *class accordance* score of two sub-batches B_0^1 and B_0^2 of classes c_1 and c_2 as:

$$C(B_0^1, B_0^2) = \mathbb{E}[cos(\boldsymbol{g_1}, \boldsymbol{g_2})], \tag{11}$$

where $\boldsymbol{g_1}$ is the gradient of the training loss of sub-batch B_0^1 w.r.t. the model parameters, and $cos(\boldsymbol{g_1}, \boldsymbol{g_2}) = (\boldsymbol{g_1}/|\boldsymbol{g_1}|) \cdot (\boldsymbol{g_2}/|\boldsymbol{g_2}|)$. Class accordance measures whether the gradient taken with respect to a sub-batch B_0^1 of class c_1 will also decrease the loss for samples in another sub-batch B_0^2 of class c_2 (Fort et al., 2019; Fu et al., 2020).

Further consider that there are N batches in one training epoch and the training samples are of k classes. The *batch accordance* score between batches B_s and B_t is defined as

$$\begin{aligned}&S_{batch\,accd}(B_s, B_t) \\ &\quad = \frac{1}{k(k-1)}\sum_{j=1}^{k}\sum_{\substack{i=1 \\ i\neq j}}^{k} C(B_s^i, B_t^j).\end{aligned} \tag{12}$$

Batch accordance quantifies the learning consistency of two batches by evaluating how the model updated on one batch affects the other. To be more specific, a positive batch accordance denotes that the measured two batches are under the same learning pace since the model updated according to each batch benefits them both. The gradient accordance of certain epoch (or a part of an epoch, namely the sub-epoch) is finally defined as

$$\begin{aligned}&S_{epoch\,accd} = \\ &\frac{1}{N(N-1)}\sum_{j=i+1}^{N}\sum_{i=1}^{N-1} S_{batch\,accd}(B_s, B_t).\end{aligned} \tag{13}$$

Gradient accordance scales the batch accordance score up from a measure of two batches to that of a sub-epoch.

Criterion for Flooding A positive gradient accordance means that the model performed gradient descent w.r.t. the certain epoch decreases the loss of its batches on average, indicating that the learning pace of most batches are in line with each other. A negative one means that the model has overfitted to some of the training batches since the update of one epoch increases the loss of its batches on average, which is right the stage we would like to identify for the model by gradient accordance. We assume that the optimal flood level lies in the range of the training loss of a model when it is about to overfit. In the following section, we empirically prove that gradient accordance is a reliable and promising criterion for flooding.

4 Experiments

In this section, we provide comprehensive analysis on Flooding-X through extensive experiments on five text classification datasets of various tasks and scales: SST (Socher et al., 2013), MRPC (Dolan and Brockett, 2005), QNLI (Rajpurkar et al., 2016), IMDB (Maas et al., 2011) and AG News (Zhang et al., 2015). We conduct experiments on BERT-base (Devlin et al., 2019) and compare robust accuracy of Flooding-X with other adversarial training algorithms to demonstrate its strength.

4.1 Baseline Methods

We compare our proposed Flooding-X with three adversarial training algorithms and one regularization method.

PGD Projected gradient descent (PGD, Madry et al., 2018) formulates adversarial training algorithms into solving a minimax problem that minimizes the empirical loss on adversarial examples that can lead to maximized adversarial risk.

FreeLB Zhu et al. (2019) propose FreeLB to improve the generalization of language models. By adding adversarial perturbations to word embeddings, FreeLB generates virtual adversarial samples inside the region around input samples.

TAVAT Token-Aware Virtual Adversarial Training (TAVAT, Li and Qiu, 2021) aims at fine-grained perturbations, leveraging a token-level accumulated perturbation vocabulary to initialize the perturbations better and constraining them within a token-level normalization ball.

InfoBERT InfoBERT (Wang et al., 2020a) leverages two mutual-information-based regularizers for robust model training, suppressing noisy mutual information while increasing mutual information between local stable features and global features.

4.2 Attack Methods and Evaluation Metrics

Three well-received attack methods are leveraged via TextAttack (Morris et al., 2020) for an extensive comparison between our proposed method and baseline algorithms.

TextFooler (Jin et al., 2020) identifies the important words for target model and repeats replacing them with synonyms until the prediction of the model is altered. Similarly, TextBugger (Li et al., 2018b) also searches for important words and modifies them by choosing an optimal perturbation from the generated several kinds of perturbations. BERTAttack (Li et al., 2020) applies BERT in a semantic-preserving way to generate substitutes for the vulnerable words detected in the given input.

We consider four evaluation metrics to measure BERT's resistance to the mentioned adversarial attacks under different defence algorithms.

Clean% The clean accuracy refers to the model's test accuracy on the original clean dataset.

Aua% Accuracy under attack measures the model's prediction accuracy on the adversarial data deliberately generated by certain attack method. A higher *Aua%* means a more robust model and a better defender.

Suc% Attack success rate is evaluated by the ratio of the number of texts successfully perturbed by a specific attack method to the number of all the involved texts. Robust models are expected to score low at *Suc%*.

#Query Number of queries denotes the average attempts the attacker queries the target model. The larger the number is, the harder the model is to be attacked.

4.3 Implementation Details

All the baseline methods are re-implemented based on their open-released codes and the results are competing to those reported. We train our models on NVIDIA RTX 3090 and RTX 2080Ti GPUs, depending on the volume of the dataset involved. Most of the parameters such as learning rate and warm-up step are in line with vanilla BERT (Devlin et al., 2019) and the baseline methods. For all of the adversarial methods we set the training step to be 5 for a fair comparison, which is a trade-off between training cost and model performance . The clean accuracy (*Clean*%) is tested on the whole test dataset. The other three metrics (e.g., *Aua*%, *Suc*% and *#Query*) are evaluated on the whole test dataset for SST-2 and MRPC, and 800 randomly chosen samples for IMDB, AG NEWS, and QNLI. We

Datasets	Methods	*Clean%*	TextFooler			BERT-Attack			TextBugger		
			Aua%	*Suc%*	*#Query*	*Aua%*	*Suc%*	*#Query*	*Aua%*	*Suc%*	*#Query*
IMDB	BERT	95.0	24.5	74.2	1533.15	20.3	76.1	2237.38	48.7	47.7	1160.35
	PGD	95.0	26.3	72.1	1194.08	21.3	77.2	1465.83	52.3	46.7	982.02
	FreeLB	97.0	29.5	69.9	1816.26	27.6	69.7	1975.21	51.6	45.9	921.35
	TAVAT	95.5	27.6	71.9	1205.80	23.1	75.1	2244.77	54.1	44.1	1022.56
	InfoBERT	96.3	27.4	72.3	1094.55	20.8	78.3	1428.67	49.8	49.3	1215.39
	Flooding-X	**97.5**	**40.5**	**58.5**	**2315.35**	**32.3**	**65.8**	**2248.71**	**62.3**	**35.8**	**2987.95**
AG NEWS	BERT	**97.0**	20.5	78.9	372.14	6.5	93.1	477.34	42.7	54.6	192.75
	PGD	94.8	37.2	60.8	428.13	**32.8**	**65.7**	**704.78**	58.2	39.1	**252.87**
	FreeLB	94.7	32.3	65.9	405.66	12.7	86.7	573.38	48.8	49.1	210.17
	TAVAT	95.2	39.7	58.3	441.11	23.7	75.2	672.52	55.9	41.5	234.01
	InfoBERT	94.6	29.2	69.1	406.32	15.6	83.3	598.25	50.7	46.7	201.66
	Flooding-X	94.9	**42.4**	**54.9**	**451.35**	27.4	71.0	690.27	**62.2**	**34.0**	222.49
SST-2	BERT	92.7	10.8	88.4	111.81	8.8	90.6	149.84	41.3	55.8	54.37
	PGD	92.8	16.6	82.1	129.33	11.7	87.7	158.80	43.7	53.8	52.49
	FreeLB	92.2	15.4	83.3	128.19	12.1	87.1	160.81	45.1	51.9	53.32
	TAVAT	93.0	19.6	79.0	132.85	14.4	85.4	122.95	43.4	54.6	48.46
	InfoBERT	92.9	18.6	79.5	114.67	16.6	82.8	138.74	43.2	53.6	50.97
	Flooding-X	**93.1**	**34.9**	**62.4**	**149.61**	**27.7**	**70.7**	**199.37**	**51.7**	**45.3**	**60.55**
QNLI	BERT	**91.6**	5.3	94.2	161.88	3.5	96.1	216.46	10.9	88.0	98.39
	PGD	90.6	**28.1**	**68.9**	**269.38**	24.0	73.6	**399.91**	**33.8**	**62.8**	**154.55**
	FreeLB	90.7	23.3	74.3	243.24	14.6	83.9	294.14	17.1	81.3	136.85
	InfoBERT	90.4	23.1	76.5	250.87	11.05	88.8	268.91	12.8	86.9	127.93
	Flooding-X	90.8	27.9	69.27	251.17	**26.2**	**71.2**	364.06	29.5	67.5	137.12
MRPC	BERT	87.8	6.4	92.8	167.59	7.4	91.5	186.97	12.0	86.2	96.82
	PGD	84.3	6.9	92.2	169.01	11.5	86.3	207.90	14.5	82.9	99.90
	FreeLB	83.8	8.2	91.0	150.23	10.3	87.7	193.67	12.5	85.1	96.61
	InfoBERT	87.7	9.1	86.6	178.16	15.0	77.9	201.26	15.9	76.5	98.87
	Flooding-X	**88.9**	**19.9**	**77.1**	**263.05**	**19.4**	**77.7**	**251.44**	**22.3**	**74.3**	**114.23**

Table 1: Experimental results of different models' defense performances on five datasets. The best performance is marked in **bold**. *Clean%* stands for the accuracy tested on the original clean dataset. *Aua%* is short for accuracy under attack, and *Sus%* is the attack success rate of the textual attack methods. Notably, a lower *Sus%* is expected for a more robust model.

train 10 epochs for each model on each dataset, among which the last epochs are selected for the comparison of adversarial robustness.

4.4 Experimental Results

The extensive results of all the above mentioned methods are summarized in Table 1. Generally, our Flooding-X method improves BERT by a large margin in terms of its resistance to adversarial attacks, surpassing the baseline adversarial training algorithms on most datasets under different attack methods.

Under TextFooler attack (Jin et al., 2020), our algorithm reaches the best robust performance on four datasets: IMDB, AG News, SST-2, and MRPC. We observe that Flooding is more effective on smaller datasets than larger ones, since the smaller datasets with shorter training sentences are easier to be memorized by the neural network and are more likely to cause overfitting. On QNLI dataset where Flooding-X fails to win, the accuracy under attack is only 0.2 points lower than the 5-step PGD. This might be explained by the mild change in gradient accordance during training on QNLI dataset, in which case the precise stage of overfitting is hard to be identified. Though we believe that a better value of flood level exists and can further boost the performance, we refuse to take on the pattern of extensive hyper-parameter searching which is against the original purpose of Flooding-X.

Notably, our method performs better than the baseline adversarial training methods by 5 to 20 points on average even without using any adversarial examples as training source, not to mention the vanilla BERT. Under most cases, our method remains the best performing algorithm facing BERTAttack (Li et al., 2020) and TextBugger (Li et al., 2018b). This proves that our method maintains effectiveness under different kinds of adversarial attacks. As a byproduct, the clean accuracy of our method is also the best among all the baseline methods, which is inherent to the vanilla Flooding that aims at better generalization. In the cases of AG News and QNLI, our re-implement the results of BERT fine-tuning to 97.0 and 91.6 respectively so Flooding-X does not surpass the reported performance, but still outperforming the baselines of our implementation.

5 Analysis and Discussion

In this section, we construct supplementary experiments to further analyze the effectiveness of Flooding-X and its building block, i.e., gradient accordance.

5.1 Does Gradient Accordance Capture Overfitting?

Influence function (Koh and Liang, 2017) inspects the influence of one single training data on the model prediction and stiffness (Fort et al., 2019) measures how the model updated according to one sample affects the model prediction on another. Based on these two works, gradient accordance is proposed as a means for identifying model overfitting at sub-epoch level.

As seen in Figure 3, during training process, the turning point of gradient accordance from negative to positive closely matches the point when the test loss is about to increase, which is well received as a signal of overfitting. Since it is computationally intractable to calculate gradient accordance after trained on every single batch, we can only figure out the range where the model is about to overfit by computing gradient accordance at sub-epoch level.

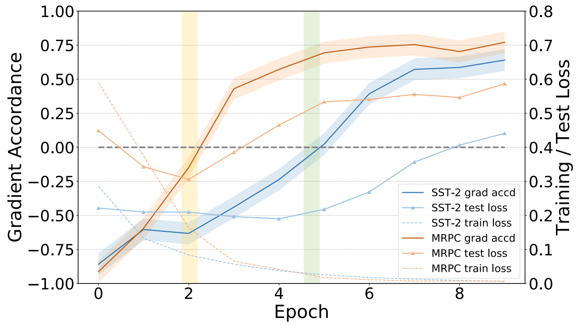

Figure 3: Gradient accordance and training/test loss of each BERT epoch finetuned on SST-2 and MRPC datasets. The grey dashed line represents zero gradient accordance, above which is the model considered to be overfitted. The region marked in yellow and green are the ranges of training iterations where the gradient accordance changes from positive to negative for MRPC and SST-2 respectively.

5.2 How does Flooding-X Help with Robustness?

Despite its outstanding performance of the last training epoch, we find that Flooding-X boosts the robustness of model at an earlier stage than standard fine-tuning and adversarial training methods like FreeLB. As is shown in Figure 4, Flooding-X improves BERT's adversarial robustness to a relatively high level at epoch 5, which is competitive with that of standard fine-tuning at the last epoch. Besides, Flooding-X accelerates the increase of robustness at late training stage. Starting from epoch 7 our method enables a steep increment on the accuracy under attack, which is due to the effect of Flooding that forces the model to perform a more fierce "random walk" since the training loss of most batches are going below the flooding level. It is also demonstrated that the training loss stops approaching zero under the constraint of Flooding-X, while the standard fine-tuning and adversarial training continues to decrease the training loss towards zero which brings about the risk of overfitting.

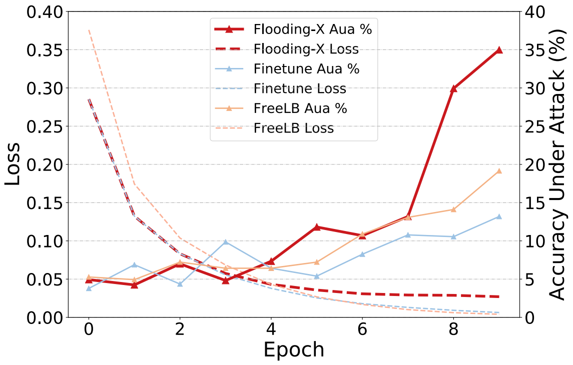

Figure 4: Loss and $Aua\%$ (accuracy under attack) of BERT trained on SST-2 under different methods. Flooding prevents the training loss from approaching zero and results in great improvement of BERT's resistance to adversarial attacks.

Method	SST-2	QNLI	IMDB
Finetune	260	1, 193	1, 059
Flooding-X	**272**	**1,222**	**1,087**
TAVAT	967	4, 105	4, 609
FreeLB	1, 041	4, 340	4, 457
PGD	1, 305	5, 571	5, 664
InfoBERT	2, 174	12, 077	19, 279

Table 2: GPU time consumption (seconds) of training one epoch on the whole dataset. Flooding-X costs nearly the same as fine-tuning and 2-15 times less than the baseline adversarial training algorithms.

5.3 Time Consumption

To further reveal the strength of Flooding-X besides its robustness performance, we compare its GPU training time consumption with baseline methods on several datasets of different sizes. For a fair comparison, every model of each dataset is trained on single NVIDIA RTX 2080Ti GPU with the same batch size, among which models on SST-2 are trained with a batch size of 32 while QNLI and IMDB are trained with 8 and 4 respectively since the training sentences are way longer than SST-2. As is demonstrated in Table 2, the time consumption (seconds) of Flooding-X is competitive with standard fine-tuning, which is far less than that of adversarial training algorithms.

6 Related Work

Adversarial Training Adversarial training (AT) is a well-received method for defending adversarial attacks. As an attempt against adversarial attacks, AT generates gradient-based adversarial samples and leverage them for further training (Goodfellow et al., 2014). A line of works try different means for the generation of adversarial examples. The PGD algorithm (Madry et al., 2018), compared as a baseline method in our experiments, involves multiple projected gradient ascent steps to find the adversarial perturbations which are then used for updating the model parameters. However, it is computationally expensive and has aroused many attempts to cut down on the cost. Shafahi et al. (2019) and Zhu et al. (2019) focus on finding better adversarial sample while maintaining a low cost.

Despite gradient-based methods which generates adversarial perturbations on the continuous input embedding, some works tailor AT for NLP fields. The adversarial examples are generated by replacing the original texts based on certain rules such as semantic similarity (Alzantot et al., 2018; Jin et al., 2020; Li et al., 2020). Ebrahimi et al. (2018) propose a perturbation strategy that conducts character insertion, deletion, and replacement. Jia and Liang (2017) mislead MRC models via a human-involved phrase generation method.

The mentioned algorithms of AT generates additional adversarial examples either by calculating gradients or by human force, which is computationally expensive and effort taking.

Overfitting and Criterion Deep neural networks are shown to suffer from overfitting to training configurations and memorise training scenarios (Takeoka et al., 2021; Rodriguez et al., 2021; Roelofs et al., 2019; Werpachowski et al., 2019), which leads to poor generalization and vulnerability towards adversarial perturbations.

One way of identifying overfitting is to see whether the generalization gap, i.e., the test minus the training loss, is increasing or not (Goodfellow et al., 2016). Ishida et al. (2020) further decompose the situation of the generalization gap increasing into two stages: The first stage is when training and test losses are both decreasing, but the former is decreasing faster then the latter. The next stage is when the training loss is decreasing but the test loss is increasing, after which the training loss continues to approach zero and memorize the training data completely (Zhang et al., 2021; Belkin et al., 2018; Arpit et al., 2017). Derived from influence function (Koh and Liang, 2017), Fort et al. (2019) propose the concept of Stiffness as a new perspective of generalization. They measure how stiff a network is by looking at how a small gradient step in the network parameters on one example affects the loss on another example. This criterion carries is theoretically proved to have a close relation with generalization and overfitting. However, from the practical perspective, it is computationally intractable to compute the stiffness between every single sample during the process of standard training where thousands of samples are involved in one batch.

7 Conclusion

In this work, we propose Flooding-X as an efficient and computational-friendly algorithm for improving BERT's resistance to adversarial attacks. We first theoretically prove that the vanilla Flooding method is able to boost model's adversarial robustness by leading it into a smooth parameter landscape. We further propose a promising and computationally tractable criterion, Gradient Accordance, to detect when the model is about to overfit and accordingly narrow down the hyperparameter space for Flooding with an optimal flood level guaranteed. Experimental results prove that gradient accordance is closely related with the phenomenon of overfitting, equipped with which Flooding-X beats the well-received adversarial training methods and achieves state-of-the-art performances on various NLP tasks facing different textual attack methods. This implies that adversarial examples, either generated by gradient-based algorithms or human efforts, are not a must for the improvement of adversarial robustness. We call for further exploring and deeper understanding in the nature of adversarial robustness and attacks.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR.

Shumeet Baluja and Ian Fischer. 2017. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*.

Mikhail Belkin, Daniel Hsu, and Partha P Mitra. 2018. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 2306–2317.

Guandan Chen, Kai Fan, Kaibo Zhang, Boxing Chen, and Zhongqiang Huang. 2021. Manifold adversarial augmentation for neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3184–3189, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, pages 9–16.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Stanislav Fort, Paweł Krzysztof Nowak, Stanislaw Jastrzebski, and Srini Narayanan. 2019. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint arXiv:1901.09491*.

Jinlan Fu, Pengfei Liu, and Qi Zhang. 2020. Rethinking generalization of neural models: A named entity recognition case study. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7732–7739.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. 2020. Do we need zero training loss after achieving zero training error? In *International Conference on Machine Learning*, pages 4604–4614. PMLR.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

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*, volume 34, pages 8018–8025.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1885–1894, Sydney, Australia. PMLR.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018a. Visualizing the loss landscape of neural nets. *Advances in Neural Information Processing Systems*, 31.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018b. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Linyang Li and Xipeng Qiu. 2021. Token-aware virtual adversarial training in natural language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8410–8418.

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 the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

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*.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Vinay Uday Prabhu, Dian Ang Yap, Joyce Xu, and John Whaley. 2019. Understanding adversarial robustness through loss landscape geometries. *arXiv preprint arXiv:1907.09061*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503, Online. Association for Computational Linguistics.

Rebecca Roelofs, Sara Fridovich-Keil, John Miller, Vaishaal Shankar, Moritz Hardt, Benjamin Recht, and Ludwig Schmidt. 2019. A meta-analysis of overfitting in machine learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 9179–9189.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32:3358–3369.

Chenglei Si, Zhengyan Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2021. Better robustness by more coverage: Adversarial and mixup data augmentation for robust finetuning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1569–1576, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Kunihiro Takeoka, Kosuke Akimoto, and Masafumi Oyamada. 2021. Low-resource taxonomy enrichment with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2747–2758, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2020a. Infobert: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*.

Xiaosen Wang, Yichen Yang, Yihe Deng, and Kun He. 2020b. Adversarial training with fast gradient projection method against synonym substitution based text attacks. *arXiv e-prints*, pages arXiv–2008.

Roman Werpachowski, András György, and Csaba Szepesvári. 2019. Detecting overfitting via adversarial examples. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 7858–7868.

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3905–3911.

Fuxun Yu, Chenchen Liu, Yanzhi Wang, Liang Zhao, and Xiang Chen. 2018. Interpreting adversarial robustness: A view from decision surface in input space. *arXiv e-prints*, pages arXiv–1810.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 6066–6080, Online. Association for Computational Linguistics.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. OpenAttack: An open-source textual adversarial attack toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 363–371, Online. Association for Computational Linguistics.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*.