Characterizing the Impacts of Instances on Robustness

Rui Zheng[1*], **Zhiheng Xi**[1*], **Qin Liu**[2], **Wenbin Lai**[1], **Tao Gui**[3†], **Qi Zhang**[1†], **Xuanjing Huang**[1], **Jin Ma**[4], **Ying Shan**[4], **Weifeng Ge**[1]

[1] School of Computer Science, Fudan University

[2] Viterbi School of Engineering, University of Southern California

[3] Institute of Modern Languages and Linguistics, Fudan University

[4] Tencent PCG

{rzheng20,tgui,qz,xjhuang}@fudan.edu.cn , qliu4147@usc.edu

{zhxi22,wblai21}@m.fudan.edu.cn

Abstract

Building robust deep neural networks (DNNs) against adversarial attacks is an important but challenging task. Previous defense approaches mainly focus on developing new model structures or training algorithms, but they do little to tap the potential of training instances, especially instances with robust patterns carring innate robustness. In this paper, we show that robust and non-robust instances in the training dataset, though are both important for test performance, have contrary impacts on robustness, which makes it possible to build a highly robust model by leveraging the training dataset in a more effective way. We propose a new method that can distinguish robust instances from non-robust ones according to the model's sensitivity to perturbations on individual instances during training. Surprisingly, we find that the model under standard training easily overfits the robust instances by relying on their simple patterns before the model completely learns their robust features. Finally, we propose a new mitigation algorithm to further release the potential of robust instances. Experimental results show that proper use of robust instances in the original dataset is a new line to achieve highly robust models. Our codes are publicly available at https://github.com/ruizheng20/robust_data.

1 Introduction

Deep neural networks (DNNs) have made significant progress in a number of fields, such as computer vision (He et al., 2016) and natural language processing (Devlin et al., 2019), but they are susceptible to adversarial examples, which are crafted by adding small, human-imperceptible adversarial perturbations to normal examples (Goodfellow et al., 2015; Alzantot et al., 2018). To improve the robustness of models, many techniques have been developed, such as robust architecture search (Guo et al., 2020; Huang et al., 2021), model pruning (Sehwag et al., 2020; Zheng et al., 2022), adversarial training (Madry et al., 2018; Zhu et al., 2020) and regularizations (Lyu et al., 2015; Wang et al., 2021). However, most of these defensive approaches focus on developing new model structures or training algorithms, ignoring the fact that training data has a decisive impact on the trained model.

It is widely believed that the more abundant the labeled data, the higher the likelihood of learning diverse features, which in turn leads to well generalized models (Swayamdipta et al., 2020). However, in practice, adversarial robustness remains a challenge that cannot be solved simply by scaling up the dataset (Xie and Yuille, 2020). On the one hand, recent theoretical work argues that training a model invariant to adversarial perturbations requires a much larger dataset than that is required for standard generalization (Schmidt et al., 2018; Alayrac et al., 2019). On the other hand, the model tends to use any available signal to maximize accuracy, and thus, adversarial examples can arise as a result of manipulating highly predictive but fragile features in the data (Ilyas et al., 2019). The above evidences indicate that, adversarial vulnerability is not only associated with the training data size, but is also an inherent property of the data. Most of existing defense methods treat all data equally, which requires us to have a closer look at the dataset whether all instances contribute equally to improving the robustness of the model.

In this paper, we focus on exploring the relationship between training data and adversarial robustness, with the aim of figuring out the following questions:

Q1: Which instances are important for adversarial robustness, and how do we find them? We delve into the training dynamics of each instance and find that instances have different robustness. Even without the help of adversarial training, a portion of the data progressively becomes more robust

[*] Equal contribution.

[†] Corresponding author.

to perturbations, and these instances are called *robust instances*.[1] When we train models on these data in isolation, they are more helpful in improving robustness than other subsets of data with the same size. Motivated by this phenomenon, we propose a metric based on the adversarial loss of each instance across the training epochs to indicate the impact of training instances on robustness. As shown in Figure 2, this metric reveals three distinct regions in the dataset: a region with inherently robust instances, a region with non-robust instances, and a region with instances that fluctuate between robust and non-robust. Based on the proposed metrics, a significant portion of robust instances can be selected from the training dataset to significantly improve the robustness of the model.

Q2: Why is the benefit of robust instances held back when mixed with other training instances? How to make the best out of them to improve robustness? DNNs exhibit memorization effects in that they first memorize easy and clean patterns, and then hard and noisy ones (Zhang et al., 2017; Wang et al., 2019). The robust instances have simple and straightforward task patterns that better align with human perception. We find that the model under standard training easily overfits the robust instances by relying on their simple patterns before the model starts to learn their robust features, which limits the power of robust instances. To address this problem, we propose a new mitigation algorithm that impedes overconfident predictions by regularizations for robust instances to avoid overfitting. The proposed method effectively releases the potential of robust instances, while other instances contribute little to robustness improvement. In particular, our contributions are:

- We show that the instances are not equally important to improve the robustness of the model. The robust instances are more critical to robustness than other instances.
- We propose a new approach to distinguish the robust instances from non-robust ones based on their sensitivity to perturbations during training.
- We find that the standard training easily overfits the robust instances relying on their simple patterns rather than learning robust features.
- We propose a new mitigation algorithm to further release the potential of robust instances. Our analysis and results are verified by extensive experiments.

[1] In this paper, a robust instance means that the model is insensitive to perturbations of this instance. In the later sections, we will show that robust instances also have a positive impact on the robustness of the model.

2 Characterizing robust instances

We find that the model have different sensitivities to perturbations of the instances during the training phase, and this property is strongly correlated with the robustness of the trained models. Based on this, we propose an approach to identify these innately robust instances and demonstrate that they contribute more to robustness when trained in isolation.

2.1 Adversarial Loss during Training

Given a C-class dataset $\mathcal{D} = \{(\mathbf{x}_i^0, \mathbf{y}_i)\}_{i=1}^N$ of size N, $\mathbf{x}_i^0$ denotes the natural input embeddings and $\mathbf{y}_i$ is the label vector. Our method assumes a model $f_{\boldsymbol{\theta}}$ whose parameters $\boldsymbol{\theta}$ are optimized to minimize the empirical risk, as in standard training, without any extra regularization. The loss function on the natural input $\mathbf{x}_i^0$ is $\ell(\mathbf{x}_i^0, \mathbf{y}_i, \boldsymbol{\theta})$. We use a stochastic gradient-based optimization procedure to optimize the model parameters, with training instances randomly ordered at each epoch, across T epochs.

To measure robustness of the instances during training, we perturb the input word embeddings.[2] The goal of an attack method is to find an adversarial example $\mathbf{x}_i$ that remains in the ϵ-ball centered at $\mathbf{x}_i^0 (\|\mathbf{x}_i - \mathbf{x}_i^0\|_F \leq \epsilon)$ but can fool the model to make an incorrect predication $(f_{\boldsymbol{\theta}}(\mathbf{x}_i) \neq \mathbf{y}_i)$. The loss function on adversarial example $\mathbf{x}_i$ can reflect to what extent the robust and useful features are preserved under adversarial perturbation (Ilyas et al., 2019):

$$\ell_{\text{adv}}(\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta}) = \max_{\|\mathbf{x}_i - \mathbf{x}_i^0\|_F \leq \epsilon} \ell(\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta}). \quad (1)$$

A wide range of attack methods have been proposed to craft adversarial examples. Projected Gradient Descent (PGD) iteratively perturbs normal input $\mathbf{x}^0$ for a number of steps K with fixed step size η. If the perturbation goes beyond the ϵ-ball, it is projected back to the ϵ-ball (Madry et al., 2018):

$$\mathbf{x}_i^k = \prod \left(\mathbf{x}_i^{k-1} + \eta \cdot \text{sign}(\nabla_{\mathbf{x}} \ell(\mathbf{x}_i^{k-1}, \mathbf{y}_i, \boldsymbol{\theta})) \right),$$

where $\mathbf{x}_i^k$ is the adversarial example at the k-th step, $\text{sign}(\cdot)$ denotes the sign function and $\prod(\cdot)$ is the projection function.

[2] In our work, the robustness of an instance refers to the robustness of the model on a specific instance.

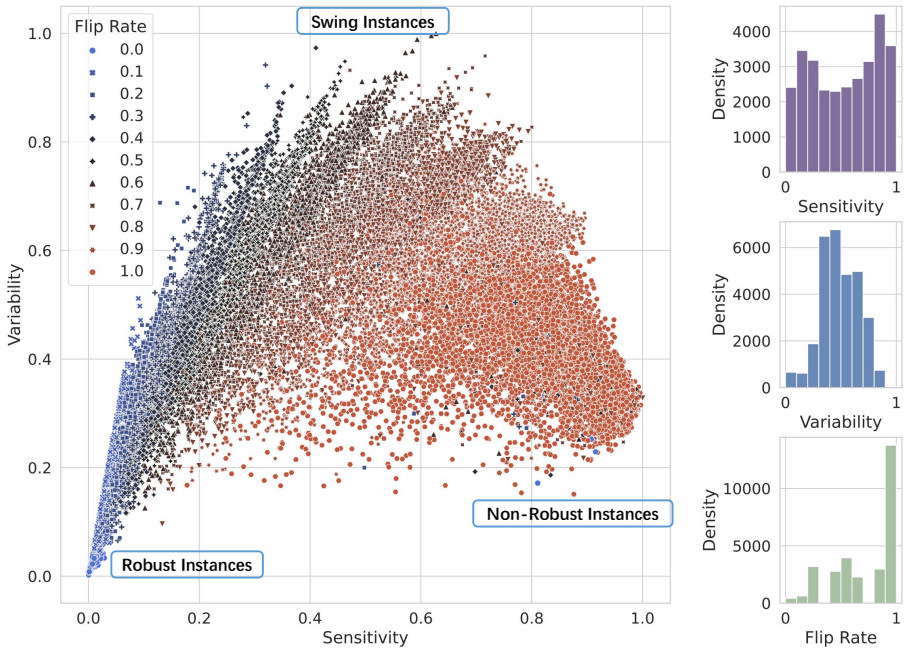


Figure 1: Data map for the SST-2 train set, based on a BERT-base model. Density plots for the three different measures (sensitivity, variability and flip rate) based on the adversarial loss of each instance during training are shown towards the right. The training instances can be roughly classified into three types: robust instances, non-robust instances and swing instances.

We characterize the evolution of robustness using statistics of adversarial losses throughout training. The first statistic aims to measure the sensitivity of the model predictions in the face of perturbations. We define **sensitivity** of individual instance $\mathbf{x}_i^0$ as mean adversarial loss across epochs:

$$\hat{\mu}_i = \frac{1}{T}\sum_{t=1}^{T} \ell_{\text{adv}}(\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta}_t), \qquad (2)$$

where $\boldsymbol{\theta}_t$ denotes the model parameters at the end of t-th epoch. We also consider a more coarse, discrete, and perhaps more intuitive statistic, the rate of times the model provides incorrect predictions when the input is perturbed, referred to as flip rate; this score has only $T+1$ possible values.

Finally, we consider **variability**, i.e., the spread of adversarial loss across epochs as measured by the standard deviation:

$$\hat{\sigma}_i = \sqrt{\frac{\sum_{t=1}^{T} \left(\ell_{\text{adv}}(\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta}_t) - \hat{\mu}_i\right)^2}{T}}. \qquad (3)$$

If the model consistently assigns the same prediction to a perturbed instance (whether correct or not), this instance will have low variability. On the contrary, if the model is indecisive, this instance will have high variability.

2.2 Data Maps

In order to better illustrate the differences in instances, we use the above statistics as coordinates to construct a data map (Swayamdipta et al., 2020). We construct data maps for three widely used benchmark datasets: SST-2 (Socher et al., 2013) – a binary classification task that needs to classify movie reviews as positive or negative; QQP (Wang et al., 2017) – a paraphrase identification task to determine if two questions are paraphrases of each other; AGNews (Zhang et al., 2015) is a text classification task that classifies news articles into one of four topics. All data maps are built using results from the models based on the BERT-base (Devlin et al., 2019) architecture.

Figure 1 shows the data map for the *SST-2* dataset. It is obvious that the data follow a bell-shaped curve with respect to sensitivity and variability. The majority of instances fall within the high sensitivity and moderate variability regions on the map (Figure 1, bottom-right). These instances are always non-robust to perturbations (for the model); therefore, we refer to them as ***non-robust instances***. The second group is smaller and consists of instances with low sensitivity and low variability (Figure 1, bottom-left). As such instances are robust to perturbations during training,

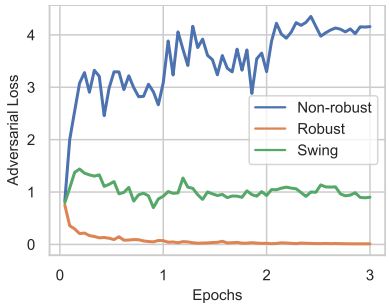


Figure 2: Adversarial losses of 10% most non-robust, robust and swing instances on SST-2 training set. The adversarial loss of robust instances is gradually decreasing, which means that they have more robust features.

we refer to them as ***robust instances***. The third group consists of instances with high variability (Figure 1, top); these instances swing between being sensitive or robust to perturbations. Therefore, we refer to them as ***swing instances***.

Robust dynamic. We consider three data subsets, i.e., 10% of the most *robust*, 10% of the most *non-robust*, and 10% of the most *swing*. Figure 2 shows the adversarial losses of instances from these three regions of the *SST-2* dataset during the training procedure. The most significant difference among the three regions is the decline rate of their adversarial losses, which is much faster in the region of *robust* instances than in the other two regions. This means that robust instances have more robust features and they become more robust to perturbations as training proceeds, without the help of robust learning methods such as adversarial training.

Case study. Table 5 shows instances of *SST-2* that belong to the different regions mentioned above. *Robust* instances have more straightforward task patterns, are better aligned with human perception, and are easy to understand. In contrast, most *non-robust* and *swing* instances are ambiguous, have no obvious task patterns, and are challenging for humans, which may explain why these instances are vulnerable to perturbations (Tsipras et al., 2019).

3 Data Selection using Data Maps

The data map shows the different regions in the dataset. It is natural to wonder what role instances from different regions play in learning and adversarial robustness. We answer this question empirically by training the model solely on instances selected from each region, and then performing standard

Dataset	Baseline	Accuracy	Robustness
SST-2	100% train	92.1	6.1
	100% FreeLB	91.7	**29.4**
	50% **non-robust**	**93.1**	4.7
	50% **swing**	91.6	17.2
	50% **robust**	91.1	23.9
QQP	100% train	90.1	20.8
	100% FreeLB	**90.2**	27.4
	50% **non-robust**	86.2	18.3
	50% **swing**	88.9	20.6
	50% **robust**	75.5	**28.7**

Table 1: Accuracy (%) and robustness (accuracy under TextFooler attack) for BERT-base models trained on three different type of instances of SST-2 and QQP. Training 50% most robust instances achieves better robustness performance, even matching the adversarial training on 100% training data.

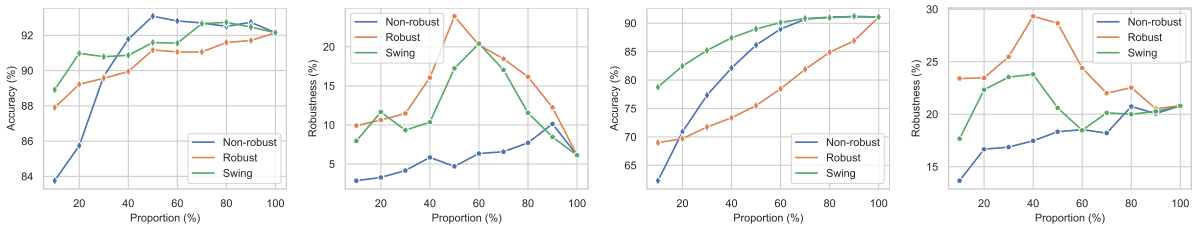
generalization (Accuracy) as well as robustness (accuracy under attack) evaluations.

The training strategy is simple and straightforward – we train the model from scratch on the subsets of the training data selected by ranking instances based on the statistics described above. We hypothesize that *robust* and *swing* instances are more important for improving the robustness of the model because they have more robust features and more stable to perturbations as training proceeds. We compare the performance of the models trained on different data regions with other baselines. All considered subsets contain 50% of the training data (to control the effect of training data size on performance).

Baselines. The most natural baseline is using all of the data (**100% train**). Our data selection baselines consider the subsets of 50% of the most *robust* (**50% robust**), 50% of the most *non-robust* (**50% non-robust**) and 50% of the most indecisive (**50% swing**), which is a trade-off between robust and non-robust instances. Finally, we also compare our models trained on data subsets with a textual adversarial training method, Freelb (Zhu et al., 2020), which is a strong defense baseline in NLP (**100% FreeLB**).

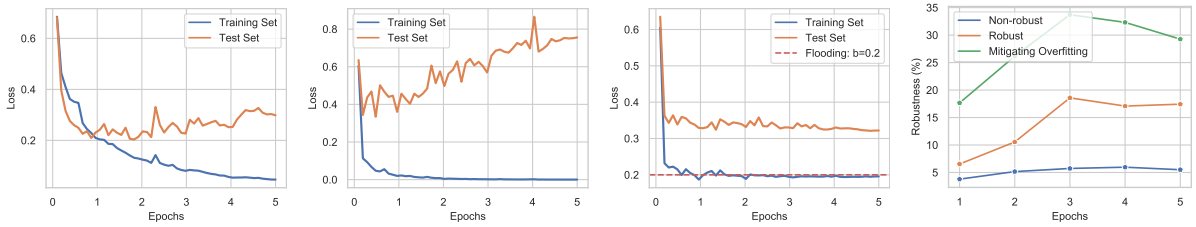
Results. We report accuracy on the test set to evaluate generalization performance, and accuracy under attack using TextFooler (Jin et al., 2020) as the attacker to measure adversarial robustness.

Table 1 shows our results on the SST-2 and QQP datasets. We can observe that: 1) Training on 50% most *robust* instances results in the best robustness performance among all data selections, ex-



(a) Accuracy on SST-2 (b) Robustness on SST-2 (c) Accuracy on QQP (d) Robustness on QQP

Figure 3: Accuracy (%) and robustness performance with increasing proportion (%) of most non-robust, robust and swing instances. 50% of the most robust instances are sufficient to achieve competitive robustness performance, while more data would impair robustness. The key finding is that only a portion of the instances is helpful for the robustness of the model.



(a) 50% Non-robust instances (b) 50% Robust instances (c) Mitigating overfitting (d) Robustness evaluation

Figure 4: Overfitting to robust instances. 1) (a) and (b) show the model easily overfits the robust instances. 2) (b) and (d) show the standard training has overfitted the robust instances before the models start learning their robust features. 3) (c) and (d) show that by mitigating the overfitting to robust instance, the model can learn their robust features better.

ceeding that of 100% train and even better than 100% FreeLB on QQP. 2) In both datasets, the robustness performance of *non-robust* instances is lower than all other baselines, which is expected because the features of these instances are non-robust. 3) The generalization performance of *non-robust* instances is better than that of *robust*, because well generalized features are more sensitive to adversarial perturbations. 4) The model trained on most *swing* instances sacrifices a bit of robustness to improve generalization compared with the performance of the model trained on *robust* instances.

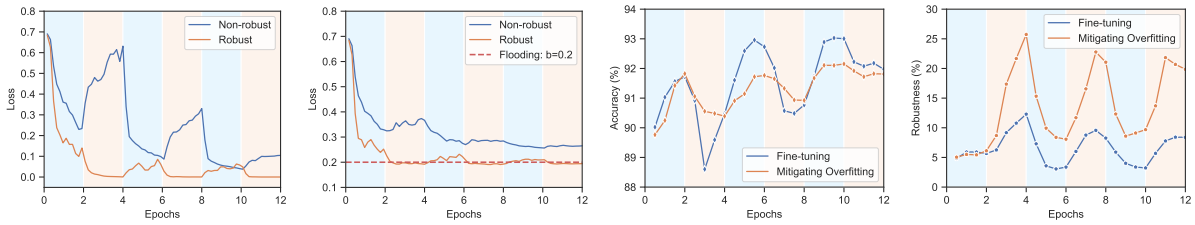
In Figure 3, we show the evolution of generalization performance and adversarial robustness when we change the size of the selected subsets of data. Each point in the figure corresponds to retraining the model from scratch (with the same hyperparameters as the base model) on an increasingly larger subset of the training data. We observe that the generalization performance improves rapidly when we increase the size of *non-robust* and *swing* subsets. This means that the original training dataset is redundant to the model, and training the model on a small portion of the data also gives excellent generalization performance. Comparatively, by increasing the number of *robust* and *swing*, 50% of the data is sufficient to achieve excellent robustness performance, while more data actually hurts robustness. On the one hand, we require sufficient data to improve the generalization, and on the other hand, excessive well-generalized data will harm the robustness. We cannot rely on data selection alone to get the best generalization and robustness at the same time, and in the next sections we will show how to leverage instances from different regions to achieve win-win results in performance.

4 Why Do Robust Instances Fail?

In the previous section, we observed that *robust* instances are import for adversarial robustness. This leads us to wonder why *robust* instances fail in competition with *non-robust* instances, and is it inevitable when we train these data together? In this section, we provide further insight into the training procedure by investigating the interactions between *robust* and *non-robust* instances.

4.1 Overfitting to Robust Instances

As shown in Figure 4, we find that *robust* instances are easy to learn and converge faster than *non-*



(a) Standard training (b) Mitigating overfitting (c) Accuracy (d) Robustness evaluation

Figure 5: Synthetic continuous learning settings for SST-2. The background color of each column indicates the training partition, with the light blue background denoting training non-robust instances and the light orange denoting training robust instances. The curves in (a) and (b) track the performance of the two partitions during interleaved training, and curves in (c) and (d) indicate the performance of the test set. The figure highlights that mitigating the overfitting of the model to robust instances can reduce the conflict between robust and non-robust instances, making the robust instances to better improve the robustness of the model.

robust instances. As the training loss of robust instances approaches zero, the test loss is increasing, which means that the model overfits the robust instances. However, the robustness of the model continues to improve, even as the training loss becomes (close to) zero. This suggests that the model under standard training easily overfits the *robust* instances before the model starts to learn their robust features. We further show the results in Figure 4(c) when we use the Flooding (Ishida et al., 2020) algorithm to mitigate overfitting to *robust* instances, where Flooding intentionally prevents further reduction of the training loss when it reaches a reasonably small value. Flooding prevents the model from memorizing and being overconfident in these instances. By mitigating the overfitting to robust instances, the benefits of robust data are further demonstrated.

The above analysis leads us to believe that, to some extent, the overfitting to *robust* instances reduces their ability to improve robustness, especially when competing with *non-robust* instances. To test this hypothesis, we conduct an experiment inspired by the standard continuous learning setup (Toneva et al., 2019). We created two equally sized datasets by extracting 50% of the most *robust* and 50% of the most *non-robust* instances, respectively. Then, we train a model for 2 epochs on each partition in an alternating fashion, while tracking generalization and robustness on the test set. The background color represents which of the two datasets is currently being used for training.

It can be concluded that: 1) Figures 5(a) and 5(d) show that even if we train the *non-robust* instances first, the *robust* instances can still improve robustness in the next 2 epochs. However, as the training loss of the *robust* instances converges to zero, the model learns less and less from the robust instances. 2) As shown in Figure 5(a), there is a significant conflict in learning between the robust and non-robust instances, which means that the model learns distinct features from them. 3) Figure 5(b) shows the conflict effect between robust and non-robust instances is reduced by mitigating the overfitting of the model to robust instances. Therefore, mitigating overfitting to robust instances allows the model to better generalize their robust features.

4.2 Mitigating Overfitting

Based on the above analysis, our aim is to mitigate the overfitting to *robust* instances in the standard training process. While overfitting has been extensively studied in the machine learning community to reduce the generalization gap, few approaches consider the impact of overfitting on adversarial robustness.

To address this problem, we propose to regularize the predictions of *robust* instances from being over-confident by integrating loss-restricted (LR) methods (Szegedy et al., 2016; Ishida et al., 2020) into the standard training framework. We believe that LR is suitable for standard training because it can be easily implemented by adding a term to the objective function. Specifically, we construct a new dataset $\mathcal{D}_r^{p\%}$ using the $p\%$ most *robust* instances in the original dataset, and other instances construct $\mathcal{D}\backslash\mathcal{D}_r^{p\%}$. The training loss on the instance $\mathbf{x}_i$ with LR can be expressed as:

$$\ell_{\mathrm{LR}}(\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta}) = \begin{cases} \ell(\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta}), & \mathbf{x}_i \in \mathcal{D}\backslash\mathcal{D}_r^{p\%}, \\ \mathcal{R}(\ell\left(\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta}\right)), & \mathbf{x}_i \in \mathcal{D}_r^{p\%}, \end{cases}$$

where $\mathcal{R}(\cdot)$ denotes regularization term. In our

method, we consider two regularizations, Flooding (Ishida et al., 2020) and Label Smoothing (Szegedy et al., 2016), to control the training loss for alleviating overfitting.

Flooding is a direct solution to the issue that the training loss becomes (near-)zero. When the training loss reaches a reasonably small value, Flooding intentionally prevents further reduction of the training loss, and the flood level corresponds to the level of training loss that the user wants to maintain. The algorithm of Flooding is simple, modifying the training loss as:

$$\mathcal{R}_{\mathrm{FL}}(\ell(\mathbf{x}_i, y_i, \boldsymbol{\theta})) = |\ell(\mathbf{x}_i, y_i, \boldsymbol{\theta}) - b| + b, \quad (4)$$

where $b > 0$ is the user-specified flood level. By using Flooding, the training loss will oscillate around the flood level. The model will continue to "random walk" with the same non-zero training loss, thus the model will move into a region with a flat loss landscape, leading to better generalization.

Label smoothing is another widely known technique to mitigate the overfitting problem by penalizing overconfident model outputs (Müller et al., 2019). For a model trained with hard labels, we minimize the expected value of the cross-entropy between the true label $\mathbf{y}_i$ and the model's output for $\mathbf{x}_i$, where $\mathbf{y}_i$ is a one-hot vector with "1" for correct class and "0" for others. For a model trained with the label smoothing, we minimize the cross-entropy between the modified label $\mathbf{y}_i^{\mathrm{LS}}$ and the model's output:

$$\mathcal{R}_{\mathrm{LS}}(\ell(\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta})) = \ell(\mathbf{x}_i, \mathbf{y}_i^{\mathrm{LS}}, \boldsymbol{\theta}), \quad (5)$$

where $\mathbf{y}_i^{\mathrm{LS}} = \mathbf{y}_i(1-\alpha)+\alpha/C$, α is the smoothing parameter and C is the number of classes .

From Table 2, we can observe that the above regularizations are valid. The robustness of the model is significantly improved by mitigating the overfitting of *robust* instances, while also achieving high accuracy. We use two regularizations to prove that the above conclusion does not depend on any specific regularization. In the experimental section, we perform more experiments to verify the effectiveness of the proposed method.

5 Experiments

In this section, we provide experimental results using BERT-base (Devlin et al., 2019) as a backbone model on the SST-2 (Socher et al., 2013), QQP (Wang et al., 2017) and AGNews (Zhang et al., 2015) datasets to validate and analyze the effectiveness of our proposed approach. Experimental implementation details and hyperparameters are provided in Appendix A.

Dataset	Baseline	Accuracy	Robustness
SST-2	Standard Training	92.1	6.1
	+ Flooding	91.9	46.0
	+ Label Smoothing	92.5	41.4
QQP	Standard Training	90.1	20.8
	+ Flooding	90.9	39.4
	+ Label Smoothing	91.0	45.1

Table 2: Accuracy (%) and robustness for BERT-base models when using Flooding and Label Smoothing to mitigate the overfitting to 50% most robust instances. By mitigating the overfitting to robust instances and using standard training on other instances, excellent accuracy and robustness can be achieved at the same time.

5.1 Robust Evaluation

The evaluation metrics used in our experimental analyses include: 1) **Clean**%: the accuracy on the clean test dataset; 2) **Aua**%: the model's prediction accuracy under attack; 3) **#Query**: the average number of times the attacker queries the victim model. For a robust model, higher accuracy under attack and higher query times are expected. The baselines we used and adversarial settings are shown in Appendix A. More experimental results and analysis are presented in Appendix B.

Results. Table 3 shows the results of the proposed method and other baselines under adversarial attack. We can observe that the proposed method achieves a significant improvement in robustness compared to other defense methods. Both Flooding and Label Smoothing work well in our approach. The proposed method improves the robustness without sacrificing accuracy, while robust tickets lose much accuracy on SST-2 despite also having a high robustness. We consistently demonstrate the effectiveness of our approach on different datasets.

6 Related Work

Text attacks typically generate adversarial examples by manipulating characters (Ebrahimi et al., 2018; Gao et al., 2018), words (Ren et al., 2019; Jin et al., 2020; Li et al., 2020; Alzantot et al., 2018; Zang et al., 2020; Maheshwary et al., 2021), phrases (Iyyer et al., 2018) of the original input, or even entire sentences (Wang et al., 2020), to

Dataset	Method	Clean%	BERT-Attack		TextFooler		TextBugger	
			Aua%	#Query	Aua%	#Query	Aua%	#Query
SST-2	Fine-tune	92.1	3.8	106.4	6.1	90.5	28.7	46.0
	PGD	92.2	13.4	151.3	18.1	118.5	44.2	53.6
	FreeLB	91.7	23.9	174.7	29.4	132.6	49.7	53.8
	InfoBERT	92.1	14.4	162.3	18.3	121.1	40.3	51.2
	RobustT	90.9	20.8	169.2	28.6	149.8	43.1	53.9
	Ours+Flooding	**92.3**	**42.4**	224.6	46.8	163.3	55.9	63.2
	Ours+Label Smoothing	91.9	41.3	**235.7**	**47.3**	**170.5**	**58.8**	**63.4**
QQP	Fine-tune	90.1	18.1	187.8	20.8	131.3	24.3	58.8
	PGD	91.2	30.5	254.1	33.6	174.2	35.9	89.2
	FreeLB	91.3	32.8	262.8	36.4	180.2	37.7	96.8
	InfoBERT	**91.5**	33.0	263.9	36.3	180.1	38.2	94.6
	RobustT	91.2	35.2	271.2	37.3	183.9	39.5	97.0
	Ours+Flooding	90.9	37.0	289.8	39.4	195.8	40.8	98.4
	Ours+Label Smoothing	91.1	**42.4**	**316.1**	**44.5**	**208.9**	**47.3**	**102.1**
AGNews	Fine-tune	94.7	4.1	412.9	14.7	306.4	40.0	166.2
	PGD	95.0	20.9	593.2	36.0	399.2	56.4	193.9
	FreeLB	95.0	19.9	581.8	33.2	396.0	52.9	201.1
	InfoBERT	94.4	11.1	517.0	25.1	374.7	47.9	193.1
	RobustT	**94.9**	21.8	617.5	35.2	415.6	49.0	206.9
	Ours+Flooding	94.5	73.1	874.2	76.6	527.5	78.7	252.9
	Ours+Label Smoothing	94.7	**75.4**	**904.0**	**79.4**	**947.3**	**82.3**	**262.6**

Table 3: Main results on adversarial robustness evaluation. The proposed method on downstream tasks achieves a significant improvement of robustness. The best performance is marked in bold.

deceive the model. The most widely used attacks are word-level attacks, which replace words in a sentence with synonyms and maintain a high-level similarity and validity in the semantic (Li et al., 2020) or embedding space (Jin et al., 2020).

To counter adversarial attacks, a number of defense methods have been developed, such as adversarial training (Madry et al., 2018; Zhu et al., 2020; Li and Qiu, 2021), information compression (Wang et al., 2021; Zhang et al., 2022), and model pruning (Zheng et al., 2022). However, most of these defensive approaches focus on developing new model structures and training algorithms, ignoring the fact that training data has a decisive impact on the robustness of the model. In this paper, we propose a new defense method from a data perspective to improve the robustness of the model by better utilizing the robust instances in the original dataset.

A body of work tends to view the existence of adversarial examples as an inevitable consequence of using high-dimensional inputs and the statistical fluctuations due to data size and data noise (Goodfellow et al., 2015; Gilmer et al., 2018). However, Ilyas et al. (2019) claim that adversarial vulnerability is a direct result of sensitivity to well-generalizing features in the data. Data-related studies in the field of robustness focus on improving the robustness of models using more unlabeled data (Carmon et al., 2019; Alayrac et al., 2019) and data augmentation (Lee et al., 2020; Rebuffi et al., 2021). Dong et al. (2021) find that low-quality data may not be useful or even detrimental to adversarial robustness. To the best of our knowledge, no work has attempted to characterize the impact of each instance in the training dataset on robustness.

7 Conclusion

In this paper, we address the challenge of understanding the impact of training instances on robustness, particularly to improve the robustness of the model. We study the adversarial losses of each instance during training and show how these losses can be used as a metric to identify robust instances. Our empirical results suggest that the proposed metric is a very promising measure for characterizing the contribution of training instances to robustness, and can be used to prune out non-robust instances to construct a dataset that is inherently robust. Furthermore, we show that standard training can easily overfit robust instances by relying on their simple patterns before the model learns the robust features. The robustness of the model can be significantly improved by mitigating the overfitting of the model to robust instances during the standard training. Further investigations in this direction may lead to new technologies for adversarial defense.

Limitations

In this work, we find that robust instances are helpful for model robustness and propose a metric to select them. However, we only applied one single criterion, i.e. the training dynamic of adversarial loss, as selection metric. More instance features can be inspected in terms of the relation with model robustness and further serve as metrics for robust data selection. Moreover, in this work, we use the selected data for standard fine-tuning with simple regularization, while the impact of data robustness on adversarial training is not studied. These two problems will be explored in future work.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No.62206057,62076069,61976056), Shanghai Rising-Star Program (23QA1400200), Natural Science Foundation of Shanghai (23ZR1403500), and CCF-Tencent Open Fund, except the third auther Qin Liu, who is funded by Graduate Fellowship from University of Southern California.

References

Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. 2019. Are labels required for improving adversarial robustness? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12192–12202.

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.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C. Duchi, and Percy Liang. 2019. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11190–11201.

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.

Chengyu Dong, Liyuan Liu, and Jingbo Shang. 2021. Data quality matters for adversarial training: An empirical study. *arXiv preprint arXiv:2102.07437*.

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.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*, pages 50–56. IEEE Computer Society.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian J. Goodfellow. 2018. Adversarial spheres. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Minghao Guo, Yuzhe Yang, Rui Xu, Ziwei Liu, and Dahua Lin. 2020. When NAS meets robustness: In search of robust architectures against adversarial attacks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 628–637. Computer Vision Foundation / IEEE.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Hanxun Huang, Yisen Wang, Sarah M. Erfani, Quanquan Gu, James Bailey, and Xingjun Ma. 2021. Exploring architectural ingredients of adversarially robust deep neural networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 5545–5559.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry.

2019. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 125–136.

Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. 2020. Do we need zero training loss after achieving zero training error? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4604–4614. PMLR.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. 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 *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

Jin-Ha Lee, Muhammad Zaigham Zaheer, Marcella Astrid, and Seung-Ik Lee. 2020. Smoothmix: a simple yet effective data augmentation to train robust classifiers. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pages 3264–3274. Computer Vision Foundation / IEEE.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society.

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 *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 8410–8418. AAAI Press.

Chunchuan Lyu, Kaizhu Huang, and Hai-Ning Liang. 2015. A unified gradient regularization family for adversarial examples. In *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*, pages 301–309. IEEE Computer Society.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021. Generating natural language attacks in a hard label black box setting. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13525–13533. AAAI Press.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4696–4705.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A. Mann. 2021. Data augmentation can improve robustness. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 29935–29948.

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.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. 2018. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5019–5031.

Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. 2020. HYDRA: pruning adversarially robust neural networks. In *Advances in Neural Information*

Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

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.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An empirical study of example forgetting during deep neural network learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021. Infobert: Improving robustness of language models from an information theoretic perspective. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020. CATgen: Improving robustness in NLP models via controlled adversarial text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5141–5146, Online. Association for Computational Linguistics.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. 2019. Symmetric cross entropy for robust learning with noisy labels. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 322–330. IEEE.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4144–4150. ijcai.org.

Cihang Xie and Alan L. Yuille. 2020. Intriguing properties of adversarial training at scale. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Cenyuan Zhang, Xiang Zhou, Yixin Wan, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. 2022. Improving the adversarial robustness of NLP models by information bottleneck. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3588–3598, Dublin, Ireland. Association for Computational Linguistics.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Rui Zheng, Bao Rong, Yuhao Zhou, Di Liang, Sirui Wang, Wei Wu, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022. Robust lottery tickets for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2211–2224, Dublin, Ireland. Association for Computational Linguistics.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Experimental Details

A.1 Implementation Details

Our implementation of the proposed method is mainly based on BERT, so most of the hyperparameter settings are based on them.[3] We use AdamW as our optimizer with the learning rate $2e^{-5}$, a batch size 32 and a linear learning rate decay schedule with a warm-up of 0.1. The dropout rate is set to 0.1 for all task-specific layers. We implement three adversarial attack methods using TextAttack framework and follow the default parameter settings.[4] The accuracy (Clean%) is tested on the whole test set. Other adversarial robustness evaluation metrics (e.g., Aua% and #Query) are evaluated on the 1000 randomly selected test instances for all datasets. All experiments are conducted using NVIDIA RTX3090 GPUs.

A.2 Hyperparameters

Our proposed method consists of two stages; the first stage finds robust instance from the training dataset based on the statistics of adversarial loss, and the second stage, uses regularizations to mitigate the overfitting of the model to robust instances during standard training. Adversarial loss objective introduces four hyperparameters: the perturbation step size η, the initial magnitude of perturbations ϵ_0, the number of adversarial steps K, and we do not constrain the bound of perturbations. In addition, we report the flood level b, smoothing parameter α and the $p\%$ most *robust* instances in the proposed overfitting mitigation method.

	Hyperparameters	SST-2	QQP	AGNEWS
Stage1	η	0.08	0.08	0.08
	ϵ_0	0.05	0.05	0.05
	K	8	8	8
	Epoch	10	10	10
Stage2	b	0.2	0.2	0.2
	α	0.8	0.8	0.8
	$p\%$	30	50	50
	Epoch	5	5	5

Table 4: Hyperparameters used in the proposed method.

A.3 Baselines

The baseline methods we use include: 1) **Fine-tune** (Zhang et al., 2015): the official BERT implementation on downstream tasks; 2) **PGD** (Madry et al., 2018): standard adversarial training with PGD attacks; 3) **FreeLB** (Zhu et al., 2020): an enhanced adversarial training to generate adversarial examples at low cost; 4) **infoBERT** (Wang et al., 2021): the information bottleneck-based approach filtering out redundant and noisy information to improve the robustness of the features; 5) **RobustT** (Zheng et al., 2022): the robust sub-network extracted from the original model with innately better robustness.

[3] https://github.com/huggingface/transformers

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

A.4 Attack Settings.

Three widely accepted attack methods are used to evaluate the robustness of the proposed approach and other baselines. **BERT-Attack** (Li et al., 2020) and **TextFooler** (Jin et al., 2020) are two word-level attackers that first identify the important words in a sentence, and then replace them with semantically similar and grammatically correct synonyms. **TextBugger** (Li et al., 2019) generates adversarial typos by using both character-level and word-level perturbations.

B Additional Results

B.1 Case Study

Table 5 shows the case study for the instances selected by the proposed metric. *Robust* instances have more straightforward task patterns, are better aligned with human perception, and are easy to understand. In contrast, most *non-robust* and *swing* instances are ambiguous, have no obvious task patterns, and are challenging for humans.

B.2 Importance of Robust Dynamic

In this paper, we propose a new metric that identifies important instances contributing to adversarial robustness based on the adversarial loss during training. To further understand the role of the adversarial loss in our approach, we compared our method with a metric based on the original training loss. From the results in Table 6, data selection based on training loss statistics can identify instances that play an important role in generalization, rather than robustness.

B.3 Mitigating Overfitting to Other Instances

In the proposed method, we use regularization to mitigate the overfitting of model to robust instances. In Table 7, we show the results when regularization is applied on other instances with different sizes. When we use regularization on robust and swing instances, the robustness of the model is significantly

Instances	Sentence	Label
Robust	a charming , funny and beautifully crafted import	Positive
	a lovely and beautifully	Positive
	have a great time	Positive
	bright shining star	Positive
	bad writing , bad direction and bad acting – the trifecta of badness	Negative
	a good time	Positive
	charming , funny and beautifully crafted import	Positive
	a lovely and beautifully photographed romance .	Positive
	's lovely and amazing	Positive
	have a good time	Positive
Non-robust	vulgarity , sex scenes , and	Negative
	hard-driving narcissism is a given	Negative
	marinated in clichés and mawkish dialogue	Negative
	as its uncanny tale of love , communal discord , and justice	Negative
	cloying messages and irksome characters	Negative
	seems a prostituted muse ...	Negative
	is tragically	Negative
	painful , horrifying and oppressively tragic	Negative
	some weird relative trots out the video he took of the family vacation to stonehenge	Negative
	cheesy backdrops , ridiculous action sequences ,	Negative
Swing	hide new secretions from the parental units	Negative
	an overall sense of brusqueness	Negative
	, dragon loses its fire midway , nearly flickering out by its perfunctory conclusion .	Negative
	your brain and your secret agent decoder ring at the door	Negative
	, two towers outdoes its spectacle .	Positive
	as-nasty -	Negative
	semi-surrealist exploration of the creative act .	Positive
	viscerally repellent	Negative
	silly – and gross – but it 's rarely as moronic as some campus gross-out films .	Negative
	bittersweet	Positive

Table 5: Examples of the most robust, non-robust and swing instances in the SST-2 training set, with gold standard labels. *Robust* instances have more straightforward task patterns, are better aligned with human perception, and are easy to understand.

improved, while using regularization on non-robust instances does not improve the robustness. This suggests that the excellent performance of the proposed work is due to our better exploitation of the robust features in the data rather than depending on regularizations. Although previous work finds that the Flooding algorithm can improve the robustness of the model, it cannot obtain a performance comparable to the proposed method. Moreover, there is no evidence to show that the robustness of the model can be improved by using label smoothing alone.

B.4 Instances from More Regions

Table 6 shows the accuracy and robustness evaluation for models trained on instances selected from different regions on the data map. The model trained on robust instances (with low sensitivity and low variability) achieves the best robustness.

B.5 Effect of Regularized Instance Proportion

Figure 6 shows the proposed method across all proportions of regularized instances. The adversarial robustness improves as the proportions of regularized instances grows until a certain threshold, then the robustness deteriorates.

B.6 Additional Data Maps

The data maps for AGNEWS and QQP are shown in Figure 7 and Figure 8, respectively.

Datasets	Metrics	Regions	Top 10%		Top 30%		Top 50%	
			Clean%	Aua%	Clean%	Aua%	Clean%	Aua%
SST-2	Adversarial Loss	Bottom-Left	87.9	9.9	89.6	**11.5**	91.1	**23.9**
		Bottom-Right	83.8	2.9	89.7	4.2	**93.1**	4.7
		Top	**88.9**	8.0	90.8	9.3	91.6	17.2
	Training Loss	Bottom-Left	83.5	**13.8**	88.2	11.2	90.7	12.7
		Bottom-Right	86.0	4.5	91.2	5.4	92.4	4.8
		Top	34.1	0.2	**91.6**	10.2	92.7	8.9
QQP	Adversarial Loss	Bottom-Left	69.0	**23.4**	71.7	**25.5**	75.5	**28.7**
		Bottom-Right	62.3	13.7	76.2	16.8	86.2	18.3
		Top	**78.8**	17.7	**85.2**	23.5	88.9	20.6
	Training Loss	Bottom-Left	58.9	15.5	66.3	11.8	76.7	14.5
		Bottom-Right	20.5	0.4	83.1	7.4	89.0	11.2
		Top	19.5	0.8	77.8	4.3	**90.3**	11.0

Table 6: Compare the performance of models trained on instances that are selected by the statistics of adversarial loss and original training loss. Our results show that adversarial loss plays a vital role in the proposed metric.

Datasets	Instances	Regularization	Top 10%		Top 30%		Top 50%	
			Clean%	Aua%	Clean%	Aua%	Clean%	Aua%
SST-2	Robust	Flooding	**92.7**	25.4	92.3	46.8	91.9	**46.0**
		Label Smoothing	92.6	**32.0**	92.4	**47.3**	**92.5**	41.4
	Non-robust	Flooding	92.4	5.4	92.3	14.0	92.0	11.9
		Label Smoothing	92.4	11.1	92.4	10.5	92.0	7.1
	Swing	Flooding	92.4	4.4	92.0	7.9	92.4	14.6
		Label Smoothing	92.4	23.8	**93.2**	32.7	92.1	26.3
QQP	Robust	Flooding	90.9	20.8	91.1	31.6	90.9	39.4
		Label Smoothing	91.1	**37.4**	90.9	**41.0**	**91.1**	**44.5**
	Non-robust	Flooding	91.2	20.2	91.1	22.2	**91.1**	27.6
		Label Smoothing	**91.4**	28.7	**91.2**	27.2	**91.1**	24.2
	Swing	Flooding	91.1	22.0	91.0	22.6	90.9	29.0
		Label Smoothing	91.2	30.2	**91.2**	32.8	91.0	35.2

Table 7: Accuracy and robustness evaluation using regularizations on $p\%$ $(p = 10, 30, 50)$ most non-robust, robust and swing instances. The robustness of the model is significantly improved when the overfitting of the model to robust and swing instances is mitigated, while using regularization on non-robust instances cannot improve the robustness.

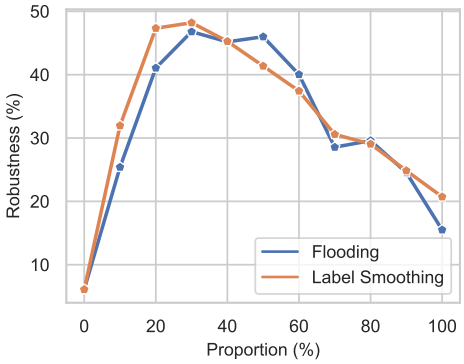


Figure 6: Robustness evaluation results of the proposed method under different proportions of regularized instances. The adversarial robustness improves as the proportion of regularized instances grows until a certain threshold, then the robustness deteriorates.

Datasets	Instances	Top 10%		Top 30%		Top 50%	
		Clean%	Aua%	Clean%	Aua%	Clean%	Aua%
SST-2	Robust	87.9	**9.9**	89.6	**11.5**	91.2	**23.9**
	Non-robust	83.8	2.9	89.7	4.2	93.1	4.7
	Low-Sensitivity	88.1	9.0	89.7	10.7	91.1	21.0
	High-Sensitivity	83.7	3.2	89.7	4.5	**93.2**	4.5
	Low-Variability	87.7	7.9	88.8	4.9	91.6	5.8
	Swing	88.9	8.0	90.8	9.3	91.6	17.2
	Small Filp Rate	62.3	9.7	88.6	9.4	88.9	12.8
	Large Filp Rate	**91.2**	7.1	**91.9**	6.3	92.5	4.7
QQP	Robust	69.0	**23.4**	71.7	**25.5**	75.5	**28.7**
	Non-robust	62.3	13.7	**86.2**	16.8	87.0	18.3
	Low-Sensitivity	68.0	19.3	70.5	21.5	75.0	24.7
	High-Sensitivity	65.2	9.0	77.3	20.9	86.2	21.9
	Low-Variability	**79.6**	16.0	84.9	14.7	88.9	10.3
	Swing	78.8	17.7	85.2	23.5	88.9	20.6
	Small Filp Rate	69.7	13.5	73.4	19.3	78.5	21.4
	Large Filp Rate	70.9	7.6	82.1	7.5	**89.0**	8.4

Table 8: Accuracy and robustness evaluation for models trained on instances selected from different regions on the data map.

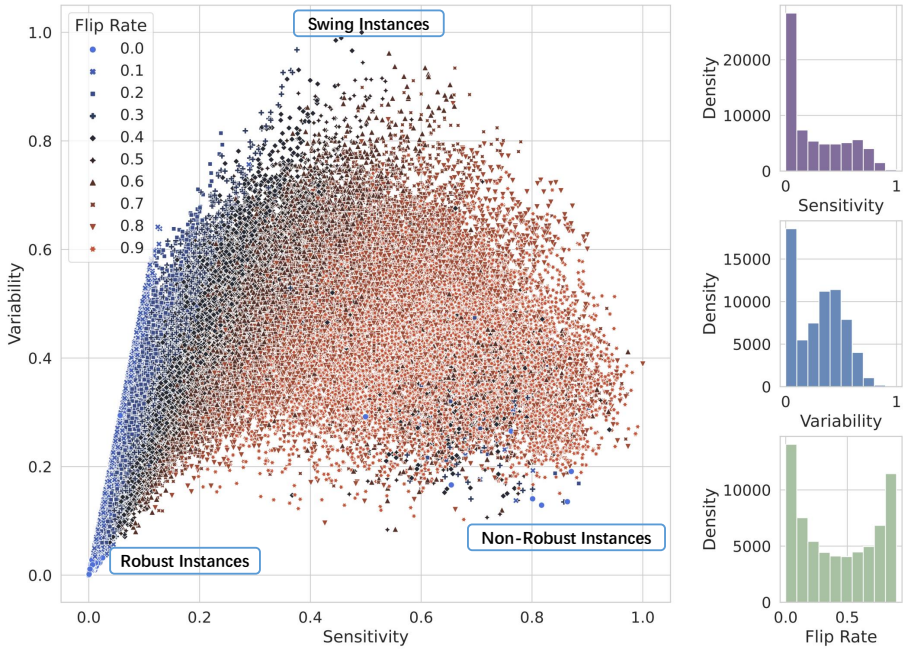


Figure 7: Data map for the AGNEWS train set, based on a BERT-base model. Density plots for the three different measures (sensitivity, variability and flip rate) based on the adversarial loss of each instance during training are shown towards the right. The training instances can be roughly classified into three types: robust instances, non-robust instances and swing instances.

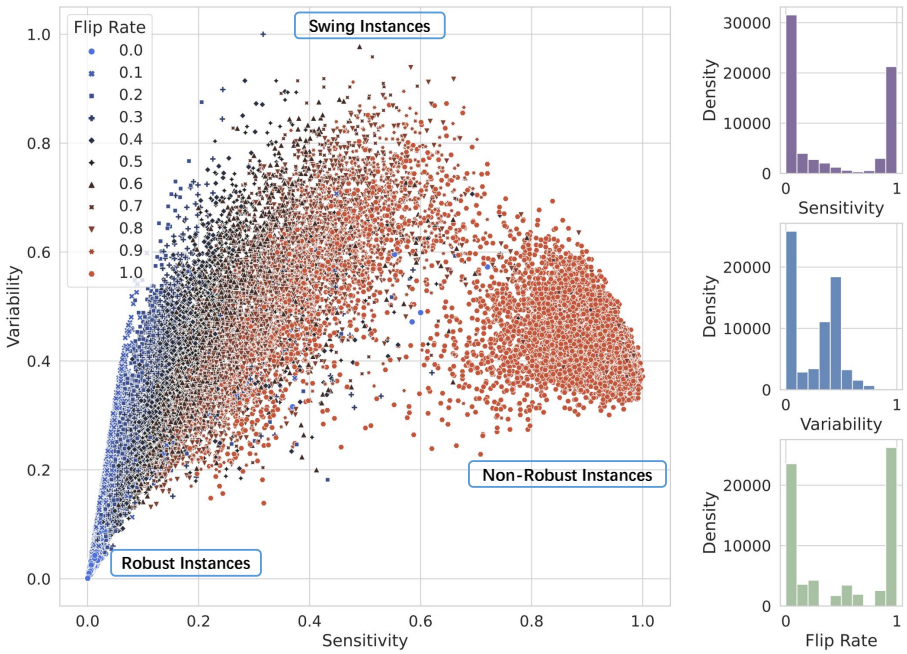


Figure 8: Data map for the QQP train set, based on a BERT-base model. Density plots for the three different measures (sensitivity, variability and flip rate) based on the adversarial loss of each instance during training are shown towards the right. The training instances can be roughly classified into three types: robust instances, non-robust instances and swing instances.