Extreme Confidence and the Illusion of Robustness in Adversarial Training

Anonymous ACL submission

Abstract

Deep learning-based Natural Language Processing (NLP) models are vulnerable to adversarial attacks, where small perturbations can cause a model to misclassify. Adversarial Training (AT) is often used to increase model robustness. Despite the challenging nature of textual inputs, numerous AT approaches have emerged for NLP models. However, we have discovered an intriguing phenomenon: deliberately or accidentally (implicitly as part of existing AT schemes) miscalibrating models such that they are extremely overconfident or underconfident in their predictions, *disrupts* adversarial attack search methods, giving rise to an apparent increase in robustness. However, we demonstrate that the observed gain in robustness is an illusion of robustness (IOR), as an adversary aware of this miscalibration can perform temperature calibration to modify the predicted model logits, allowing the adversarial attack search method to find adversarial examples whereby obviating IOR. Consequently, we urge adversarial robustness researchers to incorporate adversarial temperature scaling approaches into their evaluations to mitigate IOR.

1 Introduction

Deep learning Transformer-based Natural Language Processing (NLP) models are able to perform well in a range of tasks (Manning et al., 2014). However, these NLP models are susceptible to adversarial attacks, where clean input text samples perturbed slightly (accidentally or maliciously by an adversary) can lead to a NLP model misclassifying the perturbed input (Jia and Liang, 2017). However, the emergence of the Adversarial Training (AT) paradigm (Bai et al., 2021) has shown some success in training models to be more robust to these small adversarial perturbations. Here, the traditional training process is adapted to minimize the empirical risk associated with a "robustness loss" as opposed to the risk associated with the standard loss for clean input samples. The robustness loss is the standard loss applied to the worst-case (loss maximizing) adversarial sample for each training sample. In NLP, due to the discrete nature of the text, this adversarial training min-max formulation is particularly challenging as the inner maximization is computationally expensive (Yoo and Qi, 2021). Nevertheless, a variety of approaches have been proposed in literature, ranging from augmentation of the training set with adversarial examples for a specific model, to sophisticated token-embedding space optimizations for the inner maximization step (Wang et al., 2019a; Goyal et al., 2023).

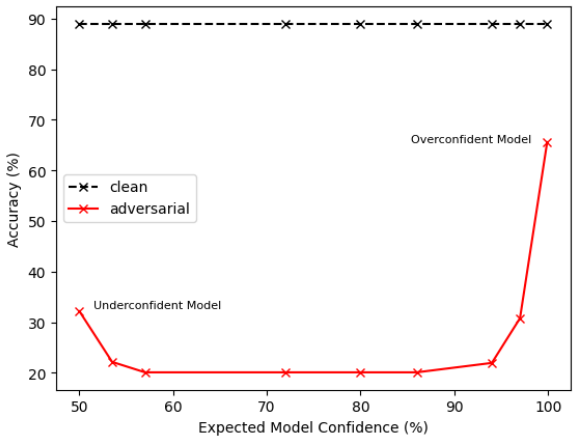

Figure 1: Accuracy on adversarial examples from out-of-the-box adversarial attack for models with different average predicted class confidence, $E_{p(\mathbf{x})}[P_{\hat{\theta}}(\hat{c}|\mathbf{x})]$. Extremely overconfident and underconfident models show increased robustness. We reveal that this increased robustness is an *illusion of robustness*.

Although many NLP AT methods are effective in boosting model robustness, we argue that, in some cases, the increased robustness is an illusion of robustness (IOR). Specifically, highly miscalibrated models, with an extreme predicted class confidence (Guo et al., 2017), present an IOR. This extreme class confidence disrupts out-of-the-box

adversarial attacks' search processes, such that the model appears robust to these out-of-the-box attacks. We identify extreme predicted class confidence as one cause of IOR by reproducing this phenomenon in a controlled manner, intentionally creating highly overconfident and underconfident models. We next demonstrate that this appears to give significant robustness gains against out-of-the-box attacks (up to a three-fold increase in adversarial accuracy). We also demonstrate that AT scheme developers can (unintentionally) develop techniques that cause high model miscalibration and thus also present an IOR — a false sense of security against adversarial attacks. We show that our findings apply to all three commonly used encoder models: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and DeBERTa (He et al., 2020).

Next, we argue that an adversary who is aware of model miscalibration used in this manner, can largely circumvent the model's perceived robustness at inference-time, such that the observed robustness gains no longer persist. We show that test-time temperature calibration approaches can be used for this purpose, and also how an adversary can use a more sophisticated and tailored temperature scaling optimization approach to better pierce a model's IOR. We then demonstrate the effectiveness of our approach at mitigating the IOR, which significantly decreases the adversarial accuracy.

In light of our findings, we urge the adversarial robustness community to adopt optimized temperature scaling approaches in all adversarial robustness evaluations to ensure they accurately reflect a proposed defense's ability to induce robustness.

2 Background

2.1 Adversarial Attacks

An untargeted adversarial attack is able to fool a classification system, $\mathcal{F}()$ with trained parameters $\hat{\theta}$, by perturbing an input sample, $\mathbf{x}$ to generate an adversarial example $\tilde{\mathbf{x}}$ to cause a change in the predicted class,

$$\mathcal{F}(\mathbf{x}; \hat{\theta}) \neq \mathcal{F}(\tilde{\mathbf{x}}; \hat{\theta}). \tag{1}$$

Traditional adversarial attack definitions (Szegedy et al., 2014) require the perturbation to be *imperceptible* as per human perception. In NLP it can be challenging to measure imperceptibility. Following Morris et al. (2020) and Raina and Gales (2023), we can separate modern NLP imperceptibility constraints into two categories: 1) pre-transformation constraints, which limit the changes that can be made to a clean sample $\mathbf{x}$, such that an adversarial example is limited to a specific set of sequences $\tilde{\mathbf{x}} \in \mathcal{A}(\mathbf{x})$; and 2) distance-based constraints, which aim to mathematically limit the distance between the original, clean sample and the adversarial example using a proxy distance measure $\mathcal{G}(\mathbf{x}, \tilde{\mathbf{x}}) \leq \epsilon$.

A plethora of adversarial attack approaches have been proposed for efficiently discovering adversarial examples for NLP models (Alzantot et al., 2018; Garg and Ramakrishnan, 2020; Li et al., 2020; Gao et al., 2018; Wang et al., 2019b; Ren et al., 2019; Jin et al., 2019; Li et al., 2018; Tan and Joty, 2021; Tan et al., 2020). Many of the popular attack approaches are implemented in the TextAttack library (Morris et al., 2020). These attack approaches can be classed as either whitebox attacks, where the adversary has full access to the model parameters or blackbox attacks, where the adversary can only access input-output pairs from the model (Tabassi et al., 2019).

2.2 Traditional Adversarial Training

Standard supervised training methods seek to find model parameters, $\hat{\theta}$ that minimises the empirical risk (for a dataset of $\mathbf{x} \sim p(\mathbf{x})$), characterised by a loss function,

$$\hat{\theta} = \arg\min_{\theta} \mathop{\mathbb{E}}_{\mathbf{x} \sim p(\mathbf{x})} [\mathcal{L}(\mathbf{x}, \theta)]. \tag{2}$$

Adversarial Training (AT) (Goodfellow et al., 2015) adapts the training scheme to minimise the empirical risk associated with the *worst-case* adversarial example, $\tilde{\mathbf{x}}$, such that we are minimising a *robust loss*

$$\hat{\theta} = \arg\min_{\theta} \mathop{\mathbb{E}}_{\mathbf{x} \sim p(\mathbf{x})} \left[\max_{\substack{\tilde{\mathbf{x}}: \\ \mathcal{G}(\mathbf{x}, \tilde{\mathbf{x}},) \leq \epsilon,\ \tilde{\mathbf{x}} \in \mathcal{A}}} \mathcal{L}(\tilde{\mathbf{x}}, \theta) \right]. \tag{3}$$

It is too computationally expensive to perform the inner maximization step to find textual adversarial examples in each step of training. A group of AT methods speed-up this optimization step by finding adversarial examples in the token embedding space, which allows for faster gradient-based approaches: PGD-K (Madry et al., 2018), FreeLB (Zhu et al., 2020), TA-VAT (Li and Qiu, 2020), InfoBERT (Wang et al., 2020). However, limited success of these approaches has been attributed to perturbations in the embedding space

being unrepresentative of real textual adversarial attacks. Hence, AT methods such as Adversarial Sparse Convex Combination (ASCC) (Dong et al., 2021) and Dirichlet Neighborhood Ensemble (DNE) (Zhou et al., 2020) identify a more sensible embedding perturbation space, which they define as the convex hull of word synonyms. Nevertheless, today the simplest and most popular AT approach in NLP is to simply to augment (once) the training set with textual adversarial examples $\tilde{\mathbf{x}}$ for each clean sample $\mathbf{x}$ using standard NLP attack mechanisms on a model trained in the standard manner (Equation 2).

2.3 Model Calibration

Modern deep learning models are often miscalibrated, where the model's confidence in the predicted class does not reflect the ground truth correctness likelihood (Guo et al., 2017). Intuitively, for 100 model predictions with a model confidence of 90%, we should expect 90% of these predictions to be correct. More formally, a model with a predicted class confidence $P_{\hat{\theta}}(\hat{c}|\mathbf{x})$, is defined as perfectly calibrated when

$$P\left(\hat{c} = c^*|P_{\hat{\theta}}(\hat{c}|\mathbf{x}) = p\right) = p, \quad \forall p \in [0,1], \quad (4)$$

where $\hat{c} = \mathcal{F}(\mathbf{x}; \hat{\theta})$ is the predicted class and the true (label) class is c^*. The extent of a model's miscalibration can be visualized on a reliability diagram (Degroot and Fienberg, 1983; Niculescu-Mizil and Caruana, 2005), displaying the sample accuracy as a function of model confidence. Any deviation from an identity function indicates miscalibration. Typical single-value summaries for the calibration error are the Expected Calibration Error (ECE) and the Maximum Calibration Error (MCE) (Naeini et al., 2015).

3 Extreme Predicted Class Confidence

The robustness gains observed for traditional AT approaches (Equation 3), may not always be due to inherent robustness gains, but can be a consequence of a high level of model miscalibration. This miscalibration can induce extreme confidence predictions, such that the model's predicted class confidence $P_{\hat{\theta}}(\hat{c}|\mathbf{x})$ is either very high (overconfident) or very low (underconfident). Figure 1 (using a standard NLP model, test dataset and adversarial attack described in Section 5) demonstrates that highly miscalibrated models with extreme confidence values in the predicted class (around 1.0 for overconfident models or $1/C$, with C as the number of classes for underconfident models) are significantly more robust to out-of-the-box adversarial attacks.

The apparent increase in robustness of extremely miscalibrated models can be explained. For both underconfident and overconfident models, the predicted class confidence has very little variance for different input sequences, $\mathbf{x}$,

$$E_{p(\mathbf{x})}[P_{\hat{\theta}}(\hat{c}|\mathbf{x}) - \mathbb{E}_{p(\mathbf{x})}[P_{\hat{\theta}}(\hat{c}|\mathbf{x})]]^2 < \zeta, \quad (5)$$

where ζ is some small variance. The narrow confidence distribution makes it challenging for an adversary to identify an appropriate search direction for adversarial examples. To illustrate this, consider a miscalibrated model with extremely high confidence in the predicted class probability, $P_{\hat{\theta}}(\hat{c}|\mathbf{x}) \approx 1.0$, then for most search directions $\mathbf{d}$ that are not in an adversarial direction $\mathbf{d} \neq \tilde{\mathbf{d}}$ (where $\tilde{\mathbf{x}} = \mathbf{x} + \tilde{\mathbf{d}}$) the model has very little sensitivity,[1] i.e.,

$$\mathbf{d}^T \nabla_{\mathbf{x}} P_{\hat{\theta}}(\hat{c}|\mathbf{x}) \approx 0. \quad (6)$$

As a consequence of this little sensitivity, any white-box adversarial attack approach looking to exploit gradients or even a blackbox attack approach measuring the sensitivity of the predicted probability, has a small confidence range to observe, meaning that the impact of any proposed perturbation gives a very *noisy* signal to its actual effect on the output. As a result, the adversarial attack search process will converge extremely slowly or fail to find the desired adversarial perturbation direction $\tilde{\mathbf{d}}$. This hypothesis is verified empirically in Appendix E.

In this work, to demonstrate that extreme confidence can cause an apparent increase in robustness against of-the-shelf adversarial attacks, we consider models that are explicitly induced with overconfidence or underconfidence (Section 3.1). We further demonstrate that standard AT approaches can also implicitly induce extreme confidence and also cause an apparent increase in robustness (Section 3.2). Section 4 shows that this increase in robustness is an illusion of robustness (IOR).

3.1 Explicit: Temperature Scaling

Let $\hat{\theta}$ be a model trained using the standard training objective, as in Equation 2. For this model with

[1] Note that these strict mathematical operations are not defined for the input text space and are simply representative of equivalent discrete textual space perturbations.

predicted logits, $l_1, \ldots, l_C$ for C output classes, the probability of a specific class is typically estimated by the Softmax function,

$$P_{\hat{\theta}}(c|\mathbf{x}) = \frac{\exp(l_c)}{\sum_i \exp(l_i)}. \quad (7)$$

However, we can intentionally miscalibrate the model and increase the model confidence at *inference time* by using a design temperature, $T = T_d$, to scale the predicted logits,

$$P_{\hat{\theta}}(c|\mathbf{x}; T) = \frac{\exp(l_c/T)}{\sum_i \exp(l_i/T)}. \quad (8)$$

A design choice of $T_d \ll 1.0$ concentrates the probability mass in the largest logit class to create an *overconfident* model, whilst conversely $T_d \gg 1.0$ creates an *underconfident* model. Hence, explicitly setting a design temperature $T^{(d)}$ at inference time can be used to serve highly miscalibrated models, which can disrupt an adversary's attack search process as described in Equation 6, whilst maintaining the simplicity of the standard training objective (Equation 2).

3.2 Implicit Overconfidence: DDi AT

Section 3.1 presents an explicit temperature scaling method to generate a highly miscalibrated system, which cause an illusion of robustness for out-of-the-box adversarial attacks. However, it is possible that implementation strategies and algorithmic features in adversarial training (AT) procedures (Equation 3) can also lead to inherently overconfident models. We now consider adversarial training techniques that implicitly induce model overconfidence.

Implicit overconfidence can be demonstrated first with the incorporation of the recently proposed Danskin Descent Direction (DDi; Latorre et al., 2023) into an AT approach. Latorre et al. (2023) adapted the standard AT paradigm of Equation 3 to identify optimal gradient update directions for increased model robustness, showing promising results in computer vision. In Appendix A, we detail how the DDi algorithm can be used to compute gradients while adversarially training NLP classifiers. In our experiments, we observe (Table 1) that the DDi gradients applied in AT for NLP classifiers induces highly overconfident models without compromising on clean accuracy, such that a model that has undergone DDi-AT almost always predicts near 100% confidence in its predicted class, $P_{\hat{\theta}}(c|\mathbf{x}) \approx 1.0$. Our ablations (Appendix B) reveal that the gradient normalization step in the DDi algorithm (Equation 12) is responsible for the induction of inherent model overconfidence. Hence, we further consider other standard AT schemes that may use gradient normalization during training. Specifically, we consider Project Gradient Descent (PGD) and Adversarial Sparce Convex Combination (ASCC), introduced in Section 2.2. Table 1 and the discussion in Appendix B demonstrate that these AT schemes also yield highly overconfident systems, and are thus at risk of IOR: they appear robust to adversarial attacks by disrupting the search process (Equation 6).

4 Piercing the Illusion

Section 3 demonstrates how intentional or accidental extreme miscalibration of a model can create extreme confidence distributions that disrupt out-of-the-box adversarial attack search methods and thus give an apparent gain in robustness. This section highlights that the observed gains in robustness are an illusion of robustness (IOR), as we propose simple approaches that an adversary can use to mitigate extreme model confidences to remove the disruption to the attack search methods.

The following approaches require an adversary to modify aspects of the output of the model to mitigate the disruption to an attack search process. Note that these modifications are only used by the adversary to create/find adversarial examples, which can then be applied to the original (unmodified) model served by the model developer.

4.1 Adversary Temperature Calibration

Highly miscalibrated models, such as the design of overconfident models in Section 3, interfere with adversarial attacks from finding meaningful search directions due to the little sensitivity in the predicted probabilities. An adversary aims to mitigate this disruption to the attack search process. The simplest solution for an adversary is to calibrate the model so that the confidences are in a sensible range and can be exploited by adversarial attacks.

A strong indicator of model miscalibration (Section 2.3) can be given by the Negative Log Likelihood (NLL; Hastie et al., 2017). Thus, assuming an adversary has access to the output model logits $l_1, \ldots, l_C$ and a labelled validation set of data $\{\mathbf{x}_i, c_i^*\}_i$, test-time temperature calibration (Guo et al., 2017) can be applied.[2] Here the adversary

[2]Note that the logits received by an adversary may already have been explicitly scaled by a model designer to intention-

optimizes an adversarial temperature, T_a to minimize the Negative Log Likelihood (NLL) of the validation set samples,

$$T_a = \arg\min_T \sum_i -\log P_{\hat{\theta}}(c_i^*|\mathbf{x}_i; T), \quad (9)$$

where $P_{\hat{\theta}}(c^*|\mathbf{x}; T)$ is the confidence of the true class after temperature scaling as in Equation 8. Due to the continuous nature of the transformation and the need to optimize a single parameter, T_a, in this work we use the standard gradient descent optimization.[3]

Other than temperature optimization, an adversary can attempt other post-training model calibration approaches such as Histogram Binning (Zadrozny and Elkan, 2001), isotonic regression (Zadrozny and Elkan, 2002) and multi-class versions of Platt scaling (Niculescu-Mizil and Caruana, 2005; Platt and Karampatziakis, 2007). However, temperature calibration is found to be the most practical and effective for an adversary seeking to mitigate a model's IOR. A more detailed discussion is presented in Appendix D.5.

4.2 Adversary Temperature Optimization

Section 4.1 outlines a temperature calibration approach an adversary can use to mitigate the disruption to out-of-the-box adversarial attack methods. However, this approach has two shortcomings:

1. The adversarial temperature, T_a is not directly tuned to minimize adversarial robustness, as it only considers the likelihood of clean examples in a validation set.

2. Learning the adversarial temperature, T_a to minimize the NLL (Equation 9) uses a gradient descent based optimization algorithm where the stability of the algorithm is sensitive to hyperparameters and does not guarantee an optimal solution.

Hence, this section outlines an algorithm that directly optimizes the adversarial temperature T_a to minimize a model's adversarial robustness. We define the adversarial accuracy, $\mathcal{Q}()$ as a function of the temperature parameter,

$$\mathcal{Q}(T) = \frac{1}{J}\sum_j \mathbb{I}\left[\mathcal{F}(\tilde{\mathbf{x}}_j(T)) = c_j^*\right], \quad (10)$$

where $\tilde{\mathbf{x}}_j(T)$ represents the adversarial example generated from an adversarial attack on the given model, $\hat{\theta}$ with the logits scaled by a temperature T as in Equation 8. Figure 1 illustrates that as the temperature parameter is swept from large to small values (increasing model confidence), the adversarial accuracy, $\mathcal{Q}()$ behaves almost as a convex function of temperature, T, such that, $\mathcal{Q}(\alpha T_1 + (1-\alpha)T_2) \leq \alpha\mathcal{Q}(T_1) + (1-\alpha)\mathcal{Q}(T_2)$, where $0 \leq \alpha \leq 1$. The optimal adversarial temperature T_a is the minimizer of the adversarial accuracy $\mathcal{Q}(T)$,

$$T_a = \arg\min_T \mathcal{Q}(T). \quad (11)$$

The minimizer, T_a can be found efficiently over the non-differentiable convex function, $\mathcal{Q}()$ using a search method such as the Golden-section search algorithm (Kiefer, 1953). In this work we use the Brent-Dekker method, an extension of Golden-section search that accounts for a potentially parabolic convergence point (Brent, 1971).

Note, as is the case for the calibration approach of Section 4.1, to optimize for T_a, an adversary is not required to query the target model multiple times as the adversary only requires the output model logits $l_1, \ldots, l_C$.

Although the temperature optimization approach in this section offers an adversarial temperature T_a optimized for adversarial robustness, the search method is significantly slower than the gradient descent approach for calibration on a clean (not adversarially attacked samples) validation set (Equation 9). The greatest computational cost can be attributed to calculation of the adversarial accuracy (Equation 10), as this requires an adversarial attack to be applied to each clean sample in the validation set, $\{\mathbf{x}_j, c_j^*\}_{j=1}^J$. Therefore, we recommend that by default, to pierce the IOR, one should adopt the calibration approach of Equation 9, but when there is access to greater computational resources Equation 11 should be followed.

5 Experiments

We first demonstrate how explicit or implicit training approaches that cause a model to become highly underconfident or overconfident (miscalibrated) suffer from an *illusion of robustness* (IOR), where the models appear robust to out-of-the-box adversarial attacks. We then show how simple approaches can be used to pierce this illusion.

ally miscalibrate the system as in Section 3.1.

[3]The optimization method is inspired by https://github.com/gpleiss/temperature_scaling/tree/master.

5.1 Experimental Setup

Data. Experiments are carried out on three standard NLP classification datasets. First, Rotten Tomatoes (Pang and Lee, 2005) is a binary sentiment classification task for movie reviews, consisting of 8530 training, 1066 validation and 1840 test samples. Next, we consider the Twitter Emotions Dataset (Saravia et al., 2018), which categorizes tweets into one of six emotions: love, joy, surprise, fear, sadness or anger, with a total of 16,000 training, 2000 validation and 2000 test samples. Finally we consider the popular AGNews dataset (Zhang et al., 2015), consisting of articles from 2000 news sources classified into one of four topics: business, sci/tech, world or sports. There are a combined 120,000 training samples and 7600 test samples. **For readability we present the results in the main paper for the Rotten Tomatoes dataset, with the equivalent results presented for the other datasets in Appendix D.1.** The same general trends are observed across the different datasets.

Models. Transformer-encoder models (Vaswani et al., 2017) give state-of-the-art performance on many NLP classification tasks. Hence, in this work we perform experiments with three Transformer-encoder base models (110M parameters). Specifically, we consider DeBERTa (He et al., 2020), RoBERTa (Liu et al., 2019) and BERT (Devlin et al., 2019). **The results in the main paper are presented for the Deberta model with equivalent results presented for the other models in Appendix D.2.** Identical trends are observed for all the models. Hyperparameter settings for training of these models are given in Appendix C. All experiments are run over three random seeds.

Adversarial attacks. We consider four popular out-of-the-box adversarial attack approaches in this work. Bert Adversarial Example (bae) (Garg and Ramakrishnan, 2020) is included as a word-level blackbox attack, where the adversary has only access to the model inputs and predictions. Next, we include the more powerful Textfooler (tf) (Jin et al., 2019) and Probability Weighted Word Saliency (pwws) (Ren et al., 2019) word-level attacks. Finally, we include the DeepWordBug (dg) (Gao et al., 2018) attack as a whitebox, *character*-level adversarial attack approach. Each adversarial attack is implemented with the default settings from TextAttack (Morris et al., 2020). To evaluate the impact of the different adversarial attacks we report the *adversarial accuracy*, which is the accuracy of the target model on adversarial examples.

AT Approaches. To demonstrate the risk of IOR we consider a range of standard AT methods. As described in Section 2.2, we first consider the Danskin Descent Direction (DDi; Latorre et al., 2023), which we show generates inherent overconfidence. We further consider PGD-K (Madry et al., 2018) and FreeLB (Zhu et al., 2020) as embedding-space AT schemes and ASCC (Dong et al., 2021) as a text-embedding combined AT approach. Finally, we consider the most popular NLP AT approach: simple augmentation of the training set with adversarial examples. In this work, to generate these adversarial examples the target model is trained in the standard manner (Equation 2) and DeepWordBug is used to attack the trained model, such that an adversarial example is found for each clean training sample. The target model architecture is then re-trained (as per Equation 2) on the training set augmented with the generated adversarial examples. Hence, for the augmentation-based AT model, DeepWordBug can be viewed as a *seen* attack and the remaining attacks as *unseen*. It would be expected that the model is relatively more robust to *seen* attacks. Hyperparameters for each individual AT baseline method are given in Appendix C.

5.2 Creating the Illusion

To illustrate the IOR, Section 3 proposes that highly miscalibrated systems with extreme predicted class confidences can be created explicitly by temperature scaling (Section 3.1). However, IOR can manifest for AT schemes that implicitly induce model miscalibration (Section 3.2). To verify this, we consider a standard model (*std*) trained in the standard manner (Equation 2). After the model is trained, we create two new versions of the *std* model using explicit design temperature scaling (Equation 8): a highly underconfident model (↓conf) with $T_d = 2000000$ and a highly overconfident model (↑conf) with $T_d = 0.005$. To demonstrate how AT schemes can implicitly create overconfidence, we include DDi-AT (ddi-at), PGD AT (pgd*), and ASCC AT (ascc*), where * indicates that gradient normalization is used during training.[4]

Table 1 verifies that models ↑conf, ddi-at, pgd* and ascc* are significantly more confident than the

[4]Appendix B shows that gradient normalization during training can implicitly lead to overconfidence.

std model, whilst the ↓conf model is far less confident, as intended. The differences in the confidence are more prominent for the adversarial examples (pwws is used to attack the test set). The clean accuracy on the test data is the same or similar to that of the *std* model.

Model	clean	$\bar{P}(\hat{c}\|\mathbf{x}_{\text{clean}})$	$\bar{P}(\hat{c}\|\mathbf{x}_{\text{adv}})$
std	88.96 ±0.30	97.08 ±0.26	86.04 ±0.68
↓conf	88.96 ±0.30	50.00007 ±0.00	50.00004 ±0.00
↑conf	88.96 ±0.30	99.98 ±0.02	99.95 ±0.01
ddi-at	87.90 ±0.49	99.97 ±0.03	99.91 ±0.01
pgd*	88.36 ±0.68	99.96 ±0.04	99.90 ±0.01
ascc*	87.80 ±0.42	99.97 ±0.04	99.92 ±0.01

Table 1: Clean accuracy (%) and model confidence (%) on clean and adversarial (pwws) examples for extreme confidence systems: high confidence (↑conf), low confidence (↓conf), ddi-at, pgd* and ascc*.

Table 2 presents the adversarial robustness of each model as measured by the adversarial accuracy under the different out-of-the-box adversarial attacks. For comparison, we include the AT approaches (aug, pgd, ascc, freelb), which have been designed to **not** be overconfident by removing gradient normalization during training (Appendix D.4). In general, the baseline AT approaches (aug, pgd, ascc, freelb) do increase model robustness across all the different attack methods, with the augmentation approach being the most effective. The low confidence model also demonstrates comparable adversarial robustness to the augmentation-based approach. However, the highly overconfident models (↑conf, ddi-at, pgd*, ascc*) indicate a significantly higher (two/three-fold increase) adversarial robustness relative to the other AT approaches.

5.3 Piercing the Illusion

We argue that the apparent increase in adversarial robustness of the extreme confidence models (↓conf, ↑conf, ddi-at, pgd*, ascc*) in Table 2 is due to the out-of-the-box attack search process being disrupted,[5] i.e. the models are actually susceptible to adversarial examples[6] but the adversarial attacks are unable to find these adversarial examples.

[5]Appendix E empirically shows that extreme confidence results in a noisier search for regular adversarial attacks.

[6]We know this must be true for the temperature-scaled models as the predicted class for any input for these models is identical to the *std* model.

Method	clean	bae	tf	pwws	dg
std	88.96 ±0.30	31.39 ±1.20	17.82 ±0.49	20.42 ±0.62	20.11 ±0.94
↓conf (§3.1)	88.96 ±0.30	31.21 ±0.94	20.98 ±0.99	25.17 ±0.89	32.18 ±2.78
↑conf (§3.1)	88.96 ±0.30	37.71 ±1.18	54.35 ±0.73	59.29 ±0.62	65.60 ±1.81
ddi-at (§3.2)	87.90 ±0.49	39.18 ±0.75	56.54 ±1.67	61.07 ±0.99	66.73 ±1.01
pgd*	88.36 ±0.68	39.94 ±0.55	58.02 ±1.04	64.45 ±0.77	67.02 ±0.83
ascc*	87.80 ±0.42	40.01 ±0.69	54.32 ±1.57	63.99 ±0.86	67.43 ±0.93
aug	87.12 ±0.39	34.74 ±1.59	22.36 ±1.83	26.11 ±2.57	37.43 ±0.75
pgd	88.24 ±0.73	33.65 ±0.57	19.92 ±0.47	26.70 ±0.87	26.05 ±0.61
ascc	87.77 ±0.36	33.61 ±0.64	15.13 ±2.17	23.50 ±0.77	26.80 ±2.11
freelb	88.74 ±0.32	32.52 ±0.52	19.51 ±1.70	24.55 ±0.70	24.52 ±0.73

Table 2: Accuracy (%) of extreme confidence systems compared to standard AT methods on out-of-the-box adversarial attacks.

Hence, the observed robustness is an IOR.

In Section 4, we presented two simple approaches an adversary could employ to mitigate the disruption of the adversarial attack search processes and remove the IOR. First, temperature calibration (*cal*) can be applied to the trained model to learn an adversarial calibrating temperature T_a. This temperature is learnt by minimizing the NLL on the validation data (Equation 9) with a gradient-descent based optimizer. The learning rate is set to 0.01 with a maximum of 5000 iterations. Alternatively, the adversary can optimize the temperature T_a (*opt*) by accounting for the adversarial examples for a validation set (Equation 11). Here, DeepWordBug is used to attack the validation set to optimize for T_a. For both approaches, the target model is modified by scaling the predicted logits by T_a and then the out-of-the-box adversarial attacks are run on the modified model to find adversarial examples. These adversarial examples are evaluated on the original, unmodified model. Table 3 shows the impact of the different adversarial approaches (*cal* and *opt*) to learn T_a on the adversarial robustness of the models. For the overconfident models, ↑conf, ddi-at, pgd* and ascc*, simple temperature calibration (cal) is sufficient to cause a significant drop in model robustness. For the low confidence model, the more computationally expensive temperature optimization approach (opt) is necessary to significantly reduce model robustness. This demonstrates that an adversary can remove the IOR of highly miscalibrated systems by optimizing for the adver-

sarial scaling temperature T_a.[7]

Method	Adv.	clean	bae	tf	pwws	dg
std	-	88.96 ±0.30	31.39 ±1.20	17.82 ±0.49	20.42 ±0.62	20.11 ±0.94
↓conf	-	88.96 ±0.30	31.21 ±0.94	20.98 ±0.99	25.17 ±0.89	32.18 ±2.78
	cal	88.96 ±0.30	31.52 ±0.34	21.89 ±0.43	27.58 ±1.31	31.52 ±0.34
	opt	88.96 ±0.30	31.44 ±1.15	17.82 ±0.49	20.86 ±0.64	21.98 ±1.66
↑conf	-	88.96 ±0.30	37.71 ±1.18	54.35 ±0.73	59.29 ±0.62	65.60 ±1.81
	cal	88.96 ±0.30	31.39 ±1.20	17.82 ±0.49	20.45 ±0.74	21.64 ±1.46
	opt	88.96 ±0.30	31.39 ±1.20	17.82 ±0.49	20.90 ±0.94	21.06 ±0.82
ddi-at	-	87.90 ±0.49	39.18 ±0.75	56.54 ±1.67	61.07 ±0.99	66.73 ±1.01
	cal	87.90 ±0.49	31.80 ±0.57	18.36 ±3.01	23.08 ±1.96	22.89 ±3.38
	opt	87.90 ±0.49	31.80 ±0.57	18.88 ±3.32	22.16 ±1.03	22.28 ±1.12
pgd*	-	88.36 ±0.68	39.94 ±0.55	58.02 ±1.04	64.45 ±0.77	67.02 ±0.83
	cal	88.36 ±0.68	33.64 ±0.61	19.95 ±1.02	26.78 ±0.73	26.22 ±0.69
ascc*	-	87.80 ±0.42	40.01 ±0.69	54.32 ±1.57	63.99 ±0.86	67.43 ±0.93
	cal	87.80 ±0.42	33.53 ±0.78	16.22 ±2.54	23.78 ±0.75	26.90 ±1.54

Table 3: Clean and adversarial accuracy (%) for the adversarial mitigation of the *Illusion of Robustness* of highly miscalibrated systems with temperature calibration (*cal*) or optimized temperature scaling (*opt*).

It is apparent that there is the risk that proposed AT approaches, such as with the naive use of the DDi gradients within AT or the use of gradient normalization (e.g. pgd* and ascc*), can give the illusion of robustness when in reality these approaches do not give inherently robust models. However, it can perhaps be argued that to expose this weakness it may not be necessary for an adversary to modify the model with adversarial temperature scaling to find adversarial examples. Instead, adversarial examples can be found for another model (e.g., *std*) and directly transferred to the target model. This follows from Demontis et al. (2018) where it is shown that similar architectures can be susceptible to the same adversarial examples. This is explored in Table 4, where adversarial examples are found for the *source* model and evaluated on the *target* model. It is clear from these results that although the transfer attack from *std* to *ddi-at* is effective in reducing the adversarial accuracy, it is unable to bring the adversarial accuracy down to the values for *std*, as is achieved by the temperature optimization approaches in Table 3.

[7]Appendix D.3 discusses the relationship between the calibration error and the model confidence.

tgt	src	clean	bae	tf	pwws	dg
std	std	88.96 ±0.30	31.39 ±1.20	17.82 ±0.49	20.42 ±0.62	20.11 ±0.94
ddi-at	ddi-at	87.90 ±0.49	39.18 ±0.75	56.54 ±1.67	61.07 ±0.99	66.73 ±1.01
ddi-at	std	87.90 ±0.49	48.91 ±0.60	52.47 ±1.15	50.00 ±1.64	48.53 ±0.99

Table 4: Transferability: adversarial examples for each attack method are generated for the source model and adversarial accuracy (%) is given for the target model.

Overall, these results demonstrate that highly miscalibrated systems can appear robust to out-of-the-box attack methods by disrupting adversarial attack search processes. However, in reality this robustness is an *illusion* as simple modifications can mitigate the disruption of the search process. Therefore, we encourage future work in adversarial robustness to incorporate model calibration or temperature optimization at test-time to ensure that any proposed AT schemes do not unintentionally include underlying mechanisms that cause extreme miscalibration and thus present an IOR, giving a false sense of security.

6 Conclusion

Modern NLP models are susceptible to adversarial attacks, where small changes in the input cause the model to predict the incorrect class. A range of Adversarial Training (AT) approaches have been proposed to encourage model robustness to adversarial attacks. However, the observed robustness gains may not be entirely due to inherent model robustness gains. In this work, we demonstrate that AT schemes can unknowingly (or intentionally) create highly miscalibrated models, such that the predicted class confidence is extreme. The extreme confidence in the class prediction disrupts out-of-the-box adversarial attack search methods, giving observed gains in robustness. However, this is an *illusion of robustness* (IOR). We propose simple approaches an adversary could use to mitigate such robustness gains. Specifically, we demonstrate that various optimized temperature scaling approaches can reduce the extremity of the class confidence, which mitigates the disruption to the adversarial attack search processes, obviating the IOR. Therefore, we recommend that future adversarial robustness evaluation frameworks incorporate adversarial temperature scaling at test-time to ensure that any observed robustness is genuine and not an *illusion*.

7 Limitations

This work demonstrates that a model developer can create an illusion of robustness (IOR) to adversarial attacks by serving highly miscalibrated systems. An aware adversary can mitigate the IOR by performing targeted temperature calibration at inference time. The following limitations have been identified for this work:

- Empirical results are presented for state-of-the-art encoder-based Transformer models. However, recently with the rise of generative models, classification tasks are being approached with the use of decoder-based models. Although many of the out-of-the-box adversarial attack approaches cannot be applied directly to decoder models, it would be useful to investigate how susceptible decoder models are to the IOR.

- In this work we consider popular Adversarial Training (AT) baselines the IOR. However, future work would benefit from considering other recently proposed alternative approaches for adversarial robustness, e.g., contrastive learning based approaches (Rim et al., 2021) and Textual Manifold Defence (Nguyen Minh and Luu, 2022), where all inputs are mapped to a robust manifold. It would be interesting to also explore to what extent these proposed approaches are offering true robustness and to what extent they may be unknowingly creating an IOR.

8 Risks and Ethics

This work presents results on the topic of adversarial training. The contributions in this work encourage the development of truly robust systems and therefore there are no identified ethical concerns.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. pages 2890–2896.

Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. 2021. Recent advances in adversarial training for adversarial robustness. *CoRR*, abs/2102.01356.

R. P. Brent. 1971. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4):422–425.

Morris H. Degroot and Stephen E. Fienberg. 1983. The comparison and evaluation of forecasters. *The Statistician*, 32:12–22.

Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. 2018. On the intriguing connections of regularization, input gradients and transferability of evasion and poisoning attacks. *CoRR*, abs/1809.02861.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186.

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2021. Towards robustness against natural language word substitutions. *CoRR*, abs/2107.13541.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. *CoRR*, abs/1801.04354.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples.

Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran. 2023. A survey of adversarial defences and robustness in nlp.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. pages 1321–1330.

Trevor Hastie, Jerome Friedman, and Robert Tisbshirani. 2017. *The elements of Statistical Learning: Data Mining, Inference, and prediction*. Springer.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

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. 2019. Is BERT really robust? natural language attack on text classification and entailment. *CoRR*, abs/1907.11932.

J. Kiefer. 1953. Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society*, 4(3):502–506.

Fabian Latorre, Igor Krawczuk, Leello Tadesse Dadi, Thomas Pethick, and Volkan Cevher. 2023. Finding actual descent directions for adversarial training. In *The Eleventh International Conference on Learning Representations*.

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

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. pages 6193–6202.

Linyang Li and Xipeng Qiu. 2020. Textat: Adversarial training for natural language understanding with token-level perturbation. *CoRR*, abs/2004.14543.

Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021a. Searching for an effective defender: Benchmarking defense against adversarial word substitution.

Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021b. Searching for an effective defender: Benchmarking defense against adversarial word substitution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3137–3147, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

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*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2019. Towards deep learning models resistant to adversarial attacks.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

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.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:2901–2907.

Dang Nguyen Minh and Anh Tuan Luu. 2022. Textual manifold-based defense against natural language adversarial examples. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6612–6625, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 625–632, New York, NY, USA. Association for Computing Machinery.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. 2012. Understanding the exploding gradient problem. *CoRR*, abs/1211.5063.

John Platt and Nikos Karampatziakis. 2007. Probabilistic outputs for svms and comparisons to regularized likelihood methods.

Vyas Raina and Mark Gales. 2023. Sample attackability in natural language adversarial attacks.

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.

Daniela N. Rim, DongNyeong Heo, and Heeyoul Choi. 2021. Adversarial training with contrastive learning in nlp.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks.

Elham Tabassi, Kevin J. Burns, Michael Hadjimichael, Andres Molina-Markham, and Julian Sexton. 2019. A taxonomy and terminology of adversarial machine learning.

Samson Tan and Shafiq Joty. 2021. Code-mixing on sesame street: Dawn of the adversarial polyglots. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3596–3616, Online. Association for Computational Linguistics.

Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. 2020. It's morphin' time! Combating linguistic discrimination with inflectional perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2920–2935, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2020. Infobert: Improving robustness of language models from an information theoretic perspective. *CoRR*, abs/2010.02329.

William Yang Wang, Sameer Singh, and Jiwei Li. 2019a. Deep adversarial learning for NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 1–5, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaosen Wang, Hao Jin, and Kun He. 2019b. Natural language adversarial attacks and defenses in word level. *CoRR*, abs/1909.06723.

Jin Yong Yoo and Yanjun Qi. 2021. Towards improving adversarial training of NLP models. *CoRR*, abs/2109.00544.

Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 609–616, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 694–699, New York, NY, USA. Association for Computing Machinery.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2020. Defense against adversarial attacks in NLP via dirichlet neighborhood ensemble. *CoRR*, abs/2006.11627.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding.

A Danksin's Descent Direction for NLP

A.1 Original Theory

Latorre et al. (2023) demonstrate that the standard formulation and implementation of AT (as in Equation 3) is potentially flawed. Specifically, solving the inner maximization to find the *worst-case* adversarial example $\tilde{\mathbf{x}}$, can give a gradient direction (in standard stochastic gradient descent approaches), that can in fact *increase* the robust loss (the new worst-case adversarial example, $\tilde{\mathbf{x}}$, with the updated model parameters, θ, can give a robust loss that is greater than before the update step), i.e. worsening the adversarial robustness of the model. This flaw is attributed to the reliance on a single adversarial example, as a parameter gradient step to reduce the model's sensitivity to a particular adversarial example does not guarantee reduction in the model's sensitivity to all adversarial examples (the model may now be less robust to other adversarial examples) for a specific sample $\mathbf{x}$. The paper argues that their exist multiple solutions to the inner-maximization for the robust loss and the optimal parameter gradient direction depends on all of those solutions. Thus, Equation 3 can theoretically be adapted to selecting the adversarial example that maximises the gradient direction in each gradient update step for a batch size of K samples,

$$\theta_{i+1} = \Phi\left(\theta_i, \boldsymbol{\gamma}^* = -\frac{\nabla_\theta g(\mathbf{x}_{1:K}, \theta_i, \hat{\tilde{\mathbf{x}}}_{1:K})}{||\nabla_\theta g(\mathbf{x}_{1:K}, \theta_i, \hat{\tilde{\mathbf{x}}}_{1:K})||_2}\right),$$
$$g(\mathbf{x}_{1:K}, \theta_i, \hat{\tilde{\mathbf{x}}}_{1:K}) = \frac{1}{K}\sum_k \mathcal{L}(\hat{\tilde{\mathbf{x}}}_k, \theta_i),$$
$$\hat{\tilde{\mathbf{x}}}_k = \underset{\tilde{\mathbf{x}} \in \mathcal{S}^*(\theta_i, \mathbf{x}_k)}{\arg\max} ||\nabla_{\theta=\theta_i}\mathcal{L}(\tilde{\mathbf{x}}, \theta)|\,|_2, \tag{12}$$

where $\Phi(\theta, \boldsymbol{\gamma})$ is the first-order stochastic gradient descent (SGD) algorithm used to update θ as per descent direction $\boldsymbol{\gamma}$, e.g. in standard SGD, $\Phi(\theta, \gamma) = \theta + \beta\boldsymbol{\gamma}$, where β is the step-size (learning rate). Further $S^*(\theta_i, \mathbf{x}_k)$ represents the set of all maximizers of the robust loss,

$$S^*(\theta, \mathbf{x}, \mathcal{G}) = \underset{\substack{\tilde{\mathbf{x}}: \\ \mathcal{G}(\mathbf{x}, \tilde{\mathbf{x}},) \leq \epsilon,\ \tilde{\mathbf{x}} \in \mathcal{A}}}{\arg\max} \mathcal{L}(\tilde{\mathbf{x}}, \theta). \tag{13}$$

This set of (robust loss) maximizers, $S^*(\theta, \mathbf{x}, \mathcal{G})$ can theoretically be infinite. However, if assume we have access to a finite set with M adversarial examples, such that they define,

$$S^{*(M)}(\theta, \mathbf{x}) = \{\tilde{\mathbf{x}}^{(1)}, \ldots, \tilde{\mathbf{x}}^{(M)}\}, \tag{14}$$

then Latorre et al. (2023) propose an efficient algorithm termed, Danskin's Descent Direction (DDi), that provides a method to approximate the steepest direction, $\boldsymbol{\gamma}^*$ as though as if we are still selecting from the infinite set S^* [8], despite only having access to $S^{*(M)}$. The optimization problem over an infinite set in Equation 12 can be solved by finding an optimal linear combination, $\boldsymbol{\alpha} \in \triangle^M$ of the gradients of the loss, $\nabla_\theta g$ for each different adversarial example. Note that $\triangle^M$ defines the M-dimensional simplex (on which $\boldsymbol{\alpha}$ lies). If we let $\nabla_\theta g(\theta, S^{*(M)}_{1:K}(\theta))$ be the matrix with columns $\nabla_\theta g(\mathbf{x}_{1:K}, \theta_i, \tilde{\mathbf{x}}^{(m)}_{1:K}))$ for $m = 1, \ldots, M$, then

$$\boldsymbol{\gamma}^* = -\frac{\nabla_\theta g(\theta, S^{*(M)}_{1:K}(\theta))\boldsymbol{\alpha}^*}{||\nabla_\theta g(\theta, S^{*(M)}_{1:K}(\theta))\boldsymbol{\alpha}^*||_2},$$
$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha} \in \triangle^M}{\arg\min} ||\nabla_\theta g(\theta, S^{*(M)}_{1:K}(\theta))\boldsymbol{\alpha}||_2^2. \tag{15}$$

A.2 DDi-AT for NLP classification

The challenge with NLP is that generating strong textual adversarial examples as per Equation 14 can be extremely slow. Hence to increase speed, we generate adversarial examples in the token embedding space, such that we follow Equation 15, but adapt Equation 12 to,

$$g(\mathbf{x}_{1:K}, \theta_i, \hat{\tilde{\mathbf{h}}}_{1:K}) = \frac{1}{K}\sum_k \mathcal{L}(\hat{\tilde{\mathbf{h}}}_k, \theta_i),$$
$$\hat{\tilde{\mathbf{h}}}_k = \underset{\tilde{\mathbf{h}} \in \mathcal{S}^*(\theta_i, \mathbf{h}_k)}{\arg\max} \left|\left|\nabla_{\theta=\theta_i}\mathcal{L}(\tilde{\mathbf{h}}, \theta)\right|\right|_2, \tag{16}$$

where $\mathbf{h}_k = \{\mathbf{h}_{k,1}, \ldots, \mathbf{h}_{k,L}\}$ represents the sequence of token embeddings for tokens $\mathbf{x}_k = \{\mathbf{x}_{k,1}, \ldots, \mathbf{x}_{k,L}\}$. We can create our proxy finite set of maximizers, $S^{*(M)}$ (Equation 14) by using a computer-vision style Projected Gradient Descent (PGD) attack (Madry et al., 2019) in each token embedding space with initialisations of the PGD attack at different points to create multiple adversarial examples,

$$S^{*(M)}(\theta, \mathbf{h}) = \{\text{PGD}^{(1)}(\theta, \mathbf{h}), \ldots, \text{PGD}^{(M)}(\theta, \mathbf{h}), \}. \tag{17}$$

In this work we refer to DDi gradients applied to PGD AT as, *DDi-AT*.

B Gradient Normalization and Overconfidence

It is shown in Table 1 that the use of the DDi gradients with the PGD AT approach (ddi-at) gives rise

[8] Theorem 3 in the paper justifies the conditions to certify that the approximation is the steepest descent direction

to a highly overconfident model, which is responsible for the IOR. This section aims to determine the route cause of this overconfidence in the DDi gradient update algorithm. Equation 12 indicates that in the DDi gradient update algorithm global gradient normalization is applied. Note that this is different to standard training algorithms where either no normalization is applied or gradient clipping is used where global gradient normalization is only applied if the global gradient norm is larger than a threshold (Pascanu et al., 2012). Table 5 demonstrates that the use of the global gradient normalization in DDi-AT is responsible for the overconfidence and thus IOR. Interestingly, Table 6 reveals that gradient normalization can also induce overconfidence for the standardly trained *std* model.

Normalization	**clean**	$\bar{P}(\hat{c}\|\mathbf{x}_{\text{clean}})$	$\bar{P}(\hat{c}\|\mathbf{x}_{\text{adv}})$
gradient norm	87.90 $_{0.49}$	99.97 $_{0.03}$	99.91 $_{0.01}$
gradient clipping	88.28 $_{0.68}$	97.16 $_{0.30}$	86.12 $_{0.72}$
none	88.20 $_{0.55}$	96.98 $_{0.42}$	86.16 $_{0.66}$

Table 5: Model Confidence on clean and adversarial (pwws) examples for DDi-AT model with different forms of gradient normalization in the DDi gradient update step. Rotten Tomatoes dataset, DeBERTa model.

Normalization	**clean**	$\bar{P}(\hat{c}\|\mathbf{x}_{\text{clean}})$	$\bar{P}(\hat{c}\|\mathbf{x}_{\text{adv}})$
gradient norm	87.93 $_{0.44}$	99.96 $_{0.04}$	99.93 $_{0.02}$
gradient clipping	88.94 $_{0.31}$	97.02 $_{0.29}$	86.74 $_{0.84}$
none	88.96 $_{0.30}$	97.08 $_{0.26}$	86.04 $_{0.68}$

Table 6: Model Confidence on clean and adversarial (pwws) examples for *std* model with different forms of gradient normalization in training. Rotten Tomatoes dataset, DeBERTa model.

C Hyperparameter selection

We train the Transformer *std* models using standard hyper-parameter settings (He et al., 2020): initial learning rate of $1e-5$; batch size of 8; total of 5 epochs; 0 warm-up steps [9]; ADAMW optimizer, with a weight decay of 0.01 and parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$.

The Adversarial Training (AT) baseline approaches are trained with the same hyperparameters as for the *std* model and AT specific hyperparameters are as described in Li et al. (2021b). The default hyperparameters for each baseline (pgd, ascc and freelb) are: 5 adversarial iterations; adversarial learning rate of 0.03; adversarial initialisation magnitude of 0.05; adversarial maximum norm of 1.0; adversarial norm type of l2; α for ascc is 10.0; and β for ascc is 40.0. For DDi-AT, DDi gradients are applied to the PGD AT approach, with $M = 3$ gradients and $K = 3$ PGD iteration steps.

C.1 DDi-AT Ablation

The main results report DDi-AT results for DDi gradients applied to PGD AT with $K = 3$ PGD steps to find each adversarial example (in the embedding space) during training and $M = 3$ adversarial examples (refer to Section A.2). Table 7 gives the impact on adversarial accuracy (with and with out adversarial temperature calibration) of varying K and M. It appears that with greater iteration steps, K, the model presents a smaller IOR and a greater true robustness as the robustness accuracy does not degrade as much after calibration.

M	K	**Adv**	**clean**	**pwws**	**dg**
3	3	-	87.90 $\pm$0.49	61.07 $\pm$0.99	66.73 $\pm$1.01
		cal	87.90 $\pm$0.49	23.08 $\pm$1.96	22.89 $\pm$3.38
3	5	-	87.87 $\pm$0.57	55.53 $\pm$10.10	61.73 $\pm$10.06
		cal	87.87 $\pm$0.57	31.08 $\pm$4.61	32.90 $\pm$6.31
3	7	-	88.12 $\pm$0.11	40.06 $\pm$12.24	44.50 $\pm$15.79
		cal	88.12 $\pm$0.11	31.21 $\pm$1.26	30.93 $\pm$0.61
5	5	-	87.65 $\pm$1.17	50.59 $\pm$21.23	54.00 $\pm$26.22
		cal	87.65 $\pm$1.17	28.08 $\pm$2.05	27.95 $\pm$4.29
5	7	-	88.15 $\pm$0.38	31.68 $\pm$2.96	34.96 $\pm$4.79
		cal	88.15 $\pm$0.38	29.92 $\pm$1.17	31.61 $\pm$0.84

Table 7: Ablation: DDi-AT with M PGD adversarial examples, with each PGD adversarial example search during training using K iteration steps.

D Further Experiments

D.1 Other Datasets

Equivalent results are presented for Twitter (6 emotion classes) in Table 8 and for the AGNews dataset (4 news classes) in Table 9.

[9] We follow TextDefender (Li et al., 2021a) (presenting benchmark comparisons for AT approaches) in setting no warm-up steps. Further, empirically validation accuracy remained the same with warm-up of 50 and 100 steps.

Method	clean	bae	tf	pwws	dg
std	93.13 ±0.24	30.17 ±0.85	5.77 ±0.55	11.80 ±2.01	8.32 ±2.98
↓conf (§3.1)	93.13 ±0.24	29.63 ±0.80	6.78 ±0.58	15.22 ±1.55	14.68 ±3.01
↑conf (§3.1)	93.13 ±0.24	30.62 ±0.76	16.62 ±0.51	28.85 ±1.01	31.03 ±2.07
ddi-at (§3.2)	93.40 ±0.18	27.92 ±1.23	9.90 ±0.79	18.57 ±0.67	18.17 ±1.65
aug	92.58 ±0.11	31.52 ±2.82	4.68 ±0.25	9.33 ±0.11	29.45 ±0.64
pgd	93.48 ±0.03	28.83 ±0.43	4.88 ±1.24	9.95 ±0.69	5.45 ±1.08
ascc	91.15 ±0.57	34.65 ±0.23	4.60 ±1.05	12.15 ±0.22	11.28 ±1.40
freelb	93.67 ±0.23	29.15 ±1.00	4.93 ±1.25	10.15 ±0.30	5.48 ±0.73

Table 8: **Twitter:** Extreme confidence systems compared to standard AT methods on out-of-the-box adversarial attacks.

Method	clean	bae	tf	pwws	dg
std	93.75 ±0.25	78.46 ±0.51	31.63 ±1.11	42.25 ±2.93	46.21 ±1.31
↓conf (§3.1)	93.75 ±0.25	81.08 ±0.51	59.17 ±0.19	70.79 ±2.24	75.71 ±1.06
↑conf (§3.1)	93.75 ±0.25	85.71 ±0.80	84.79 ±0.89	88.21 ±0.36	88.17 ±0.31
ddi-at (§3.2)	94.25 ±0.33	88.00 ±0.75	88.08 ±1.00	88.96 ±0.36	89.25 ±0.13
aug	94.13 ±0.43	74.58 ±1.63	33.92 ±0.19	50.33 ±1.25	56.38 ±0.38
pgd	94.00 ±0.50	85.13 ±0.50	45.86 ±1.27	59.58 ±0.95	57.00 ±1.44
ascc	94.03 ±0.46	83.19 ±0.87	49.80 ±1.95	54.04 ±1.86	58.70 ±1.32
freelb	93.58 ±0.07	83.46 ±0.71	44.13 ±0.66	58.13 ±1.73	54.25 ±2.05

Table 9: **AGNews:** Extreme confidence systems compared to standard AT methods on out-of-the-box adversarial attacks. **Evaluation on 1000 samples.*

D.2 Other Models

The *illusion of robustness* is presented for an overconfident, underconfident and DDi-AT *DeBERTa* model in the main paper in Table 2. The same trends are observed for other popular Transformer-encoder (*base*) models: RoBERTa (Table 10); and BERT (Table 11).

Method	clean	bae	tf	pwws	dg
std	88.27 ±0.47	32.46 ±0.74	17.01 ±0.72	21.23 ±0.05	24.30 ±1.71
↓conf	88.27 ±0.47	31.77 ±0.33	20.42 ±1.27	24.92 ±1.43	32.99 ±1.33
↑conf	88.27 ±0.47	37.65 ±0.76	53.63 ±0.94	58.66 ±0.61	66.32 ±0.92
ddi-at	88.06 ±0.62	36.24 ±0.85	50.84 ±0.41	54.85 ±1.25	62.76 ±1.27

Table 10: **RoBERTa** Model: Robustness of Mis-calibrated systems.

Method	clean	bae	tf	pwws	dg
std	85.08 ±0.50	30.52 ±0.76	21.01 ±0.32	21.20 ±0.34	23.14 ±2.14
↓conf	85.08 ±0.50	29.74 ±0.19	20.95 ±0.53	24.58 ±1.36	30.64 ±0.24
↑conf	85.08 ±0.50	35.08 ±1.11	45.84 ±0.85	53.25 ±1.37	57.50 ±2.06
ddi-at	85.55 ±0.43	36.80 ±0.29	48.09 ±0.69	51.50 ±1.04	56.60 ±1.16

Table 11: **BERT** Model: Robustness of Mis-calibrated systems.

D.3 Calibration Error

In Table 12 we verify that the calibration approaches are effective in calibrating the models. We report the metrics: Expected Calibration Error (ECE) and Maximum Calibration Error (MCE).

Method	ECE	MCE	$\bar{P}(\hat{c}\|\mathbf{x}_{\text{clean}})$	$\bar{P}(\hat{c}\|\mathbf{x}_{\text{adv}})$
std	48.82 ±0.62	51.98 ±1.15	97.08 ±0.26	86.04 ±0.68
↓conf	38.96* ±0.30	38.96* ±0.30	50.00007 ±0.00	50.00004 ±0.00
+cal	38.96* ±0.30	38.96* ±0.30	50.00004 ±0.00	50.00002 ±0.00
↑conf	51.31 ±1.03	62.62 ±11.8	99.98 ±0.02	99.95 ±0.01
+cal	42.30 ±0.91	48.28 ±1.04	90.36 ±0.45	75.88 ±0.58
ddi-at	52.41 ±0.57	74.87 ±20.97	99.97 ±0.03	99.91 ±0.05
+cal	42.60 ±0.58	62.73 ±18.36	90.13 ±0.11	87.54 ±0.80

Table 12: Calibration Error and Average Predicted Confidence (on clean and adv-pwws). N.B. std is across seeds. *off-the-shelf calibration error computation fails here as all confidences very close to 50%, so manual computation of CE here: *accuracy - 50%*.

D.4 IOR in AT Approaches

The main results demonstrate that highly miscalibrated systems have an *illusion of robustness* (IOR), where an adversary's temperature calibration can mitigate this illusion of robustness. Considering the rotten tomatoes dataset and the DeBERTa model, Table 13 demonstrates that standard AT approaches considered in this work can also suffer from the IOR, when global gradient normalization is included in the training algorithm (Note that Table 6 shows that gradient normalization can be a source of model overonfidence). Nevertheless, Table 14 demonstrates that when global gradient normalization is excluded from the training algorithm, the baseline AT approaches considered in this work no longer present IOR as calibration does not degrade their adversarial accuracy.

Method	Adv	clean	bae	tf	pwws	dg
pgd*	-	88.36 ±0.68	39.94 ±0.55	58.02 ±1.04	64.45 ±0.77	67.02 ±0.83
	cal	88.36 ±0.68	33.64 ±0.61	19.95 ±1.02	26.78 ±0.73	26.22 ±0.69
ascc*	-	87.80 ±0.42	40.01 ±0.69	54.32 ±1.57	63.99 ±0.86	67.43 ±0.93
	cal	87.80 ±0.42	33.53 ±0.78	16.22 ±2.54	23.78 ±0.75	26.90 ±1.54

Table 13: Baseline AT approach (PGD and ASCC results here) can also suffer from IOR (calibration reduces observed adversarial robustness) when global gradient normalization used in the training algorithm. The IOR was also observed for aug and freelb AT schemes.

Method	Adv	clean	bae	tf	pwws	dg
std	-	88.96 ±0.30	31.39 ±1.20	17.82 ±0.49	20.42 ±0.62	20.11 ±0.94
	cal	88.96 ±0.30	31.39 ±1.20	17.80 ±0.51	20.46 ±0.66	20.05 ±0.88
aug	-	87.12 ±0.39	34.74 ±1.59	22.36 ±1.83	26.11 ±2.57	37.43 ±0.75
	cal	87.12 ±0.39	34.74 ±1.59	22.36 ±1.81	25.98 ±2.32	37.45 ±0.74
pgd	-	88.24 ±0.73	33.65 ±0.57	19.92 ±0.47	26.70 ±0.87	26.05 ±0.61
	cal	88.24 ±0.73	33.65 ±0.57	19.90 ±0.46	26.74 ±0.90	26.10 ±0.54
ascc	-	87.77 ±0.36	33.61 ±0.64	15.13 ±2.17	23.50 ±0.77	26.80 ±2.11
	cal	87.77 ±0.36	33.60 ±0.63	15.10 ±2.19	23.49 ±0.79	26.75 ±2.03
freelb	-	88.74 ±0.32	32.52 ±0.52	19.51 ±1.70	24.55 ±0.70	24.52 ±0.73
	cal	88.74 ±0.32	88.74 ±0.32	19.50 ±1.72	24.35 ±0.55	24.54 ±0.75

Table 14: Baseline AT approach can be freed of the IOR when global gradient normalization is not used in the training algorithm.

D.5 Alternative Calibration Approaches

In the main results, temperature calibration was implemented to detect adversarial examples based on two central considerations: 1) Temperature calibration effectively facilitates the adversarial attack search, especially for obviously mis-calibrated models; and 2) Temperature calibration preserves the rank order of logits, thereby ensuring transferability of adversarial examples from the calibrated to the original uncalibrated model. To broaden the analytical scope, alternative calibration techniques are examined. The goal is to assess their potential in mitigating the disruption to the adversarial attack search processes and to determine the potency of the resulting adversarial examples on the uncalibrated model. Binning-based calibration is deemed unsuitable due to its intrinsic non-differentiability, which could prevent the adversarial search process. Hence, the multi-class version of Platt Scaling is explored as a viable calibration strategy and subsequently contrasted against the benchmark temperature calibration approach from the main results. The performance of the calibration results is shown in Table 15, where it is evident that the Platt scaling approach is far less stable than temperature calibration and can in fact excessively enhance the *illusion of robustness*.

For automatic calibration, standard training hyperparameters were employed. Specifically, the temperature calibration protocol was set at 5,000 iterations with a learning rate of 0.01. Similarly, the Platt scaling protocol was also designed for 5000 iterations with a learning rate of 0.01. A point to note for practical implementation: adversaries might need to refine calibrator hyperparameters to minimize the Expected Calibration Error (ECE) on a specified validation set. However, ECE determination is nuanced, largely due to its sensitivity to chosen bin widths, as highlighted in Table 12 for instances of underconfidence.

Method	Adv	clean	bae	tf	pwws	dg
std	-	88.96 ±0.30	31.39 ±1.20	17.82 ±0.49	20.42 ±0.62	20.11 ±0.94
↓conf	-	88.96 ±0.30	31.21 ±0.94	20.98 ±0.99	25.17 ±0.89	32.18 ±2.78
	temp	88.96 ±0.30	31.52 ±0.34	21.89 ±0.43	27.58 ±1.31	31.52 ±0.34
	platt	88.96 ±0.30	72.08 ±12.15	70.33 ±18.00	72.70 ±16.72	74.73 ±17.11
↑conf	-	88.96 ±0.30	37.71 ±1.18	54.35 ±0.73	59.29 ±0.62	65.60 ±1.81
	temp	88.96 ±0.30	31.39 ±1.20	17.82 ±0.49	20.45 ±0.74	21.64 ±1.46
	platt	88.96 ±0.30	37.21 ±3.73	34.55 ±17.90	37.46 ±19.70	41.09 ±19.59
ddi-at	-	87.90 ±0.49	39.18 ±0.75	56.54 ±1.67	61.07 ±0.99	66.73 ±1.01
	temp	87.90 ±0.49	31.80 ±0.57	18.36 ±3.01	23.08 ±1.96	22.89 ±3.38
	platt	87.90 ±0.49	43.34 ±19.42	38.77 ±32.23	42.25 ±31.66	42.72 ±32.72

Table 15: Adversarial mitigation of highly miscalibrated systems using different test-time calibration approaches.

E Extreme miscalibration causes noisy gradients

Section 3 argues that for heavily miscalibrated systems, the 'gradients' of the output probabilities with respect to the input are extremely noisy. Therefore, of-the-shelf adversarial attack methods, that use these gradients to select which tokens in the input sequence to attack, receive noisy signals and fail to operate. In this section, we demonstrate that extreme miscalibration does indeed cause noisy gradients for of-the-shelf-adversarial attacks.

We consider two systems: the standard *std* system from the main paper and the heavily miscalibrated, overconfident system, ↑conf in the main paper. Experiments are on the *rt* dataset and we consider specifically the PWWS attack and Textfooler attack. These off-the-shelf adversarial attack approach rank all tokens w_i in the input sequence $\mathbf{x}$ by their influence on the output of the model (N.B. this is considered an approximation for the gradient of the output with respect to each input token). The PWWS attack refers to this influence as *saliency*, whilst the Textfooler attack calls it *importance*. To assess the impact of heavy miscalibration on the rank ordering, Table 16 reports the Spearman Rank Correlation between the rank of all input tokens (in the first iteration of the attack) as per the two models: *std* and ↑conf. The average correlation and standard deviation are given over the entire dataset. The average rank correlation is 0.28 for PWWS and 0.29 Textfooler, which is very low and demonstrates that by simply having heavy miscalibration there is a significant impact on the attack mechanism. Further, the standard deviation is also large, suggesting that for many input sequences, the correlation is even lower.

Attack	Rank Correlation
pwws	$0.28_{\pm 0.24}$
textfooler	$0.29_{\pm 0.26}$

Table 16: Spearman Rank Correlation of input tokens' importance with (overonfident model) and without (*std* model) heavy miscalibration. The low rank correlation demonstrates that the token importance is strongly impacted by extreme confidence, which can explain the observed IOR for highly miscalibrated models.