DiffuseDef: Improved Robustness to Adversarial Attacks via Iterative Denoising

Anonymous ACL submission

Abstract

Pretrained language models have significantly advanced performance across various natural language processing tasks. However, adversarial attacks continue to pose a critical challenge to system built using these models, as they can be exploited with carefully crafted adversarial texts. Inspired by the ability of diffusion models to predict and reduce noise in computer vision, we propose a novel and flexible adversarial defense method for language classification tasks, `DiffuseDef`[1], which incorporates a diffusion layer as a denoiser between the encoder and the classifier. The diffusion layer is trained on top of the existing classifier, ensuring seamless integration with any model in a plug-and-play manner. During inference, the adversarial hidden state is first combined with sampled noise, then denoised iteratively and finally ensembled to produce a robust text representation. By integrating adversarial training, denoising, and ensembling techniques, we show that `DiffuseDef` improves over existing adversarial defense methods and achieves state-of-the-art performance against common black-box and white-box adversarial attacks.

1 Introduction

Pretrained language models (PLM) have significantly advanced the performance of various natural language processing (NLP) tasks. Despite such improvements, current NLP systems remain susceptible to adversarial attacks where carefully crafted text perturbations can lead to incorrect model outputs (Alzantot et al., 2018; Jin et al., 2020; Li et al., 2020). In order to improve robustness to adversarial attacks, various defense methods have been proposed, such as adversarial training (Zhu et al., 2020; Si et al., 2021; Zhou et al., 2021; Xi et al., 2022), text denoising (Nguyen Minh and Luu, 2022; Wang et al., 2023), ensembling (Zhou et al., 2021; Zeng et al., 2023; Li et al., 2023), etc. However, existing defense methods either assume the test-time perturbation/attack set is similar to that used in training (Li et al., 2021), or are limited to specific architectures (Xi et al., 2022), or at inference time require large computational cost, thereby limiting their practical applicability.

Diffusion models are commonly used in computer vision (CV) to generate high-quality images by predicting and removing noise from a sampled noisy image. Therefore, they can be adopted to remove noise from adversarial images and thus improve robustness to attacks (Im et al., 2016; Nie et al., 2022; Gulsen et al., 2024). However, in NLP very limited research has investigated adversarial defense with diffusion models due to the discrete and contextual nature of text data. Li et al. (2023) adopt the idea of iterative denoising and reconstruct adversarial texts from masked texts, while Yuan et al. (2024) use a diffusion model as a classifier and perform reverse diffusion steps on the label vector, conditioning on the input text. Inspired by the general noise prediction and reduction capability of diffusion models, we propose `DiffuseDef`, a novel adversarial defense method which employs diffusion training to denoise hidden representations of adversarial texts. Unlike Li et al. (2023) and Yuan et al. (2024) which apply diffusion on texts or labels, `DiffuseDef` directly removes noise to the hidden states, providing a more effective and robust text representation to defend against adversarial texts. Different from diffusion-based defenses in CV (Nie et al., 2022; Gulsen et al., 2024) that applies purification to input images, `DiffuseDef` only performs denoising at the final hidden layer, resulting in improvement in efficiency. This improvement is further enhanced when combined with ensembling as it eliminates the need for a forward pass through all classifier parameters for each ensemble.

`DiffuseDef` combines adversarial training with

[1] Codes anonymized at: `https://anonymous.4open.science/r/Diffuse-Def`

diffusion training, where the diffusion layer is trained to predict randomly sampled noise at a given timestep. During inference, the diffusion layer serves as a denoiser, iteratively removing noise from adversarial hidden states to yield a robust hidden representation. Moreover, we adopt the ensembling strategy by first adding random noise to text hidden states to create multiple variants then denoising them via the diffusion layer. The model output is made by averaging all denoised hidden states. Since ensembling happens solely at the diffusion layer, `DiffuseDef` is more efficient than traditional ensembling-based methods (Ye et al., 2020; Zeng et al., 2023), which require a full forward pass through all model parameters.

Through systematic experimentation, we demonstrate that `DiffuseDef` outperforms strong defense methods and is able to defend against multiple types of black-box and white-box adversarial attacks, while preserving performance on clean texts. Our analysis also reveals that the ensembling diffused representations provide a stronger defense against finding vulnerable words to attack and can reduce the distance in latent space between adversarial texts and their clean text counterpart.

Our contributions can be summarised as follows:

- We propose `DiffuseDef`, a novel and flexible adversarial defense method that can be added on top of any existing adversarial defense methods to further improve robustness to adversarial attacks.

- `DiffuseDef` outperforms existing adversarial methods and achieves state-of-the-art performance against prevalent black-box and white-box adversarial attacks.

- Through extensive analysis, we demonstrate the effectiveness of the ensembling diffused representation and the efficiency of `DiffuseDef` compared to existing ensembling-based methods.

2 Related Work

2.1 Textual Adversarial Attacks

Textual adversarial attacks focus on constructing adversarial examples from an original text that maximize the likelihood of incorrect predictions by a neural network. These attacks require adversarial examples to be perceptually similar to the original text, which is typically achieved by introducing subtle perturbations to the original text, such as character swapping (Gao et al., 2018; Ebrahimi et al., 2018), synonym-substitutions (Ren et al., 2019; Yoo and Qi, 2021; Liu et al., 2023), and paraphrasing (Gan and Ng, 2019; Huang and Chang, 2021). Taking the text classification task as an example, given a classifier $\mathcal{C}(\mathbf{x})$ that maps an input sequence of words $\mathbf{x} = [w_1, w_2, ..., w_L]$ to its designated label y, the goal of the attack model is to construct an adversarial example $\mathbf{x}' = \mathbf{x} + \delta$ to fool the classifier, where δ is a subtle adversarial perturbation constrained by $||\delta|| < \omega$. The adversarial example $\mathbf{x}'$ is considered a successful attack if it leads to an incorrect prediction $\mathcal{C}(\mathbf{x}') \neq y$. The attacker can iteratively generate multiple adversarial examples and query the classifier to obtain a successful attack, whereas the classifier must consistently return the correct prediction within a specified number of query attempts to be considered robust.

Common textual adversarial attack methods adopt a two-stage process to construct effective adversarial examples: *word importance ranking* and *word substitution*. In the first stage, words or subwords are ranked based on their influence on the model's prediction. This is measured by leveraging either gradient information (Liu et al., 2022) or changes in prediction probabilities when words are removed (Jin et al., 2020) or masked (Ren et al., 2019; Li et al., 2020). In the second stage, candidate words are substituted with synonyms (Zang et al., 2020), perturbed variants (Gao et al., 2018), or outputs from masked language models (Garg and Ramakrishnan, 2020; Li et al., 2020). The substitution process is guided by various constraints to ensure the adversarial example remains natural and semantically equivalent to the original text. Common constraints include thresholding the similarity between the replacement word embedding and the substituted word embedding, or ensuring the semantic similarity between sentence vectors modeled from Universal Sentence Encoder (Cer et al., 2018). Despite these constraints, current textual adversarial attacks still pose significant challenges to NLP models (Liu et al., 2022; Xu et al., 2021; Yuan et al., 2023), highlighting the necessity for defense methods for better adversarial robustness.

2.2 Adversarial Defense Methods

To mitigate the performance degradation caused by adversarial attacks, various adversarial defense methods have been developed. They can be grouped into three categories: *training-based*, *ensembling-based*, and *denoising-based* methods. Adversarial training improves the robustness of the model to adversarial examples through strategies like data augmentation (Si et al., 2021; Bao et al., 2021) and adversarial regularisation (Madry et al., 2018; Zhu et al., 2020; Wang et al., 2021; Xi et al., 2022; Gao et al., 2023; Formento et al., 2024). However, adversarial training methods are limited as they assume similar train-test adversarial examples, and thus tend to overfit to specific types of adversarial attacks. Ensembling-based methods generate multiple variants of the input text at inference time and ensemble model predictions over all the variants (Ye et al., 2020; Zhou et al., 2021; Zeng et al., 2023; Li et al., 2023), but they can be inefficient given that model predictions are needed on every ensemble, increasing the inference time with the number of ensembles. More recently, denoising-based methods have been proposed to improve adversarial robustness by mapping the vector representation of the adversarial text to another point in the latent space that is close to the clean text (Nguyen Minh and Luu, 2022; Wang et al., 2023; Moon et al., 2023; Yuan et al., 2024; Ji et al., 2024). The denoised representation makes it more difficult to find vulnerable words to attack, thus improving adversarial robustness (Wang et al., 2023). Nevertheless, denoising might lead to very different representations of clean text and adversarial text, therefore changing the semantic meanings.

The proposed `DiffuseDef` builds on these three approaches and can use any adversarially trained classifier as the base, applying denoising via a diffusion layer, and ensembling the diffused representations with a small number of ensembles. Using a diffusion layer as a denoiser addresses the overfitting problem from adversarial training and mitigates the efficiency problem by performing ensembling only at the diffusion layer. By averaging denoised hidden states across all ensembles, `DiffuseDef` also addresses the issue stemming from denoising, maintaining good performance on clean texts.

3 DiffuseDef

3.1 Training

The proposed diffusion defense model consists of a pretrained encoder for feature extraction, a transformer-based diffusion layer for noise prediction and reduction, and a classifier layer for output generation. The training process is split into two stages: adversarial training and diffusion training (Figure 1). The **adversarial training** stage employs any neural network-based adversarial training methods like FreeLB++ (Li et al., 2021) and RSMI (Moon et al., 2023), which optimise the encoder and classifier for robustness by perturbing the latent representation of the text input.

In the **diffusion training** stage, only the diffusion layer is trained to predict random noise added to the clean text hidden state at different timesteps, enabling it to denoise the adversarial hidden state at inference time. The pretrained encoder, however is frozen during this stage. Since the pretrained encoder is only used for feature extraction, the diffusion training method is compatible with any neural network-based adversarial training method.

Given an input sequence of tokens $\mathbf{x} \in \mathbb{R}^L$, the pretrained encoder extracts the hidden state $h \in \mathbb{R}^{L \times D}$. A random Gaussian noise ϵ is sampled to perturb hidden state h. Sohl-Dickstein et al. (2015) define the forward diffusion process as a Markov Chain where at each timestep a Gaussian noise is sampled and added to the previous latent feature: $h_t = \sqrt{1-\beta_t} h_{t-1} + \sqrt{\beta}\epsilon$, where $\epsilon \in \mathcal{N}(0, \mathcal{I})$, h_t is the noisy hidden state at step t and β is a pre-calculated variance schedule changing with t. As shown by Ho et al. (2020), this equation can be reformulated to calculate h_t directly from h by defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha} = \prod_{i=1}^{t} \alpha_i$, thus

$$h_t = \sqrt{\bar{\alpha}_t} h + \sqrt{1 - \bar{\alpha}_t}\epsilon \quad (1)$$

At each training step, a random forward diffusion timestep t is sampled from a uniform distribution. Therefore, the noisy hidden state h_t is created from h, t, and ϵ. The diffusion layer θ consists of a time embedding and a transformer layer. The time embedding receives the diffusion timestep t as input and produces an embedding e_t, which is added to h_t as input for the transformer layer. Finally, the transformer layer outputs the predicted noise $\epsilon_\theta(h_t, t)$, and mean square error is used to compute the loss between predicted noise $\epsilon_\theta(h_t, t)$ and actual sampled noise ϵ.

$$L = \mathbb{E}_{t,h,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} h + \sqrt{1 - \bar{\alpha}_t}\epsilon)\right\|^2\right] \quad (2)$$

3.2 Inference

Leveraging the diffusion layer's ability to predict noise at a given timestep t, we utilise it as a denoiser during inference by iteratively performing

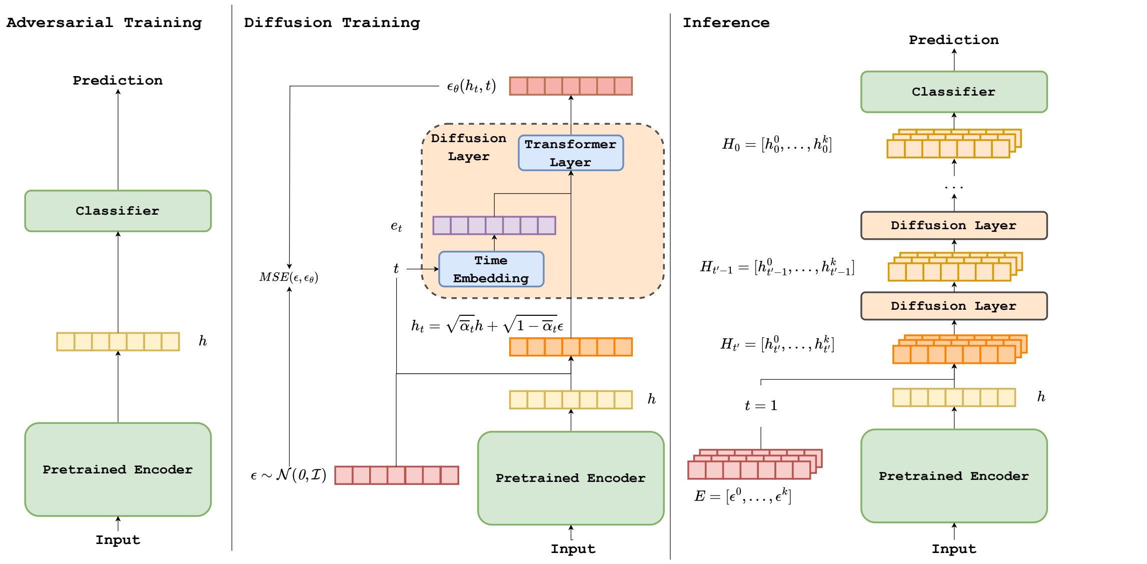
Adversarial Training
Prediction
Classifier
h
Pretrained Encoder
Input
Diffusion Training
$\epsilon_\theta(h_t, t)$
Diffusion Layer
Transformer Layer
e_t
t
Time Embedding
$MSE(\epsilon, \epsilon_\theta)$
$h_t = \sqrt{\overline{\alpha}_t}h + \sqrt{1-\overline{\alpha}_t}\epsilon$
h
$\epsilon \sim \mathcal{N}(0, \mathcal{I})$
Pretrained Encoder
Input
Inference
Prediction
Classifier
$H_0 = [h_0^0, \dots, h_0^k]$
. . .
Diffusion Layer
$H_{t'-1} = [h_{t'-1}^0, \dots, h_{t'-1}^k]$
Diffusion Layer
$H_{t'} = [h_{t'}^0, \dots, h_{t'}^k]$
$t = 1$
h
$E = [\epsilon^0, \dots, \epsilon^k]$
Pretrained Encoder
Input

Figure 1: Training and inference of DiffuseDef model. The *adversarial training* stage trains the pretrained encoder and classifier with perturbed input for adversarial robustness. The *diffusion training* trains the diffusion layer to predict injected noise at a given timestep t. At *inference* time, the text hidden state is first noised by 1 step and then denoised by t' steps to create the denoised hidden states, which are ensembled to make the final prediction.

the reverse diffusion steps, which sample from $p_\theta(h_{t-1}|h_t) = \mathcal{N}(h_{t-1}; \mu_\theta(h_t, t), \Sigma_\theta(h_t, t))$ to produce the denoised hidden state

$$\mu_\theta(h_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(h_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right) \tag{3}$$

$$\Sigma_\theta(h_t, t) = \sigma_t^2 \mathcal{I} \tag{4}$$

where ϵ_t is the predicted noise from diffusion layer and $\sigma_t^2 = \beta_t$. The denoised hidden state can thus be computed with

$$h_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(h_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right) + \sigma_t z \tag{5}$$

where $z \in \mathcal{N}(0, \mathcal{I})$.

Inference in DiffuseDef combines a one-step **noising**, a multi-step **denoising**, and an **ensembling** step. After the pretrained encoder extracts its hidden state h, a set of k Gaussian noise vectors $E = [\epsilon^0, \epsilon^1, ..., \epsilon^k]$ are sampled to perform a single forward diffusion step. These noise vectors E are then added to the hidden state h following equation 1, resulting in a set of noisy hidden states $H_{t'} = [h_{t'}^0, h_{t'}^1, ..., h_{t'}^k]$, where t' denotes the number of denoising steps. The noisy hidden states $H_{t'}$ are subsequently denoised through t' reverse diffusion steps, where noise is predicted by the diffusion layer and subtracted from the previous noisy hidden states. Unlike Ho et al. (2020) where the reverse diffusion step starts with pure noise sampled from standard normal distribution, we assume the noisy hidden state $H_{t'}$ is already an intermediate state in the reverse diffusion steps. This allows us to use a smaller number of t' than the training timestep t to prevent the denoised hidden states from diverging substantially from the initial hidden state h. This sequence of denoising steps creates the final denoised hidden states $H_0 = [h_0^0, h_0^1, ..., h_0^k]$, which are averaged and used by the classifier to output the final predicted label. This process is summarised in Algorithm 1.

Algorithm 1: Inference of DiffuseDef

```
Data: Input text $\mathbf{x}$
Result: Predicted label $y'$
$h \leftarrow Enc(\mathbf{x})$;
Sample $E = [\epsilon^0, \epsilon^1, ..., \epsilon^k]$, $\epsilon \sim \mathcal{N}(0, \mathcal{I})$;
$H_{t'} \leftarrow \sqrt{\bar{\alpha}_1}h + \sqrt{1-\bar{\alpha}_1}E$;
for $i \leftarrow 0$ to $t' - 1$ do
    $E_{t'-i} \leftarrow \epsilon_\theta(H_{t'-i}, t' - i)$;
    $H_{t'-i-1} \leftarrow \frac{1}{\sqrt{\alpha_{t'-i}}}\left(H_{t'-i} - \frac{1-\alpha_{t'-i}}{\sqrt{1-\bar{\alpha}_{t'-i}}}E_{t'-i}\right) + \sigma_{t'-i}z$;
end
$y' \leftarrow CLS\,(avg(H_0))$;
```

4 Experiments

Datasets. We focus on two common NLP tasks in our experiments: topic classification and natural language inference (NLI). We conduct experiments on three common datasets: AG News (Zhang et al., 2015a), IMDB (Maas et al., 2011a), and QNLI datasets (Wang et al., 2018). We randomly split AGNews, IMDB, and QNLI datasets into train, validation, and test splits. Appendix A gives details on data preparation.

Evaluation. Following previous work on adversarial defense, we use three benchmarking black-box attack methods to evaluate the robustness of `DiffuseDef`: TextFooler (TF) (Jin et al., 2020), TextBugger (TB) (Li et al., 2019), and Bert-Attack (BA) (Li et al., 2020). We also evaluated the robustness against two white-box attacks: T-PGD (Yuan et al., 2023) and SemAttack (Wang et al., 2022). Regarding evaluation metrics, we measure the clean accuracy (**Clean%**) on the test set, the accuracy under attack (**AUA%**), and the number of adversarial queries (**#Query**) needed for a successful attack. Higher scores on the three metrics denote a better robustness performance of a defense method. The accuracy on clean data is measured across the entire test set. The accuracy under attack and number of queries, due to the lengthy attacking process, is measured on a randomly sampled subset of 1000 examples from the test set. We use the `TextAttack` library as the adversarial evaluation framework. To ensure a fair comparison and high-quality adversarial examples, we follow the same evaluation constraints as in Li et al. (2021). The evaluation metrics are averaged based on experiments run with 5 random seeds.

4.1 Comparison to SOTA

We compare our proposed method with state-of-the-art adversarial defense approaches, trained using both BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) as backbones: **Fine-tune**: Fine-tuning pretrained models on downstream task with no defense method applied[2]. **InfoBERT** (Wang et al., 2021): Applying mutual-information-based regularizers during fine-tuning of pretrained models to improve robustness. **FreeLB++** (Li et al., 2021): An adversarial training method improving on FreeLB(Zhu et al., 2020), which adds adversarial perturbations to word embedding during fine-tuning. **EarlyRobust**[3] (Xi et al., 2022): Extracting early-bird subnetworks and pruning pretrained models for efficient adversarial training. **RMLM** (Wang et al., 2023): A denoising-based model combined with adversarial text detection. **ATINTER** (Gupta et al., 2023): A fine-tuned T5 model that re-writes the input adversarial texts and the classifier predicts labels from the re-written texts. **RSMI** (Moon et al., 2023): A two-stage training method that combines randomised smoothing and masked inference to improve adversarial robustness.

4.2 Implementation and Settings

We train two `DiffuseDef` variants using FreeLB++ and RSMI models as base models considering their robust adversarial defense capabilities. In the diffusion layer, only one transformer encoder layer (Vaswani et al., 2017) is used. The maximum noising timestep t during training is set to 30 for AG-News and QNLI datasets, and 10 for IMDB dataset, while at inference time, we only apply 5 denoising steps for t'. We follow (Ho et al., 2020) to use a linear β_t schedule from $\beta_1 = 10^{-4}$ to $\beta_t = 0.02$. The base classifier is first fine-tuned for 10 epochs, and the diffusion layer is trained for 100 epochs, with the base classifier parameters frozen for efficiency. During the diffusion training stage, the same train-dev splits are used as in the adversarial training stage, thus ensuring no data leakage. At inference time, the number of ensembles is set to 10. Appendix C lists the training hyper-parameters.

5 Results and Analysis

5.1 Adversarial Robustness

In Table 1, we compare the adversarial robustness of `DiffuseDef` with baselines and SOTA methods on AGNews and IMDB datasets trained with BERT (results for RoBERTa in Appendix E). `DiffuseDef` consistently outperforms all other methods on all three datasets across the three attacks, exhibiting substantial improvements in accuracy under attack. After applying diffusion training, the AUA score for both FreeLB++ and RSMI models improves significantly, with an average increase of 30% AUA against the three attack methods. When comparing the clean accuracies to its base model (i.e. FreeLB++ and RSMI), `DiffuseDef` only shows a minor decline, between 0.2 and 0.7 accuracy score, which indicates that it can preserve the clean text performance while improving adversarial robustness. Moreover, models trained with `DiffuseDef` show a much smaller gap between clean accuracy and accuracy under attack, and such difference can be reduced to less than 10% AUA.

[2] "Fine-tuned" is a baseline approach used to illustrate the effect of adversarial attacks.

[3] We only run EarlyRobust with BERT as its implementation with RoBERTa has not been released.

Dataset	Method	Clean%	AUA%			#Query		
			TF	TB	BA	TF	TB	BA
AGNews	Fine-Tuned	94.4	10.2	25.4	27.1	348	372	379
	InfoBERT (Wang et al., 2021)	95.0	35.5	39.1	42.6	377	397	397
	FreeLB++ (Li et al., 2021)	95.0	54.7	56.5	44.6	426	430	390
	EarlyRobust (Xi et al., 2022)	94.4	35.6	37.2	45.7	475	516	533
	RMLM (Wang et al., 2023)	94.3	45.6	44.4	54.1	642	738	705
	ATINTER (Gupta et al., 2023)	94.2	68.0	59.0	81.0	527	235	122
	RSMI (Moon et al., 2023)	94.3	52.6	56.7	55.4	680	737	687
	`DiffuseDef-FreeLB++` (Ours)	94.8	**84.5**	**86.0**	**84.6**	877	972	910
	`DiffuseDef-RSMI` (Ours)	93.8	82.7	83.3	84.4	**894**	**1029**	**930**
IMDB	Fine-Tuned	93.3	7.7	8.3	10.5	540	534	378
	InfoBERT (Wang et al., 2021)	93.9	29.2	25.4	30.7	642	644	390
	FreeLB++ (Li et al., 2021)	94.3	44.2	39.6	40.6	784	829	426
	EarlyRobust (Xi et al., 2022)	92.7	49.7	46.8	43.8	2267	2788	1841
	RMLM (Wang et al., 2023)	93.6	48.8	47.9	47.1	2554	3258	2064
	ATINTER (Gupta et al., 2023)	94.3	25.0	15.0	39.0	1173	645	637
	RSMI (Moon et al., 2023)	90.9	60.0	54.4	51.1	2840	3455	2070
	`DiffuseDef-FreeLB++` (Ours)	94.4	**82.1**	**83.0**	**84.0**	3174	4348	2842
	`DiffuseDef-RSMI` (Ours)	90.2	80.9	79.8	79.8	**3590**	**4748**	**2901**
QNLI	Fine-Tuned	90.8	21.5	15.5	13.3	195	206	177
	InfoBERT (Wang et al., 2021)	91.2	27.8	22.8	18.7	217	232	201
	FreeLB++ (Li et al., 2021)	90.3	45.6	40.2	30.5	253	279	226
	EarlyRobust (Xi et al., 2022)	89.2	24.3	21.2	19.1	265	292	255
	ATINTER (Gupta et al., 2023)	90.4	43.0	24.8	34.0	185	115	89
	RSMI (Moon et al., 2023)	87.4	35.2	30.9	28.2	314	353	314
	`DiffuseDef-FreeLB++` (Ours)	90.3	**66.7**	**65.3**	**64.4**	**485**	**587**	**543**
	`DiffuseDef-RSMI` (Ours)	86.4	55.5	54.8	57.2	459	569	528

Table 1: Main adversarial robustness results on classification tasks with BERT. *Clean*: accuracy on clean test set. *TF*: TextFooler. *TB*: TextBugger. *BA*: BertAttack.

Another benefit of `DiffuseDef` is the increased number of adversarial queries needed to obtain a successful attack. Models applying `DiffuseDef` require over twice the number of queries on all three datasets compared to the other methods. This increase is even larger on the IMDB dataset due to the longer text length. For example, `DiffuseDef` model requires on average over 3000 queries to achieve a successful attack while FreeLB++ only needs 400 to 800 queries. The substantial increase suggests that even if the attackers manage to construct a successful adversarial attack, they need 2x to 3x more time to find the attack on `DiffuseDef` than other models, affirming the improved robustness from diffusion training. In addition, we observe that the number of queries for denoising-based methods (i.e. RSMI, RMLM, and `DiffuseDef`) is generally higher than adversarial training-based methods (i.e. InfoBERT, FreeLB++). This is because denoising-based methods transform the hidden representations of the adversarial texts into a non-deterministic representation. The introduction of randomness in hidden states results in uncertainty in model logits, thus increasing the difficulty of finding vulnerable words to attack (Wang et al., 2023).

5.2 Robustness against White-box Attacks

In addition to black-box attacks, we also measure the robustness against white-box attacks, where the attackers can access the defender model's gradients and parameters, which is a more challenging adversarial attack scenario than black-box attacks. In Table 2, we report AUA score under T-PGD (Yuan et al., 2023) and SemAttack (Wang et al., 2022), the former being a variant of PGD attack (Madry et al., 2019) for text while the later adopt C&W attack (Carlini and Wagner, 2017) on embedding layers. We observe that white-box attacks result in lower AUA for all defense methods. For example, FreeLB++ performs poorly on IMDB when attacked by SemAttack, with AUA score being 0.03. However, applying diffusion training still significantly improves adversarial robustness to white-box attacks. Note that even RSMI shows very robust performance against T-PGD[4], it can

[4] RSMI is incompatible with SemAttack because it requires gradients of embedding layer but SemAttack skips the embed-

still further improve from DiffuseDef.

Method	AGNews		IMDB	
	T-PGD	Sem	T-PGD	Sem
Fine-Tuned	8.8	41.5	3.0	1.3
InfoBERT	32.1	47.1	25.4	2.0
FreeLB++	19.6	58.2	15.5	3.4
RSMI	79.6	n/a	43.3	n/a
DiffuseDef-FreeLB	59.4	**68.1**	**50.3**	**28.2**
DiffuseDef-RSMI	**81.7**	n/a	48.4	n/a

Table 2: Accuracy under white-box attacks.

5.3 DiffuseDef + Adversarial Augmentation

In Table 2, we combine DiffuseDef with adversarial data augmentation method and evaluate the robustness on AGNews. We first generate adversarial augmentation data using Textfooler and adversarially train the classifier with augmented training data. The augmented classifier is further performed with diffusion training, resulting in a DiffuseDef variant. We notice that adversarial training on augmented data help improve the performance on all three attacks. After applying diffusion training, the adversarial performance significantly improves over the augmented model, which proves the flexibility of DiffuseDef to be applied with different types of adversarial training methods.

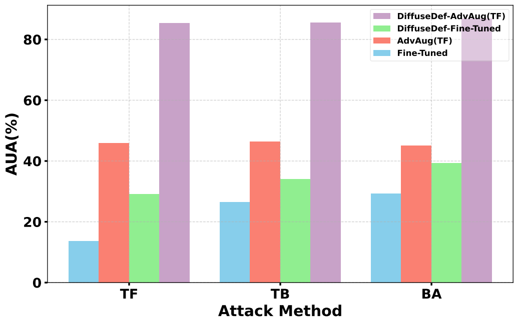

Figure 2: Robustness of DiffuseDef with textual adversarial augmentation method.

5.4 Effect of Additional Denoising Steps

In Figure 3 we study how the inference denoising steps t' can affect the adversarial performance. For the DiffuseDef model without ensembling, both AUA score and the number of queries required to attack increase as the inference denoising step is larger. As the denoising step t' grows from 1 to 30, the AUA score improves from 58 to 65 while the number of attack queries grows from 430 to 780. In contrast, for DiffuseDef with ensembling, the model maintains a stable but robust performance in AUA and number of queries, regardless of the increase of t'. Considering that the ensembling introduces a notable performance increase, the DiffuseDef model is likely to be hitting an upperbound in both metrics, thus no further improvement is reached by increasing the denoising steps. However, it also shows that with ensembling, DiffuseDef can be applied with a smaller t' for better efficiency while maintaining a robust adversarial performance.

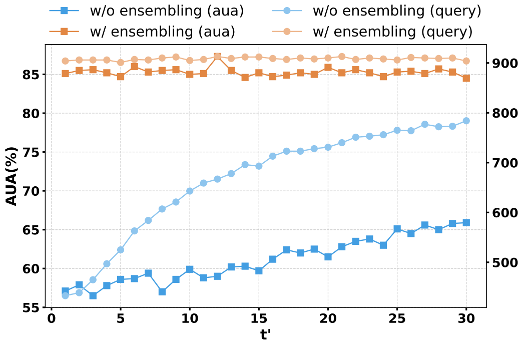

Figure 3: AUA and #Query (TextFooler) w.r.t inference denoising step for DiffuseDef w/ and w/o ensembling.

5.5 Ensembling Diffused Hidden Representations

In DiffuseDef the text hidden state is diffused and ensembled to form a denoised hidden representation, which contributes significantly to the improved adversarial robustness. In this section, we study how the ensembling diffused hidden representation helps defend against adversarial attacks.

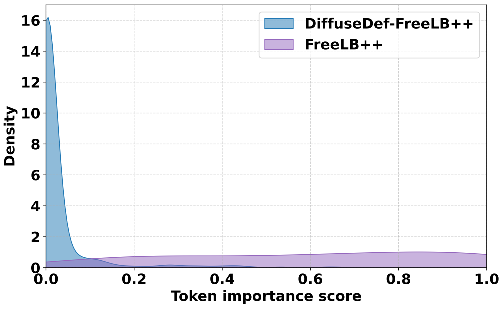

Figure 4: Distribution of max token importance score in the AGNews test set.

As mentioned in Section 2.1, attack methods need to first rank token importance based on its influence on prediction. Specifically, the importance score is calculated by comparing the change of model prediction probablities after removing

ding layer by sending embeddings directly to the target model.

each word. In Figure 4, we compare the density distribution of max token importance score between FreeLB++ and its DiffuseDef counterpart. DiffuseDef shows a notably lower percentage than FreeLB++ when the max importance score is high, where the attacker can easily find the vulnerable token to construct adversarial examples. This difference shows that DiffuseDef can complicate the process of important word searching, which accounts for the increased number of queries required for a successful attack.

Method	L2	Cosine
FreeLB++	12.53	0.35
DiffuseDef-FreeLB++	10.66	0.27
RSMI	9.72	0.24
DiffuseDef-RSMI	8.61	0.21

Table 3: L2 and cosine distance between hidden states for clean and adversarial texts.

In addition, DiffuseDef mitigates the difference between clean and adversarial texts by reducing the distance between their hidden states. In Table 3, we report the L2 and cosine distance between clean and adversarial hidden states for FreeLB++ and RSMI. Both show lower L2 and cosine distance after applying DiffuseDef, indicating that ensembling diffused representation repositions the adversarial example closer to the clean example, leading to the model maintaining its predictions.

5.6 Efficiency of DiffuseDef

Method	Params	FLOPS	Train
Fine-Tuned (BERT)	110M	46G	1x
EarlyRobust	82M	32G	0.8x
FreeLB++	110M	46G	10.5x
InfoBERT	110M	46G	6.5x
RSMI	110M	92G	1.25x
RanMask ($k = 10$)	110M	459G	1.2x
SAFER ($k = 10$)	110M	459G	1x
DiffuseDef ($t' = 1, k = 10$)	120M	96G	1.1x
DiffuseDef ($t' = 5, k = 10$)	120M	267G	1.1x

Table 4: Efficiency comparison of DiffuseDef and other methods. Params: number of model parameters. FLOPS: number of floating point operations per second at inference time, calculated with batch size of 1 and sequence length of 256. Train: training time.

Given that DiffuseDef adds additional denoising and ensembling steps during inference, it inevitably increases the computation time compared to its base model. To study its efficiency, we report the number of model parameters, inference FLOPS, and training time in Table 4. In addition to the defense methods in Table 1, we also compare the efficiency of DiffuseDef with two other SOTA ensembling-based defense methods, i.e. RanMask (Zeng et al., 2023) and SAFER (Ye et al., 2020).

All SOTA models have the same number of parameters as the fine-tuned BERT model, except EarlyRobust which applies attention head pruning for better efficiency. DiffuseDef requires more inference FLOPS than non ensembling-based baselines such as FreeLB++ and EarlyRobust, due to the diffusion steps. With $t' = 1$ and $k = 10$, the FLOPS for DiffuseDef doubles from 46G to 96G, nevertheless, this number is close to RSMI model as it requires gradient information during inference. Despite this increase, DiffuseDef is more efficient than ensembling-based methods like RanMask and SAFER which require a full forward pass for all ensembles. With the same ensembling number of 10, both RanMask and SAFER require 459G FLOPS, which is 10x the number for BERT baseline. In contrast, even with t' increased to 5, DiffuseDef can be run faster with 267G FLOPS, mitigating the efficiency problem from ensembling while maintaining the benefit of improved robustness.

Regarding training time, training the diffusion layer although requires more epochs, it cost similar time as training the classifier due to the small size of the diffusion layer. By contrast, adversarial training methods such as FreeLB++ and InfoBERT require much longer training time.

6 Conclusions

We propose a novel adversarial defense method, DiffuseDef, which combines adversarial training, diffusion training, and ensembling to improve model robustness to adversarial attacks. DiffuseDef can build on any existing adversarial training method, training an additional diffusion layer to predict and remove randomly sampled noise at a given timestep. During inference, the diffusion layer is used to denoise the adversarial hidden states, which are ensembled to construct a robust text representation. Our experiments validate the effectiveness and efficiency of DiffuseDef, which significantly outperforms SOTA on common black-box and white-box adversarial attack methods. Analysis shows that DiffuseDef makes it difficult to find vulnerable tokens to attack, and also reduces the difference between the hidden representations of clean and adversarial texts.

7 Limitations

Despite the fact that DiffuseDef is more efficient than existing ensembling-based methods, it still requires more model parameters and inference FLOPS than non-ensembling-based models to achieve a better robustness. Future directions of this work might involve efforts to reduce the size of diffusion layer and number of ensembles to make DiffuseDef more efficient.

8 Ethical Considerations

In this paper we propose a new method DiffuseDef which uses a diffusion layer as a denoiser to provide robust and efficient text representation. We demonstrate that the proposed method could significantly improve the robustness of NLP systems to adversarial attacks. However, DiffuseDef cannot defend against all adversarial attacks without limitations (e.g. number of perturbed words, semantic similarity between original and adversarial examples). Potential risks might include creation of new adversarial attacks devised specifically for DiffuseDef.

References

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

Rongzhou Bao, Jiayi Wang, and Hai Zhao. 2021. Defending pre-trained language models from adversarial word substitution without performance sacrifice. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3248–3258, Online. Association for Computational Linguistics.

Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. *Preprint*, arXiv:1608.04644.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

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

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Brian Formento, Wenjie Feng, Chuan-Sheng Foo, Anh Tuan Luu, and See-Kiong Ng. 2024. SemRoDe: Macro adversarial training to learn representations that are robust to word-level attacks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8005–8028, Mexico City, Mexico. Association for Computational Linguistics.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy. Association for Computational Linguistics.

J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.

SongYang Gao, Shihan Dou, Yan Liu, Xiao Wang, Qi Zhang, Zhongyu Wei, Jin Ma, and Ying Shan. 2023. DSRM: Boost textual adversarial training with distribution shift risk minimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12177–12189, Toronto, Canada. Association for Computational Linguistics.

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.

Mert Gulsen, Batuhan Cengiz, Yusuf H. Sahin, and Gozde Unal. 2024. Pcld: Point cloud layerwise diffusion for adversarial purification. *Preprint*, arXiv:2403.06698.

Ashim Gupta, Carter Wood Blum, Temma Choji, Yingjie Fei, Shalin Shah, Alakananda Vempala, and Vivek Srikumar. 2023. Don't retrain, just rewrite: Countering adversarial perturbations by rewriting text. *arXiv preprint arXiv:2305.16444*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1022–1033, Online. Association for Computational Linguistics.

Daniel Jiwoong Im, Sungjin Ahn, Roland Memisevic, and Yoshua Bengio. 2016. Denoising criterion for variational auto-encoding framework. *Preprint*, arXiv:1511.06406.

Jiabao Ji, Bairu Hou, Zhen Zhang, Guanhua Zhang, Wenqi Fan, Qing Li, Yang Zhang, Gaowen Liu, Sijia Liu, and Shiyu Chang. 2024. Advancing the robustness of large language models through self-denoised smoothing. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 246–257, Mexico City, Mexico. 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.

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, Demin Song, and Xipeng Qiu. 2023. Text adversarial purification as defense against adversarial attacks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 338–350, Toronto, Canada. Association for Computational Linguistics.

Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021. 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.

Aiwei Liu, Honghai Yu, Xuming Hu, Shu'ang Li, Li Lin, Fukun Ma, Yawen Yang, and Lijie Wen. 2022. Character-level white-box adversarial attacks against transformers via attachable subwords substitution. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7664–7676, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Han Liu, Zhi Xu, Xiaotong Zhang, Xiaoming Xu, Feng Zhang, Fenglong Ma, Hongyang Chen, Hong Yu, and Xianchao Zhang. 2023. Sspattack: A simple and sweet paradigm for black-box hard-label textual adversarial attack. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):13228–13235.

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. *Preprint*, arXiv:1907.11692.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011a. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011b. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2019. Towards deep learning models resistant to adversarial attacks. *Preprint*, arXiv:1706.06083.

Han Cheol Moon, Shafiq Joty, Ruochen Zhao, Megh Thakkar, and Chi Xu. 2023. Randomized smoothing with masked inference for adversarially robust text classifications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5145–5165, Toronto, Canada. 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, Online. Association for Computational Linguistics.

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.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. 2022. Diffusion models for adversarial purification. In *International Conference on Machine Learning (ICML)*.

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.

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

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

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 *International Conference on Learning Representations*.

Boxin Wang, Chejian Xu, Xiangyu Liu, Yu Cheng, and Bo Li. 2022. SemAttack: Natural textual attacks via different semantic spaces. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 176–205, Seattle, United States. Association for Computational Linguistics.

Zhaoyang Wang, Zhiyue Liu, Xiaopeng Zheng, Qinliang Su, and Jiahai Wang. 2023. RMLM: A flexible defense framework for proactively mitigating word-level adversarial attacks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2757–2774, Toronto, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhiheng Xi, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022. Efficient adversarial training with robust early-bird tickets. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8318–8331, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ying Xu, Xu Zhong, Antonio Jimeno Yepes, and Jey Han Lau. 2021. Grey-box adversarial attack and defence for sentiment classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4078–4087, Online. Association for Computational Linguistics.

Mao Ye, Chengyue Gong, and Qiang Liu. 2020. SAFER: A structure-free approach for certified robustness to adversarial word substitutions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3465–3475, Online. Association for Computational Linguistics.

Jin Yong Yoo and Yanjun Qi. 2021. Towards improving adversarial training of NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 945–956, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lifan Yuan, YiChi Zhang, Yangyi Chen, and Wei Wei. 2023. Bridge the gap between CV and NLP! a gradient-based textual adversarial attack framework. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7132–7146, Toronto, Canada. Association for Computational Linguistics.

Shilong Yuan, Wei Yuan, Hongzhi Yin, and Tieke He. 2024. Roic-dm: Robust text inference and classification via diffusion model. *Preprint*, arXiv:2401.03514.

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.

Jiehang Zeng, Jianhan Xu, Xiaoqing Zheng, and Xuanjing Huang. 2023. Certified robustness to text adversarial attacks by randomized [MASK]. *Computational Linguistics*, 49(2):395–427.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015a. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA. MIT Press.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2021. Defense against synonym substitution-based adversarial attacks via Dirichlet neighborhood ensemble. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5482–5492, Online. 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 *International Conference on Learning Representations*.

A Data Preparation

Dataset	Train	Valid	Test	Avg Len
AGNews	108K	12K	7K	51.3
IMDB	40K	5K	5K	311.9
QNLI	94K	10K	5K	47.2

Table 5: Dataset statistics. The average text length is counted with BertTokenizer.

Table 5 presents the number of examples in train/valid/test splits and the average token length for the three datasets used in the experiments. For QNLI and AGNews datasets, we randomly split the training set into our train/valid splits, with a ratio of 0.9/0.1, and use its test set as our test split. For IMDB dataset, we randomly split the dataset into train/valid/test splits with a ratio of 0.8/0.1/0.1. All train/valid/test splitting is performed using a random seed of 42.

B Evaluation Constraints

Dataset	$\varepsilon_{\mathbf{min}}$	$\mathbf{K}_{\mathbf{max}}$	$\rho_{\mathbf{max}}$
AGNews	0.84	50	0.3
IMDB	0.84	50	0.1
QNLI	0.84	50	0.2

Table 6: Evaluation parameters for each dataset.

When evaluating with adversarial attack, We follow the parameter settings for TextAttack as suggested in (Li et al., 2021). The minimum semantic similarity $\varepsilon_{\mathbf{min}}$ between the clean text and adversarial text is set to 0.84, with the score computed using Universal Sentence Encoder (Cer et al., 2018). The maximum number of candidate substitution $\mathbf{K}_{\mathbf{max}}$ from attacker is 50, thus the maximum number of queries $\mathbf{Q}_{\mathbf{max}} = \mathbf{K}_{\mathbf{max}} \times \mathbf{L}$ where $\mathbf{L}$ is the number of tokens. Finally, the maximum percentage of changed tokens $\rho_{\mathbf{max}}$ is set to 0.3/0.1/0.2 for AGNews, IMDB, and QNLI dataset respectively.

C Training

	AGNews	IMDB	QNLI
Epochs	100	100	100
Batch size	64	64	64
Sequence len	128	256	256
Dropout	0.1	0.1	0.1
Optimizer	AdamW	AdamW	AdamW
Lr	2e-5	2e-5	2e-5
t	30	10	30
t'	5	5	5
k	10	10	10

Table 7: Hyperparameters for training `DiffuseDef`.

The details on hyper-parameters of diffusion training can be found in Table 7. All models are trained on a single RTX A6000 GPU. The diffusion training of 100 epochs takes 6/4/3 hours on AGNews, IMDB, QNLI datasets respectively.

D License for Scientific Artifacts

Table 8 lists the scientific artifacts including data, codes, and models used in this paper. The use of

Artifact	License
AGNews (Zhang et al., 2015b)	Custom (non-commercial)
IMDB (Maas et al., 2011b)	-
QNLI (Wang et al., 2018)	CC BY-SA 4.0
transformers (Wolf et al., 2020)	Apache License 2.0
TextAttack (Morris et al., 2020)	MIT License
BERT (Devlin et al., 2019)	Apache License 2.0
RoBERTa (Liu et al., 2019)	MIT License

Table 8: Licenses of scientific artifacts used in this paper.

these artifacts in this paper is consistent with their intended use, i.e. for scientific research only. The data used in the experiment is in English and does not contain personally identifying info or offensive content.

E Robustness with RoBERTa

In Table 9, we report the black-box attack results on RoBERTa-based models. Similar conclusion holds for BERT-based models, and DiffuseDef significantly outperforms other defense methods on all 3 datasets.

F Robustness w.r.t Token Length

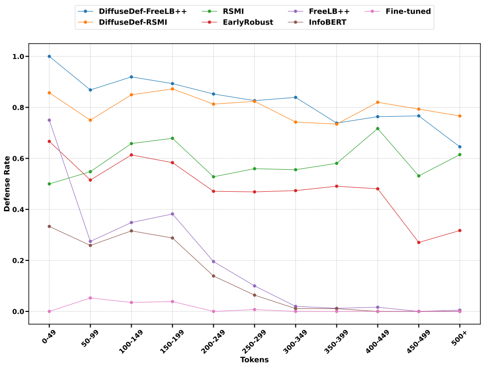

Figure 5: Defense rate (against TextFooler) w.r.t token length for different models on IMDB dataset.

Figure 5 provides comparison of defense rate for different models by token length on the IMDB dataset. The defense rate is calculated as the percentage of test examples in which TextFooler fails to construct a successful attack. All models except RSMI show a consistent trend that the defense rate declines as the texts lengthen. This trend can be attributed to the nature of adversarial attacks as longer texts allow for the generation of more adversarial examples. Specifically, adversarial training defense methods like InfoBERT and FreeLB++ show poor performance on longer texts (more than 300 tokens), with the defense rate reduced to near 0. This drastic decline indicates that given an adequate number of queries, the attacker is guaranteed to find a successful attack to fool these models. Similarly, EarlyRobust exhibits a performance drop on long texts as it is based on FreeLB training. RSMI, however, performs worse on short texts, but its defense rate increases as the text length grows. Compared to all SOTA defense approaches, the two DiffuseDef variants show a more steadily declining trend and maintain a higher defense rate across all token lengths, i.e. DiffuseDef is more robust to input text length.

G Example of noising and denoising in DiffuseDef

Adding and removing noise to hidden states are essential features in DiffuseDef which contribute to the improved adversarial robustness. To study how adding or removing noise can affect the semantic meaning of the text, we feed the hidden states to the pretrained BERT model with masked language modeling (MLM) head to generate the text output.

In Table 10, we present the MLM outputs from hidden states added with different steps of noise and the MLM outputs from noise hidden states denoised with same number of steps. In the example shown, with more noise added some semantic information can be lost and replaced by symbols or function words like "." or "the". In contrast, denoising for the same number of steps help alleviate such information lost. For example, the word "IBM" can be recovered from the noise.

However, in practise it is not possible to assume number of denoising steps therefore in Table 11 we show the MLM outputs of denoised hidden states directly from clean and adversarial texts. On clean text, we observe that a higher number of denoising steps can result in more abstraction of the texts. For example, more words are replaced with "the" in the MLM outputs as t' grows. However, words related to the topic (e.g. "Manchester United", "Liverpool") are kept during the denoising process, thus the model can predict correctly. Similarly, the trend of abstraction can be also found on adversarial text while we observe that the denoising can help remove the adversarial noise / perturbation and recover the word "united" from "nation", thus resulting its correct prediction on the adversarial text.

Dataset	PLM	Method	Clean%	AUA%			#Query		
				TF	TB	BA	TF	TB	BA
AGNews	RoBERTa-base	Fine-Tuned	94.9	34.1	36.9	43.6	372	396	410
		InfoBERT	95.5	40.2	45.2	48.6	392	421	430
		FreeLB++	95.4	57.5	62.9	55.9	444	467	447
		RSMI	93.1	64.2	66.4	67.4	774	861	808
		`DiffuseDef-FreeLB++` (Ours)	95.3	**85.6**	**87.6**	**85.3**	880	**976**	906
		`DiffuseDef-RSMI` (Ours)	92.9	82.9	83.5	82.2	**905**	925	**1047**
IMDB	RoBERTa-base	Fine-Tuned	94.6	21.3	17.9	13.6	587	671	493
		InfoBERT	94.8	30.9	27.9	21.8	681	760	549
		FreeLB++	95.3	46.0	42.1	33.9	829	974	637
		RSMI	92.7	77.9	74.3	70.6	3443	4342	2619
		`DiffuseDef-FreeLB++` (Ours)	95.0	**86.2**	**85.9**	**86.8**	3573	4663	2941
		`DiffuseDef-RSMI` (Ours)	92.4	84.7	84.1	84.3	**3673**	**4782**	**3007**
QNLI	RoBERTa-base	Fine-Tuned	92.8	26.6	22.5	20.3	204	219	188
		InfoBERT	92.5	29.1	25.8	20.3	205	223	189
		FreeLB++	92.8	33.9	27.7	20	227	244	200
		RSMI	89.3	38.8	33.2	30.0	340	375	343
		`DiffuseDef-FreeLB++` (Ours)	92.7	64.6	64.3	61.5	473	579	524
		`DiffuseDef-RSMI` (Ours)	88.8	57.7	55.7	53.9	469	578	518

Table 9: Main adversarial robustness results on classification tasks with RoBERTa. *Clean*: accuracy on clean test set. *TF*: TextFooler. *TB*: TextBugger. *BA*: BertAttack.

t'	MLM Output (add noise)	MLM Output (add noise then denoise)
0	IBM Chips May Someday Heal Themselves New technology applies electrical fuses to help identify and repair faults.	-
5	the ibm chips may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.	the ibm chips may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.
6	) ibm chips may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.	the ibm chips may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.
7	the. chips may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.	the. chips may someday heal themselves new technology uses electrical fuses to help identify and repair faults.
8	).. may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.	the ibm. may someday heal themselves new technology uses electrical fuses to help identify and repair faults.
9	the ibm chips may someday heal themselves new technology uses electrical fuses to help identify and repair faults.	the ibm chips may someday heal themselves new technology introduces electrical fuses to help identify and repair faults.
10	the. chips may someday heal themselves new technology extends electrical fuses to help identify and repair faults.	the ibm. may someday heal themselves new technology develops electrical fuses to help identify and repair faults.

Table 10: MLM outputs from hidden states with noise added and hidden states with first noise added but then denoised. We only report t' above 5 as the MLM outputs with smaller t' are identical to the clean text.

H Confusion Matrix under Attack

Figure 6 and 7 present the confusion matrixes of models prediction on clean text and on adversarial texts (successful attack example) on AGNews and IMDB test sets respectively.

t′	Clean Text / MLM Output	Adv Text / MLM Output	Pred clean	Pred adv
0	United Apology over Website Abuse Manchester United have been forced to issue an embarrassing apology to Liverpool for an ill-advised attack on the Anfield outfit on its own website.	United Apology over Website Abuse Manchester Nations have been forced to issue an embarrassing apology to Liverpool for an ill-advised attack on the Anfield outfit on its own website.	Sports	World
1	football. apology over website abuse manchester united have been - to issue an embarrassing apology to liverpool for an the - advised attack on the anfield outfit on its own website.	the. apology over website abuse manchester nations have been the to issue an embarrassing apology to liverpool for an the - advised attack on the anfield outfit on its own website.	Sports	World
2	the. apology over website abuse manchester united have been - to issue an embarrassing apology to liverpool for an the - advised attack on the anfield outfit on its own website.	the. apology over website abuse manchester nations have been the to issue an embarrassing apology to liverpool for an the - advised attack on the anfield outfit on its own website.	Sports	World
3	the. apology over website abuse manchester united have been the to issue an embarrassing apology to liverpool for an the - advised attack on the anfield outfit on its own website.	the. apology over website abuse manchester s have been the to issue an embarrassing apology to liverpool for an the - advised attack on the anfield outfit on its own website.	Sports	Sports
4	the. apology over website abuse manchester united have the - to issue an embarrassing apology to liverpool for an the - advised attack on the anfield outfit on its own website.	the. apology over website abuse manchester s have been the to issue an embarrassing apology to liverpool for an the - advised attack on the anfield outfit on its own website.	Sports	Sports
5	the. apology over website abuse manchester united have the the to issue an a apology to liverpool for an the - advised attack on the anfield outfit on its own website.	the. apology over website abuse manchester united have been the to issue an the apology to liverpool for an’- advised attack on the anfield outfit on its own website.	Sports	Sports

Table 11: MLM outputs and FreeLB++ model predictions from ensembling diffused hidden states at different denoising steps.

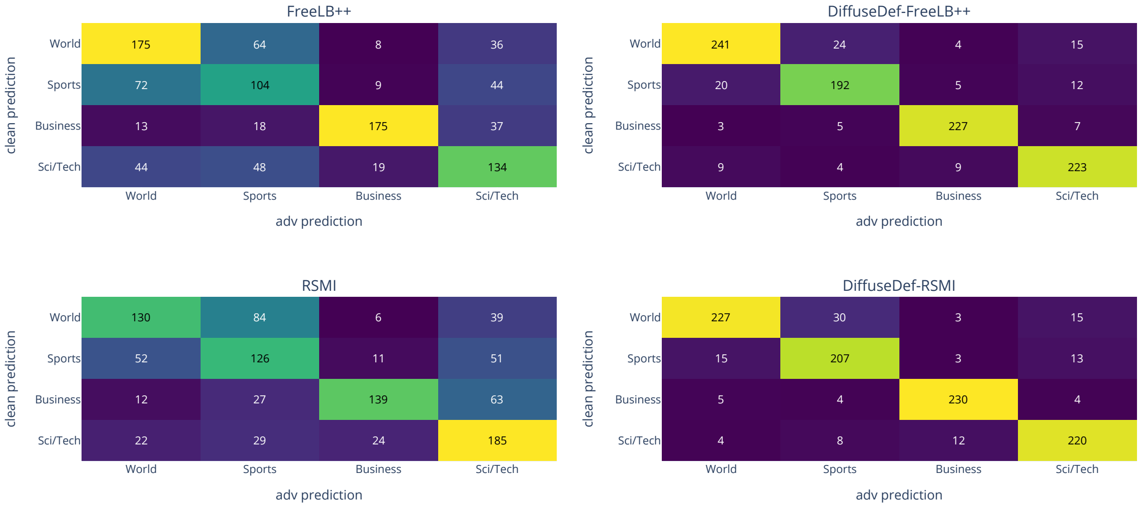

Figure 6: Confusion matrix of models under attack on AGNews test set.

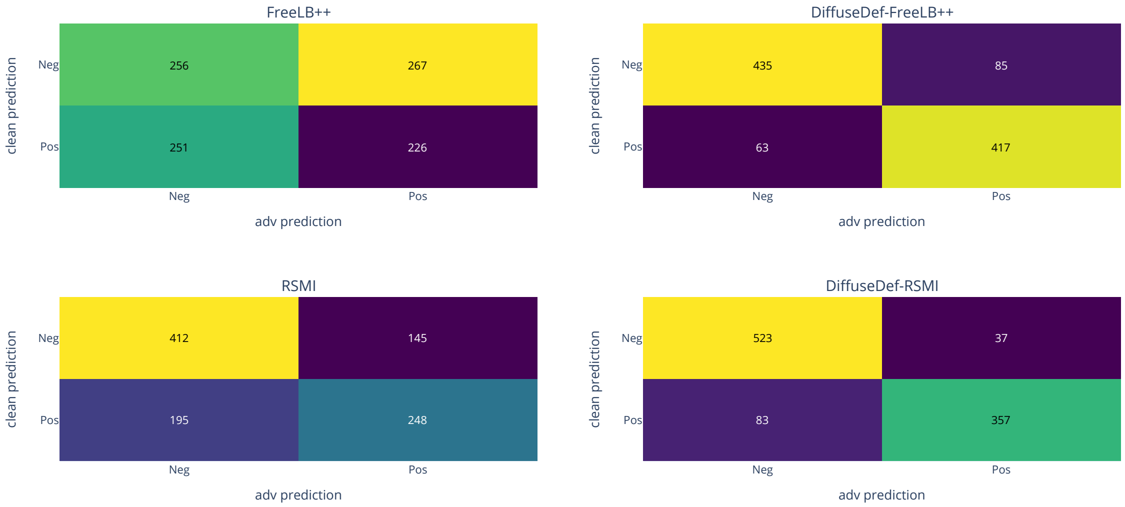

Figure 7: Confusion matrix of models under attack on IMDB test set.