DA^3: A Distribution-Aware Adversarial Attack against Language Models

Anonymous ACL submission

Abstract

Language models can be manipulated by adversarial attacks, which introduce subtle perturbations to input data. While recent attack methods can achieve a relatively high attack success rate (ASR), we've observed that the generated adversarial examples have a different data distribution compared with the original examples. Specifically, these adversarial examples exhibit reduced confidence levels and greater divergence from the training data distribution. Consequently, they are easy to detect using straightforward detection methods, diminishing the efficacy of such attacks. To address this issue, we propose a Distribution-Aware Adversarial Attack (DA^3) method. DA^3 considers the distribution shifts of adversarial examples to improve attacks' effectiveness under detection methods. We further design a novel evaluation metric, the Non-detectable Attack Success Rate (NASR), which integrates both ASR and detectability for the attack task. We conduct experiments on four widely used datasets to validate the attack effectiveness and transferability of adversarial examples generated by DA^3 against both the white-box BERT-BASE and ROBERTA-BASE models and the black-box LLAMA2-7B model[1].

1 Introduction

Language models (LMs), despite their remarkable accuracy and human-like capabilities in many applications, face vulnerability to adversarial attacks and exhibit high sensitivity to subtle input perturbations, which can potentially cause failures (Jia and Liang, 2017; Belinkov and Bisk, 2018; Wallace et al., 2019). Recently, an increasing number of adversarial attacks have been proposed, employing techniques such as insertion, deletion, swapping, and substitution at character, word, or sentence levels (Ren et al., 2019; Jin et al., 2020; Garg and Ramakrishnan, 2020; Ribeiro et al., 2020). These thoroughly crafted adversarial examples are imperceptible to humans yet can deceive victim models, thereby raising concerns regarding the robustness and security of LMs. For example, chatbots may misunderstand user intent or sentiment, resulting in inappropriate responses (Perez et al., 2022).

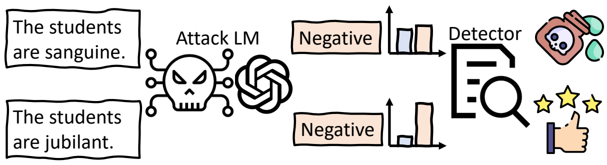


Figure 1: Toy examples of two adversarial sentences in a sentiment analysis task. Although both sentences successfully attack the victim model, the top one is flagged by the detector, while the bottom one is not detected. In our task, we aim to generate adversarial examples that are hard to detect.

However, while existing adversarial attacks can achieve a relatively high attack success rate (Gao et al., 2018; Belinkov and Bisk, 2018; Li et al., 2020), our experimental observations detailed in §3 reveal notable distribution shifts between adversarial examples and original examples, rendering high detectability of adversarial examples. On one hand, adversarial examples exhibit different confidence levels compared to their original counterparts. Typically, the Maximum Softmax Probability (MSP), a metric indicating prediction confidence, is higher for original examples than for adversarial examples. On the other hand, there is a disparity in the distance to the training data distribution between adversarial and original examples. Specifically, the Mahalanobis Distance (MD) to training data distribution for original examples is shorter than that for adversarial examples. Based on these two observations, we conclude that adversarial examples generated by previous attack methods, such as BERT-Attack (Li et al., 2020), can be easily detected through score-based detection techniques like MSP detection (Hendrycks and Gimpel, 2017)

[1] Our codes are available at https://anonymous.4open.science/r/DALA-A16D/.

and embedding-based detection methods like MD detection (Lee et al., 2018). Thus, the efficacy of previous attack methods is diminished when considering Out-of-distribution (OOD) detection, as shown in Figure 1.

To address the aforementioned problems, we propose a **D**istribution-**A**ware **A**dversarial **A**ttack (DA^3) method with Data Alignment Loss (DAL), which is a novel attack method that can generate hard-to-detect adversarial examples. The DA^3 framework comprises two phases. Firstly, DA^3 fine-tunes a LoRA-based LM by combining the Masked Language Modeling task and the downstream classification task using DAL. This fine-tuning phase enables the LoRA-based LM to generate adversarial examples closely resembling original examples in terms of MSP and MD. Subsequently, the LoRA-based LM is used during inference to generate adversarial examples.

To measure the detectability of adversarial examples, we propose a new evaluation metric: **N**on-detectable **A**ttack **S**uccess **R**ate (NASR), which combines Attack Success Rate (ASR) with OOD detection. We conduct experiments on four datasets to assess whether DA^3 can effectively attack white-box LMs using ASR and NASR. Furthermore, given the widespread use of Large Language Models (LLMs) and their costly fine-tuning process, coupled with the limited availability of open-source models, we also evaluate the attack transferability of adversarial examples on black-box LLMs. The results show that DA^3 achieves competitive attack performance on the white-box BERT-BASE (Devlin et al., 2019) and RoBERTa-BASE (Liu et al., 2019) models and superior transferability on the black-box LLaMA2-7B (Touvron et al., 2023).

Our work has the following contributions:

- We analyze the distribution of adversarial and original examples, revealing the existence of distribution shifts in terms of MSP and MD.
- We propose a novel Distribution-Aware Adversarial Attack method with Data Alignment Loss, which is capable of generating adversarial examples that effectively undermine victim models while remaining difficult to detect.
- We design a new evaluation metric – NASR – for the attack task, which considers the detectability of adversarial examples.
- We conduct comprehensive experiments to compare DA^3 with baselines on four datasets, demonstrating that DA^3 achieves competitive attack capabilities and better transferability.

2 Related Work

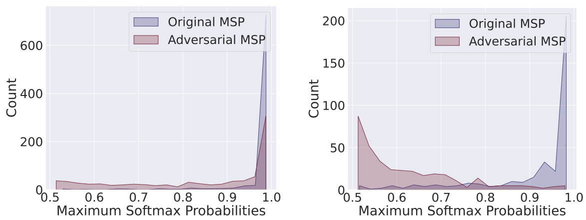
2.1 Adversarial Attacks in NLP

Adversarial attacks have been extensively studied to explore the robustness of LMs. Current methods fall into character-level, word-level, sentence-level, and multi-level (Goyal et al., 2023). Character-level methods manipulate texts by incorporating typos or errors into words, such as deleting, repeating, replacing, swapping, flipping, inserting, and allowing variations in characters for specific words (Gao et al., 2018; Belinkov and Bisk, 2018). Word-level attacks alter entire words rather than individual characters within words. Common manipulation includes addition, deletion, and substitution with synonyms to mislead language models while the manipulated words are selected based on gradients or importance scores (Ren et al., 2019; Jin et al., 2020; Li et al., 2020; Garg and Ramakrishnan, 2020). Sentence-level attacks typically involve inserting or rewriting sentences within a text, all while preserving the original meaning (Zhao et al., 2018; Iyyer et al., 2018; Ribeiro et al., 2020). Multi-level attacks combine multiple perturbation techniques to achieve both imperceptibility and a high success rate in the attack (Song et al., 2021).

2.2 Out-of-distribution Detection in NLP

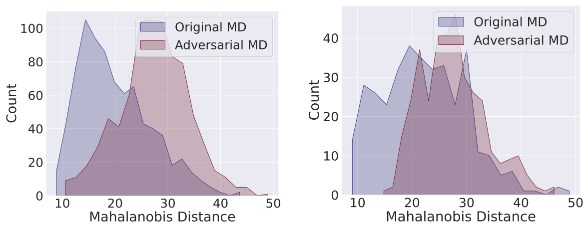
Out-of-distribution (OOD) detection methods have been widely explored in NLP, like machine translation (Arora et al., 2021; Ren et al., 2022; Adila and Kang, 2022). OOD detection methods in NLP can be roughly categorized into two types: (1) score-based methods and (2) embedding-based methods. Score-based methods use maximum softmax probability (Hendrycks and Gimpel, 2017), perplexity score (Arora et al., 2021), beam score (Wang et al., 2019b), sequence probability (Wang et al., 2019b), BLEU variance (Xiao et al., 2020), or energy-based scores (Liu et al., 2020). Embedding-based methods measure the distance to in-distribution data in the embedding space for OOD detection. For example, Lee et al. (2018) uses Mahalanobis distance; Ren et al. (2021) proposes to use relative Mahalanobis distance; Sun et al. (2022) proposes a nearest-neighbor-based OOD detection method.

We select the simple, representative, and widely-used OOD detection methods of these two categories: MSP detection (Hendrycks and Gimpel, 2017) and MD detection (Lee et al., 2018), respec-



(a) MSP on SST-2 dataset. (b) MSP on MRPC dataset.

Figure 2: Visualization of the distribution shift between original data and adversarial data generated by BERT-Attack when attacking BERT-BASE regarding MSP.



(a) MD on SST-2 dataset. (b) MD on MRPC dataset.

Figure 3: Visualization of the distribution shift between original data and adversarial data generated by BERT-Attack when attacking BERT-BASE regarding MD.

tively. This selection serves to highlight a significant issue within the community – the ability to detect adversarial examples using such basic and commonly employed OOD detection methods underscores the criticality of detectability. These two methods are then incorporated with the ASR to assess the robustness and detectability of adversarial examples generated by different attack models.

3 Understanding Distribution Shifts of Adversarial Examples

This section showcases distribution shifts between adversarial and original examples, suggesting that the original examples are in-distribution examples while adversarial examples are Out-of-Distribution (OOD) examples. Due to space constraints, we focus our analysis on adversarial examples generated by BERT-Attack on SST-2 (Socher et al., 2013) and MRPC (Dolan and Brockett, 2005); the complete results are available in Appendix G.

Maximum Softmax Probability (MSP). Maximum Softmax Probability (MSP) is a metric to evaluate prediction confidence, rendering it a widely used score-based method for OOD detection, where lower confidence values often signify OOD examples. To assess MSP, we visualize the MSP distribution of adversarial examples generated by BERT-Attack and original examples from SST-2 and MRPC datasets in Figure 2. Our observation reveals that in both datasets, the majority of original examples have an MSP exceeding 0.9, indicating a significantly higher MSP compared to adversarial examples overall. This distribution shift is particularly notable in the MRPC dataset, whereby most adversarial examples exhibit MSP below 0.6, highlighting a clear distinction from the original examples.

Mahalanobis Distance (MD). Mahalanobis Distance (MD) is a metric used to measure the distance between a data point and a distribution, making it a highly suitable and widespread method for OOD detection. A high MD between an example and the in-distribution data (training data) indicates that the example is probably an OOD instance. To assess the MD difference between adversarial and original examples, we visualize the MD distribution of adversarial examples generated by BERT-Attack and original examples from the SST-2 and MRPC datasets in Figure 3. From Figure 3, we can observe that distribution shifts exist between original and adversarial examples in both datasets. This dissimilarity is more noticeable on the SST-2 dataset and not as conspicuous on the MRPC dataset.

Summary. These observations regarding MSP and MD highlight clear distinctions between original and adversarial examples generated by one of the state-of-the-art methods, BERT-Attack. Compared to the original examples, the adversarial examples exhibit a more pronounced OOD nature in either MSP or MD, meaning that adversarial examples are easy to detect and the practical effectiveness of previous attack methods is diminished.

4 Methodology

In this section, we define the attack task (§4.1), propose a novel attack method called Distribution-Aware Adversarial Attack (§4.2), and introduce the new Data Alignment Loss (§4.3).

4.1 Problem Formulation

Given an original sentence $x^{orig} \in \mathcal{X}$ and its corresponding original label $y^{orig} \in \mathcal{Y}$, our objective is to generate an adversarial sentence x^{adv} such that the prediction of the victim model corresponds to $y^{adv} \in \mathcal{Y}$ and $y^{adv} \neq y^{orig}$.

4.2 Distribution-Aware Adversarial Attack

Motivated by the observed distribution shifts of adversarial examples, we propose a Distribution-

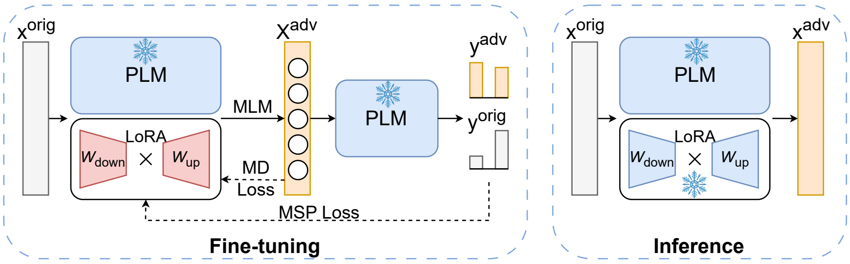


Figure 4: The model architecture of DA[3] comprises two phases: fine-tuning and inference. During fine-tuning, a LoRA-based PLM is fine-tuned to develop the ability to generate adversarial examples resembling original examples in terms of MSP and MD. During inference, the LoRA-based PLM is used to generate adversarial examples.

Aware Adversarial Attack (DA^3) method. The key idea of DA^3 is to consider the distribution of the generated adversarial examples and attempt to achieve a closer alignment between distributions of adversarial and original examples in terms of MSP and MD. DA^3 is composed of two phases: fine-tuning and inference, as shown in Figure 4.

Fine-tuning Phase. The fine-tuning phase aims to fine-tune a LoRA-based Pre-trained Language Model (PLM) to make it capable of generating adversarial examples through the Masked Language Modeling (MLM) task. We employ LoRA-based PLM because it is efficient to finetune and the frozen PLM can serve in both MLM and downstream classification tasks. First, the original sentence x^{orig} undergoes the MLM task through a LoRA-based PLM to generate the adversarial embedding X^{adv}, during which the parameters of the PLM are frozen, and the parameters of LoRA (Hu et al., 2021) are tunable. Then, the generated adversarial embedding X^{adv} is fed into the frozen PLM to perform the corresponding downstream classification task, producing logits of original ground truth label y^{orig} and adversarial label y^{adv}. The loss is computed based on X^{adv}, $P(y^{orig}|X^{adv},\theta)$, and $P(y^{adv}|X^{adv},\theta)$ to update the parameters of LoRA, where θ is the model parameters. Details are discussed in §4.3.

Inference Phase. The inference phase aims to generate adversarial examples with minimal perturbation. The original sentence x^{orig} is first tokenized, and a ranked token list is obtained through token importance (Li et al., 2020). Then, a token is selected from the token list to be masked. Subsequently, the MLM task of the frozen LoRA-based PLM is employed to generate a candidate list for the masked token. A word is then chosen from the list to replace the masked token until a successful attack on the victim model is achieved or the candidate list is exhausted. If the attack is unsuccessful, another token is chosen from the token list until a successful attack is achieved or the termination condition is met. The termination condition is set as the percentage of the tokens.

4.3 Model Learning

The Data Alignment Loss, denoted as $\mathcal{L}_{DAL}$, is used to minimize the discrepancy between distributions of adversarial examples and original examples in terms of MSP and MD. $\mathcal{L}_{DAL}$ is composed of two losses: $\mathcal{L}_{MSP}$ and $\mathcal{L}_{MD}$.

$\mathcal{L}_{MSP}$ aims to increase the difference between $P(y^{adv}|X^{adv},\theta)$ and $P(y^{orig}|X^{adv},\theta)$. $\mathcal{L}_{MSP}$ is formulated as

$$\mathcal{L}_{MSP} = \sum_{X^{adv}} \frac{exp(P(y^{orig}|X^{adv},\theta))}{exp(P(y^{orig}|X^{adv},\theta))+exp(P(y^{adv}|X^{adv},\theta))}.$$

According to our observation experiments in Figure 2, original examples have higher MSP than adversarial examples. Minimizing $\mathcal{L}_{MSP}$ increases MSP of adversarial examples. Thus, minimizing $\mathcal{L}_{MSP}$ makes generated adversarial examples more similar to original examples concerning MSP.

$\mathcal{L}_{MD}$ aims to reduce MD between adversarial input and the training data distribution. $\mathcal{L}_{MD}$ is formulated as:

$$\mathcal{L}_{MD} = \sum_{X^{adv}} log\sqrt{(X^{adv}-\mu)\sum^{-1}(X^{adv}-\mu)^\intercal},$$

where μ and $\sum^{-1}$ are the mean and covariance embedding of the in-distribution (training) data respectively. MD is a robust metric for OOD detection and adversarial data detection. In general, adversarial data has higher MD than original data, as shown in Figure 3. Therefore, minimizing $\mathcal{L}_{MD}$ encourages the generated adversarial examples to resemble original examples in terms of MD. $\mathcal{L}_{MD}$ is constrained to the logarithmic space for consistency with the scale of $\mathcal{L}_{MSP}$.

Thus, Data Alignment Loss is represented as

$$\mathcal{L}_{DAL} = \mathcal{L}_{MSP} + \mathcal{L}_{MD}, \tag{1}$$

and DA^3 is trained by optimizing $\mathcal{L}_{DAL}$.

5 Automatic Evaluation Metrics

Given the observations of distribution shifts analyzed in Section 3, we adopt a widely-used metric – Attack Success Rate (ASR) – and design a new metric – Non-detectable Attack Success Rate (NASR) – to evaluate attack performance. We also report the Percentage of Perturbed Words (%Words) and Semantic Similarity (SS) to evaluate the impact of text perturbation. Detailed explanations of ASR, %Words, and SS are shown in Appendix A.

Non-detectable Attack Success Rate (NASR). Considering the detectability of adversarial examples generated by attack methods, we define a new evaluation metric – Non-Detectable Attack Success Rate (NASR). This metric considers both ASR and OOD detection. Specifically, NASR posits that a successful adversarial example is characterized by its ability to deceive the victim model while simultaneously evading OOD detection methods.

We utilize two established and commonly employed OOD detection techniques – MSP detection (Hendrycks and Gimpel, 2017) and MD detection (Lee et al., 2018). MSP detection relies on logits and utilizes a probability distribution-based approach, while MD detection is a distance-based approach. For MSP detection, we use Negative MSPs, calculated as $-\max_{y \in \mathcal{Y}} P(y \mid X, \theta)$. For MD detection, we compute $\sqrt{(X-\mu)\sum^{-1}(X-\mu)^{\intercal}}$. NASRs under MSP detection and MD detection are denoted as $\mathbf{NASR}_{MSP}$ and $\mathbf{NASR}_{MD}$.

Thus, NASR is formulated as:

$$\text{NASR}_k = 1 - \frac{|\{x^{orig}|y^{adv}=y^{orig}, x^{orig}\in\mathcal{X}\}|+|\mathcal{D}_k|}{|\mathcal{X}|},$$

where $\mathcal{D}_k$ denotes the set of examples that successfully attack the victim model but are detected by the detection method $k \in \{MSP, MD\}$.

In this context, adversarial examples are considered as OOD examples (positive), while original examples are considered as in-distribution examples (negative). To avoid misdetecting original examples as adversarial examples from a defender's view, we use the negative MSP and MD value at 99% False Positive Rate of the training data as thresholds. Values exceeding these thresholds are considered positive, while those falling below are classified as negative.

6 Experimental Settings

Attack Baselines. We use two character-level attack methods, DeepWordBug (Gao et al., 2018) and TextBugger (Jinfeng et al., 2019), and three word-level attack methods, TextFooler (Jin et al., 2020), BERT-Attack (Li et al., 2020) and A2T (Yoo and Qi, 2021). Detailed descriptions are listed in Appendix B.1.

Datasets. We evaluate DA^3 on four different types of tasks: sentiment analysis task – SST-2 (Socher et al., 2013), grammar correctness task – CoLA (Warstadt et al., 2019), textual entailment task – RTE (Wang et al., 2019a), and textual similarity task – MRPC (Dolan and Brockett, 2005). Detailed descriptions and statistics of each dataset are shown in Appendix B.2.

Implementation Details The backbone models of DA^3 are BERT-BASE or ROBERTA-BASE models fine-tuned on corresponding downstream datasets. We use BERT-BASE and ROBERTA-BASE as white-box victim models and LLAMA2-7B as the black-box victim model. More detailed information about hyperparameters and settings is in Appendix B.3. The prompts used for the black-box LLAMA2-7B are listed in Appendix B.4

7 Experimental Results and Analysis

In this section, we conduct experiments and analysis to answer five research questions:

- **RQ1** Will DA^3 effectively attack the white-box language models?
- **RQ2** Are generated adversarial examples transferable to the black-box LLAMA2-7B model?
- **RQ3** Will human judges find the quality of the generated adversarial examples reasonable?
- **RQ4** How do $\mathcal{L}_{DAL}$ components impact DA^3?
- **RQ5** Does $\mathcal{L}_{DAL}$ outperform other attack losses?

7.1 Automatic Evaluation Results

We use the adversarial examples generated by DA^3 with BERT-BASE or ROBERTA-BASE as the backbone to attack the white-box BERT-BASE and ROBERTA-BASE models, respectively. White-box models have been fine-tuned on the corresponding datasets and are accessible during our fine-tuning phase. Besides, considering that LLMs are widely used, expensive to fine-tune, and often not open

Table 1: Evaluation results on the white-box victim models. BERT-BASE and ROBERTA-BASE models are finetuned on the corresponding dataset. ACC represents model accuracy. We highlight the **best** and the $\boxed{\text{second-best}}$ results.

Dataset	Model	BERT-BASE				ROBERTA-BASE			
		ACC↓	ASR↑	NASR$_{MSP}$ ↑	NASR$_{MD}$ ↑	ACC↓	ASR↑	NASR$_{MSP}$ ↑	NASR$_{MD}$ ↑
SST-2	Original	92.43				94.04			
	TextFooler	4.47	95.16	53.47	**91.94**	4.7	95.0	73.29	92.93
	TextBugger	29.01	68.61	37.34	66.87	36.70	60.98	44.02	60.37
	DeepWordBug	16.74	81.89	**57.57**	$\boxed{80.77}$	16.97	81.95	68.17	81.10
	BERT-Attack	38.42	58.44	33.62	54.96	2.06	97.80	$\boxed{74.02}$	**94.76**
	A2T	55.16	40.32	20.72	11.79	59.63	36.59	26.10	35.73
	DA^3 (ours)	21.10	77.17	$\boxed{54.22}$	75.06	4.82	94.88	**75.98**	$\boxed{94.27}$
CoLA	Original	81.21				85.04			
	TextFooler	1.92	97.64	**95.63**	**94.92**	5.56	93.46	$\boxed{90.98}$	$\boxed{89.18}$
	TextBugger	12.18	85.01	81.23	77.69	15.63	81.62	75.87	73.28
	DeepWordBug	7.09	91.26	88.78	86.19	11.02	87.03	84.10	74.18
	BERT-Attack	12.46	84.65	79.22	79.93	2.21	97.41	**91.43**	**90.98**
	A2T	20.44	74.82	71.63	48.82	19.75	76.78	72.72	71.82
	DA^3 (ours)	2.78	96.58	$\boxed{93.74}$	$\boxed{93.27}$	6.33	92.56	87.60	85.91
RTE	Original	72.56				78.34			
	TextFooler	1.44	98.01	68.66	79.60	5.05	93.55	67.74	87.56
	TextBugger	2.53	96.52	68.66	83.08	9.75	87.56	70.05	81.57
	DeepWordBug	4.33	94.03	**79.60**	**88.06**	16.25	79.26	69.59	76.04
	BERT-Attack	3.61	95.02	67.16	72.64	1.44	98.16	$\boxed{70.51}$	**90.32**
	A2T	8.66	88.06	62.69	25.87	16.97	78.34	67.28	77.88
	DA^3 (ours)	1.08	98.51	$\boxed{72.14}$	$\boxed{86.07}$	7.22	90.78	**71.43**	$\boxed{88.94}$
MRPC	Original	87.75				91.18			
	TextFooler	2.94	96.65	58.38	$\boxed{91.62}$	4.90	94.62	35.48	94.62
	TextBugger	7.35	91.60	62.85	87.15	9.80	89.25	34.68	89.25
	DeepWordBug	10.05	88.55	$\boxed{72.35}$	86.31	12.01	86.83	$\boxed{47.31}$	86.83
	BERT-Attack	9.56	89.11	55.31	61.39	2.45	97.31	34.95	$\boxed{97.04}$
	A2T	30.88	64.80	46.65	26.54	49.51	45.70	21.51	45.43
	DA^3 (ours)	0.74	99.16	**74.86**	**93.29**	0.49	99.46	**50.27**	**99.46**

source, we evaluate the attack transferability of the adversarial examples, which are generated by DA^3 with BERT-BASE as the backbone, on the black-box LLAMA2-7B model, which is not available during DA^3 fine-tuning. The experimental results on ACC, ASR, and NASR are shown in Table 1.

Attack Performance (RQ1). When attacking white-box models, DA^3 obtains the best or second-to-best performance regarding NASR on most datasets. Aside from DA^3, some baseline methods perform well on one of the victim models. For example, TextFooler works well on BERT-BASE, while its NASR$_{MSP}$ decreases drastically compared to ASR on SST-2, RTE, and MRPC. Similarly, BERT-Attack shows good performance on ROBERTA-BASE, while its NASR$_{MSP}$ is notably lower than its ASR, especially on SST-2, RTE, and MRPC. This phenomenon indicates these adversarial examples are relatively easy to detect using MSP detection. Considering the results of both victim models, DA^3 consistently produces reasonable and favorable outcomes when attacking white-box models, which proves the effectiveness of DA^3.

We also report %Words and SS in Appendix C. DA^3 achieves best or second-to-best %Words and comparable SS compared to baselines across datasets on both victim models.

Transferability to LLMs (RQ2). [2] When attacking the black-box LLAMA2-7B model, DA^3 performs the best on SST-2, RTE, and MRPC, outperforming baselines in all evaluation metrics. On CoLA, DA^3 achieves second-to-best results on NASR. Further analysis and visualization of attack performance on LLAMA2-7B across five different prompts are displayed in Appendix F. DA^3 consistently surpasses all baselines across five prompts.

The experimental results underscore the substantial advantage of our model when generalizing generated adversarial examples to the black-box LLAMA2-7B model, compared to baselines.

[2] We also present results on MISTRAL-7B and the analysis on why the generated samples can be transferred to another LLMs in Appendix C. The results show DA^3 achieves the best performance in most cases when attacking MISTRAL-7B.

Table 2: Evaluation results on the black-box LLaMA2-7B model. Results of LLaMA2-7B are the average of zero-shot prompting with five different prompts.

Dataset	Model	LLaMA2-7B			
		ACC↓	ASR↑	NASR$_{MSP}$↑	NASR$_{MD}$↑
SST-2	Original	89.91			
	TextFooler	68.97	23.81	22.97	23.58
	TextBugger	84.50	6.89	6.51	6.69
	DeepWordBug	81.97	9.49	9.01	9.39
	BERT-Attack	66.42	26.61	$\boxed{25.81}$	$\boxed{26.38}$
	A2T	81.33	10.63	10.14	10.15
	DA^3 (ours)	64.19	29.42	**28.68**	**29.14**
CoLA	Original	70.97			
	TextFooler	31.95	57.65	52.13	57.09
	TextBugger	39.41	48.22	42.49	47.22
	DeepWordBug	31.93	61.23	**56.67**	**60.58**
	BERT-Attack	39.98	46.07	40.97	45.68
	A2T	40.38	45.09	39.81	37.75
	DA^3 (ours)	33.06	58.51	$\boxed{53.39}$	$\boxed{57.69}$
RTE	Original	57.76			
	TextFooler	53.29	12.62	10.54	12.11
	TextBugger	56.39	5.62	3.77	5.10
	DeepWordBug	51.05	12.78	9.76	12.39
	BERT-Attack	44.33	24.96	$\boxed{20.30}$	$\boxed{24.05}$
	A2T	48.52	21.40	17.45	19.72
	DA^3 (ours)	42.81	28.95	**24.26**	**26.87**
MRPC	Original	67.94			
	TextFooler	61.96	14.32	9.69	7.74
	TextBugger	65.25	8.60	6.71	7.21
	DeepWordBug	63.97	9.59	6.77	8.87
	BERT-Attack	60.64	15.47	10.99	$\boxed{14.82}$
	A2T	60.19	15.40	$\boxed{11.06}$	14.17
	DA^3 (ours)	59.85	17.92	**12.22**	**16.84**

Table 3: Grammar correctness, prediction accuracy and semantic preservation of original examples (denoted as Orig.) and adversarial examples generated by DA^3.

Dataset	Grammar		Accuracy		Semantic	
	DA^3	Orig.	DA^3	Orig.	DA^3	TextFooler
SST-2	4.12	4.37	0.68	0.74	0.71	0.66
MRPC	4.62	4.86	0.68	0.76	0.88	0.84

Table 4: Ablation study on BERT-BASE regarding MSP.

Dataset	Model	ACC↓	ASR↑	NASR$_{MSP}$↑	DR$_{MSP}$↓
SST-2	DA^3	21.10	77.17	**54.22**	**29.74**
	(w/o MSP)	1.61	98.26	47.27	51.89
CoLA	DA^3	2.78	96.58	**93.74**	**2.93**
	(w/o MSP)	2.11	97.40	93.15	4.36
RTE	DA^3	1.08	98.51	**72.14**	**26.77**
	(w/o MSP)	1.08	98.51	70.65	28.28
MRPC	DA^3	0.74	99.16	**74.86**	**24.51**
	(w/o MSP)	0.74	99.16	73.18	26.20

Table 5: Ablation study on BERT-BASE regarding MD.

Dataset	Model	ACC↓	ASR↑	NASR$_{MD}$↑	DR$_{MD}$↓
SST-2	DA^3	21.10	77.17	75.06	**2.73**
	(w/o MD)	15.60	83.13	**80.77**	2.84
CoLA	DA^3	2.78	96.58	**93.27**	**3.42**
	(w/o MD)	2.30	97.17	90.55	6.80
RTE	DA^3	1.08	98.51	**86.07**	**12.63**
	(w/o MD)	1.08	98.51	85.57	13.13
MRPC	DA^3	0.74	99.16	**93.29**	**5.90**
	(w/o MD)	1.72	98.04	90.22	7.98

7.2 Human Evaluation (RQ3)

Given that our goal is to generate high-quality adversarial examples that preserve the original semantics and remain imperceptible to humans, we perform human evaluations to assess the adversarial examples generated by DA^3 using BERT-BASE as the backbone. These evaluations focus on grammar, prediction accuracy, and semantic preservation on SST-2 and MRPC datasets. For this purpose, three human judges evaluate 50 randomly selected original-adversarial pairs from each dataset. Detailed annotation guidelines are in Appendix D.

First, human raters are tasked with evaluating the grammar correctness and making predictions of a shuffled mix of the sampled original and adversarial examples. Grammar correctness is scored from 1-5 (Li et al., 2020; Jin et al., 2020). Then, human judges assess the semantic preservation of adversarial examples, determining whether they maintain the original semantics. We follow Jin et al. (2020) and ask human judges to classify adversarial examples as similar (1), ambiguous (0.5), or dissimilar (0) to the original examples. We compare DA^3 with the best baseline model, TextFooler, on semantic preservation for better evaluation. We take the average scores among human raters for grammar correctness and semantic preservation and take the majority class as the predicted label.

As shown in Table 3, grammar correctness scores of adversarial examples generated by DA^3 are similar to those of original examples. While word perturbations make predictions more challenging, adversarial examples generated by DA^3 still show decent accuracy. Compared to TextFooler, DA^3 can better preserve semantic similarity to original examples. Some generated adversarial examples are displayed in Appendix E.

7.3 Ablation Study (RQ4)

To analyze the effectiveness of different components of $\mathcal{L}_{DAL}$, we conduct an ablation study on BERT-BASE. The results of the ablation study are shown in Table 4 and Table 5.

MSP Loss. We ablate $\mathcal{L}_{MSP}$ during fine-tuning to assess the efficacy of $\mathcal{L}_{MSP}$. $\mathcal{L}_{MSP}$ helps improve NASR$_{MSP}$ and MSP Detection Rate (DR$_{MSP}$), which is the ratio of $|\mathcal{D}_{MSP}|$ to the total number of successful adversarial examples, across all datasets. An interesting finding is that on SST-2 and CoLA, although models without $\mathcal{L}_{MSP}$ perform better in terms of ASR, the situation deteriorates when considering detectability, leading to lower NASR$_{MSP}$ and higher DR$_{MSP}$ compared to the model with $\mathcal{L}_{DAL}$.

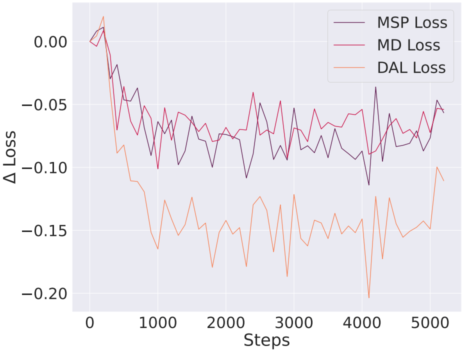


Figure 5: The change of $\mathcal{L}_{MSP}$, $\mathcal{L}_{MD}$, and $\mathcal{L}_{DAL}$ throughout the fine-tuning phase of DA^3 with BERT-BASE as backbone on SST-2. The x-axis represents fine-tuning steps; the y-axis represents the change of loss compared to the initial loss.

MD Loss. We ablate $\mathcal{L}_{MD}$ during fine-tuning to assess the efficacy of $\mathcal{L}_{MD}$. $\mathcal{L}_{MD}$ helps improve MD Detection Rate (DR$_{MD}$), which is the ratio of $|\mathcal{D}_{MD}|$ to the number of successful adversarial examples, across all datasets. $\mathcal{L}_{MD}$ also improves NASR$_{MD}$ on all datasets except SST-2. A similar finding on CoLA exists that although models without $\mathcal{L}_{MD}$ perform better on ASR, the performance worsens when considering detectability.

The ablation study shows that both $\mathcal{L}_{MSP}$ and $\mathcal{L}_{MD}$ are effective on most datasets.

7.4 Loss Visualization and Analysis (RQ4)

To better understand how different loss components contribute to DA^3, we visualize the changes of $\mathcal{L}_{MSP}$, $\mathcal{L}_{MD}$, and $\mathcal{L}_{DAL}$ throughout the fine-tuning phase of DA^3 with BERT-BASE as backbone on SST-2 dataset, as illustrated in Figure 5.

We observe that all three losses exhibit oscillating descent and eventual convergence. Although the overall trends of $\mathcal{L}_{MSP}$ and $\mathcal{L}_{MD}$ are consistent, a closer examination reveals that they often exhibit opposite trends at each step, especially in the initial stages. Despite both losses sharing a common goal of reducing distribution shifts between adversarial examples and original examples, this observation reveals a potential trade-off relationship between them. One possible interpretation is that, on the one hand, minimizing $\mathcal{L}_{MSP}$ increases the confidence of wrong predictions, aligning with the objective of the adversarial attack task to induce incorrect predictions. On the other hand, minimizing $\mathcal{L}_{MD}$ encourages the generated adversarial sentences to resemble the original ones more closely, loosely akin to the objective of the masked language modeling task to restore masked tokens to their original values. While these two objectives are not inherently conflicting, an extreme standpoint reveals that when the latter objective is fully satisfied – meaning the model generates identical examples to the original ones – the former objective naturally becomes untenable.

Table 6: Comparison of DA^3 using BERT-BASE as backbone with loss variants.

Dataset	Model	ACC↓	ASR↑	MSP		MD	
				NASR↑	DR↓	NASR↑	DR↓
SST-2	w/ $\mathcal{L}_{NCE}$	18.23	80.27	55.71	30.60	76.30	4.95
	w/ $\mathcal{L}_{FCE}$	17.66	80.89	**63.03**	**22.09**	**78.04**	3.53
	ours	21.10	77.17	54.22	29.74	75.06	**2.73**
CoLA	w/ $\mathcal{L}_{NCE}$	2.03	97.52	**94.10**	3.51	92.80	4.84
	w/ $\mathcal{L}_{FCE}$	3.07	96.22	93.98	**2.33**	91.97	4.42
	ours	2.78	96.58	93.74	2.93	**93.27**	**3.42**
RTE	w/ $\mathcal{L}_{NCE}$	1.08	98.51	71.14	27.78	85.57	13.13
	w/ $\mathcal{L}_{FCE}$	1.44	98.01	69.65	28.93	85.07	13.20
	ours	1.08	98.51	**72.14**	**26.77**	**86.07**	**12.63**
MRPC	w/ $\mathcal{L}_{NCE}$	2.45	97.21	71.79	26.15	89.39	8.05
	w/ $\mathcal{L}_{FCE}$	0.74	99.16	68.99	30.42	91.34	7.89
	ours	0.74	99.16	**74.86**	**24.51**	**93.29**	**5.90**

7.5 Loss Comparison (RQ5)

Other than using our $\mathcal{L}_{DAL}$, we also explore other loss variants: $\mathcal{L}_{NCE}$ and $\mathcal{L}_{FCE}$.

Minimizing the negative of regular cross-entropy loss (denoted as $\mathcal{L}_{NCE}$) or minimizing the cross-entropy loss of flipped adversarial labels (denoted as $\mathcal{L}_{FCE}$) are two simple ideas as baseline attack methods. We replace $\mathcal{L}_{DAL}$ with $\mathcal{L}_{NCE}$ or $\mathcal{L}_{FCE}$ during the fine-tuning phase to assess the efficacy of our loss $\mathcal{L}_{DAL}$. The results in Table 6 show that $\mathcal{L}_{DAL}$ outperforms the other two losses across all evaluation metrics on RTE and MRPC datasets. On CoLA dataset, $\mathcal{L}_{DAL}$ achieves better or similar performance compared to $\mathcal{L}_{NCE}$ and $\mathcal{L}_{FCE}$. While $\mathcal{L}_{DAL}$ may not perform as well as $\mathcal{L}_{NCE}$ and $\mathcal{L}_{FCE}$ on SST-2, given its superior performance on the majority of datasets, we believe $\mathcal{L}_{DAL}$ is more effective than $\mathcal{L}_{NCE}$ and $\mathcal{L}_{FCE}$ generally.

8 Conclusion

We analyze the adversarial examples generated by previous attack methods and identify distribution shifts between adversarial examples and original examples in terms of MSP and MD. To address this, we propose a Distribution-Aware Adversarial Attack (DA^3) method with the Data Alignment Loss and introduce a novel evaluation metric, NASR, which integrates out-of-distribution detection into the assessment of successful attacks. Our experiments validate the attack effectiveness of DA^3 on BERT-BASE and ROBERTA-BASE and the transferability of adversarial examples generated by DA^3 on the black-box LLAMA2-7B.

Limitations

We analyze the distribution shifts between adversarial examples and original examples in terms of MSP and MD, which exist in most datasets. Nevertheless, the MD distribution shift is not very obvious in some datasets like MRPC. This indicates that MD detection may not always effectively identify adversarial examples. However, we believe that since such a distribution shift is present in many datasets, we still need to consider MD detection. Furthermore, our experiments demonstrate that considering distribution shift is not only effective for NASR but also enhances the performance of the model in ASR.

Ethics Statement

There exists a potential risk associated with our proposed attack methods – they could be used maliciously to launch adversarial attacks against off-the-shelf systems. Despite this risk, we emphasize the necessity of conducting studies on adversarial attacks. Understanding these attack models is crucial for the research community to develop effective defenses against such attacks.

References

Dyah Adila and Dongyeop Kang. 2022. Understanding out-of-distribution: A perspective of data dynamics. In *I (Still) Can't Believe It's Not Better! Workshop at NeurIPS 2021*, pages 1–8. PMLR.

Udit Arora, William Huang, and He He. 2021. Types of out-of-distribution texts and how to detect them. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 10687–10701. Association for Computational Linguistics (ACL).

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

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

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.

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.

Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran. 2023. A survey of adversarial defenses and robustness in nlp. *ACM Comput. Surv.*, 55(14s).

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

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

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.

Li Jinfeng, Ji Shouling, Du Tianyu, Li Bo, and Wang Ting. 2019. Textbugger: Generating adversarial text against real-world applications. *Proceedings 2019 Network and Distributed System Security Symposium*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31.

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.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475.

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. *arXiv preprint arXiv:1907.11692*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. 2021. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022*.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. 2022. Out-of-distribution detection and selective generation for conditional language models. In *The Eleventh International Conference on Learning Representations*.

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.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

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

Liwei Song, Xinwei Yu, Hsuan-Tung Peng, and Karthik Narasimhan. 2021. Universal adversarial attacks with natural triggers for text classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3724–3733, Online. Association for Computational Linguistics.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. 2022. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pages 20827–20840. PMLR.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019b. Improving back-translation with uncertainty-based confidence estimation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 791–802, Hong Kong, China. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Tim Z Xiao, Aidan N Gomez, and Yarin Gal. 2020. Wat zei je? detecting out-of-distribution translations with variational transformers. *arXiv preprint arXiv:2006.08344*.

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.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *International Conference on Learning Representations*.

Appendix

A Evaluation Metrics

Percentage of Perturbed Words (%Words). Percentage of Perturbed Words (%Words) is used to measure how much a text has been altered or perturbed from its original form. %Words is formulated as

$$\%\text{Words} = \frac{\text{Number of Perturbed Words}}{\text{Total Number of Words}} \times 100.$$

Semantic Similarity (SS). We calculate Semantic Similarity (SS) using sentence semantic similarity between x^{orig} and x^{adv}. Specifically, we transform the two sentences into high-dimensional sentence embeddings using the Universal Sentence Encoder (USE) (Cer et al., 2018). We then approximate their semantic similarity by calculating the cosine similarity score between these vectors.

Attack Success Rate (ASR). Attack Success Rate (ASR) is defined as the percentage of generated adversarial examples that successfully deceive model predictions. Thus, ASR is formulated as

$$\text{ASR} = \frac{|\{x^{orig} \mid y^{adv} \neq y^{orig}, x^{orig} \in \mathcal{X}\}|}{|\mathcal{X}|}.$$

These definitions are consistent with prior work.

B More Implementation Details

B.1 Baselines

DeepWordBug (Gao et al., 2018) uses two scoring functions to determine the most important words and then adds perturbations through random substation, deletion, insertion, and swapping letters in the word while constrained by the edit distance.

TextBugger (Jinfeng et al., 2019) finds important words through the Jacobian matrix or scoring function and then uses insertion, deletion, swapping, substitution with visually similar words, and substitution with semantically similar words.

TextFooler (Jin et al., 2020) uses the prediction change before and after deleting the word as the word importance score and then replaces each word in the sentence with synonyms until the prediction label of the target model changes.

BERT-Attack (Li et al., 2020) finds the vulnerable words through logits from the target model and then uses BERT to generate perturbations based on the top-K predictions.

Table 7: Dataset statistics.

Dataset	Train	Validation	Description
SST-2	67,300	872	Sentiment analysis
CoLA	8,550	1,043	Grammar correctness
RTE	2,490	277	Textual entailment
MRPC	3,670	408	Textual similarity

Table 8: Hyperparameters of different datasets.

Backbone	Hyperparameter	SST-2	CoLA	RTE	MRPC
BERT-BASE	batch size	128	128	32	128
	learning rate	1e-4	5e-5	1e-5	1e-3
	% masked tokens	30	30	30	30
RoBERTA-BASE	batch size	128	128	32	128
	learning rate	5e-5	1e-4	1e-5	1e-3
	% masked tokens	30	30	30	30

A2T (Yoo and Qi, 2021) employs a gradient-based method for ranking word importance, iteratively replacing each word with top synonyms generated from counter-fitting word embeddings (Mrkšić et al., 2016).

For the implementation of baselines, we use the TextAttack[3] package with its default parameters.

B.2 Datasets

SST-2. The Stanford Sentiment Treebank (Socher et al., 2013) is a binary sentiment classification task. It consists of sentences extracted from movie reviews with human-annotated sentiment labels.

CoLA. The Corpus of Linguistic Acceptability (Warstadt et al., 2019) contains English sentences extracted from published linguistics literature, aiming to check grammar correctness.

RTE. The Recognizing Textual Entailment dataset (Wang et al., 2019a) is derived from a combination of news and Wikipedia sources, aiming to determine whether the given pair of sentences entail each other.

MRPC. The Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005) comprises sentence pairs sourced from online news articles. These pairs are annotated to indicate whether the sentences are semantically equivalent.

Data statistics for each dataset are shown in Table 7.

B.3 Hyperparameters and More Settings

For each experiment, the DA^3 fine-tuning phrase is executed for a total of 20 epochs. The learning rate is searched from $[1e-5, 1e-3]$. Up to 30% of

[3] https://github.com/QData/TextAttack (MIT License).

Table 9: Prompt template for different datasets. {instruct} is replaced by different instructions in Table 10, while {text} is replaced with input sentence.

Dataset	Prompt
SST-2	"{instruct} Respond with 'positive' or 'negative' in lowercase, only one word. \nInput: {text}\nAnswer:"
CoLA	"{instruct} Respond with 'acceptable' or 'unacceptable' in lowercase, only one word.\nInput: {text}\nAnswer:",
RTE	"{instruct} Respond with 'entailment' or 'not_entailment' in lowercase, only one word.\nInput: {text}\nAnswer:
MRPC	"{instruct} Respond with 'equivalent' or 'not_equivalent' in lowercase, only one word.\nInput: {text} \nAnswer:

Table 10: Different instructions used for different runs.

Dataset	Prompt
SST-2	"Evaluate the sentiment of the given text."
	"Please identify the emotional tone of this passage."
	"Determine the overall sentiment of this sentence."
	"After examining the following expression, label its emotion."
	"Assess the mood of the following quote."
CoLA	"Assess the grammatical structure of the given text."
	"Assess the following sentence and determine if it is grammatically correct."
	"Examine the given sentence and decide if it is grammatically sound."
	"Check the grammar of the following sentence."
	"Analyze the provided sentence and classify its grammatical correctness."
RTE	"Assess the relationship between sentence1 and sentence2."
	"Review the sentence1 and sentence2 and categorize their relationship."
	"Considering the sentence1 and sentence2, identify their relationship."
	"Please classify the relationship between sentence1 and sentence2."
	"Indicate the connection between sentence1 and sentence2."
MRPC	"Assess whether sentence1 and sentence2 share the same semantic meaning."
	"Compare sentence1 and sentence2 and determine if they share the same semantic meaning."
	"Do sentence1 and sentence2 have the same underlying meaning?"
	"Do the meanings of sentence1 and sentence2 align?"
	"Please analyze sentence1 and sentence2 and indicate if their meanings are the same."

the tokens are masked during the fine-tuning phrase. The rank of the update matrices of LoRA is set to 8; LoRA scaling factor is 32; LoRA dropout value is set as 0.1. The inference termination condition is set as 40% of the tokens.

Table 8 shows the hyperparameters used in experiments.

White-box experiments are conducted on two NVIDIA GeForce RTX 3090ti GPUs, and black-box experiments are conducted on two NVIDIA RTX A5000 24GB GPUs.

B.4 Prompts Used for the Black-box LLM

The constructed prompt templates used for the Black-box LLM (LLaMA2-7B[4]) are shown in Table 9. For each run, {instruct} in the prompt template is replaced by different instructions in Table 10, while {text} is replaced with the input sentence.

[4]LLaMA2 Community License

C More Automatic Evaluation Results

Experimental results of %Words and SS on the white-box victim models BERT-BASE and RoBERTa-BASE are shown in Table 12 and Table 13. DA^3 achieves best or second-to-best %Words and comparable SS compared to baselines across datasets on both victim models.

The results of the generated adversarial examples by DA^3 with BERT-BASE as the backbone on attacking the white-box MISTRAL-7B model on CoLA, RTE, and MRPC are shown in Table 11. Our proposed DA^3 outperforms all other baselines.

Although BERT-BASE, LLaMA2-7B, and MISTRAL-7B have different structures and parameters, they are both trained on large text corpora.

Table 11: Evaluation results on the black-box MISTRAL-7B models. Results of MISTRAL-7B are the average of zero-shot prompting with five different prompts.

Dataset	Model	MISTRAL-7B			
		ACC↓	ASR↑	NASR_{MSP} ↑	NASR_{MD} ↑
CoLA	Original	79.35			
	TextFooler	27.84	66.20	$\boxed{57.59}$	**63.57**
	TextBugger	38.28	52.52	46.36	48.26
	DeepWordBug	34.67	58.99	51.69	53.87
	BERT-Attack	33.25	59.58	52.23	55.96
	A2T	35.70	56.36	49.26	51.86
	DA^3 (ours)	29.11	66.12	**63.41**	$\boxed{62.49}$
RTE	Original	80.94			
	TextFooler	65.20	24.35	24.35	24.17
	TextBugger	77.91	6.95	6.95	6.86
	DeepWordBug	77.98	6.33	6.33	6.24
	BERT-Attack	56.73	33.18	$\boxed{33.18}$	$\boxed{33.12}$
	A2T	57.69	32.11	32.11	32.11
	DA^3 (ours)	54.08	3598	**35.71**	**35.45**
MRPC	Original	79.31			
	TextFooler	63.09	25.00	24.81	22.97
	TextBugger	78.68	4.52	4.52	4.52
	DeepWordBug	78.33	4.46	4.46	4.40
	BERT-Attack	56.22	34.58	$\boxed{33.72}$	$\boxed{34.60}$
	A2T	61.91	26.52	26.03	26.52
	DA^3 (ours)	56.18	35.30	**35.07**	**35.38**

Thus, they share similar knowledge. From Table 2 and Table 11, we can see that BERT-based models (BERT-Attack and DA^3) perform better than other models in most cases, which confirms our explanations. Besides, the best transferability also shows that our proposed DA^3 can generate high-quality adversarial examples that are robust to the black-box LLMs.

D Annotation Guidelines

Here we provide the annotation guidelines for annotators:

Grammar. Rate the grammaticality and fluency of the text between 1-5; the higher the score, the better the grammar of the text.

Prediction. For SSTS-2 dataset, classify the sentiment of the text into negative (0) or positive (1); For MRPC dataset, classify if the two sentences are equivalent (1) or not_equivalent (0).

Semantic. Compare the semantic similarity between text1 and text2, and label with similar (1), ambiguous (0.5), and dissimilar (0).

E Examples of Generated Adversarial Sentences

Table 14 displays some original examples and the corresponding adversarial examples generated by DA^3. The table also shows the predicted results of the original or adversarial sentence using BERT-BASE. Blue words are perturbed into the red words. Table 14 shows that DA^3 only perturbs a very small number of words, leading to model prediction failure. Besides, the adversarial examples generally preserve similar semantic meanings to their original inputs.

F Results Visualization Across Different Prompts

We display the individual attack performance of five runs with different prompts on the MRPC dataset in Figure 6. The figure illustrates that DA^3 consistently surpasses other baseline methods for each run.

G Observation Experiments

The observation experiments on previous attack methods TextFooler, TextBugger, DeepWordBug, and BERT-Attack are shown in Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, and Figure 14.

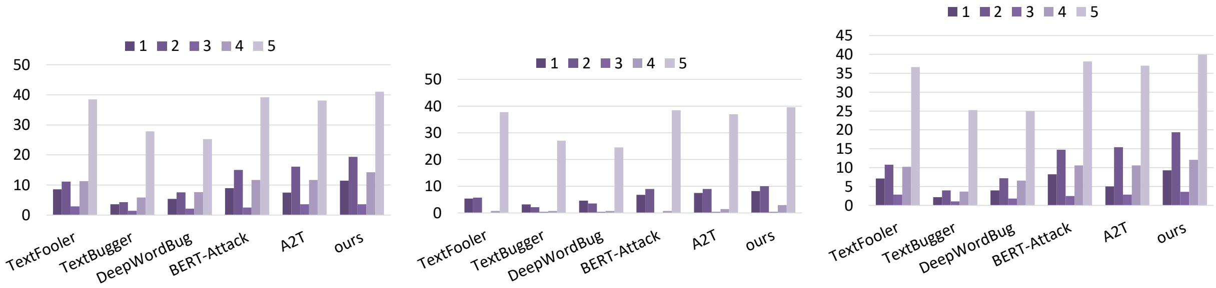
The distribution shift between adversarial examples and original examples is more evident in terms of MSP across all the datasets. The distribution shift between adversarial examples and original examples in terms of MD is clear only on SST-2 dataset and MRPC dataset. Although this shift is not always present in terms of MD, it is imperative to address this issue given its presence in certain datasets.

Table 12: %Words and SS results on the BERT-BASE victim model.

Dataset	SST-2						CoLA					
Model	TextFooler	TextBugger	DeepWordBug	BERT-Attack	A2T	DA^3	TextFooler	TextBugger	DeepWordBug	BERT-Attack	A2T	DA^3
%Words	17.58	15.35	19.11	13.42	$\boxed{11.06}$	**10.72**	19.16	19.16	18.53	$\boxed{18.34}$	19.04	**16.83**
SS	82.32	**90.98**	80.03	89.89	$\boxed{90.25}$	87.78	82.09	**91.36**	83.60	$\boxed{90.65}$	88.62	86.95
Dataset	**RTE**						**MRPC**					
Model	TextFooler	TextBugger	DeepWordBug	BERT-Attack	A2T	DA^3	TextFooler	TextBugger	DeepWordBug	BERT-Attack	A2T	DA^3
%Words	6.01	12.07	6.59	6.97	**4.41**	$\boxed{4.75}$	9.69	19.09	8.32	11.66	**6.2**	$\boxed{6.64}$
SS	96.80	**97.26**	96.72	96.32	$\boxed{97.18}$	96.37	94.04	$\boxed{95.60}$	94.56	93.07	**96.10**	93.86

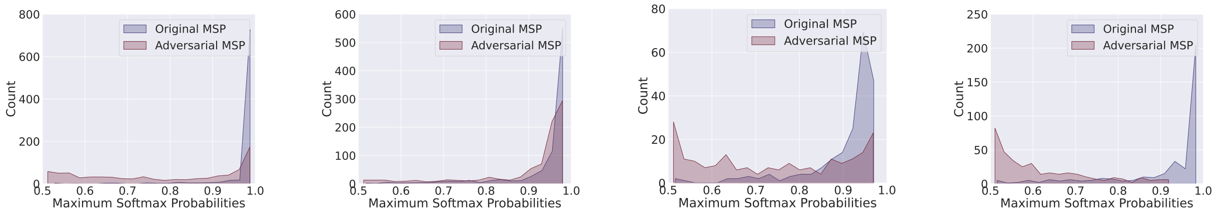
Table 13: %Words and SS results on the ROBERTA-BASE victim model.

Dataset	SST-2						CoLA					
Model	TextFooler	TextBugger	DeepWordBug	BERT-Attack	A2T	DA^3	TextFooler	TextBugger	DeepWordBug	BERT-Attack	A2T	DA^3
%Words	18.73	18.03	22.70	14.33	**12.30**	$\boxed{12.58}$	19.07	18.40	19.10	$\boxed{17.31}$	17.60	**17.29**
SS	81.58	**90.37**	75.26	86.44	$\boxed{89.48}$	86.98	83.31	**91.90**	83.22	$\boxed{90.49}$	90.15	85.99
Dataset	**RTE**						**MRPC**					
Model	TextFooler	TextBugger	DeepWordBug	BERT-Attack	A2T	DA^3	TextFooler	TextBugger	DeepWordBug	BERT-Attack	A2T	DA^3
%Words	6.96	7.93	$\boxed{5.27}$	6.59	**3.93**	6.38	12.50	18.84	13.18	10.09	**7.04**	$\boxed{8.10}$
SS	96.35	$\boxed{97.32}$	96.93	96.67	**97.69**	94.88	92.12	93.28	90.44	93.13	**95.96**	$\boxed{94.12}$



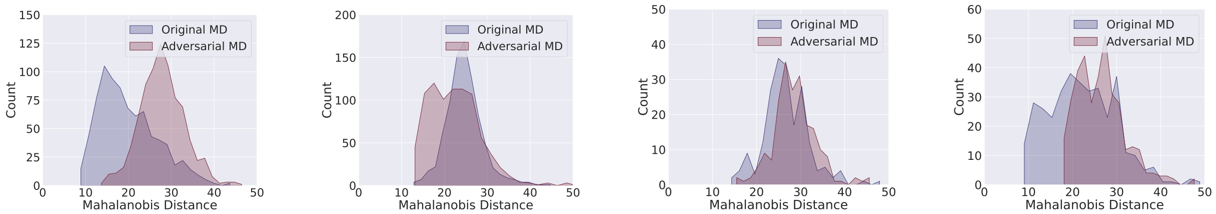
(a) ASR (b) NASR (MSP) (c) NASR (MD)

Figure 6: Results of LLAMA2-7B across five different prompts on MRPC.



(a) MSP on SST-2 dataset. (b) MSP on CoLA dataset. (c) MSP on RTE dataset. (d) MSP on MRPC dataset.

Figure 7: Visualization of the distribution shift between original data and adversarial data generated by TextFooler when attacking BERT-BASE regarding Maximum Softmax Probability.

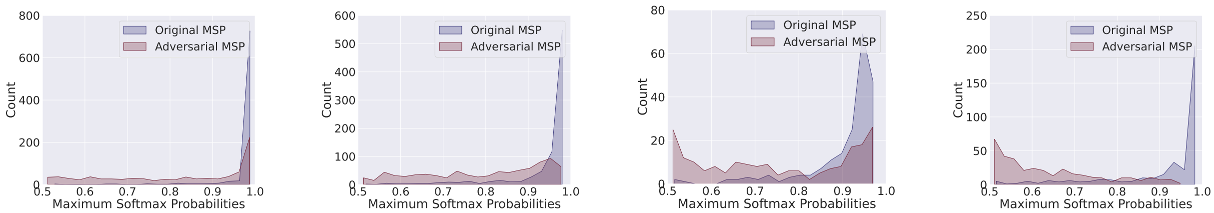


(a) MD on SST-2 dataset. (b) MD on CoLA dataset. (c) MD on RTE dataset. (d) MD on MRPC dataset.

Figure 8: Visualization of the distribution shift between original data and adversarial data generated by TextFooler when attacking BERT-BASE regarding Mahalanobis Distance.

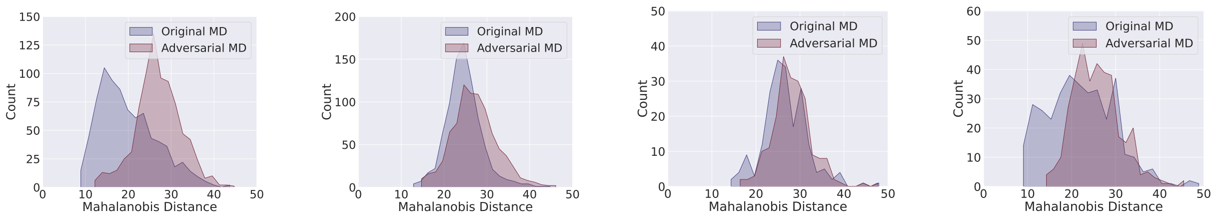
Table 14: Examples of generated adversarial sentences

	Sentence	Prediction
Ori	/ but daphne , you 're too buff / fred thinks he 's tough / and velma - wow , you 've lost weight !	Negative
Adv	/ but daphne , you 're too buff / fred thinks he 's tough / and velma - wow , you 've corrected weight !	Positive
Ori	The car was driven by John to Maine.	Acceptable
Adv	The car was amounted by John to Maine.	Unacceptable
Ori	The sailors rode the breeze clear of the rocks.	Acceptable
Adv	The sailors wandered the breeze clear of the rocks.	Unacceptable
Ori	The more Fred is obnoxious, the less attention you should pay to him.	Acceptable
Adv	The more Fred is obnoxious, the less noticed you should pay to him.	Unacceptable
Ori	Sentence1: And, despite its own suggestions to the contrary, Oracle will sell PeopleSoft and JD Edwards financial software through reseller channels to new customers.<SPLIT>Sentence2: Oracle sells financial software.	Not_entailment
Adv	Sentence1: And, despite its own suggestions to the contrary, Oracle will sell PeopleSoft and JD Edwards financial software through reseller channels to new customers.<SPLIT>Sentence2: Oracle sells another software.	Entailment
Ori	Sentence1: Ms Stewart , the chief executive , was not expected to attend .<SPLIT>Sentence2: Ms Stewart , 61 , its chief executive officer and chairwoman , did not attend .	Equivalent
Adv	Sentence1: Ms Stewart , the chief executive , was not expected to visiting .<SPLIT>Sentence2: Ms Stewart , 61 , its chief executive officer and chairwoman , did not attend .	Not_equivalent
Ori	Sentence1: Sen. Patrick Leahy of Vermont , the committee 's senior Democrat , later said the problem is serious but called Hatch 's suggestion too drastic .<SPLIT>Sentence2: Sen. Patrick Leahy , the committee 's senior Democrat , later said the problem is serious but called Hatch 's idea too drastic a remedy to be considered .	Equivalent
Adv	Sentence1: Sen. Patrick Leahy of Vermont , the committee 's senior Democrat , later said the problem is serious but called Hatch 's suggestion too drastic .<SPLIT>Sentence2: Sen. Patrick Leahy , the committee 's senior Democrat , later said the problem is serious but called Hatch 's idea too drastic a remedy to be counted .	Not_equivalent



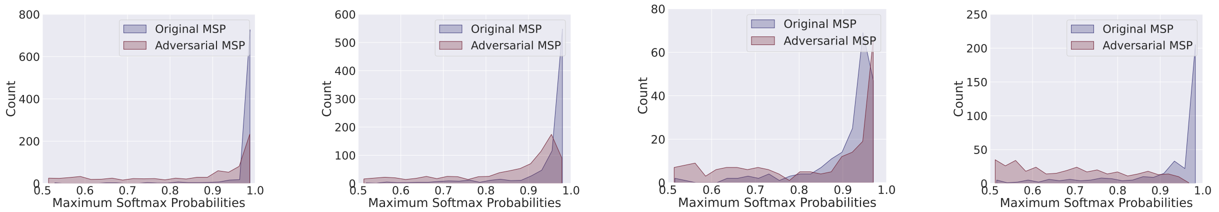
(a) MSP on SST-2 dataset. (b) MSP on CoLA dataset. (c) MSP on RTE dataset. (d) MSP on MRPC dataset.

Figure 9: Visualization of the distribution shift between original data and adversarial data generated by TextBugger when attacking BERT-BASE regarding Maximum Softmax Probability.



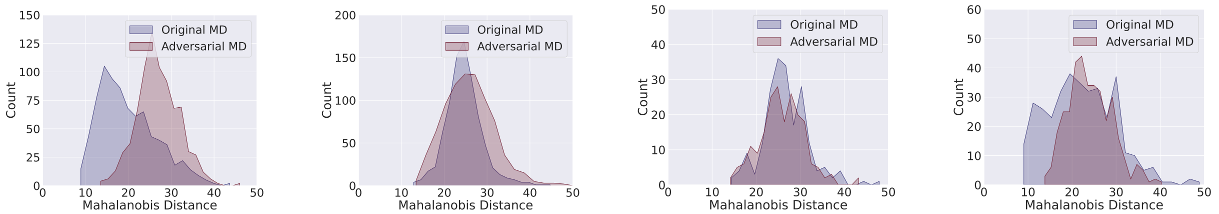
(a) MD on SST-2 dataset. (b) MD on CoLA dataset. (c) MD on RTE dataset. (d) MD on MRPC dataset.

Figure 10: Visualization of the distribution shift between original data and adversarial data generated by TextBugger when attacking BERT-BASE regarding Mahalanobis Distance.



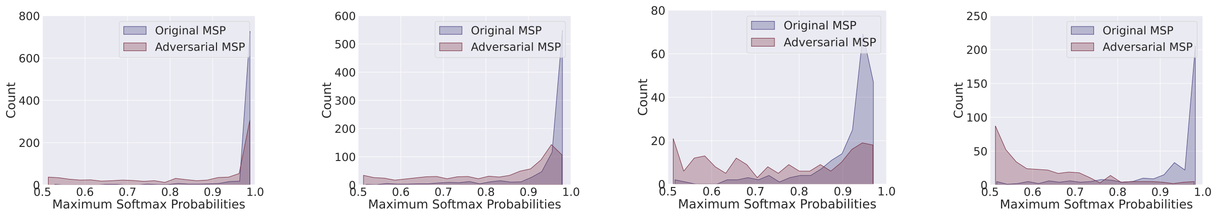
(a) MSP on SST-2 dataset. (b) MSP on CoLA dataset. (c) MSP on RTE dataset. (d) MSP on MRPC dataset.

Figure 11: Visualization of the distribution shift between original data and adversarial data generated by DeepWordBug when attacking BERT-BASE regarding Maximum Softmax Probability.



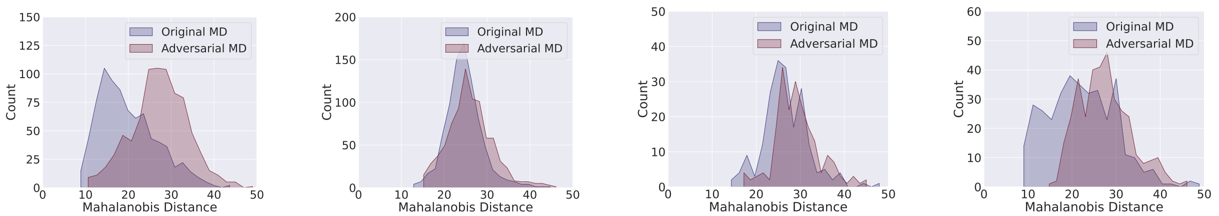
(a) MD on SST-2 dataset. (b) MD on CoLA dataset. (c) MD on RTE dataset. (d) MD on MRPC dataset.

Figure 12: Visualization of the distribution shift between original data and adversarial data generated by DeepWordBug when attacking BERT-BASE regarding Mahalanobis Distance.



(a) MSP on SST-2 dataset. (b) MSP on CoLA dataset. (c) MSP on RTE dataset. (d) MSP on MRPC dataset.

Figure 13: Visualization of the distribution shift between original data and adversarial data generated by BERT-Attack when attacking BERT-BASE regarding Maximum Softmax Probability.



(a) MD on SST-2 dataset. (b) MD between on CoLA dataset. (c) MD on RTE dataset. (d) MD on MRPC dataset.

Figure 14: Visualization of the distribution shift between original data and adversarial data generated by BERT-Attack when attacking BERT-BASE regarding Mahalanobis Distance.