DALA: A Distribution-Aware LoRA-Based Adversarial Attack against Language Models

Anonymous ACL submission

Abstract

Language models (LMs) are susceptible to adversarial attack methods that generate adversarial examples with minor perturbations. Although recent attack methods can achieve a relatively high attack success rate (ASR), we find that the generated adversarial examples have a different data distribution compared with the original examples. Specifically, these adversarial examples exhibit lower confidence levels and higher distance to the training data distribution. As a result, they are easy to detect using straightforward detection methods, diminishing the effectiveness of these attack methods. To overcome this problem, we propose a Distribution-Aware LoRA-based Adversarial Attack (DALA) method, which considers the distribution shift of adversarial examples to improve attack effectiveness under detection methods. We further design a new evaluation metric, Non-detectable Attack Success Rate (NASR), combining ASR and detection for the attack task. We conduct experiments on four widely-used datasets and validate the attack effectiveness and transferability of the adversarial examples generated by DALA on the white-box BERT-BASE model and the black-box LLAMA2-7B model.

1 Introduction

Language models (LMs), despite their capacity for remarkable accuracy and human-like performance in many applications, face vulnerability to adversarial attacks and exhibit high sensitivity to subtle input perturbations, which can potentially lead to failure (Jia and Liang, 2017; Belinkov and Bisk, 2018; Wallace et al., 2019). Recently, an increasing number of adversarial attacks have been proposed, taking forms of 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 meticulously crafted adversarial examples are imperceptible to humans but can deceive targeted models, raising concerns about the robustness and security of LMs. For example, chatbots may misunderstand user intent or sentiment and generate inappropriate responses (Perez et al., 2022).

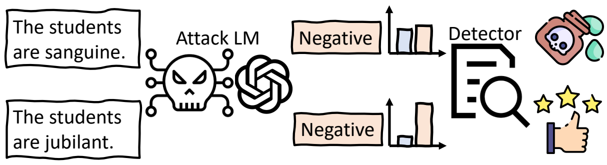


Figure 1: Toy examples of two adversarial sentences on a sentiment analysis task. Although both sentences successfully attack the victim model, the top one is detected by the detector, while the bottom one is not detected. In our task, we aim to generate adversarial examples hard to detect.

However, while existing adversarial attack methods can achieve a relatively high attack success rate on victim models (Gao et al., 2018; Belinkov and Bisk, 2018; Li et al., 2020), our experimental observations detailed in §3 reveal 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) and embedding-based detection methods like MD detection (Lee et al., 2018). Thus, the efficacy of previous attack methods is diminished when out-of-distribution detection is considered, as shown in Figure 1.

To address these problems, we propose a **D**istribution-**A**ware **L**oRA-based **A**ttack (DALA) method with Data Alignment Loss (DAL), which is a new attack method that can generate elusive adversarial examples that are hard to identify by existing detection methods. The framework of DALA consists of two phases. Firstly, DALA finetunes a LoRA-based LM by combining the Masked Language Modeling task and the downstream classification task using the Data Alignment Loss. The fine-tuning phase enables the LoRA-based LM to generate adversarial examples closely resembling original examples in terms of MSP and MD. Then, the LoRA-based LM is used during inference to generate adversarial examples.

To measure the detectability of adversarial examples generated by attack methods, we propose a new evaluation metric: **N**on-detectable **A**ttack **S**uccess **R**ate (NASR), which combines Attack Success Rate (ASR) with Out-of-Distribution (OOD) detection. We conduct experiments on four datasets to verify whether DALA can effectively attack white-box LMs using ASR and NASR. Furthermore, given the widespread use of Large Language Models (LLMs) and the fact that LLMs are expensive to fine-tune and many of them are not open source, we also evaluate the attack transferability of adversarial examples on the black-box LLMs. Our experiments show that DALA achieves competitive attack performance on the white-box BERT-BASE and the best transferability on the black-box LLAMA2-7B compared with baselines.

Our work has the following contributions:

- We analyze the distribution of adversarial and original examples and find that distribution shift exists in terms of MSP and MD.
- We propose a new Distribution-Aware LoRA-based Attack method with Data Alignment Loss, which can generate adversarial examples that effectively attack victim models.
- We design a new evaluation metric – NASR – for the attack task, which considers the detectability of adversarial examples.
- We conduct comprehensive experiments to compare the performance between DALA and baseline models on four datasets, where we find DALA achieves competitive attack capabilities and better transferability under the consideration of detection.

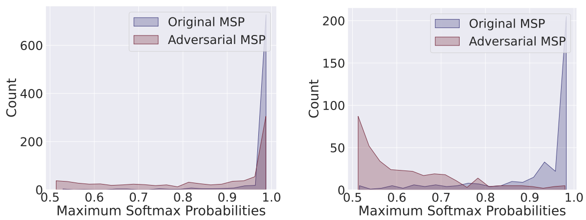
2 Related Work

2.1 Adversarial Attack in NLP

Adversarial attacks have been extensively studied to explore the robustness of language models. 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). While these attacks are effective and can achieve a high success rate, they can be easily detected through a grammar checker. Word-level attacks alter entire words rather than individual characters within words, which tend to be less perceptible to humans than character-level attacks. 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).

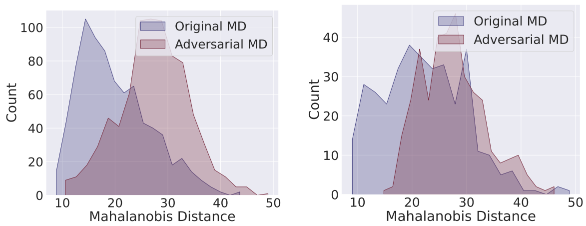
2.2 Out-of-distribution Detection in NLP

Out-of-distribution (OOD) detection methods have been widely explored in NLP problems, 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 probabilities (Hendrycks and Gimpel), 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.



(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 Maximum Softmax Probability.



(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 Mahalanobis Distance.

We select the simple, representative, and widely-used OOD detection methods of these two categories: MSP detection (Hendrycks and Gimpel) and MD detection (Lee et al., 2018), respectively. These two methods are then incorporated with the Attack Success Rate to assess the robustness and detectability of adversarial examples generated by different attack models.

3 Understanding Distribution Shift of Adversarial Examples

This section showcases the empirical observations from our analysis of adversarial examples generated by previous attack methods. Specifically, we find distribution shifts exist between adversarial and original examples, which implies that the original examples are in-distribution examples while adversarial examples are Out-of-Distribution (OOD) examples. Due to limited space, we only present the analysis of adversarial examples generated by BERT-Attack on SST-2 and MRPC; the complete results are available in Appendix E.

Maximum Softmax Probability (MSP). Maximum Softmax Probability (MSP) is a measure to evaluate prediction confidence, rendering it a widely employed score-based method for OOD detection, with diminished confidence correlating to OOD examples. To assess the difference of MSP, we visualize the MSP distribution of adversarial examples generated by BERT-Attack (Li et al., 2020) and original examples on SST-2 (Socher et al., 2013) and MRPC dataset (Dolan and Brockett, 2005) in Figure 2. We observe that on both datasets, most of the original examples have an MSP over 0.9, indicating a significantly higher MSP compared to adversarial examples overall. This distribution shift is particularly pronounced in the MRPC dataset, whereby most adversarial examples exhibit MSP below 0.6, highlighting a distinct contrast with the original examples.

Mahalanobis Distance (MD). Mahalanobis Distance (MD) is a measure of distance between one data point and a distribution, which serves as a highly suitable and widespread method for OOD detection. The higher MD of an example to in-distribution data (training data) indicates that the example may be 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 on the SST-2 and MRPC datasets in Figure 3. From Figure 3, we can observe that a distribution shift exists between original and adversarial examples on both datasets. This dissimilarity is more noticeable on the SST-2 dataset and not as conspicuous on the MRPC dataset.

Overall. These observations for MSP and MD indicate 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 LoRA-based Attack (§4.2), and introduce the new Data Alignment Loss (§4.3).

4.1 Problem Formulation

Given an original sentence $x_i^{orig} \in \mathcal{X}$ and an original label $y_i^{orig} \in \mathcal{Y}$, our objective is to obtain an adversarial sentence x_i^{adv} such that the prediction

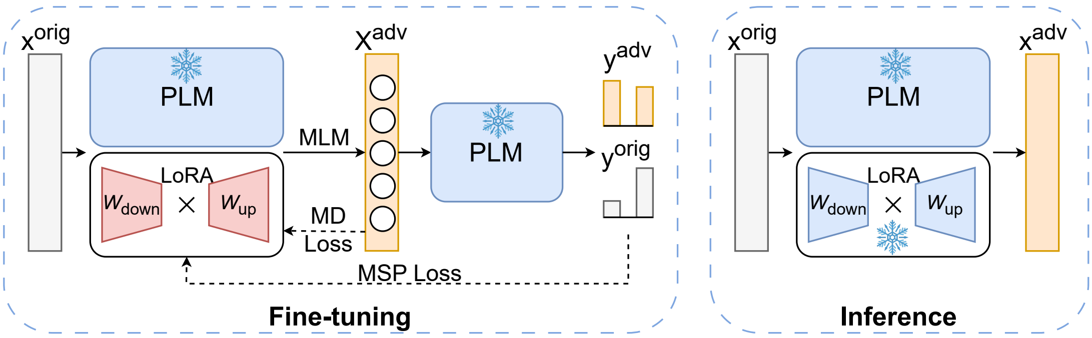


Figure 4: The overall model architecture of DALA. DALA consists of two phases: fine-tuning and inference. During fine-tuning, a LoRA-based PLM is fine-tuned to possess the ability to generate adversarial examples resembling original examples in terms of MSP and MD. During the inference phase, the LoRA-based PLM is used to generate adversarial examples.

of the victim model corresponds to $y_i^{adv} \in \mathcal{Y}$ and $y_i^{adv} \neq y_i^{orig}$.

4.2 Distribution-Aware LoRA-based Attack

Motivated by the distribution shift of adversarial examples, we propose a Distribution-Aware LoRA-based Attack (DALA) method. The key idea of DALA is to consider the distribution of the generated adversarial examples and attempt to bring about a closer alignment between the distributions of adversarial examples and original examples in terms of MSP and MD. DALA is composed of two phases: fine-tuning and inference. DALA model structure is 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. First, the original sentence x_i^{orig} undergoes the MLM task through a LoRA-based PLM to generate the adversarial embedding X_i^{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_i^{adv} is subjected to the corresponding downstream task through the frozen PLM and outputs logits of original ground truth label y_i^{orig} and adversarial label y_i^{adv}. The loss is calculated from X_i^{adv}, $P(y_i^{orig}|X_i^{adv}, \theta)$, and $P(y_i^{adv}|X_i^{adv}, \theta)$ to update the parameters of LoRA. Details are discussed in §4.3.

Inference Phase. The inference phase aims to generate adversarial examples with minimal perturbation. The original sentence x_i^{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

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

$\mathcal{L}_{MSP}$ is the complementary number of the sigmoid of the Softmax probability difference between the adversarial label and the original label given adversarial input. $\mathcal{L}_{MSP}$ is formulated as:

$$\mathcal{L}_{MSP} = 1 - \frac{1}{1+e^{-[P(y_i^{adv}|X_i^{adv},\theta)-P(y_i^{orig}|X_i^{adv},\theta)]}}, \quad (1)$$

where θ is the model parameters. According to our observation experiments in Figure 2, original data has higher Maximum Softmax Probabilities (confidence) than adversarial data. Thus, minimizing $\mathcal{L}_{MSP}$ makes generated adversarial examples more similar to original examples concerning MSP.

$\mathcal{L}_{MD}$ is the log of Mahalanobis Distance (MD) (Lee et al., 2018) of adversarial input to the training data distribution. $\mathcal{L}_{MD}$ is formulated as:

$$\mathcal{L}_{MD} = log\sqrt{(X_i^{adv} - \mu)\sum{}^{-1}(X_i^{adv} - \mu)^\intercal}, \quad (2)$$

where μ and $\sum^{-1}$ are the mean and covariance embedding of the in-distribution (training) data respectively. MD is a robust metric for out-of-distribution detection and adversarial data detection. In general, adversarial data has higher MD than original data, as shown in Figure 3. Thus, minimizing $\mathcal{L}_{MD}$ generates adversarial examples more similar to original examples in terms of MD. $\mathcal{L}_{MD}$ is constrained to the log space to be consistent with the scale of $\mathcal{L}_{MSP}$.

Thus, Data Alignment Loss is represented as

$$\mathcal{L}_{DAL} = \mathcal{L}_{MSP} + \mathcal{L}_{MD}, \tag{3}$$

and DALA is trained by optimizing $\mathcal{L}_{DAL}$.

5 Attack Performance Evaluation Metrics

Considering the observations of distribution shift analyzed in Section 3, we adopt a widely-used metric – Attack Success Rate – and design a new metric – Non-detectable Attack Success Rate – to evaluate attack performance.

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

$$\text{ASR} = \frac{|\{x_i^{orig} \mid y_i^{adv} \neq y_i^{orig}, x_i^{orig} \in \mathcal{X}\}|}{|\mathcal{X}|}. \tag{4}$$

This definition is consistent with prior work.

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 new metric considers both ASR and Out-Of-Distribution (OOD) detection. Specifically, NASR posits that the indicative criterion for a successful adversarial example resides in the capacity to cause failure in the victim model while concurrently eluding OOD detection methods.

We utilize two established and commonly employed OOD detection techniques – MSP detection (Hendrycks and Gimpel) and MD detection (Lee et al., 2018). MSP detection relies on logits and constitutes a method based on probability distributions, while MD detection is a distance-based approach. We use Negative MSPs, $-max_{y_i \in \mathcal{Y}} P(y_i \mid X_i, \theta)$, for MSP detection and $\sqrt{(X_i - \mu)\sum^{-1}(X_i - \mu)^\intercal}$ for MD detection, where μ and $\sum^{-1}$ are the mean and covariance value of the in distribution (training) data respectively. NASRs under MSP detection and MD detection are denoted as $\textbf{NASR}_{MSP}$ and $\textbf{NASR}_{MD}$.

Thus, NASR is formulated as:

$$\text{NASR}_k = 1 - \frac{|\{x_i^{orig} | y_i^{adv} = y_i^{orig}, x_i^{orig} \in \mathcal{X}\}| + |\mathcal{D}_k|}{|\mathcal{X}|}, \tag{5}$$

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

Adversarial examples are considered as OOD examples (positive), while original examples are considered as in-distribution examples (negative). To avoid misdetecting original examples into adversarial examples from a defender's view, we use the negative MSP and MD value at 99% False Positive Rate of the training data, where values exceeding the threshold are considered positive, and those less than the threshold are considered negative.

6 Experimental Settings

6.1 Baselines and Datasets

Attack Baselines. We use two character-level attack methods, DeepWordBug (Gao et al., 2018) and TextBugger (Jinfeng et al., 2019), and two word-level attack methods, TextFooler (Jin et al., 2020) and BERT-Attack (Li et al., 2020). Detailed descriptions for each baseline method are listed in Appendix A.1.

Datasets. We evaluate DALA 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 A.2.

6.2 Implementation Details

The backbone models of DALA are BERT-BASE (Devlin et al., 2019) models fine-tuned on corresponding downstream datasets. We use BERT-BASE as white-box victim models and LLAMA2-7B as black-box victim models. For each experiment, the DALA fine-tuning phrase is executed for a total of 20 epochs. The learning rate is searched from $[1e-5, 1e-3]$. 30% of 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

Table 1: Evaluation results on the white-box and black-box victim models. BERT-BASE models are finetuned on the corresponding dataset. Results of LLaMA2-7B are the average of zero-shot prompting with five different prompts (individual analysis is in Appendix D). ACC represents model accuracy. We highlight the **best** and the $\boxed{\text{second-best}}$ results.

Dataset	Model	BERT-BASE (white-box)				LLaMA2-7B (black-box)			
		ACC↓	ASR↑	NASR$_{MSP}$ ↑	NASR$_{MD}$ ↑	ACC↓	ASR↑	NASR$_{MSP}$ ↑	NASR$_{MD}$ ↑
SST-2	Original	92.43				89.91			
	TextFooler	**4.47**	**95.16**	53.47	**91.94**	68.97	23.81	22.97	23.58
	TextBugger	29.01	68.61	37.34	66.87	84.50	6.89	6.51	6.69
	DeepWordBug	$\boxed{16.74}$	$\boxed{81.89}$	**57.57**	$\boxed{80.77}$	81.97	9.49	9.01	9.39
	BERT-Attack	38.42	58.44	33.62	54.96	$\boxed{66.42}$	$\boxed{26.61}$	$\boxed{25.81}$	$\boxed{26.38}$
	DALA (ours)	21.10	77.17	$\boxed{54.22}$	75.06	**64.19**	**29.42**	**28.68**	**29.14**
CoLA	Original	81.21				70.97			
	TextFooler	**1.92**	**97.64**	**95.63**	**94.92**	$\boxed{31.95}$	57.65	52.13	57.09
	TextBugger	12.18	85.01	81.23	77.69	39.41	48.22	42.49	47.22
	DeepWordBug	7.09	91.26	88.78	86.19	**31.93**	**61.23**	**56.67**	**60.58**
	BERT-Attack	12.46	84.65	79.22	79.93	39.98	46.07	40.97	45.68
	DALA (ours)	$\boxed{2.78}$	$\boxed{96.58}$	$\boxed{93.74}$	$\boxed{93.27}$	33.06	$\boxed{58.51}$	$\boxed{53.39}$	$\boxed{57.69}$
RTE	Original	72.56				57.76			
	TextFooler	$\boxed{1.44}$	$\boxed{98.01}$	68.66	79.60	53.29	12.62	10.54	12.11
	TextBugger	2.53	96.52	68.66	83.08	56.39	5.62	3.77	5.10
	DeepWordBug	4.33	94.03	**79.60**	**88.06**	51.05	12.78	9.76	12.39
	BERT-Attack	3.61	95.02	67.16	72.64	$\boxed{44.33}$	$\boxed{24.96}$	$\boxed{20.30}$	$\boxed{24.05}$
	DALA (ours)	**1.08**	**98.51**	$\boxed{72.14}$	$\boxed{86.07}$	**42.81**	**28.95**	**24.26**	**26.87**
MRPC	Original	87.75				67.94			
	TextFooler	$\boxed{2.94}$	$\boxed{96.65}$	58.38	$\boxed{91.62}$	61.96	14.32	9.69	7.74
	TextBugger	7.35	91.60	62.85	87.15	65.25	8.60	6.71	7.21
	DeepWordBug	10.05	88.55	$\boxed{72.35}$	86.31	63.97	9.59	6.77	8.87
	BERT-Attack	9.56	89.11	55.31	80.17	$\boxed{60.64}$	$\boxed{15.47}$	$\boxed{10.99}$	$\boxed{14.82}$
	DALA (ours)	**0.74**	**99.16**	**74.86**	**93.29**	**59.85**	**17.92**	**12.22**	**16.84**

value is set as 0.1. The inference termination condition is set as 40% of the tokens. More detailed information about hyperparameters is described in Appendix A.3. The prompts used for LLaMA2-7B are listed in Appendix A.4

BERT-BASE related experiments are conducted on two NVIDIA GeForce RTX 3090ti GPUs, and LLaMA2-7B related experiments are conducted on two NVIDIA RTX A5000 24GB GPUs.

7 Experimental Results and Analysis

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

- **RQ1** Will DALA effectively attack BERT-BASE?
- **RQ2** Are generated adversarial examples transferable to the black-box LLaMA2-7B model?
- **RQ3** Will human judges find the quality of generated adversarial examples reasonable?
- **RQ4** How do $\mathcal{L}_{DAL}$ components impact DALA?
- **RQ5** Does $\mathcal{L}_{DAL}$ outperform other attack losses?

7.1 Automatic Evaluation Results

We use the adversarial examples generated by DALA to attack the white-box BERT-BASE models, which 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 source, we evaluate the attack transferability of the generated adversarial examples on the black-box LLaMA2-7B model, which are not available during DALA fine-tuning. The experimental results on ACC, ASR, and NASR compared with baselines are shown in Table 1.

Attack Performance (RQ1). When attacking the white-box models, DALA obtains the best or second-to-best performance regarding ACC, ASR, and NASR on CoLA, RTE, and MRPC datasets. On SST-2 dataset, although DALA's performance is not the best, NASRs of DALA experience a relatively minor decrease from ASR compared with baselines, implying that adversarial examples generated by DALA are more challenging to detect. Aside from DALA, some baseline methods like TextFooler work well on some datasets. However, NASR$_{MSP}$ of TextFooler on SST-2 and MRPC drops drastically compared to ASR, indicating these adversarial examples are relatively easy to

Table 2: Grammar correctness, prediction accuracy and semantic preservation of original examples (denoted as Orig.) and adversarial examples generated by DALA.

Dataset	Grammar		Accuracy		Semantic	
	DALA	Orig.	DALA	Orig.	DALA	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

detect using MSP detection.

The experimental results indicate that DALA yields reasonable outcomes when attacking a white-box model, and the results remain favorable when considering detectability.

Transferability to LLMs (RQ2). When attacking the black-box LLAMA2-7B model, DALA consistently performs well on SST-2, RTE, and MRPC, outperforming baselines in every evaluation metric. On CoLA, DALA achieves second-to-best results on most evaluation metrics. Further analysis and visualization of attack performance on LLAMA2-7B across five different prompts are displayed in Appendix D.

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

7.2 Human Evaluation (RQ3)

Given that our objective is to generate high-quality adversarial examples with similar semantic meaning to the original examples and imperceptible to humans, we perform human evaluations to assess the generated adversarial examples in terms of grammar, prediction accuracy, and semantic preservation on SST-2 and MRPC datasets. We ask three human judges to evaluate 50 randomly sampled original-adversarial pairs from each dataset. Detailed annotation guidelines are provided in Appendix B.

First, we ask human raters to evaluate the grammar correctness and make predictions of the 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, we ask human judges to assess the semantic preservation of adversarial examples—whether each one maintains the meaning of the original example. We follow Jin et al. (2020) and ask human judges to decide whether the adversarial example is similar (1), ambiguous (0.5), or dissimilar (0) to the corresponding original example. We compare DALA with the best baseline model, TextFooler, on semantic preservation for better evaluation. We take the average score among human raters for grammar correctness and semantic preservation and take the majority class as the predicted label.

Table 3: Ablation study on BERT-BASE regarding MSP.

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

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

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

As shown in Table 2, the grammar correctness scores of adversarial examples generated by DALA are similar to those of original examples. Word perturbations make predictions more challenging, but adversarial examples generated by DALA still show decent accuracy. Compared to TextFooler, DALA can better preserve semantic similarity to original examples. Some generated adversarial examples are displayed in Appendix C.

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 3 and Table 4.

MSP Loss. We ablate $\mathcal{L}_{MSP}$ during fine-tuning phase 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 $|\mathcal{D}_{MSP}|$ and the number of all successful adversarial examples, across all datasets. An interesting finding is that on SST-2 and CoLA, although the model without $\mathcal{L}_{MSP}$ performs 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 $\mathcal{L}_{DAL}$.

MD Loss. We ablate $\mathcal{L}_{MD}$ during the fine-tuning phase to assess the efficacy of $\mathcal{L}_{MD}$. $\mathcal{L}_{MD}$ helps

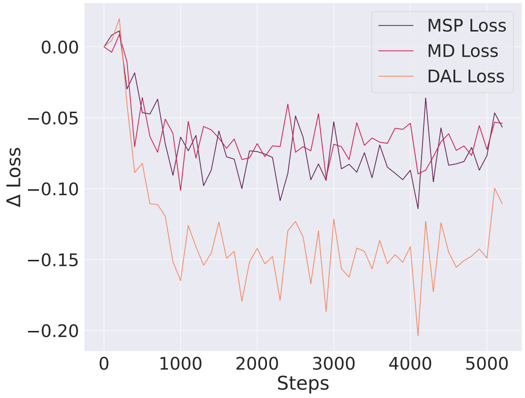


Figure 5: The change of $\mathcal{L}_{MSP}$, $\mathcal{L}_{MD}$, and $\mathcal{L}_{DAL}$ throughout the fine-tuning phase of DALA on SST-2. The x-axis represents fine-tuning steps; the y-axis represents the change of loss compared to the initial loss.

improve MD Detection Rate (DR_{MD}), which is the ratio of $|\mathcal{D}_{MD}|$ and the number of successful adversarial examples, across all the datasets. $\mathcal{L}_{MD}$ also improves NASR_{MD} on all the datasets except SST-2. A similar finding on CoLA also exists that although the model without $\mathcal{L}_{MD}$ performs 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 (RQ4)

To better understand how different loss components contribute to DALA, we visualize the change of $\mathcal{L}_{MSP}$, $\mathcal{L}_{MD}$, and $\mathcal{L}_{DAL}$ throughout the fine-tuning phase of DALA 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, upon closer examination, 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, and the adversarial attack task aims to lead victim models to wrong predictions. Thus, minimizing $\mathcal{L}_{MSP}$ aligns with the objective of the attack task. On the other hand, minimizing $\mathcal{L}_{MD}$ pushes the generated adversarial sentences more like original sentences, and the masked language modeling task is to restore masked tokens to the original tokens. Thus, minimizing $\mathcal{L}_{MD}$ is loosely akin to the objective of the masked language modeling task. While these two objectives are not inherently conflicting, an extreme standpoint reveals that when the latter objective is fully satisfied – meaning the model generates the same examples as the original ones – the former objective naturally becomes untenable.

Table 5: Comparison of DALA with loss variants.

Dataset	Model	ACC↓	ASR↑	MSP NASR↑	MSP DR↓	MD NASR↑	MD 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 $\mathcal{L}_{NCE}$ or $\mathcal{L}_{FCE}$ with $\mathcal{L}_{DAL}$ during the fine-tuning phase to assess the efficacy of our loss $\mathcal{L}_{DAL}$. The results in Table 5 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 find that distribution shifts exist between adversarial examples and original examples in terms of MSP and MD. Thus, we propose a Distribution-Aware LoRA-based Adversarial Attack (DALA) method with the Data Alignment Loss (DAL) and introduce a novel evaluation metric, NASR, which incorporates OOD detection into consideration within a successful attack. Our experiments validate the attack effectiveness of DALA on BERT-BASE and the transferability of DALA 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*.

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. 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.

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.

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*.

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

Appendix

A More Implementation Details

A.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.

For the implementation of baselines, we use the TextAttack[1] package with its default parameters.

A.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 6.

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

Table 6: 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 7: Hyperparameters of different datasets.

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

A.3 Hyperparameters

The hyperparameters used in experiments are shown in Table 7.

A.4 Prompts used for LLAMA2-7B

The constructed prompt templates used for LLAMA2-7B are shown in Table 8. For each run, {instruct} in the prompt template is replaced by different instructions in Table 9, while {text} is replaced with the input sentence.

B 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).

C Examples of Generated Adversarial Sentences

Table 10 displays some original examples and the corresponding adversarial examples generated by DALA. 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 10 shows that DALA only perturbs a very small number of words, leading to model prediction

Table 8: Prompt template for different datasets. {instruct} is replaced by different instructions in Table 9, 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 9: 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."

failure. Besides, the adversarial examples generally preserve similar semantic meanings to their original inputs.

D 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 DALA consistently surpasses other baseline methods for each run.

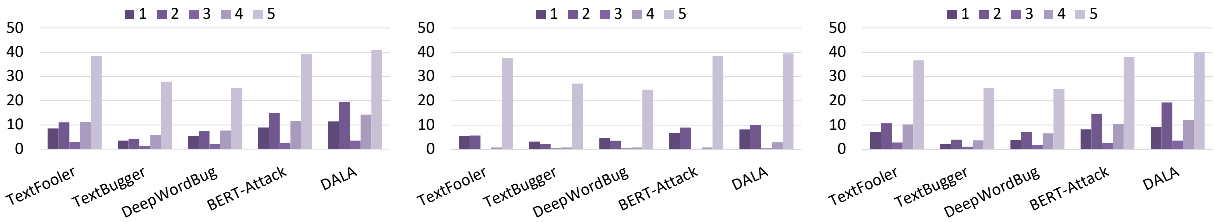
E Observation Experiments

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.

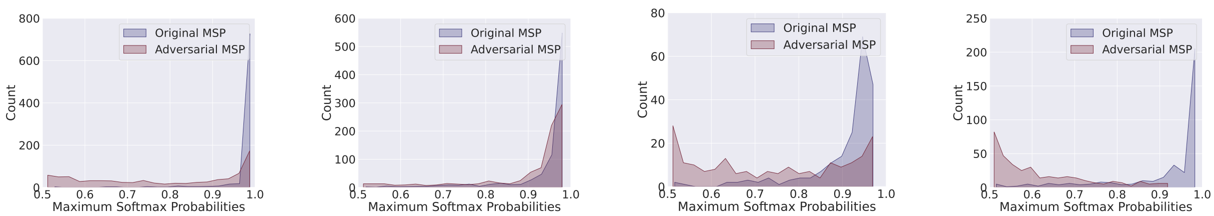
Table 10: Examples of generated adversarial sentences

	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



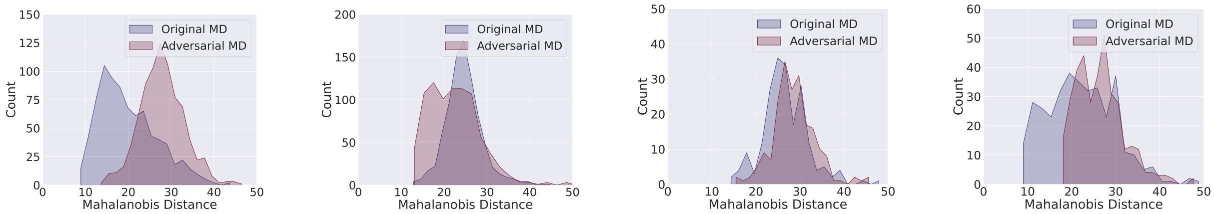
(a) ASR (b) NASR (MSP) (c) NASR (MD)

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



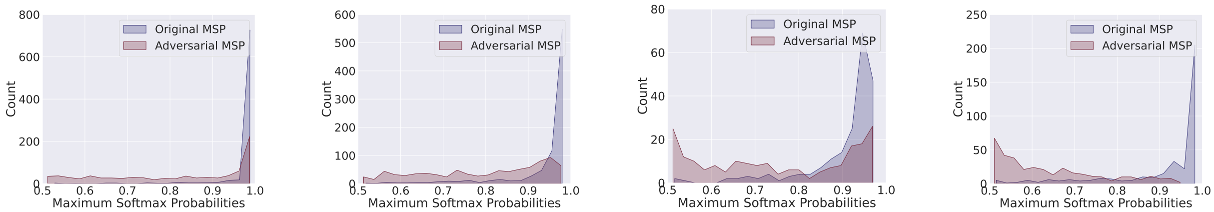
(a) MSP on SST-2 dataset. (b) MSP on CoLA dataset. (c) MSP on RTE dataset. (d) MSP on MRPC dataset.

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



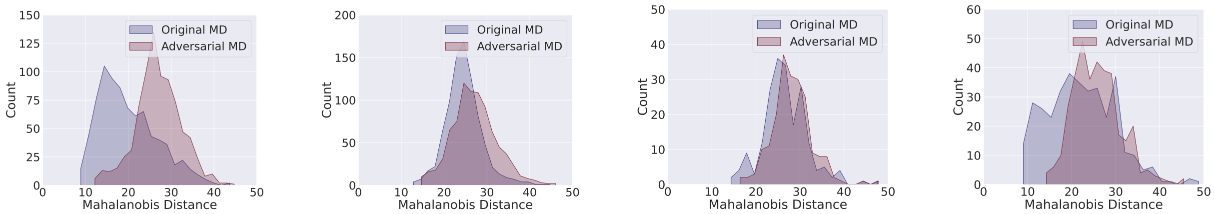
(a) MD on SST-2 dataset. (b) MD on CoLA dataset. (c) MD on RTE dataset. (d) MD on MRPC dataset.

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



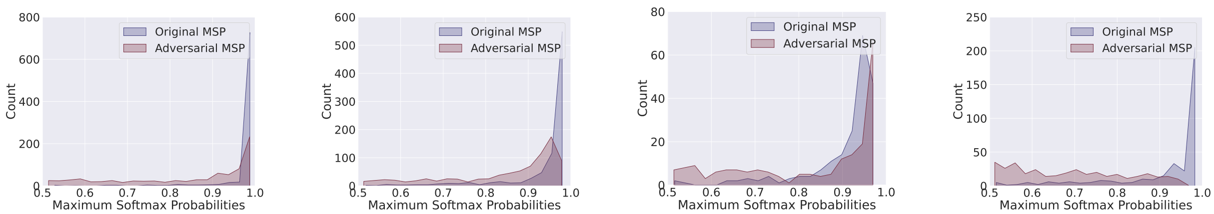
(a) MSP on SST-2 dataset. (b) MSP on CoLA dataset. (c) MSP on RTE dataset. (d) MSP on MRPC dataset.

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



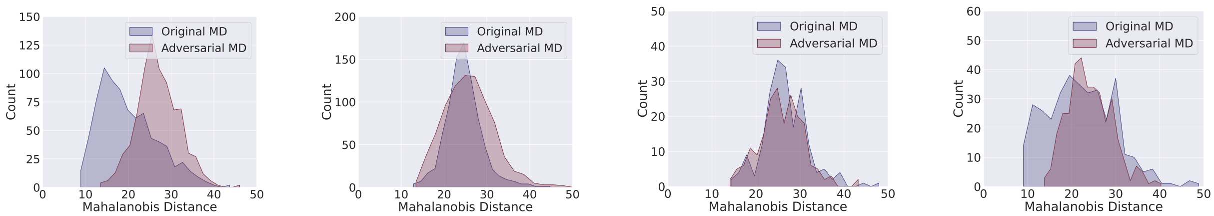
(a) MD on SST-2 dataset. (b) MD on CoLA dataset. (c) MD on RTE dataset. (d) MD on MRPC dataset.

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



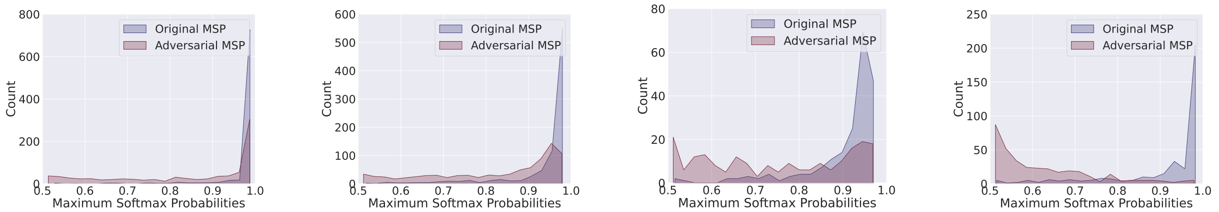
(a) MSP on SST-2 dataset. (b) MSP on CoLA dataset. (c) MSP on RTE dataset. (d) MSP on MRPC dataset.

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.



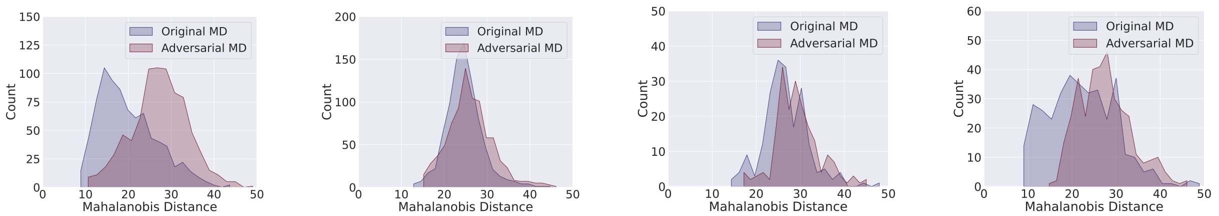
(a) MSP on SST-2 dataset.

(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.