Is Adversarial Vulnerability a Real Trade-off in LoRA Fine-Tuned Models?

Anonymous ACL submission

Abstract

Parameter-Efficient Fine-Tuning (PEFT) methods like Low-Rank Adaptation (LoRA) have emerged as efficient alternatives to full model fine-tuning for adapting large language models to downstream tasks. However, it remains unclear how these lightweight tuning methods fare against adversarial attacks in natural language processing (NLP). We investigate this by testing the adversarial robustness of LoRA-tuned models and comparing against fully fine-tuned models on text classification tasks. We conduct experiments on multiple transformer architectures (LLaMA3, GPT-Neo, BERT, DistilBERT, ALBERT and RoBERTa) fine-tuned on three benchmark datasets (SST-2, AG-News, IMDB). We evaluate model robustness under three strong textual attack algorithms – TextFooler, TextBugger and BAE. Additionally we aim to see if LoRA models lose ability in terms of attack feedback. Specifically, we conduct experiments with the feedback models differing from the attack target models. Our results show that FFT models show no advantage in terms of robustness over LoRA models. For some attacks (e.g. TextFooler) model accuracy is reduced to near-zero for all models. These findings suggest that LoRA's efficiency gains do not inherently weaken model security.

1 Introduction

Large Pre-trained Language Models (PLMs) such as LLaMA (Touvron et al., 2023), GPTneo (Yenduri et al., 2023) and RoBERTa (Zhuang et al., 2021) have achieved remarkable performance on many NLP tasks. Fine-tuning these models on task-specific data can bridge performance gaps, but fully fine-tuning all models for each task is resource-intensive. Parameter-Efficient Fine-Tuning (PEFT) (Xu et al., 2023) methods have been proposed to address this by updating only a small fraction of model parameters while reusing most of the pre-trained weights. One prominent PEFT method is Low-Rank Adaptation (LoRA) (Hu et al., 2021b), which injects a small number of trainable low-rank matrices into each layer's weights instead of modifying the full weight matrices. LoRA drastically reduces training overhead (updating 1% of parameters) while maintaining performance close to full fine-tuning. With fewer parameters updated, the question of robustness arises. Specifically, **are LoRA models more vulnerable to standard adversarial attacks?**

Modern NLP models have been shown be vulnerable to adversarial examples (Goodfellow et al., 2015), inputs perturbed in small ways that cause misclassifications while preserving the original meaning. Numerous attack methods have been developed to expose these weaknesses. For example, TextFooler (Jin et al., 2020) replaces important words with synonyms to fool the classifier while preserving semantics, achieving very high success rates on text classification tasks. Prior studies have demonstrated that fully fine-tuned models can be vulnerable to such attacks – for example, (Jin et al., 2020) reports BERT's accuracy can be driven to near zero by a word-substitution attack. What remains less explored is whether using PEFT methods (like LoRA) alters a model's adversarial surface. On one hand, updating fewer parameters might lead to underfitting, potentially making the model more vulnerable. On the other hand, constraining weight updates might prevent overfitting, possibly improving generalization and robustness on adversarial inputs. One additional aspect that has not been studied in the context of LoRA, is attack transferability across models. That is, **is information that might be useful to an attacker in transfer attacks, lost in LoRA models?**

In this work, we aim to answer both aforementioned questions on robustness and transferability. Our key contributions are:

- We present an extensive empirical study of

adversarial robustness in LoRA versus fully fine-tuned models. We fine-tune a suite of transformer models using both strategies on multiple datasets and evaluate them against strong attacks (TextFooler, TextBugger, BAE). We find that LoRA-tuned models are as robust as FFT models. In fact, LoRA sometimes yields slight robustness improvements on larger architectures, whereas on smaller models it can modestly hurt robustness.

- We evaluate both finetuning approach transfer (LoRA-to-LoRA and Full-to-Full) and cross-finetuning transfer (LoRA-to-Full and Full-to-LoRA), ensuring that cross-finetuning approach patterns of susceptibility are also captured. We find no loss of transferability when using LoRA for transferability.

To our knowledge, this is the first systematic analysis of LoRA's adversarial robustness in NLP. We hope our findings inform the deployment of PEFT models in security-critical applications and spur development of defense techniques compatible with parameter-efficient training.

2 Related Work

Parameter-Efficient Fine-Tuning. LoRA (Hu et al., 2021a) is a popular PEFT method that injects low-rank adapters into transformers, reducing trainable parameters by more than 98% while often matching full fine-tuning performance. Razuvayevskaya et al. (Razuvayevskaya et al., 2024) showed in multilingual news classification that LoRA and adapters reached comparable accuracy to full fine-tuning while training $140\times$ fewer parameters and cutting training time by 32–44%. Accuracy drops were typically below 1%, and in low-resource settings LoRA sometimes outperformed full fine-tuning.

Structural Differences and Forgetting. Shuttleworth et al. (Shuttleworth et al., 2024) found that while LoRA and full fine-tuning achieve similar downstream accuracy, their internal updates diverge. Spectral analysis revealed "intruder dimensions" introduced by LoRA that do not appear in full fine-tuning. These helped LoRA retain pretraining knowledge (less forgetting) but, in continual learning scenarios, accumulated intruder directions degraded performance compared to full fine-tuning. Thus, LoRA trades off efficiency and retention against potential generalization issues.

Transferability of Adversarial Attacks. Prior work shows that adversarial examples often transfer across models, enabling black-box attacks via surrogate-to-target transfer (Papernot et al., 2016; Liang et al., 2018). Motivated by this, we evaluate transferability between LoRA and fully fine-tuned models, testing both cross-strategy (Full→LoRA, LoRA→Full) and within-strategy (Full→Full, LoRA→LoRA) transfers. This probes whether vulnerabilities are strategy-specific or shared across architectures and training regimes.

Robustness. Robustness has been less studied in PEFT. Liang et al. (Liang et al., 2025) reported that LoRA is more resilient to backdoor triggers but more vulnerable to untargeted data poisoning, due to its restricted update geometry. However, prior work has not systematically tested LoRA vs. full fine-tuning under adversarial attacks at inference. Our study fills this gap by benchmarking robustness across six transformer models and three datasets under strong attacks (TextFooler, TextBugger, BAE), showing that adversarial vulnerabilities remain high, though LoRA can slightly improve robustness in larger models.

3 Experiments

We aim to address two main questions in this research:

1. Are LoRA models more vulnerable to standard adversarial attacks?
2. Is information that might be useful to an attacker in transfer attacks, lost in LoRA models?

To evaluate the first question, we run standard adversarial attacks (Section 3.2), on both LoRA and FFT versions of the same models. To evaluate the second question, we use the adversarial examples generated by an attack targeting one model, as adversarial examples for another. The experimental design for this study is deliberately aligned with established traditional adversarial NLP research (Jin et al., 2020; Garg and Ramakrishnan, 2020; Li et al., 2019). Next, we expand on the models, datasets, attacks, and metrics used for these evaluations.

3.1 Models and Fine-Tuning Strategies

We experiment with six transformer-based language models spanning a range of sizes and architectures: GPT-Neo (125M, GPT-3 style decoder model released by EleutherAI), LLaMA (1.3B, an open large language model), BERT-base (110M

parameters), DistilBERT (66M, a distilled smaller BERT) (Sanh et al., 2020), ALBERT-base (12M, uses parameter sharing) (Lan et al., 2020) and RoBERTa-base (125M, robustly optimized BERT). Each model is fine-tuned for text classification on three datasets, We refrain from using substantially larger LLMs, as smaller-scale models (typically under 2B parameters) are more representative of those commonly deployed in production settings. The dataset which we choose are selected based on their sequence length ranging from shorter, medium to longer sequences.

- **SST-2:** Stanford Sentiment Treebank v2, binary sentiment classification dataset derived from movie reviews. Average sequence length is approximately 20 tokens (sst).
- **AG-News:** 4-class news topic classification dataset with categories: World, Sports, Business, and Science/Technology. Average sequence length is around 100 tokens (agn).
- **IMDB:** Binary sentiment classification on movie reviews, containing 50,000 reviews. Average sequence length is about 234 words, with a standard deviation of 173 words (Maas et al., 2011).

As noted, we follow two finetuning strategies for our models:

- **Full Fine-Tuning (FFT):** All model parameters are updated during training on the task data.
- **LoRA Fine-Tuning:** The original model weights are frozen, and LoRA adapters with rank $r = 8$ are inserted into every layer's attention modules (applied to the query and value projection matrices). Only the LoRA adapter weights are trained, accounting for roughly 0.5–1.5% of total parameters, along with the task classification head.

We use the HuggingFace `transformers` (Vaswani et al., 2023) library and an open-source LoRA implementation to ensure consistent training pipelines. Each model is fine-tuned with same hyperparameter and choosing model which reaches high accuracy on the clean validation set (typically in the high 90% range for these datasets), so that we have a strong baseline before adversarial evaluation. This controlled setup allows for fair comparison of robustness without confounding effects from poor base accuracy.

Shared Training Hyperparameters

Hyperparameter	Value
Number of epochs	3
Batch size (train/eval)	32
Warm-up steps	500
Learning rate	3e-5
Weight decay	0.01
Logging steps	10
Evaluation/saving strategy	epoch
Load best model at end	`True`
Optimizer	AdamW
LR scheduler	Linear with warm-up

Table 1: Shared hyperparameters used across training.

3.2 Adversarial Attacks

For this study the examined attacks are black-box attacks. Specifically, the attacker can query a target model and receive feedback in terms of labels and probabilities, but do not have knowledge of the architecture or weights of a model. Note the goal of these attacks is to produce examples which cause misclassification while preserving meaning for humans. This is different than jailbreak and backdoor attacks which we *exclude* as they target different threat models and need not preserve meaning.

We evaluate robustness using three prominent attack algorithms: TextFooler, TextBugger and BAE. We choose these attacks because they remain state-of-the-art for semantic word-level adversarial evaluations, while many newer methods are oriented toward backdoor or jailbreak settings. All attacks are implemented using the `TextAttack` framework (Morris et al., 2020) for reproducibility. Following previous research (Jin et al., 2020; Li et al., 2019, 2020), we evaluate adversarial robustness using a fixed set of 1,000 test samples per dataset, balanced across classes.

We adopt the default attack parameters from the original authors. Briefly:

- **TextFooler:** (Jin et al., 2020) uses a word-importance ranking based on the classifier's prediction probabilities and replaces words with top semantically similar alternatives until the prediction changes.
- **TextBugger:** (Li et al., 2019)uses a combination of character-level operations (e.g., inserting a space or misspelling) and word replacements; it greedily applies perturbations until misclassification is achieved.
- **BAE:** (Garg and Ramakrishnan, 2020) generates up to K candidate substitutions or inser-

tions for each masked position (using a BERT masked language model) and greedily applies the change that yields the largest drop in the model's confidence, iterating until the label flips.

These attacks collectively cover a range of perturbation styles (character-level noise, word-level synonyms and fluent paraphrase-like edits). All attacks are untargeted, aiming to cause any label change. We consider an attack successful if the model's predicted label is incorrect and different from the original (i.e., the attack must actually fool the model, not merely exploit an already incorrect prediction).

3.3 Evaluation Metrics

We measure model robustness with three metrics: Accuracy Under Attack (AUA), Attack Success Rate (ASR), and Average Queries. AUA measures the accuracy of the model on adversarial samples. More robust models should have higher AUA. ASR measures the effectiveness of an attack. It is an alternative measure that takes original model accuracy into account. Formally,

$$\text{ASR} = \frac{\text{Original_Acc.} - \text{AUA}}{\text{Original_Acc.}} \quad (1)$$

A robust model would have a lower ASR. AQ measures the average number of queries sent to the target model by the attacker during one adversarial example. AQ(Average Number of Queries) remains a meaningful secondary signal of resistance under practical constraints. Intuitively, *fewer* queries to achieve a successful attack (lower AQ) indicate an easier-to-exploit model and thus *lower* robustness, whereas *more* required queries (higher AQ) suggest greater resistance to black-box probing.

3.4 Transfer-attack metrics.

We quantify *transferability* by generating adversarial examples on a **source** model and evaluating them on a **target** model with *no additional target queries*. For each attack, we compute transfer ASR over the subset of inputs correctly classified by both source and target, conditioning on source-side attack success (i.e., what fraction still fool the target). We then macro-average across relevant source→target groups to summarize cross-model alignment. Specifically, **F→F** denotes adversarial examples generated on Full-fine-tuned source models and evaluated on Full-fine-tuned target models, irrespective of their backbone architecture, while **L→L** indicates examples generated on LoRA-tuned source models and evaluated on LoRA-tuned target models, also irrespective of architecture. Similarly, **L→F** represents attacks generated from LoRA sources and tested on Full targets, and **F→L** the reverse. We further report **L (S.A.)/F (S.A.)** to capture transfers within the same architecture family (e.g., encoder-only→encoder-only) and **L (D.A.)/F (D.A.)** for transfers across different architecture families (e.g., encoder-only→decoder-only). Higher transfer ASR indicates stronger feature overlap and thus lower robustness generalization across models, whereas Average Queries (AQ) pertain solely to the source model since target evaluation incurs zero additional queries.

4 Robustness of LoRA Models

We run the three noted attacks against both the six Fully Fine-Tuned (FFT) models and LoRA fine-tuned models. The main results for SST-2, AGNEWS and IMDB can be found in Tables 2, 3, 4, respectively. Overall, we observe that models display no more robustness against adversarial attacks compared to LoRA fine-tuned models.

4.1 ASR Comparison

Across SST-2, AGNEWS, and IMDB, we observe that LoRA-tuned and fully fine-tuned models exhibit broadly comparable levels of vulnerability to adversarial attacks, with differences in Attack Success Rate (ASR) typically limited to a few percentage points. On SST-2, LoRA configurations produced the higher ASR in 7 out of 18 model–attack settings, while Full configurations were higher in 10 (with 1 tie). For instance, GPT-NEO LoRA reached 96.97% ASR under TextFooler, notably higher than its Full counterpart at 84.15%. In contrast, larger models such as LLAMA-3 show only marginal differences (e.g., SST-2 BAE: 65.6% vs. 69.9%), indicating that the effect of fine-tuning strategy diminishes with scale.

On AGNEWS, the results are more balanced: LoRA achieved higher ASR in 9 out of 18 settings, while Full had the advantage in the remaining 9. Most differences were modest, such as LoRA BERT under TextFooler (84.55%) versus Full BERT (85.86%). However, an anomaly is observed with GPT-NEO: LoRA yielded 91.23% ASR under TextFooler compared to 85.71% for Full, suggesting that low-rank adaptation can am-

MT	MC	**TextFooler** (ASR/AUA/AQ)	**TextBugger** (ASR/AUA/AQ)	**BAE** (ASR/AUA/AQ)
Lora	GPTneo	96.97 / 2.64 / 79.80	75.07 / 21.67 / **35.65**	**66.89 / 28.78** / 51.00
Full	GPTneo	**84.15 / 5.16 / 84.15**	**69.82 / 26.38** / 39.67	68.24 / 27.75 / **51.01**
Lora	Llama3	**94.36 / 5.39 / 101.14**	**52.22 / 45.64 / 45.68**	**65.55 / 32.91 / 52.67**
Full	Llama3	96.13 / 3.44 / 76.37	70.84 / 25.92 / 40.42	69.94 / 26.72 / 49.10
Lora	BERT	**94.85** / 4.70 / 85.88	70.23 / 27.18 / 39.43	67.96 / 29.24 / 50.54
Full	BERT	95.16 / **4.47 / 87.57**	**68.73 / 28.90 / 41.47**	**66.38 / 31.08 / 52.71**
Lora	DistillBert	**94.71** / 4.70 / **80.57**	**69.42** / 27.18 / **39.44**	69.55 / 27.06 / **50.73**
Full	DistillBert	96.94 / **2.75** / 79.35	72.10 / **25.11** / 39.36	**66.37 / 30.28** / 50.24
Lora	Albert	**94.78** / 4.70 / **89.99**	**64.63 / 31.88** / 42.68	66.54 / 30.16 / **52.46**
Full	Albert	95.54 / **4.13** / 80.91	72.52 / 25.46 / **39.57**	**66.54 / 30.73** / 50.39
Lora	Roberta	**94.89** / 4.60 / 88.23	62.89 / 35.26 / **23.56**	66.29 / 31.65 / 45.11
Full	Roberta	95.00 / **4.70 / 92.47**	**60.98 / 36.70** / 44.05	**65.73 / 32.22 / 51.55**

Table 2: **SST-2:** Attack Success Rate (ASR %), Accuracy Under Attack (AUA %), and Average Queries (AQ) for TextFooler, TextBugger, and BAE across all models. Per-cell **bold** indicates the *lower ASR* (i.e., more robust) between LoRA and FULL configurations for the same model.

plify instability in smaller-capacity models. This dataset highlights that robustness outcomes are not systematically tied to the fine-tuning method, and instead fluctuate based on architecture and attack type.

On IMDB, the overall picture is one of near-complete vulnerability regardless of fine-tuning strategy. Both LoRA and Full models consistently reached ∼100% ASR under TextFooler, with only slight differences (e.g., LoRA BERT at 100.0% vs. Full BERT at 98.4%). LoRA recorded higher ASR in 11 out of 18 cases, including under TextBugger and BAE for DISTILBERT and ALBERT, though gaps were small (e.g., LoRA ALBERT BAE: 72.0% vs. Full 67.0%). Overall, out of 54 total model–dataset–attack comparisons, LoRA surpassed Full in 27 cases, while Full was higher in 26 (with 1 tie). This confirms that LoRA provides efficiency gains without systematically degrading robustness, with anomalies primarily confined to smaller models and a normalization effect evident in larger architectures.

4.2 Query Robustness

Beyond attack success rates, we examine the average number of queries (AQ) required to generate successful adversarial examples. Overall, LORA-tuned models tend to be more query-efficiently attacked compared to their fully fine-tuned counterparts, though exceptions exist in larger models. On SST-2, most LoRA models require slightly fewer queries than Full (e.g., BERT LoRA: 85.9 vs. Full: 87.6), while DISTILBERT shows a minor reversal (LoRA: 80.6 vs. Full: 79.4). A striking anomaly arises with ROBERTA under TextBugger, where LoRA requires only 23.6 queries versus Full at 44.1, suggesting certain low-rank adaptations can make models easier to perturb.

On AGNEWS, query counts remain high across the board (typically >300 for TextFooler), but LoRA often requires fewer attempts (e.g., GPT-NEO LoRA: 282 vs. Full: 316; DISTILLBERT LoRA: 319 vs. Full: 336). While the differences are modest, they reinforce the trend that adversaries need slightly fewer queries to compromise LoRA models. However, some reversals occur, such as DISTILBERT, where LoRA: 319 vs. Full: 336, indicating dataset-specific variation.

The IMDB dataset highlights the strongest anomalies. LoRA GPT-NEO reaches 100% ASR with only 370 queries, compared to Full requiring 472.9. Similarly, LoRA BERT (546 vs. Full 748) and LoRA ALBERT (660 vs. Full 833) also reduce the query cost substantially. Yet, in larger architectures such as ROBERTA, the trend reverses: LoRA requires 648 queries versus Full at 621, and in LLAMA-3, LoRA can need even more queries than Full under TextFooler (736 vs. 450). These results suggest that while LORA generally reduces the number of queries needed for a successful attack—likely due to "softer" decision boundaries introduced by low-rank decomposition, larger and more stable models can offset this effect, sometimes making Full models more query-efficiently attacked. Across all three datasets, LoRA required fewer queries in roughly 70% of the cases, with exceptions concentrated in ROBERTA and LLAMA-3, underscoring that query efficiency improvements

MT	MC	**TextFooler** (ASR/AUA/AQ)	**TextBugger** (ASR/AUA/AQ)	**BAE** (ASR/AUA/AQ)
Lora	GPTneo	91.23 / **8.0** / 282.19	64.80 / 32.16 / 142.46	18.86 / 74.0 / **104.89**
Full	GPTneo	**85.71** / 3.58 / **315.74**	**53.86 / 43.65 / 152.28**	**17.81 / 74.57** / 104.12
Lora	Llama3	**72.78 / 5.77 / 390.25**	**34.11 / 62.23 / 177.33**	**10.38 / 84.6 / 111.88**
Full	Llama3	83.49 / 5.58 / 319.64	57.61 / 39.85 / 138.49	14.91 / 79.9 / 102.98
Lora	BERT	**84.55 / 14.0 / 327.11**	58.50 / 37.6 / **159.80**	17.44 / 74.8 / 111.79
Full	BERT	85.86 / 14.1 / 320.11	**58.38 / 41.5** / 164.58	**16.35 / 83.4 / 116.26**
Lora	DistillBert	87.57 / 11.1 / 319.00	57.33 / 38.1 / **152.36**	17.92 / 73.3 / **113.31**
Full	DistillBert	**81.82 / 17.6 / 336.76**	**53.41 / 45.1** / 156.12	**15.60 / 81.7** / 110.08
Lora	Albert	83.20 / 14.9 / **310.00**	**61.22** / 34.4 / **153.06**	**19.95** / 71.0 / 111.77
Full	Albert	**82.71 / 16.7** / 301.37	72.26 / **26.8** / 151.87	24.84 / **72.6 / 117.32**
Lora	Roberta	**80.54** / 18.0 / **358.89**	**49.41 / 46.8** / 166.64	**17.19** / 76.6 / 115.21
Full	Roberta	81.46 / **18.3** / 336.00	55.22 / 44.2 / **181.34**	21.78 / **77.2 / 116.96**

Table 3: AGNEWS: Attack Success Rate (ASR %), Accuracy Under Attack (AUA %), and Average Queries (AQ) for TextFooler, TextBugger, and BAE. Per-cell bold: Lower value between Lora and Full for the same model. We use ASR as the primary metric for evaluating model robustness.

MT	MC	**TextFooler** (ASR/AUA/AQ)	**TextBugger** (ASR/AUA/AQ)	**BAE** (ASR/AUA/AQ)
Lora	GPTneo	100.00 / **0** / 370	97.90 / **1.9 / 298.63**	86.39 / **12.3 / 277.67**
Full	GPTneo	**99.89** / 0.19 / **472.96**	**96.46** / 3.2 / 334.04	**79.29** / 18.79 / 352.92
Lora	Llama3	**99.38** / 0.69 / **736.05**	**89.28** / 10.3 / **466**	**56.61 / 41.75 / 395.72**
Full	Llama3	99.58 / **0.4** / 450	96.76 / **3.1** / 337.32	67.78 / 30.8 / 361.42
Lora	BERT	100.00 / **0.0 / 546.21**	96.67 / **3.1 / 427.53**	74.36 / **23.9 / 359.02**
Full	BERT	**98.40** / 1.7 / 747.97	**85.47** / 14.3 / 551.82	**64.94** / 34.5 / 379.64
Lora	DistillBert	**99.78** / 0.2 / 411.96	97.12 / **2.6 / 339.40**	79.42 / **18.6 / 289.69**
Full	DistillBert	99.90 / **0.1 / 459.08**	**93.98** / 5.9 / 392.71	**73.67** / 25.8 / 330.51
Lora	Albert	99.57 / **0.4** / 660.00	93.17 / **6.3 / 504.00**	72.02 / **25.8 / 369.34**
Full	Albert	**95.38** / 4.5 / **833.00**	**81.03** / 18.5 / **667.00**	**66.97** / 32.2 / **388.70**
Lora	Roberta	**99.57** / 0.4 / **648.66**	**86.60** / 12.6 / 491.00	**62.70** / 35.0 / **374.90**
Full	Roberta	99.69 / **0.3** / 621.00	89.06 / **10.7 / 546.30**	71.57 / **27.8** / 358.51

Table 4: IMDB: Attack Success Rate (ASR %), Accuracy Under Attack (AUA %), and Average Queries (AQ) for TextFooler, TextBugger, and BAE. Per-cell bold: Lower value between Lora and Full for the same model. We use ASR as the primary metric for evaluating model robustness.

are more pronounced in smaller and mid-sized models.

4.3 Architecture Differences

Across datasets, consistent architectural patterns emerge. Larger models such as LLAMA-3 (1.3B) show more stable robustness, with LORA and Full fine-tuning yielding very similar ASR values, suggesting a normalization effect at scale. In contrast, smaller models (<500M parameters, e.g., DISTILBERT, ALBERT) tend to be more susceptible, with LORA sometimes slightly worsening robustness compared to Full. ROBERTA models maintain strong clean accuracy but remain highly vulnerable under attack, confirming that pretraining depth alone does not guarantee adversarial robustness. GPT-NEO behaves comparably to BERT-style classifiers on IMDB and SST-2, confirming its susceptibility to lexical perturbations. Cross-dataset trends further reinforce these findings: models that are highly vulnerable on one dataset (e.g., DISTILBERT showing ∼97% ASR under TextFooler on SST-2) also fail on the others, while larger models such as LLAMA-3 show relatively lower ASR across datasets under weaker attacks like BAE, highlighting that scale moderates but does not eliminate vulnerability. Importantly, the ranking of attack strength is preserved across datasets TextFooler > TextBugger > BAE showing that the attack methods generalize in effectiveness regardless of dataset or model family.

Atk	Model	F→L	L→F	F→F	L→L	L (S.A.)	F (S.A.)	L (D.A.)	F (D.A.)
SST-2 Dataset									
TextFooler	GPT-Neo	45.21	31.82	37.78	49.33	51.57	44.22	43.12	42.65
	LLaMA3	32.74	19.59	31.82	41.29	33.79	34.54	40.68	38.85
	BERT	36.55	41.66	47.25	51.20	81.83	71.64	39.35	35.95
	DistBERT	44.29	42.46	52.54	53.39	66.99	74.78	44.11	43.14
	ALBERT	36.07	39.73	49.75	53.39	63.48	47.87	44.81	48.52
	RoBERTa	32.73	37.78	39.73	49.33	38.64	40.19	45.71	43.45
TextBugger	GPT-Neo	33.61	26.43	17.68	25.00	20.56	12.42	22.66	20.53
	LLaMA3	18.69	8.14	26.43	20.21	11.87	11.21	20.36	17.93
	BERT	15.15	23.82	26.62	30.79	54.85	44.11	21.80	16.24
	DistBERT	27.03	24.61	35.81	32.94	43.10	50.13	25.91	27.67
	ALBERT	12.50	19.61	25.85	32.94	32.84	23.60	27.96	32.98
	RoBERTa	11.18	17.68	19.61	25.00	13.36	10.83	24.10	20.85
BAE	GPT-Neo	46.52	39.71	43.80	47.50	67.54	61.42	40.07	39.49
	LLaMA3	39.16	33.90	39.71	44.60	59.12	58.88	41.22	40.56
	BERT	38.56	42.95	46.64	47.03	81.58	72.59	37.67	36.60
	DistBERT	58.03	41.37	61.05	48.30	73.53	78.49	39.10	55.75
	ALBERT	39.10	41.80	47.18	48.30	72.52	68.16	39.30	57.82
	RoBERTa	40.34	43.80	41.80	47.50	59.40	60.22	41.70	39.73

Table 5: SST-2: Transferability of adversarial attacks based on sementic word perturbation percentage across fine-tuning strategies and architectures. Metrics report the averaged Attack Success Rate (ASR).

Atk	Model	F→L	L→F	F→F	L→L	L (S.A.)	F (S.A.)	L (D.A.)	F (D.A.)
AGNEWS Dataset									
TextFooler	GPT-Neo	7.75	15.11	17.23	7.02	5.54	6.57	14.17	9.48
	LLaMA3	22.24	11.93	15.11	7.27	5.73	8.11	13.55	14.73
	BERT	14.76	18.96	6.05	8.69	34.71	10.98	9.65	10.29
	DistBERT	13.80	16.85	4.28	9.86	35.33	4.81	8.96	9.89
	ALBERT	13.01	18.45	4.98	9.86	15.05	4.77	13.02	9.90
	RoBERTa	17.27	17.23	18.45	7.02	5.17	6.18	14.25	9.55
TextBugger	GPT-Neo	15.13	22.72	15.93	8.87	6.34	8.97	17.82	15.92
	LLaMA3	5.25	18.03	22.72	14.93	12.43	12.94	16.03	20.55
	BERT	23.11	25.83	10.65	10.95	49.88	15.67	12.09	17.12
	DistBERT	19.86	26.77	7.83	15.57	53.00	5.70	14.80	15.47
	ALBERT	20.79	22.95	8.73	15.57	23.61	6.44	20.68	15.33
	RoBERTa	24.36	15.93	22.95	8.87	6.77	7.83	17.73	16.14
BAE	GPT-Neo	3.58	6.54	8.64	4.10	2.56	2.69	7.06	3.71
	LLaMA3	0.00	3.71	6.54	4.62	3.45	5.32	7.27	6.58
	BERT	4.97	8.07	1.77	5.29	11.10	3.11	5.79	3.42
	DistBERT	4.56	6.47	0.94	5.76	11.77	2.30	4.98	2.84
	ALBERT	5.17	8.52	1.91	5.76	4.98	1.83	6.34	2.93
	RoBERTa	7.46	8.64	8.52	4.10	2.89	2.86	6.99	3.68

Table 6: AGNEWS: Transferability of adversarial attacks based on word perturbation percentage across fine-tuning strategies and architectures. Metrics report the averaged Attack Success Rate (ASR).

5 Transferability of LoRA

We test different models' feedback for attack transferability on the same 3 datasets and 5 models. The descriptions of the transfers measured can be found in Section 3.4.

Overall, we observe that LoRA models tend to exhibit **lower average ASR** when targeted by attacks generated from fully fine-tuned models. Conversely, when LoRA models serve as the source (feedback) model, the ASR on fully fine-tuned models increases. This trend is consistent across datasets, suggesting that LoRA models can act as more effective feedback models for transfer attacks. There are some anomalies: for example, the Llama3 model behaves differently, where a fully fine-tuned Llama3 attacking a LoRA Llama3 yields an averaged ASR of 32.74, whereas a LoRA Llama3 attacking a full Llama3 results in 19.56 on the SST-2 dataset. Despite these exceptions, LoRA models generally maintain robustness comparable to fully fine-tuned models and in some cases they can even be more robust.

5.1 Transferability Across Architectures

Across all three datasets, adversarial attacks demonstrate a clear advantage when transferred within the same architecture (S.A.) compared to across different architectures (D.A.). For example, in SST-2 under TextFooler (Table 5), BERT shows

Atk	Model	F→L	L→F	F→F	L→L	L (S.A.)	F (S.A.)	L (D.A.)	F (D.A.)
IMDB Dataset									
TextFooler	GPT-Neo	15.56	24.91	21.26	7.60	4.60	5.46	15.20	13.09
	LLaMA3	25.23	14.63	24.91	10.09	9.20	7.58	16.97	19.57
	BERT	19.61	21.32	4.16	9.08	47.67	7.12	8.71	12.84
	DistBERT	18.33	23.67	2.16	11.57	55.51	5.47	10.05	11.20
	ALBERT	18.95	19.27	4.68	11.57	7.50	4.69	19.65	11.36
	RoBERTa	23.78	21.26	19.27	7.60	6.38	7.70	14.85	12.64
TextBugger	GPT-Neo	17.79	24.10	24.06	8.03	4.33	6.65	17.33	16.90
	LLaMA3	28.15	15.90	24.10	10.43	9.95	8.90	18.70	23.55
	BERT	21.87	24.31	6.54	9.32	54.50	7.48	9.28	15.55
	DistBERT	21.83	25.17	4.76	11.30	57.80	7.15	10.32	14.52
	ALBERT	23.04	22.31	7.34	11.30	7.21	5.11	20.44	14.93
	RoBERTa	26.83	24.06	22.31	8.03	6.45	10.47	16.91	16.14
BAE	GPT-Neo	12.24	26.60	23.41	12.87	7.81	12.01	19.58	18.64
	LLaMA3	0.00	11.94	26.60	14.62	13.93	17.98	20.03	28.34
	BERT	22.13	21.83	10.40	12.75	38.21	10.10	13.11	17.50
	DistBERT	21.38	28.08	10.20	17.84	48.45	11.26	17.87	16.69
	ALBERT	22.80	22.37	12.27	17.84	12.66	8.25	25.02	17.30
	RoBERTa	31.82	23.41	22.37	12.87	9.96	14.92	19.15	18.06

Table 7: IMDB: Transferability of adversarial attacks based on sentence-level prediction flips across fine-tuning strategies and architectures. Metrics report the averaged Attack Success Rate (ASR).

a striking 81.83% success rate when transferring within LoRA (L S.A.), and 71.64% under Full fine-tuning (F S.A.), but cross-architecture transferability drops sharply to 39.35% and 35.95% for L(D.A.) and F(D.A.), respectively. DistilBERT follows the same pattern, with high S.A. transferability (66.99% and 74.78%) that collapses to near 44% on D.A. columns. This suggests that adversarial perturbations strongly exploit shared representational features inside the same backbone, while architectural differences weaken the alignment of perturbations with target decision boundaries.

The same-architecture dominance is consistent across AGNEWS and IMDB as well. On AGNEWS (Table 6), BERT again shows much higher same-architecture rates (34.71% L S.A., 10.98% F S.A.) compared to cross-architecture values (9.65% and 10.29%). DistilBERT exhibits an even sharper divide, with S.A. transferability peaking above 53.00% for L S.A. under TextBugger, while D.A. values remain closer to 15%. IMDB (Table 7) replicates this gulf: DistilBERT records 55.51% for L S.A. but only 10.05–11.20% for D.A. under TextFooler, with BERT showing a similar gap (47.67% vs. under 13%). These results, consistent across datasets and attacks, confirm that adversarial examples flourish most when both source and target share the same architecture, whereas cross-architecture transferability remains weaker and rarely crosses into high success ranges. A few anomalies stand out in the results. In SST-2 (Table 5), DistilBERT under BAE shows unusually high cross-architecture transferability (F(D.A.) 55.75%, L(D.A.) 39.10%), exceeding typical baselines. Likewise, in IMDB (Table 7), RoBERTa maintains moderate transfer across architectures with less severe drop-offs between S.A. and D.A. These isolated cases suggest that certain model or dataset characteristics may occasionally enhance perturbation generalization, though overall, same-architecture transferability remains dominant.

6 LoRA Rank Effect on Robustness

We conduct experiments to examine if LoRA's rank affects the robustness of the model. Specifically, we fine-tuned BERT models using LoRA at varying ranks (2, 4, 8, and 16) and evaluated them under thesame attack algorithms. Overall we find no change in robustness based on rank. The full results can be found in Appendix A.

7 Conclusion

We conducted an empirical study on the adversarial robustness of LoRA-tuned versus FFT models across three datasets and six transformer architectures. While all models remain vulnerable to adversarial attacks, LoRA did not degrade robustness and in some larger models even improved it, likely due to its regularization effect. Smaller models, however, showed mild underfitting under LoRA. Adversarial examples were highly transferable within architectures but less so across them, emphasizing the role of model design in robustness. Overall, combining LoRA with adversarial training may produce models that are both efficient and resilient.

Limitations

We restricted our study to text classification with untargeted attacks. Other tasks (e.g., question answering or text generation) or targeted attacks may yield different results. Other PEFT methods such as Adapters or Prompt-Tuning could behave differently. We also treated models independently rather than in ensembles, leaving system-level defenses unexplored.

Ethics Statement

This work adheres to the ACL Code of Ethics and ARR guidelines. We study semantics-preserving, black-box adversarial attacks using only public benchmarks (SST-2, AGNEWS, IMDB); no human subjects or personally identifiable information were collected, so IRB approval was not required. To mitigate dual-use risks, we do not probe production systems and provide methodological detail sufficient for scientific reproducibility while omitting operational specifics that would enable automated abuse. We acknowledge that benchmark corpora can encode socio-linguistic biases that may affect both clean and adversarial performance and we recommend auditing on domain and group-representative data prior to any deployment. To reduce environmental impact, we reuse checkpoints, apply parameter-efficient fine-tuning (LoRA), and fix evaluation budgets; we encourage reporting energy/carbon estimates when feasible. Our conclusions are scoped to text classification under untargeted, semantics-preserving, black-box and transfer-attack settings and may not generalize to other tasks or threat models.

Acknowledgments

We thank our advisors and colleagues for their guidance and constructive feedback throughout this work. We also acknowledge the support of the open-source community, whose implementations and datasets made this research possible.

References

Ag-news dataset. https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset. Accessed: 2025-10-05.

Sst-2 dataset. https://huggingface.co/datasets/stanfordnlp/sst2. Accessed: 2025-10-05.

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)*. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *Preprint*, arXiv:1412.6572.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021a. LoRA: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021b. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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. *Preprint*, arXiv:1907.11932.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *Preprint*, arXiv:1909.11942.

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

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *Preprint*, arXiv:2004.09984.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4208–4215.

Zi Liang, Haibo Hu, Qingqing Ye, Yaxin Xiao, and Ronghua Li. 2025. Does low rank adaptation lead to lower robustness against training-time attacks? In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*. To appear. arXiv:2505.12871.

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

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack:

A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *Preprint*, arXiv:2005.05909.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 601–618. IEEE.

Olesya Razuvayevskaya, Ben Wu, Jo ao A. Leite, Freddy Heppell, Ivan Srba, Carolina Scarton, Kalina Bontcheva, and Xingyi Song. 2024. Comparison between parameter-efficient techniques and full fine-tuning: A case study on multilingual news article classification. *PLOS ONE*, 19(5):e0301738.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. 2024. LoRA vs Full Fine-tuning: An Illusion of Equivalence. *arXiv preprint arXiv:2410.21228*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *Preprint*, arXiv:2312.12148.

Gokul Yenduri, Ramalingam M, Chemmalar Selvi G, Supriya Y, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H Jhaveri, Prabadevi B, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. 2023. Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *Preprint*, arXiv:2305.10435.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

AT	MT	MC	AQ	SA	FA	SKA	PWP %	OA	AUA	ASR
TextBugger	LoRA8	BERT	159.80	530	376	94	36.61	90.6	37.6	58.5
TextBugger	LoRA2	BERT	156.84	519	400	81	36.52	91.9	40.0	56.47
TextBugger	LoRA4	BERT	158.11	522	391	87	35.27	91.3	39.1	57.17
TextBugger	LoRA16	BERT	158.59	478	430	92	37.22	90.8	43.0	52.64
TextBugger	Full	BERT	164.58	582	415	3	33.80	99.7	41.5	58.38
TextFooler	LoRA8	BERT	327.11	766	140	94	24.40	90.6	14.0	84.55
TextFooler	LoRA2	BERT	301.68	798	121	81	23.31	91.9	12.1	86.83
TextFooler	LoRA4	BERT	302.30	766	147	87	22.84	91.3	14.7	83.9
TextFooler	LoRA16	BERT	301.89	792	116	92	23.97	90.8	11.6	87.22
TextFooler	Full	BERT	320.11	856	141	3	23.06	99.7	14.1	85.86
BAE	LoRA8	BERT	111.79	158	748	94	8.46	90.6	74.8	17.44
BAE	LoRA2	BERT	107.68	154	765	81	6.58	91.9	76.5	16.76
BAE	LoRA4	BERT	108.74	141	772	87	7.58	91.3	77.2	15.44
BAE	LoRA16	BERT	108.22	139	769	92	7.47	90.8	76.9	15.31
BAE	Full	BERT	116.26	163	834	3	8.60	99.7	83.4	16.35

Table 8: Adversarial attack performance of LoRA- and fully fine-tuned BERT models on the AG News dataset across three attack types. Metrics include Average Queries (AQ), Successful Attacks (SA), Failed Attacks (FA), SKA, Percentage of Word Perturbation (PWP%), Overall Accuracy (OA), Accuracy Under Attack (AUA), and Attack Success Rate (ASR).

A Effect of LoRA Rank on Robustness

We conducted experiments using the AG News dataset Table 8, a well-established benchmark for topic classification that consists of a large number of samples and four distinct output classes. The dataset's diversity and scale make it a suitable choice for evaluating fine-tuned models under adversarial conditions. We fine-tuned BERT models using LoRA at varying ranks specifically, rank 2, 4, 8, and 16 and evaluated them under three widely used adversarial attack algorithms: TextBugger, TextFooler, and BAE.

Across all attacks and ranks, the models exhibited nearly identical attack success rates (ASRs), with only minor fluctuations. For instance, under the TextBugger attack, ASRs for LoRA configurations ranged narrowly from 52.64% (rank 16) to 58.5% (rank 8), while under the stronger TextFooler attack, ASRs remained in a tight band between 83.9% and 87.22% across all LoRA ranks. Even the subtler BAE attack showed minimal differences in ASR across ranks, all lying between 15% and 17%. These trends persisted despite differences in the average number of queries or perturbed words across attack types, reinforcing that the rank of the LoRA adaptation module does not significantly influence the model's vulnerability to adversarial attacks. Notably, these LoRA models matched the robustness of a fully fine-tuned BERT baseline while using significantly fewer trainable parameters. This strongly supports the conclusion that low-rank LoRA configurations (even as low as rank 2) can be deployed without sacrificing robustness, making them highly efficient for downstream tasks, especially in resource-constrained deployment settings. The results from the AG News dataset thus provide compelling evidence that LoRA rank selection has minimal, if any, bearing on adversarial performance, and lower ranks can be safely utilized to reduce overhead without compromising model security.