AdvChar: Attacking Interpretable NLP Systems

Eldor Abdukhamidov, Tamer Abuhmed, Joanna C. S. Santos, and Mohammed Abuhamad

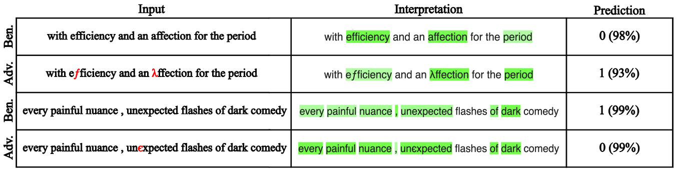
***Abstract*—Studies have shown that machine learning systems are vulnerable to adversarial examples in theory and practice. Where previous attacks have focused mainly on visual models that exploit the difference between human and machine perception, text-based models have also fallen victim to these attacks. However, these attacks often fail to maintain the semantic meaning of the text and similarity. This paper introduces AdvChar, a black-box attack on Interpretable Natural Language Processing Systems, designed to mislead the classifier while keeping the interpretation similar to benign inputs, thus exploiting trust in system transparency. AdvChar achieves this by making less noticeable modifications to text input, forcing the deep learning classifier to make incorrect predictions and preserve the original interpretation. We use an interpretation-focused scoring approach to determine the most critical tokens that, when changed, can cause the classifier to misclassify the input. We apply simple character-level modifications to measure the importance of tokens, minimizing the difference between the original and new text while generating adversarial interpretations similar to benign ones. We thoroughly evaluated AdvChar by testing it against seven NLP models and three interpretation models using benchmark datasets for the classification task. Our experiments show that AdvChar can significantly reduce the prediction accuracy of current deep learning models by altering just two characters on average in input samples.**

***Index Terms*—Adversarial Machine Learning, Interpretable Deep Learning, Black-box Attacks, NLP**

I. Introduction

Deep learning models, particularly in Natural Language Processing (NLP), have revolutionized how machines understand and interact with human language. These advancements have enabled various applications, from simple spellcheck and keyword search to complex tasks such as sentiment analysis [1], machine translation [2], and chatbot interactions [3]. The integration of NLP into our daily digital interactions, such as through search engines, virtual assistants, and recommendation systems, highlights its importance. However, these models are shown to be susceptible to adversarial attacks [4].

Adversarial attacks in NLP, which involve careful manipulations of input data leading to incorrect model outputs, are a growing concern. These attacks are especially stealthy because of the complex nature of human language, which is filled with idioms, metaphors, and context-dependent meanings [5]. It is essential to rigorously research and develop effective methods to counter these adversarial attacks, given the growing integration of such models in various NLP applications across different sectors. For example, adversarial attacks on AI systems for social media moderation, opinion analysis, customer reviews, or market trends could result in failures to identify harmful content, misinterpretation of benign posts, and misleading analyses, ultimately leading to poor decisions [6].



	Input	Interpretation	Prediction
Ben.	with efficiency and an affection for the period	with efficiency and an affection for the period	0 (98%)
Adv.	with eƒficiency and an λffection for the period	with eƒficiency and an λffection for the period	1 (93%)
Ben.	every painful nuance , unexpected flashes of dark comedy	every painful nuance , unexpected flashes of dark comedy	1 (99%)
Adv.	every painful nuance , unϵxpected flashes of dark comedy	every painful nuance , unϵxpected flashes of dark comedy	0 (99%)

Fig. 1: Example texts comparing a benign sample and a sample subject to our proposed attack, along with their corresponding interpretations based on LIME interpreter. Both inputs generate similar interpretations regardless of differences.

By coupling NLP classifiers with interpreters, *i.e.,* creating Interpretable NLP Systems (INLPS), adversarial manipulations can be recognized by an observer (see Figure 1). INLPS offer transparency in the decision-making process, which is especially crucial for applications where understanding the reasoning behind decisions is as important as the decisions themselves. Although this transparency is designed to promote trust and accountability, it can inadvertently introduce vulnerabilities, offering potential attackers a roadmap to manipulate the outcomes. This vulnerability is not just a theoretical concern but a practical challenge that needs to be addressed to ensure the reliability and safety of AI applications. In highlighting the vulnerabilities introduced by INLPS, our research goes a step further by demonstrating these theoretical concerns in a practical context. Our attack, AdvChar, targets INLPS, which integrate NLP classifiers with interpreters to deliver predictions and explanations. The attack has two objectives: (1) to mislead the classifier into misclassifying the input, and (2) to ensure that the interpreter produces an explanation similar to that of a benign input. We focus on text classification tasks using large language models (LLMs) and diverse interpretation methods. By changing certain words at the character level, AdvChar is designed to mislead both the primary NLP classifier and its interpreter. AdvChar demonstrates that the strengths of INLPS can also become weaknesses (as shown in Figure 1), emphasizing the need for robust countermeasures.

We evaluate our attack using three distinct datasets: the Stanford Sentiment Treebank (SST-2), AG News, and Yahoo Answers datasets. SST-2 (with binary sentiment classification) serves as a basic test for our attack, while AG News (with four-category news topic classification) and Yahoo Answers (with ten-category question classification) provide more complex scenarios to assess the effectiveness of the attack.

We selected a diverse range of LLM models, *i.e.,* GPT-2, BERT, DistilBERT, Electra, CANINE, FNet, and XLM-

Eldor Abdukhamidov and Tamer Abuhmed are with the Department of Computer Science and Engineering, Sungkyunkwan University, Suwon, South Korea.(E-mail: abdukhamidov@skku.edu, tamer@skku.edu). Joanna C. S. Santos is with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, USA. (joannacss@nd.edu) Mohammed Abuhamad is with the Department of Computer Science, Loyola University, Chicago, USA.(E-mail: mabuhamad@luc.edu).

R, to provide a comprehensive analysis across these models. Furthermore, our study incorporates three interpretation models, *i.e.,* SHAP [7], Saliency Maps [8], and LIME [9], each providing unique insights into the model decision-making process. Our findings show that the attack reaches a success rate of 79% in the AG News dataset using LIME interpreter with CANINE model. In the SST-2 and Yahoo Answers datasets, it achieves a success rate of 79% (with Saliency Map interpreter and BERT model) and 80% (with LIME interpreter and BERT model), respectively. The effectiveness of the attack varies by models and interpreters, with LIME generally outperforming others, especially on complex tasks.

Contributions. We summarize our contributions as follows:

- We propose a stealthy and query-efficient black-box attack targeting INLPS. We evaluate the attack performance against seven classifiers when coupled with three different interpreters using three datasets. The evaluations show that the proposed attack achieves considerable success rates, highlighting the attack's effectiveness in various scenarios.
- We investigate the transferability of adversarial examples across various classifiers and interpreters to assess whether adversarial samples designed for a certain model and interpreter can expose vulnerabilities in other models and interpreters, highlighting areas of concern in the field.
- We perform a comparative analysis to examine several factors (*i.e.,* input length, amount of perturbation, *etc.*) that might influence the performance of the attack.

Organization. Our paper is organized as follows: §II reviews related research studies; §III describes the notations and terms used in the paper; § III presents the proposed attack and its underlying mechanisms; § V provides the results of the attack in terms of effectiveness, robustness, and transferability against LLMs and interpretation models; § VI explains the existing limitations and future work; and § VII concludes the paper.

II. Related Work

The evolution of NLP techniques has advanced significantly with the emergence of deep learning-based LLMs. Despite their strength, deep learning models have demonstrated certain vulnerabilities, particularly when faced with adversarial perturbations. Several studies [10], [11], [12], [13], [14], [15] have highlighted how these models, despite their complexity, can be deceived by carefully crafted perturbations in the input.

NLP presents a unique challenge compared to other domains, such as computer vision, due to the nature of its data. Textual data is inherently discrete, contrasting with images' continuous nature. For example, a sentence like '*I do not dislike this movie*' in NLP can be challenging to interpret. While '*dislike*' generally suggests something negative, the addition of '*do not*' shifts the meaning to positive or neutral. This highlights how text meaning can change significantly based on word arrangement, emphasizing its discrete nature. In contrast, CNN-based image classifiers typically show robustness against minor modifications like changing one or a few pixels, unless these modifications are carefully designed as adversarial perturbations [16], [17], [18], [19]. This fundamental distinction highlights why adversarial attacks on textual data are inherently challenging. Even minor textual perturbations can substantially change the intended meaning and model predictions, making subtle adversarial modifications notably harder to achieve [20], [21], [22], [23].

Using word embeddings/representations in NLP adds complexity to the problem of adversarial attacks. Word embeddings convert words into continuous vector spaces, which can be modified to introduce perturbations. By slightly adjusting the vectors associated with words, attackers can mislead a model into interpreting the input differently, resulting in incorrect outputs. This aspect of adversarial attacks, especially in the context of word embeddings, has been extensively explored in research studies such as [24], [25], [26].

Various tasks in NLP range from basic predictive tasks, *e.g.,* sentiment analysis [27] and text classification [28], to more complex generative tasks, *e.g.,* machine translation [29], and question answering [30]. This diversity of tasks also means that adversarial attacks can occur in different scenarios and threat models. Each task has its unique vulnerabilities that attackers exploit with creativity. In our work, we focus mainly on text classification tasks.

III. Notations and Definitions

This section introduces the notation and definitions used throughout the paper to describe our attack and its associated concepts. The symbols and notation used in this paper are summarized in Table I. The samples are denoted as pairs (x,c). Specifically, $[x_1, x_2, x_3, ..., x_n] = x \subset X$, where x refers to an input text sequence of n tokens. These tokens can be words or characters, depending on the specific model used. Meanwhile, $c = 1, ..., M$ represents a label of M classes. The deep learning model is illustrated as $F : X \rightarrow C$, which maps the input set to the associated labels.

TABLE I: Notations used in the paper.

Symbol	**Description**
x	Original text sequence.
x'	Adversarial text sequence generated by the attack.
x_i	i^{th} token in the sequence x.
c	Output label of benign input text.
c'	Output label of adversarial input text.
g	Benign interpretation map.
g'	Adversarial interpretation map.
F	Target deep learning classifier.
G	Target interpreter.
T	General notation for token transformation function.
S	Similarity function.
m	Number of top important tokens to be perturbed.
$T_{\text{sub}}(x_i)$	Substitution transformation applied to token x_i.

Classifier. In this study, our primary focus is on predictive tasks, such as text classification, in which a deep learning model F classifies an input x into one of the predefined classes C, with the result being $F(x) = c$ that belongs to C.

Interpreter. There exist two distinct approaches to improving the interpretability of deep learning models: one involves building models with inherent comprehensibility, while the other involves the adoption of post hoc explanation techniques. Using post hoc techniques does not require any modifications to the model's architecture or configuration, and therefore, they are often more desirable. Our experiments primarily focus on post hoc explanations, which explain how a model works after it has made a decision, focusing specifically on individual examples. We examine how the model, which we denote as F, processes a given input x and how it makes a decision based on that input, referred to as $F(x)$. We interpret the connections between x and $F(x)$ as interpretation maps. We use an interpreter, G, to create a map $g = G(x, F)$ for each input x using the model F. In this map, g, each element $g[i]$ highlights the importance of the i-th component of the input x in generating the output $F(x)$.

Threat Model. This work assumes a black-box scenario: the adversary does not have direct knowledge of the inner workings of the classifier F and/or the interpreter G, including their structures and parameters. This assumption represents scenarios in which the internal details of the system are hidden from potential adversaries, which is practical since modern deep learning models are often used as a service that takes input from a user and produces a related output. An adversary/user interacts with the system by providing inputs to the classifier F and receiving the corresponding outputs. In a typical use case, the user submits an input x, the classifier processes this input, and the user receives the output $c = F(x)$. Additionally, the interpreter G can be queried (by the system/observer) to provide explanations $g = G(x, F)$ for the classifier's output.

Adversarial Capabilities and Constraints. The adversary has the following capabilities and constraints:

- Query the classifier F and interpreter G with any input x.
- Observe the classifier's output $c = F(x)$.
- Observe the interpreter's output $g = G(x, F)$.
- Does not know/access training data or methods for F or G.
- Can not access the parameters or architecture of F or G.
- Can not modify the internal components of F or G.

Adversarial Objectives. The adversary aims to design the inputs x' such that the classifier's output $c' = F(x')$ is incorrect, while x' appears similar to a benign input. The adversary seeks to maintain the original explanations provided by the interpreter G to hide the true nature of the classifier's decision-making process or to mislead users regarding the classifier's reliability.

Taking into account these scenarios, capabilities, constraints, and objectives, this work evaluates the robustness of the system against adversarial attacks.

IV. Methodology

This section provides an overview of the attack and its execution on three different types of interpreters.

A. Attack Formulation

In AdvChar, the interpretable model is considered as a central target of the attack. It is used to guide character-level perturbations and preserve overall interpretation, ensuring that adversarial input remains undetectable even under explanation-based inspection. Adversarial attacks generate an adversarial sample by producing a perturbation Δx within a specific threshold ϵ. AdvChar considers more constraints since the target systems employ interpretable models for the decision-making process. The goal of AdvChar is twofold: to trick the classifier and to ensure that the interpreter produces interpretation maps that resemble those of benign cases:

1) Ensuring model misclassification, *i.e.,* $F(x') \to c'$, $c' \neq c$.

$$F(x) \to c, \quad F(x') \to c', \quad c' \neq c \tag{1}$$

2) Triggering an interpreter G to generate target interpretation maps, *i.e.,* $G(x'; F) \to g' : g' \approx g$, where $G(x; F) = g$.

$$G(x, F) \to g, \quad G(x', F) \to g', \quad g' \approx g \tag{2}$$

3) The variation between x and x', denoted as $\Delta(x, x')$, should not be noticeable.

Unlike traditional attacks, such as TextFooler [31] or PWWS [11] that score word importance based on prediction confidence shifts, AdvChar leverages the output of interpretation models to identify influential characters. This enables the attack to simultaneously influence both the classifier's decision and its explanation, which is crucial for targeting INLPS.

To make the perturbation small enough to be unnoticeable to humans, we implement AdvChar in the character level to achieve an adversarial attack. In other words, AdvChar generates adversarial characters visually similar to the target character but unfamiliar to the target NLP model. The attack is universal, meaning it can be applied to any NLP model, and it generates adversarial samples that produce interpretations similar to the benign input, making it challenging for humans to detect the adversarial samples.

Our adversarial strategy involves the following steps:

1) Token Importance Evaluation. Using an interpreter output from the INLPS, such as visualized attention or color intensity maps, we estimate each token's importance without relying on model probabilities or gradients. These outputs are converted into numerical importance scores, providing a practical and realistic way to identify influential tokens. This enables AdvChar to function effectively even in black-box API scenarios. Therefore, we can derive an importance score for each element (token) $\mathcal{I}(t_i)$. These scores can be organized in ascending order to target the most influential elements.

Let us denote the set of elements (*i.e.,* tokens) from the input x_i as $x_i = \{t_1, t_2, ..., t_n\}$. Upon computing their respective importance scores, we can re-arrange them into a sorted set x_i^* such that: $x_i^* = \{t_1^*, t_2^*, ..., t_n^*\}$ where $\mathcal{I}(t_1^*) \geq \mathcal{I}(t_2^*) \geq ... \geq \mathcal{I}(t_n^*)$. In this representation, t_1^* is the element with the highest importance score and t_n' has the least. This ordered set facilitates a structured approach, allowing us to prioritize and perturb the most influential elements first.

Since interpreters have varied output ranges, importance scores are adjusted to typically lie between 0 and 1. This ensures that no particular element or score disproportionately influences the model due to its magnitude. Normalization is performed as follows: $\mathcal{I}(t_i) = [\mathcal{I}(t_i) - \min(\mathcal{I}(x_i))] \div [\max(\mathcal{I}(x_i)) - \min(\mathcal{I}(x_i))]$.

2) Adversarial Perturbation. Starting with the essential token, we introduce slight modifications. These perturbations are crafted to mislead the classifier while the token remains semantically readable and keeps its importance. Given:

Algorithm 1: AdvChar attack's main algorithm

```
Data: Classifier $F$, interpreter $G$, input text $x$, similarity function $S$, benign
    category $c$, interpretation map $m$, perturbation threshold $\epsilon$.
Result: Adversarial text $x'$
Initialization: Setting $x' \leftarrow x$;
begin
    $g \leftarrow G(x, F)$ ;          /* extract interpretation map */
    $scores \leftarrow Convert(g)$ ; /* convert map into importance
      scores */
    $x_{ordered} \leftarrow Sort(x_1, x_2, x_3, ..., x_n, scores_{x_i})$ ;       /* sort
      tokens based on importance score */
    for $x_i$ in $x_{ordered}$ do
        $x'_i \leftarrow SubstituteChar(x_i)$ ;  /* replace a character
          in a word */
        $x' \leftarrow$ replace $x_i$ with $x'_i$ in $x'$ ;
        if $S(x, x') > \epsilon$ then
            return None ;     /* Perturbation threshold is
              reached */
        else if $F(x') \neq c$ then
            return $x'$ ;            /* Solution is found */
        else
            continue
        end
    end
    return None ;                       /* Attack failed */
end
```

(1) $C(t_i)$ as the set of characters in token t_i

(2) $R(t_i, \zeta)$ as a function that replaces character ζ in input token t_i with a new character

(3) $L(F, x_i, c)$ as the classification loss for input x_i and true label c using classifier F

(4) $S(x_i, x'_i)$ as a similarity function that measures the difference in the order of importance of the tokens before and after perturbation for the benign sample x_i.

The token perturbed by our attack can be represented as:

$$t'_i = R(t_i, \zeta_r) \text{ s.t. } \zeta_r \in C(t_i)$$

We define the similarity function as:

$$S(x_i, x'_i) = \|\text{argsort}_{t_i \in x_i}[\mathcal{I}(t_i, F, G)] - \text{argsort}_{t'_i \in x'_i}[\mathcal{I}(t'_i, F, G)]\| \quad \text{s.t.} \quad S(x_i, x'_i) \leq \theta,$$

where $\text{argsort}_{t_i \in x_i}[\mathcal{I}(t_i, F, G)]$ represents the indices of tokens in x_i sorted by their importance scores $\mathcal{I}(t_i, F, G)$ in descending order. Here, the threshold θ limits permissible changes in token ordering after perturbation. AdvChar continues to perturb tokens following this order until the perturbation threshold θ is reached or the attack succeeds, *i.e.,* $F(x') \neq c$.

3) Iterative Attack. If the classifier is not deceived by perturbing the essential token, AdvChar keeps the modification and proceeds to the next token in the order of importance. This iterative process continues until the classifier is fooled, the similarity exceeds the perturbation threshold ϵ, or all tokens have been considered. In this process, all tokens in the input text are considered for perturbation to ensure comprehensive coverage and maximize the chances of deceiving the classifier.

The details of the algorithm are presented in Algorithm 1.

B. NLP Classification Models and Interpreters

Classification Models. Diverse NLP models are employed to test our attack. Our experiment includes GPT-2 [32], BERT [33], DistilBERT [34], Electra [35], CANINE [36], FNet [37], and XLM-R [**?**] models. The selected models offer a comprehensive overview of the advancements in NLP models over recent years. They collectively represent a range of architectures and methodologies, from lightweight and efficient versions to those optimized for multilingual tasks or generative capabilities. While some models are designed for classification tasks, others have been fine-tuned and transformed as classification models. The performance of the models on three datasets is provided in Table II.

TABLE II: Performance of NLP models on three different datasets. The performance is provided in terms of accuracy.

	GPT-2	BERT	DistilBERT	Electra	CANINE	FNet	XLM-R
SST-2	0.92	0.91	0.97	0.95	0.85	0.89	0.93
AG	0.94	0.95	0.95	0.94	0.91	0.91	0.92
Yahoo Answers	0.90	0.91	0.89	0.89	0.87	0.88	0.89

Interpretation Models. Our study uses three interpreters: SHAP [38], Saliency Maps (SM) [39], and LIME [9]. Each interpreter has a unique methodology and provides valuable insights into model behavior and feature importance. The use of thesde different interpreters enables a more comprehensive and deeper analysis of model performance and vulnerability to attacks, as each provides a unique perspective:

- **SHAP**: By assigning consistent and fairly distributed importance values to each feature, SHAP helps identify which features are most influential in making predictions. This can reveal how the attack manipulates these key features to degrade model performance.
- **Saliency Maps**: By visualizing the gradient of the output with respect to the input features, Saliency Maps highlight which parts of the input data are most affected by the attack. This direct visualization allows us to see how the attack modifies the input to influence the model's decisions, providing insight into the attack's impact on critical features.
- **LIME**: By creating interpretable surrogate models for specific predictions, LIME helps understand how the attack affects local decision boundaries. This localized view shows how the attack changes the model's behavior in specific regions of the input space, highlighting potential weaknesses and areas where the model is vulnerable.

C. Evaluation Metrics

We evaluated the attack success against DNN classifiers and interpretation models using multiple metrics:

- **Attack Success Rate** This metric measures the effectiveness of the adversarial attack. It is calculated as the percentage of instances where the adversarial input successfully deceived the model, leading it to produce an incorrect output. A higher attack success rate indicates a more effective attack.
- **Number of Queries (#)** This metric represents the number of queries made to the target model during the attack. It indicates of the efficiency of the attack. Fewer queries suggest that the attack can deceive the model with minimal interaction. This is especially important in scenarios where the number of queries are limited or costly.
- **Misclassification Confidence** When the adversarial input is provided, this metric measures the confidence level of the model's incorrect prediction. A higher misclassification confidence indicates that the model is not only fooled

by the adversarial input but is also highly sure about its wrong decision. This can provide insights into the model's vulnerability to the attack.

- **IoU (Intersection over Union)** Originally a metric used in computer vision tasks, IoU measures the overlap between two samples. In the context of NLP adversarial attacks, it can be used to quantify the similarity of interpretations between the original input and the adversarial input. A higher IoU indicates that the adversarial input retains much of the benign input's interpretation structure, suggesting a subtle and potentially more deceptive attack.
- **Avg. adversarial characters** This metric calculates the average number of characters in the input altered to craft the adversarial sample. It provides insights into the magnitude of changes required to deceive the model. A lower average suggests that minor perturbations are sufficient to mislead the model, indicating potential vulnerabilities in the model's understanding of the input.
- **Qualitative Metric (Manual Interpretation Comparison)** We manually compare the interpretations of samples in both their adversarial and benign forms. This involves assessing how the model's understanding or interpretation of the input changes due to the adversarial perturbations. Two human evaluators assess the similarity between benign and adversarial interpretation maps generated by interpreters. The criterion is *interpretation similarity*, defined as the degree to which top-m tokens retain similar importance rankings and weights, indicating a successful attack when no significant difference is observed. This metric is conducted on random samples to help us understand the effectiveness of the attack, how much of the original meaning is maintained.

V. Experiments and Evaluation

Dataset. For our experiment, we use three different datasets: Binary Stanford Sentiment Treebank (SST-2) [40], AG News [41] and Yahoo Answers [41]. The Stanford Sentiment Treebank offers a detailed look into the sentiment structure of language through its labeled parse trees. The dataset contains 11,855 movie review sentences. Notably, it covers over 215,000 unique phrases, each evaluated by three human annotators. It is commonly termed SST-2 or SST binary for binary sentiment classification. The AG dataset contains over a million news articles with four news topic categories. The Yahoo Answers dataset, comprising approximately 1.4 million user-generated questions labeled for 10-category topic classification, introduces a highly challenging environment with its diverse, informal content. We use its test set to validate the attack's effectiveness in complex, multi-class settings.

A. Attack Effectiveness against Classifiers

Table III presents the attack results across the seven NLP models on SST-2, AG News, and Yahoo Answers, highlighting its effectiveness with different interpreters. We also compare our results with the existing attacks TextBugger [42] and TextFooler [43]. For this comparison, we use the default settings provided in the attacks' original implementation. TextBugger serves as a character level black box baseline, while TextFooler is a word substitution black box baseline for text classification tasks.

For the SST-2 dataset, the SHAP interpreter has an attack success rate of approximately 64.71%. It required, on average, 20.68 queries with misclassification confidence of around 89.91% and an average number of 1.89 adversarial characters. The attack with SM interpreter performed better, with the highest attack success rate of 73.43%. However, a considerable number of queries, averaging 60.93, were required. Its misclassification confidence and average adversarial characters were approximately 89.50% and 2.14, respectively. While achieving an attack success rate of about 69.57%, the LIME interpreter was more efficient than the others regarding the number of queries, averaging only 17.76. It has the highest misclassification confidence at around 90.38% and the lowest average adversarial characters at 1.72.

For the AG News dataset, the SHAP performance showed an attack success rate of 53.29%, the lowest among the interpreters. It required an average of 40.75 queries and had a misclassification confidence of around 56.91%. It also generated more adversarial characters for the dataset, averaging 3.65. With an attack success rate of roughly 69.00%, the SM interpreter had a significantly high number of queries, averaging 258.84, but had misclassification confidence of around 58.51% and average adversarial characters of 2.13. The LIME interpreter was the most effective for this dataset, resulting in the highest attack success rate of 74.57%. It was also efficient, requiring the fewest queries, with an average of 34.45. Its misclassification confidence was around 81.78%, with the lowest average adversarial characters at 1.80.

Regarding the SST-2 dataset classifiers, BERT showed superior performance, with a 75.00% success rate and an average of 32.91 queries per attack. The misclassification confidence was at 88.22%, with an average of 1.87 adversarial characters. With an attack success rate of 71.33%, CANINE demonstrated efficiency by averaging 31.55 queries. Interestingly, the attack required the least number of adversarial character perturbations, averaging 1.73 against this model with high misclassification confidence of 93.27%. Contrarily, DistilBERT was found to be more robust regarding misclassification, achieving a success rate of 69.33% with a misclassification confidence of 79.75%. Electra has a success rate of 68.00% and higher query usage of an average of 34.63. FNet and GPT-2 classifiers had 66.00% and 68.33% attack success rates, respectively. Meanwhile, XLM-R had an average attack success rate of 66.67% but maintained high misclassification confidence, reaching 96.30%.

For the classifiers applied to the AG News dataset, their behavior demonstrated significant changes. While maintaining an attack success rate of 67.33%, BERT required an increased 110.71 queries on average, showing a dataset-specific behavior. Its misclassification confidence was 56.14%. CANINE, reporting an attack success rate of 68.67%, performed with misclassification confidence of 72.38%, while DistilBERT provided the lowest misclassification confidence of 53.94% despite its success rate of 66.33%. Electra's attack success rate was 65.67%, but its query requirement was high at an average of 116.63. The FNet model presented a success

TABLE III: Attack results against different models coupled with three interpreters using three datasets. Results of Textbugger and TextFooler are repeated for different interpreters, as the attacks do not consider attacking interpreter. ASR, MC, PA, and QC denote attack success rate, misclassification confidence, character perturbation amount, and query count, respectively.

	Models	**AdvChar**				**Textbugger**				**TextFooler**			
		ASR	**CQ**	**MC**	**PA**	**ASR**	**QC**	**MC**	**PA**	**ASR**	**QC**	**MC**	**PA**
SST-2													
SHAP	GPT-2	0.69	20.99	93.18	1.91	0.60	61.50	91.21	14.50	0.70	195.24	89.17	31.42
	BERT	0.75	20.66	87.99	1.85	0.79	32.99	93.84	8.33	0.89	166.38	92.05	18.23
	DistilBERT	0.62	21.39	80.45	1.87	0.61	33.08	90.12	7.32	0.70	166.11	88.29	16.37
	Electra	0.64	20.51	88.53	1.89	0.74	36.70	92.27	8.66	0.83	170.47	90.12	19.08
	CANINE	0.62	19.40	93.00	1.78	0.65	34.54	89.74	8.46	0.74	168.33	88.54	18.19
	FNet	0.60	20.35	90.28	1.87	0.71	33.79	91.63	7.32	0.80	167.26	90.31	16.48
	XLM-R	0.61	21.47	95.97	2.03	0.72	34.29	94.36	8.15	0.81	167.03	93.42	18.07
SM	GPT-2	0.72	62.38	91.05	2.32	0.60	61.50	91.21	14.50	0.70	195.24	89.17	31.42
	BERT	0.79	60.31	88.62	2.04	0.79	32.99	93.84	8.33	0.89	166.38	92.05	18.23
	DistilBERT	0.74	59.74	78.77	2.04	0.61	33.08	90.12	7.32	0.70	166.11	88.29	16.37
	Electra	0.71	65.21	87.41	2.54	0.74	36.70	92.27	8.66	0.83	170.47	90.12	19.08
	CANINE	0.78	58.06	93.98	1.79	0.65	34.54	89.74	8.46	0.74	168.33	88.54	18.19
	FNet	0.70	58.72	90.83	1.90	0.71	33.79	91.63	7.32	0.80	167.26	90.31	16.48
	XLM-R	0.70	62.06	95.83	2.37	0.72	34.29	94.36	8.15	0.81	167.03	93.42	18.07
LIME	GPT-2	0.64	18.12	93.48	1.81	0.60	61.50	91.21	14.50	0.69	195.24	89.17	31.42
	BERT	0.71	17.75	88.04	1.72	0.79	32.99	93.84	8.33	0.89	166.38	92.05	18.23
	DistilBERT	0.72	17.78	80.04	1.68	0.61	33.08	90.12	7.32	0.70	166.11	88.29	16.37
	Electra	0.69	18.16	90.32	1.77	0.74	36.70	92.27	8.66	0.83	170.47	90.12	19.08
	CANINE	0.74	17.18	92.84	1.63	0.65	34.54	89.74	8.46	0.74	168.33	88.54	18.19
	FNet	0.68	17.94	90.84	1.73	0.71	33.79	91.63	7.32	0.80	167.26	90.31	16.48
	XLM-R	0.69	17.41	97.11	1.73	0.72	34.29	94.36	8.15	0.81	167.03	93.42	18.07
AG													
SHAP	GPT-2	0.49	45.15	54.85	3.73	0.40	134.00	67.12	105.25	0.55	465.19	76.27	97.44
	BERT	0.51	41.49	51.54	3.61	0.73	102.92	70.95	50.17	0.88	357.62	73.41	46.18
	DistilBERT	0.52	39.78	56.72	3.65	0.60	95.11	68.73	48.03	0.74	330.08	78.36	44.21
	Electra	0.56	37.63	59.58	3.43	0.52	93.72	71.88	54.04	0.67	325.77	81.09	50.34
	CANINE	0.53	39.47	57.84	3.61	0.58	94.30	69.82	46.70	0.72	327.55	79.22	43.18
	FNet	0.51	41.64	61.48	3.92	0.59	92.85	72.66	47.12	0.73	322.49	83.07	43.09
	XLM-R	0.61	40.07	56.37	3.57	0.66	91.34	74.01	45.88	0.81	317.31	78.44	42.16
SM	GPT-2	0.65	265.02	59.53	2.31	0.40	134.00	67.12	105.25	0.55	465.19	76.27	97.44
	BERT	0.75	256.23	57.94	2.03	0.73	102.92	70.95	50.17	0.88	357.62	73.41	46.18
	DistilBERT	0.70	253.81	51.50	2.03	0.60	95.11	68.73	48.03	0.74	330.08	78.36	44.21
	Electra	0.67	277.05	57.15	2.53	0.52	93.72	71.88	54.04	0.67	325.77	81.09	50.34
	CANINE	0.74	246.67	61.44	1.78	0.58	94.30	69.82	46.70	0.72	327.55	79.22	43.18
	FNet	0.66	249.47	59.38	1.89	0.59	92.85	72.66	47.12	0.73	322.49	83.07	43.09
	XLM-R	0.66	263.66	62.65	2.36	0.66	91.34	74.01	45.88	0.81	317.31	78.44	42.16
LIME	GPT-2	0.69	35.14	72.59	1.84	0.40	134.00	67.12	105.25	0.55	465.19	76.27	97.44
	BERT	0.76	34.42	58.95	1.72	0.73	102.92	70.95	50.17	0.88	357.62	73.41	46.18
	DistilBERT	0.77	34.48	53.59	1.78	0.60	95.11	68.73	48.03	0.74	330.08	78.36	44.21
	Electra	0.74	35.22	95.20	1.88	0.52	93.72	71.88	54.04	0.67	325.77	81.09	50.34
	CANINE	0.79	33.31	97.86	1.73	0.58	94.30	69.82	46.70	0.72	327.55	79.22	43.18
	FNet	0.73	34.79	95.75	1.84	0.59	92.85	72.66	47.12	0.73	322.49	83.07	43.09
	XLM-R	0.74	33.76	98.53	1.84	0.66	91.34	74.01	45.88	0.81	317.31	78.44	42.16
Yahoo Answers													
SHAP	GPT-2	0.45	50.25	50.12	4.10	0.32	200.50	55.23	120.33	0.47	534.28	78.51	199.63
	BERT	0.48	46.52	48.75	3.85	0.59	150.74	61.41	81.27	0.80	410.37	75.22	95.48
	DistilBERT	0.49	44.30	52.40	3.92	0.46	161.39	57.48	85.40	0.67	379.46	80.34	91.27
	Electra	0.52	42.75	55.61	3.63	0.50	155.22	61.37	90.14	0.59	373.58	83.45	102.61
	CANINE	0.50	44.94	53.20	3.78	0.49	156.44	58.10	88.72	0.65	376.12	81.09	88.37
	FNet	0.48	46.12	57.33	4.17	0.40	165.26	58.75	93.20	0.66	370.24	85.38	89.15
	XLM-R	0.55	45.49	52.83	3.82	0.53	164.54	60.67	84.32	0.73	364.73	80.16	87.29
SM	GPT-2	0.60	300.52	54.23	2.55	0.32	200.50	55.23	120.33	0.47	534.28	78.51	199.63
	BERT	0.71	290.72	53.19	2.22	0.59	150.74	61.41	81.27	0.80	410.37	75.22	95.48
	DistilBERT	0.65	285.20	47.93	3.04	0.46	161.39	57.48	85.40	0.67	379.46	80.34	91.27
	Electra	0.61	311.28	50.41	2.97	0.50	155.22	61.37	90.14	0.59	373.58	83.45	102.61
	CANINE	0.68	280.76	56.84	2.18	0.49	156.44	58.10	88.72	0.65	376.12	81.09	88.37
	FNet	0.60	288.69	54.35	2.58	0.40	165.26	58.75	93.20	0.66	370.24	85.38	89.15
	XLM-R	0.62	297.24	57.47	2.85	0.53	164.54	60.67	84.32	0.73	364.73	80.16	87.29
LIME	GPT-2	0.76	40.18	66.38	2.04	0.32	200.50	55.23	120.33	0.47	534.28	78.51	199.63
	BERT	0.80	40.40	54.59	2.13	0.59	150.74	61.41	81.27	0.80	410.37	75.22	95.48
	DistilBERT	0.80	38.40	57.83	1.97	0.46	161.39	57.48	85.40	0.67	379.46	80.34	91.27
	Electra	0.79	43.23	91.47	2.31	0.50	155.22	61.37	90.14	0.59	373.58	83.45	102.61
	CANINE	0.81	36.77	94.90	1.98	0.49	156.44	58.10	88.72	0.65	376.12	81.09	88.37
	FNet	0.75	40.67	91.14	2.08	0.40	165.26	58.75	93.20	0.66	370.24	85.38	89.15
	XLM-R	0.77	39.96	94.23	1.98	0.53	164.54	60.67	84.32	0.73	364.73	80.16	87.29

rate of 63.33%, while GPT-2 displayed a 61.00% success rate with a misclassification confidence of 62.32%. XLM-R achieved a 67.00% success rate while maintaining a 72.52% misclassification confidence.

Extending the analysis to the Yahoo Answers dataset, we observe that the attack's performance varies across interpreters and models. The SHAP interpreter exhibits lower attack success rates, ranging from 45% to 55%, with higher query requirements (approximately 42 to 50) and more adversarial characters (around 3.63 to 4.17) compared to its performance on SST-2. This could be attributed to the increased complexity and length of the dataset inputs, which may pose greater challenges for generating effective adversarial examples. The SM interpreter achieves success rates between 60% and 71%, but at the cost of a substantial increase in the number of queries, averaging around 280 to 311, indicating a trade-off between success and efficiency. The LIME interpreter achieves the highest success rates, from 75% to 81%, while maintaining relatively low query counts (approximately 36 to 43) and minimal adversarial perturbations (around 1.97 to 2.31), suggesting it adapts particularly well to this dataset.

Our method consistently outperforms TextBugger across the datasets of SST-2, AG News, and Yahoo Answers. On SST-2, it achieves comparable or higher success rates (*e.g.,* 0.79 vs. 0.79 for BERT with SM), while requiring significantly fewer queries (*e.g.,* 17.18 vs. 34.54 for CANINE with LIME) and fewer adversarial character modifications (*e.g.,* 1.63 vs. 8.46), along with higher misclassification confidence (*e.g.,*

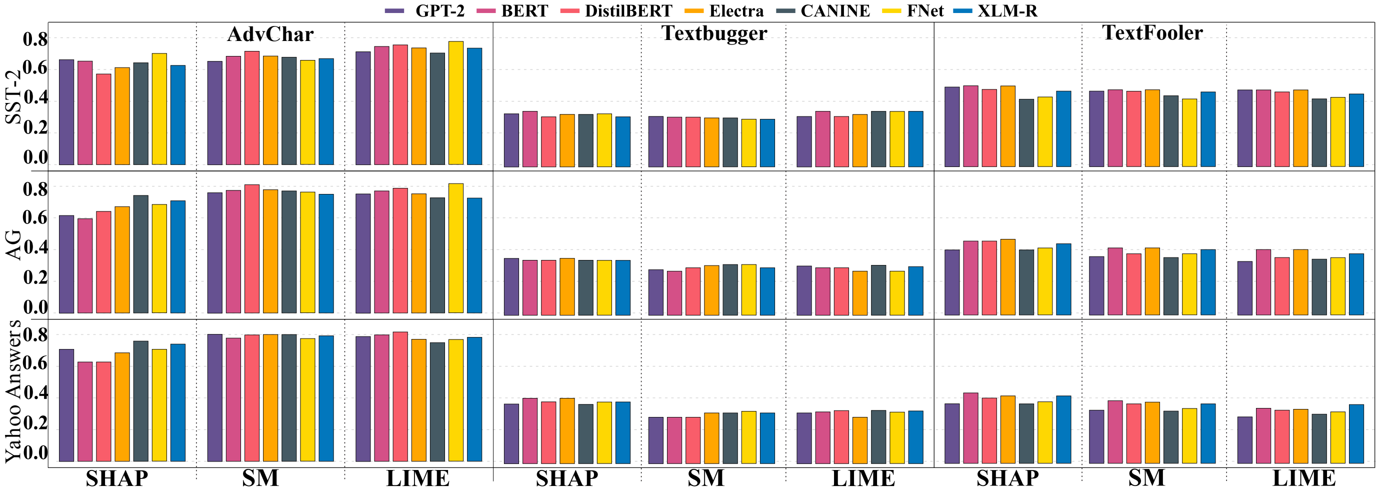


Fig. 2: IoU scores of AdvChar, TextBugger, and TextFooler against seven classifiers with three interpreters across datasets.

97.11% vs. 94.36% for XLM-R with LIME). For AG News, AdvChar demonstrates a higher success rate (*e.g.,* 0.79 vs. 0.58 for CANINE with LIME) with fewer queries (*e.g.,* 33.31 vs. 94.30) and minimal perturbations (*e.g.,* 1.73 vs. 46.70). However, SM occasionally requires more queries (*e.g.,* 256.23 vs. 102.92 for BERT). On Yahoo Answers, our method achieves superior success (*e.g.,* 0.81 vs. 0.49 for CANINE with LIME), lower query counts (*e.g.,* 36.77 vs. 156.44), and reduced perturbations (*e.g.,* 1.98 vs. 88.72), while also producing higher confidence scores (*e.g.,* 94.90% vs. 58.10%). Notably, the LIME interpreter contributes significantly to this efficiency across datasets, while SHAP and SM offer robust success rates despite occasional increases in query count, highlighting AdvChar's adaptability and efficiency over TextBugger. We also include TextFooler as a strong word-level baseline. While it can reach high ASR, it typically requires significantly more queries and larger edit budgets than AdvChar. In contrast, AdvChar achieves strong ASR with tens of queries and few character edits (≈ 2 on average), while delivering higher confidence at lower cost. Compared to TextFooler misclassification confidence drop under deletion and greedily substitutes top-ranked tokens with POS-consistent synonyms from counter-fitted embeddings under a sentence-similarity constraint, AdvChar uses interpreter-guided, *character*-level edits on the most influential tokens. This eliminates heavy candidate-generation loops, lowers query and edit budgets, and preserves explanation similarity by providing comparable or better ASR with higher efficiency.

The superior performance of AdvChar over TextBugger and TextFooler can be attributed to three key factors. First, our method leverages interpreter-driven importance scores from SHAP, SM, and LIME to target semantically critical tokens, ensuring minimal character changes significantly alter model predictions. Second, modern NLP models, pretrained on noisy web-scale data, can be robust to obvious perturbations generated by TextBugger and TextFooler, reducing their effectiveness. Third, by avoiding random perturbations and preserving interpreter consistency, our method maintains stealthiness and efficiency, outperforming other methods in various metrics across diverse models.

B. Attack Effectiveness against Interpreters

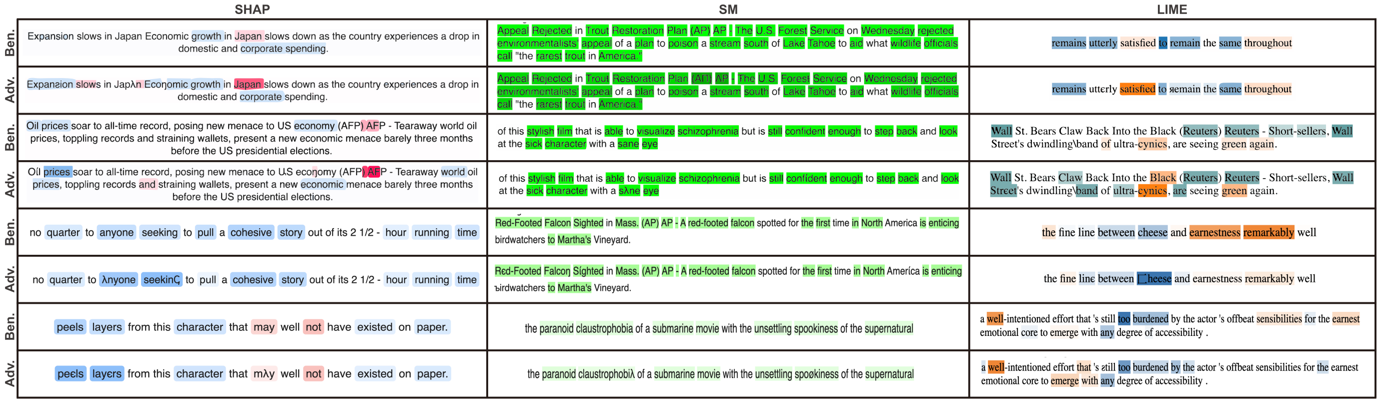
Our analysis employed the Intersection over Union (IoU) scores to measure the similarity between adversarial and benign interpretations across various NLP models. Seven models were assessed on the SST-2 and AG News datasets using three interpreters: SHAP, SM, and LIME. Figure 2 displays the results of IoU scores in our experiment.

On the SST-2 dataset with the SHAP interpreter, the FNet model registered the peak IoU score at 0.70, with GPT-2 and BERT closely following at 0.66 and 0.65, respectively. DistilBERT indicated adversarial interpretation with the lowest score of 0.57. Under the SM interpreter, DistilBERT achieved the highest IoU score of 0.71, while models like GPT-2, BERT, and Electra produced scores ranging from 0.65 to 0.68. When using the LIME interpreter, FNet scored the highest, 0.77, while GPT-2 and BERT scored above 0.70.

For the AG News dataset, CANINE with the SHAP interpreter performed with an IoU score of 0.72, contrasting BERT at the lower range with 0.59. The top performers among SM interpreters were DistilBERT, with a score of 0.81, followed by BERT and Electra, with scores of 0.78 each. FNet and DistilBERT scored 0.81 and 0.79 with the LIME, respectively.

For the Yahoo Answers dataset, AdvChar continues to demonstrate high interpretation similarity across all models and interpreters. Under SHAP, IoU scores range from 0.61 to 0.69, with FNet and XLM-R achieving the highest similarity. Using SM, most models produce IoU scores between 0.75 and 0.81. The LIME interpreter again shows the highest IoU values, with several models including BERT, GPT-2, and Electra reaching or exceeding 0.80.

In comparison with TextBugger [42] using the same experimental setup, Figure 2 shows that TextBugger produces lower IoU scores across all datasets. For example, on SST-2, IoU scores of TextBugger range between 0.28 to 0.36, indicating limited similarity between adversarial and benign interpretations. This contrast is especially noticeable under the LIME interpreter, where our method consistently exceeds 0.75, while TextBugger remains below 0.36 for all models. Similar results are observed on AG News and Yahoo Answers datasets.



	SHAP	SM	LIME
Ben.	Expansion slows in Japan Economic growth in Japan slows down as the country experiences a drop in domestic and corporate spending.	Appeal Rejected in Trout Restoration Plan (AP) AP - The U.S. Forest Service on Wednesday rejected environmentalists' appeal of a plan to poison a stream south of Lake Tahoe to aid what wildlife officials call "the rarest trout in America."	remains utterly satisfied to remain the same throughout
Adv.	Expanɜion slows in Japλn Ecoŋomic growth in Japan slows down as the country experiences a drop in domestic and corporate spending.	Appeal Rejected in Trout Restoration Plan (AΠ) ΛP - The U.S. Forest Service on Wednesday rejected environmentalists' appeal of a plan to poison a stream south of Lake Tahoe to aid what wildlife officials call "the rarest trout in America."	remains utterly satisfied to яemain the same throughout
Ben.	Oil prices soar to all-time record, posing new menace to US economy (AFP) AFP - Tearaway world oil prices, toppling records and straining wallets, present a new economic menace barely three months before the US presidential elections.	of this stylish film that is able to visualize schizophrenia but is still confident enough to step back and look at the sick character with a sane eye	Wall St. Bears Claw Back Into the Black (Reuters) Reuters - Short-sellers, Wall Street's dwindling\band of ultra-cynics, are seeing green again.
Adv.	Oíl prices soar to all-time record, posing new menace to U.S ecoŋomy (AFP) AFP - Tearaway world oil prices, toppling records and straining wallets, present a new economic menace barely three months before the US presidential elections.	of this stylish film that is able to visualize schizophrenia but is still confídent enough to step back and look at the sick character with a sλne eye	Wall St. Bears Claw Back Into the Black (Reuters) Reuters - Short-sellers, Wall Street's dwindling\band of ultra-cynics, are seeing green again.
Ben.	no quarter to anyone seeking to pull a cohesive story out of its 2 1/2 - hour running time	Red-Footed Falcon Sighted in Mass. (AP) AP - A red-footed falcon spotted for the first time in North America is enticing birdwatchers to Martha's Vineyard.	the fine line between cheese and earnestness remarkably well
Adv.	no quarter to λnyone seekinϚ to pull a cohesive story out of its 2 1/2 - hour running time	Rɛd-Footed Falcoŋ Síghted in Mass. (AP) AP - A red-footed falcon spotted for the first time in North Ameríca is enticing ъirdwatchers to Martha's Vineyard.	the fine linɛ between ⊏heese and earnestness remarkably well
Ben.	peels layers from this character that may well not have existed on paper.	the paranoid claustrophobia of a submarine movie with the unsettling spookiness of the supernatural	a well-intentioned effort that 's still too burdened by the actor 's offbeat sensibilities for the earnest emotional core to emerge with any degree of accessibility .
Adv.	peɛls layers from this character that mλy well not have existed on paper.	the paranoid claustrophobiλ of a submarine movie with the unsettling spøøkiness of the supernatural	a well-intentioned effort that 's ƨtill too burdened by the actor 's offbeat sensibilities for the earnest emotional core to emerge with any degree of accessibility .

Fig. 3: Results of interpreters on benign and adversarial texts generated by AdvChar against BERT, GPT and Electra models. Ben. and Adv. stand for benign and adversarial texts, respectively. Samples are selected randomly from SST-2 and AG datasets.

For the IoU of TextFooler [31] across SST-2, AG News, and Yahoo Answers, there is moderate overlap between benign and adversarial interpretations. The IoU scores of TextFooler are higher than those of TextBugger, as word-level synonym substitutions preserve much of the original saliency, but lower than our method (see Figure 2). The trend mirrors our overall findings: IoU is highest on the shorter SST-2 inputs and decreases on the longer, multi-class AG News and Yahoo datasets. Across interpreters, LIME generally produces the largest overlaps (*i.e.,* least robust), followed by SM, with SHAP showing lower overlaps (*i.e.,* more robust). Model-wise, BERT-family encoders typically keep slightly more overlap than GPT-2, reflecting greater tokenization and representation stability under synonym substitutions.

We also evaluate the capability of our attack to create interpretation maps comparable to benign interpretation maps. We conduct a qualitative evaluation by manual checking to observe the similarity of interpretations produced by adversarial and the corresponding benign samples. By qualitatively comparing the interpretations of benign and adversarial samples, our attack-generated interpretations are visually highly similar to their corresponding benign sample inputs. Figure 3 shows examples of observed attribution maps obtained using SHAP, SM, and LIME. The examples show adversarial interpretations and corresponding benign ones. The adversarial and benign attribution maps are highly similar as displayed in Figure 3.

C. Attack Transferability

Adversarial inputs have a desirable property of transferability. This means that an adversarial input effective against one deep neural network (DNN) can also be effective against other DNNs. Our study aims to explore the transferability of our attacks across interpreters and classifiers.

Interpreters. We generate a set of adversarial inputs against the source interpreter, denoted by G, and use them to calculate their attribution maps using target interpreters, denoted by G'. In exploring the transferability of adversarial inputs using the SST-2 and AG News datasets, distinct patterns emerged. The IoU scores quantitatively measured how similar adversarial examples were recognized between different interpreters.

A detailed overview of the IoU scores across different interpreters is provided in Table IV. As shown in the table, BERT demonstrated higher similarity in its attribution maps with an average IoU score of 0.3792. This suggests that adversarial examples generated for BERT tend to keep similarity by other interpreters. However, CANINE showed more varied interpretability with an average IoU score of 0.2192, indicating reduced transferability of its adversarial examples across interpreters. GPT-2 also showed low transferability in specific scenarios, *e.g.,* adversarial examples from LIME measured by SHAP, resulting in a low IoU score of 0.12.

TABLE IV: IoU scores of the transferability of adversarial inputs from source interpreters to target interpreters on different SST-2, AG News, and Yahoo Answers datasets classifiers.

Source	Target	Classifiers						
		GPT-2	BERT	DistilBERT	Electra	CANINE	FNet	XLM-R
SST-2								
SHAP	LIME	0.23	0.30	0.15	0.13	0.15	0.16	0.15
	SM	0.21	0.27	0.13	0.12	0.14	0.13	0.13
SM	SHAP	0.19	0.34	0.23	0.26	0.22	0.20	0.27
	LIME	0.23	0.62	0.31	0.24	0.23	0.21	0.28
LIME	SHAP	0.12	0.12	0.14	0.11	0.13	0.14	0.12
	SM	0.17	0.15	0.18	0.14	0.16	0.16	0.16
AG								
SHAP	LIME	0.31	0.57	0.24	0.23	0.27	0.30	0.27
	SM	0.22	0.51	0.21	0.20	0.20	0.21	0.21
SM	SHAP	0.34	0.52	0.39	0.34	0.35	0.34	0.46
	LIME	0.35	0.65	0.56	0.46	0.30	0.38	0.53
LIME	SHAP	0.24	0.21	0.27	0.26	0.27	0.25	0.26
	SM	0.32	0.29	0.32	0.22	0.21	0.22	0.29
Yahoo Answers								
SHAP	LIME	0.35	0.62	0.28	0.27	0.31	0.34	0.31
	SM	0.25	0.58	0.26	0.25	0.24	0.24	0.24
SM	SHAP	0.38	0.56	0.44	0.39	0.39	0.39	0.50
	LIME	0.40	0.71	0.61	0.51	0.34	0.42	0.59
LIME	SHAP	0.27	0.24	0.31	0.29	0.30	0.26	0.29
	SM	0.36	0.33	0.36	0.28	0.25	0.25	0.32

When we explore the usage of specific interpreter combinations, the SM-LIME pairing frequently indicated high transferability, particularly in the SST-2 dataset, where it achieved IoU scores as high as 0.62 for BERT. However, adversarial examples from LIME, when measured by SHAP, often showed lower transferability, with IoU scores dropping to 0.12 for BERT and GPT-2 in the SST-2 dataset.

For the AG News dataset, both SM-LIME and SM-SHAP combinations showcased strong transferability, with IoU scores reaching 0.65 and 0.52 for BERT, respectively. The difference between the datasets can be the result of their unique characteristics. SST-2 focuses on sentiment analysis, while AG News involves news article categorization. Such differences can influence the generation and interpretability of adversarial examples.

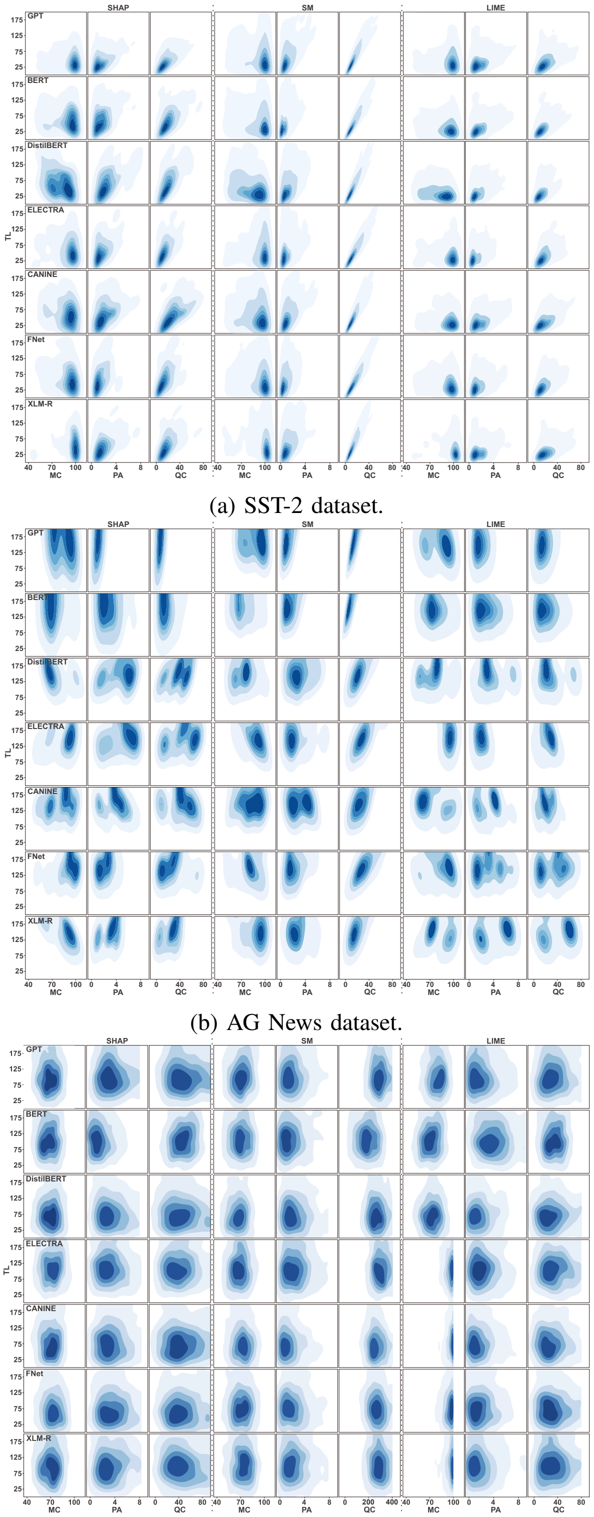
TABLE V: Transferability of adversarial inputs across NLP classifiers, measured by Attack Success Rate (ASR) and Misclassification Confidence (MC) on SST-2, AG News, and Yahoo Answers datasets. The first and second columns represent the source classifiers with their coupled interpreters, while the first row shows the target classifiers.

		GPT-2		BERT		DistilBERT		Electra		CANINE		FNet		XLM-R	
		ASR	MC	ASR	MC	ASR	MC	ASR	MC	ASR	MC	ASR	MC	ASR	MC
SST-2															
SHAP	GPT-2	-	-	0.44	0.40	0.45	0.37	0.37	0.33	0.41	0.38	0.51	0.44	0.31	0.30
	BERT	0.26	0.25	-	-	0.56	0.45	0.42	0.39	0.35	0.33	0.48	0.41	0.21	0.20
	DistilBERT	0.27	0.26	0.58	0.53	-	-	0.41	0.38	0.39	0.38	0.52	0.46	0.20	0.19
	Electra	0.43	0.41	0.72	0.66	0.65	0.54	-	-	0.43	0.41	0.50	0.44	0.30	0.30
	CANINE	0.10	0.19	0.26	0.24	0.22	0.19	0.21	0.19	-	-	0.35	0.32	0.10	0.10
	FNet	0.31	0.30	0.36	0.32	0.36	0.29	0.31	0.29	0.27	0.25	-	-	0.28	0.27
	XLM-R	0.39	0.36	0.47	0.42	0.44	0.37	0.40	0.35	0.34	0.31	0.51	0.44	-	-
SM	GPT-2	-	-	0.36	0.33	0.35	0.28	0.29	0.26	0.32	0.30	0.44	0.38	0.26	0.26
	BERT	0.27	0.25	-	-	0.57	0.46	0.43	0.39	0.39	0.36	0.45	0.39	0.24	0.23
	DistilBERT	0.28	0.27	0.63	0.58	-	-	0.43	0.39	0.46	0.43	0.45	0.39	0.23	0.22
	Electra	0.30	0.29	0.63	0.58	0.54	0.45	-	-	0.45	0.42	0.43	0.37	0.24	0.23
	CANINE	0.15	0.14	0.26	0.23	0.25	0.20	0.21	0.18	-	-	0.35	0.29	0.14	0.16
	FNet	0.28	0.27	0.34	0.30	0.31	0.25	0.26	0.24	0.28	0.26	-	-	0.22	0.21
	XLM-R	0.32	0.30	0.45	0.40	0.41	0.34	0.34	0.30	0.39	0.37	0.45	0.40	-	-
LIME	GPT-2	-	-	0.34	0.31	0.36	0.30	0.41	0.36	0.31	0.29	0.46	0.40	0.29	0.27
	BERT	0.32	0.30	-	-	0.60	0.50	0.55	0.51	0.43	0.40	0.47	0.40	0.26	0.25
	DistilBERT	0.32	0.30	0.67	0.61	-	-	0.54	0.50	0.43	0.40	0.46	0.39	0.23	0.22
	Electra	0.33	0.30	0.60	0.53	0.52	0.43	-	-	0.38	0.35	0.44	0.38	0.25	0.26
	CANINE	0.20	0.19	0.32	0.28	0.29	0.24	0.32	0.28	-	-	0.35	0.29	0.18	0.17
	FNet	0.35	0.34	0.39	0.35	0.34	0.29	0.34	0.32	0.31	0.29	-	-	0.29	0.28
	XLM-R	0.37	0.34	0.42	0.38	0.38	0.34	0.44	0.40	0.38	0.35	0.48	0.40	-	-
AG															
SHAP	GPT-2	-	-	0.45	0.30	0.43	0.26	0.34	0.25	0.35	0.27	0.41	0.31	0.28	0.23
	BERT	0.12	0.10	-	-	0.58	0.32	0.41	0.27	0.31	0.24	0.43	0.30	0.19	0.14
	DistilBERT	0.13	0.10	0.42	0.38	-	-	0.38	0.27	0.30	0.27	0.41	0.35	0.17	0.13
	Electra	0.19	0.12	0.37	0.46	0.66	0.40	-	-	0.37	0.30	0.45	0.32	0.25	0.23
	CANINE	0.05	0.17	0.25	0.18	0.20	0.13	0.20	0.14	-	-	0.31	0.23	0.09	0.07
	FNet	0.15	0.08	0.37	0.23	0.32	0.21	0.28	0.20	0.22	0.18	-	-	0.25	0.19
	XLM-R	0.19	0.13	0.43	0.30	0.40	0.26	0.39	0.25	0.28	0.22	0.45	0.33	-	-
SM	GPT-2	-	-	0.38	0.19	0.08	0.06	0.17	0.06	0.13	0.06	0.09	0.08	0.05	0.10
	BERT	0.06	0.13	-	-	0.12	0.09	0.10	0.08	0.12	0.08	0.10	0.09	0.05	0.07
	DistilBERT	0.06	0.11	0.14	0.12	-	-	0.09	0.08	0.09	0.11	0.09	0.10	0.05	0.14
	Electra	0.06	0.10	0.13	0.12	0.11	0.09	-	-	0.09	0.15	0.09	0.11	0.05	0.17
	CANINE	0.06	0.11	0.05	0.12	0.06	0.19	0.04	0.17	-	-	0.07	0.16	0.07	0.10
	FNet	0.06	0.15	0.07	0.16	0.07	0.15	0.06	0.15	0.06	0.19	-	-	0.05	0.11
	XLM-R	0.07	0.11	0.10	0.08	0.09	0.07	0.17	0.12	0.08	0.19	0.09	0.15	-	-
LIME	GPT-2	-	-	0.41	0.29	0.12	0.10	0.14	0.12	0.11	0.10	0.17	0.14	0.10	0.09
	BERT	0.12	0.10	-	-	0.22	0.18	0.20	0.17	0.15	0.15	0.18	0.14	0.10	0.11
	DistilBERT	0.12	0.16	0.23	0.25	-	-	0.20	0.17	0.14	0.15	0.15	0.13	0.08	0.10
	Electra	0.11	0.18	0.23	0.19	0.19	0.15	-	-	0.14	0.12	0.15	0.13	0.09	0.12
	CANINE	0.08	0.16	0.11	0.10	0.12	0.08	0.12	0.10	-	-	0.13	0.11	0.07	0.16
	FNet	0.13	0.12	0.13	0.15	0.11	0.14	0.12	0.17	0.11	0.18	-	-	0.10	0.12
	XLM-R	0.13	0.12	0.16	0.13	0.14	0.12	0.17	0.14	0.14	0.13	0.17	0.14	-	-
Yahoo Answers															
SHAP	GPT-2	-	-	0.42	0.32	0.43	0.30	0.35	0.28	0.38	0.30	0.46	0.35	0.29	0.25
	BERT	0.25	0.22	-	-	0.55	0.40	0.40	0.35	0.33	0.30	0.45	0.38	0.20	0.18
	DistilBERT	0.26	0.23	0.56	0.41	-	-	0.39	0.34	0.34	0.31	0.46	0.39	0.19	0.17
	Electra	0.32	0.29	0.50	0.45	0.48	0.40	-	-	0.36	0.32	0.42	0.36	0.28	0.26
	CANINE	0.12	0.10	0.24	0.20	0.22	0.18	0.20	0.17	-	-	0.30	0.25	0.11	0.09
	FNet	0.28	0.25	0.32	0.28	0.30	0.25	0.26	0.23	0.25	0.22	-	-	0.24	0.21
	XLM-R	0.35	0.30	0.40	0.35	0.38	0.32	0.36	0.31	0.32	0.28	0.42	0.36	-	-
SM	GPT-2	-	-	0.35	0.28	0.34	0.27	0.28	0.24	0.30	0.25	0.40	0.32	0.24	0.22
	BERT	0.26	0.23	-	-	0.50	0.38	0.38	0.34	0.32	0.29	0.42	0.36	0.22	0.20
	DistilBERT	0.27	0.24	0.52	0.40	-	-	0.37	0.33	0.33	0.30	0.43	0.37	0.21	0.19
	Electra	0.30	0.27	0.48	0.42	0.46	0.38	-	-	0.34	0.30	0.40	0.34	0.26	0.24
	CANINE	0.14	0.12	0.22	0.18	0.20	0.16	0.18	0.15	-	-	0.28	0.23	0.12	0.10
	FNet	0.26	0.23	0.30	0.26	0.28	0.23	0.24	0.21	0.23	0.20	-	-	0.22	0.19
	XLM-R	0.33	0.28	0.38	0.33	0.36	0.30	0.34	0.29	0.30	0.26	0.40	0.34	-	-
LIME	GPT-2	-	-	0.38	0.30	0.37	0.29	0.31	0.27	0.33	0.28	0.43	0.35	0.27	0.25
	BERT	0.28	0.25	-	-	0.53	0.40	0.41	0.37	0.35	0.32	0.45	0.39	0.24	0.22
	DistilBERT	0.29	0.26	0.55	0.42	-	-	0.40	0.36	0.36	0.33	0.46	0.40	0.23	0.21
	Electra	0.32	0.29	0.50	0.45	0.48	0.40	-	-	0.36	0.32	0.42	0.36	0.28	0.26
	CANINE	0.16	0.14	0.24	0.20	0.22	0.18	0.20	0.17	-	-	0.30	0.25	0.14	0.12
	FNet	0.28	0.25	0.32	0.28	0.30	0.25	0.26	0.23	0.25	0.22	-	-	0.24	0.21
	XLM-R	0.35	0.30	0.40	0.35	0.38	0.32	0.36	0.31	0.32	0.28	0.42	0.36	-	-

For the Yahoo Answers dataset, transferability patterns align with those observed in AG News, with BERT showing high similarity in attribution maps, achieving IoU scores up to 0.71 for SM to LIME and 0.62 for SHAP to LIME. The SM-LIME pairing remains the most transferable, with scores like 0.61 for DistilBERT and 0.59 for XLM-R, while LIME to SHAP produces low transferability, with scores as low as 0.24 for BERT and 0.27 for GPT-2. CANINE continues to exhibit varied interpretability, with IoU scores ranging from 0.24 (SHAP to SM) to 0.39 (SM to SHAP), and GPT-2 shows moderate transferability, reaching 0.40 for SM to LIME.

The transferability among interpreters varies due to their distinct attribution mechanisms. The SM interpreter achieves the highest transferability, particularly against LIME, due to their shared focus on local token importance as AdvChar preserves localized interpretation maps with character-level perturbations. LIME, relying on local linear approximations through perturbation-based sampling, shows lower transferability to SM, as its less precise saliency estimation diverges from gradient-based sensitivity. In contrast, SHAP accounts for global feature interactions which enables moderate transferability to LIME. However, there is poor transferability from LIME to SHAP as localized focus often fails to maintain global coherence in SHAP interpretations.

Classifiers. After examining the transferability of adversarial inputs across various NLP classifiers using the SST-2 and AG News datasets, Table V provides insights into the effectiveness of adversarial attacks and the confidence with which classifiers make misclassifications. The SHAP interpreter paired



(a) SST-2 dataset.

(b) AG News dataset.

(c) Yahoo Answers dataset.

Fig. 4: Density plots showcasing the distribution of input text length (TL) concerning misclassification confidence, perturbation amount, and query count (MC, PA, and QC) for different models, using three interpreters - SHAP, SM, and LIME.

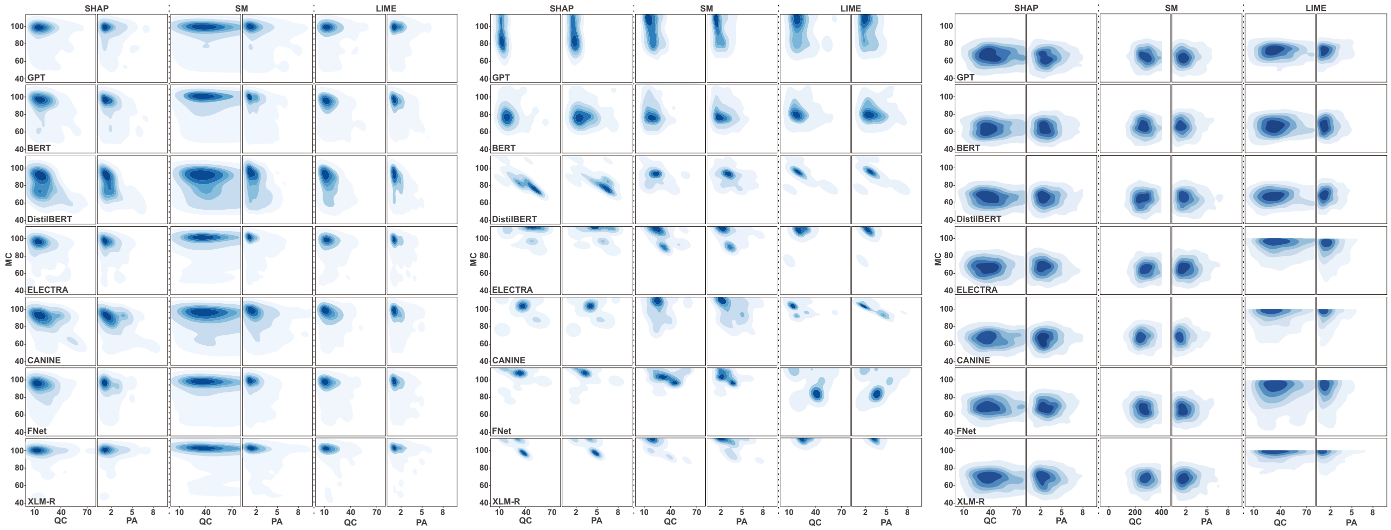
with Electra as the source classifier showed high adversarial transferability, especially when targeting BERT. In the SST-2 dataset, this pairing achieved an ASR of 0.72 against BERT, while in the AG News dataset, an ASR of 0.66 was recorded against DistilBERT.

The SST-2 dataset highlighted SHAP with CANINE targeting GPT-2 and XLM-R as the least transferable combination with an ASR of 0.1. In contrast, the SM interpreter with CANINE targeting Electra in the AG News dataset achieved a minimal ASR of 0.04. For the Yahoo Answers dataset, SHAP with Electra performs strong transferability, achieving an ASR of 0.50 against BERT (MC 0.45), consistent with SST-2 and AG News trends. LIME with DistilBERT also shows high transferability, with an ASR of 0.55 against BERT (MC 0.42). In contrast, SHAP and SM with CANINE are the least transferable, with ASRs as low as 0.11–0.12 (MC 0.09–0.10). The results show the significant adversarial transferability potential of certain combinations of models, with BERT often emerging as a susceptible target.

D. *Comparative Metric Analysis*

Identifying trends can offer critical insights into optimal model selection and tuning when examining different architectures and their reactions to various input conditions. We embarked on a comparative analysis to shed light on this, focusing primarily on model architectures such as GPT, BERT, DistilBERT, ELECTRA, CANINE, FNet, and XLM-R. We calculated the correlation between input text length (TL), misclassification confidence (MC), query count (QC), and perturbation amount (PA) through a detailed assessment. Figures 4a–5b present a visual representation of our findings, highlighting the specific behaviors of each model across different interpreters (SHAP, SM, and LIME) on two datasets.

In Figure 4a, when analyzed using SHAP on the SST-2 dataset, most models display a concentrated cluster of values in the mid- to high-range of MC scores, regardless of the length of the input text. Notably, the density plots exhibit a vertical pattern, which suggests that the MC score is usually consistent, irrespective of the input text's length. The behavior of models analyzed using SM is more varied compared to SHAP. Density plots show horizontal and vertical orientations, indicating that the interpretation heavily depends on the model and its interaction with the input text length. LIME interpretations reveal a combination of patterns found in SHAP and SM. Some models show dense concentration around specific MC values (like SHAP), while others exhibit variations across text lengths (like SM). When analyzed using SHAP and LIME, models like GPT and BERT show a high concentration around mid-range MC values, regardless of input text length. This suggests a consistent confidence level in misclassification maintained across different text lengths. The behavior of the DistilBERT and ELECTRA models seems to be influenced by the interpretation method. At the same time, SHAP interpretations result in consistent MC values, and SM and LIME exhibit more varied patterns, particularly about text length. CANINE, FNet, and XLM-R models show diverse performances, especially under SM. Density plots indicate MC and input text length influence interpretation. In Figure 4b, When analyzed using SHAP on the AG News dataset, the models demonstrate distinct vertical patterns, particularly GPT, BERT, and XLM-R. This indicates a consistency in MC regardless of the input length. For DistilBERT and ELECTRA,



(a) SST-2 dataset.

(b) AG News dataset.

(c) Yahoo Answers dataset.

Fig. 5: Density plots showcasing the distribution of misclassification confidence (MC) concerning query count and perturbation amount (QC and PA) for different models, using three interpreters - SHAP, SM, and LIME.

there is a noticeable split in the density, which implies two potential clusters or behaviors based on MC, leading to non-uniform model responses across data points. The density patterns of SM vary more than SHAP. GPT and BERT show diverse patterns based on text length, while ELECTRA and CANINE display dual peaks, indicating two potential behaviors. A combination of dense and dispersed areas can be observed when using LIME with models such as GPT and BERT. However, the pattern of CANINE is remarkably distinct. It has an oval shape with high density at mid-range MC values, which suggests a strong correlation between MC and a specific range of text lengths.

In Figure 4c, the SHAP-based analysis on the Yahoo Answers dataset reveals that models such as GPT-2, BERT, and XLM-R exhibit vertically aligned density patterns, with misclassification confidence (MC) values concentrated in the mid to high range (approximately 40–80), irrespective of input text length. This suggests stable confidence in misclassification across varying input sizes. In contrast, DistilBERT and ELECTRA show more dispersed MC distributions, indicating greater sensitivity to input length. Under the SM interpreter, density patterns become more diverse, for instance, GPT-2 and BERT demonstrate dual MC peaks around 20 and 70, while CANINE and FNet exhibit broader density regions associated with high query counts (QC), ranging from 200 to 400 with the increased computational cost. The LIME-based results also show a mix of behaviors: GPT-2 and BERT maintain consistent MC values in the 40–80 range, whereas CANINE forms a distinct oval-shaped cluster centered around MC values of 40–60 and text lengths between 75 and 125.

In Figure 5a, The MC distribution on the SST-2 dataset for all models in SHAP displays primarily elliptical patterns. For models like GPT and BERT, the graphs based on QC have a solid vertical alignment, suggesting consistent misclassification confidence for varied query counts. Models with SM produce broader and more diverse patterns, whereas GPT and BERT still exhibit vertical concentrations. CANINE and FNet, on the other hand, exhibit bilateral symmetry around the QC metric. LIME results show dense regions skewed towards lower QC and PA values, especially for models like DistilBERT and ELECTRA. GPT and BERT models exhibit consistent behavior across interpretation methods, with vertical solid clusters in the QC metric, suggesting consistent MC values irrespective of query counts. DistilBERT and ELECTRA models exhibit different patterns. While their SHAP plots show vertical clusters, LIME displays more horizontally aligned behaviors, especially about the PA metric. CANINE, FNet, and XLM-R models have unique interpretations. CANINE displays a symmetrical pattern in the SM plot, while FNet and XLM-R exhibit scattered regions, especially with the LIME interpretation method. In Figure 5b, The distributions of most models with SHAP on the AG News dataset are vertically stretched, indicating consistent MC across various QC values, particularly in the GPT model. Patterns in the SM method are more diverse, with BERT and DistilBERT showing a distinct split, particularly with the QC metric, while GPT retains its vertical orientation. Using the LIME methodology, most models, particularly BERT and DistilBERT, tend to shift towards lower values on the QC metric, indicating fewer queries are required for the misclassification. The GPT model displays consistent vertical concentration across interpretation methods, suggesting stable MC regardless of QC or PA. BERT and DistilBERT models exhibit distinct patterns in each interpretation. SHAP and SM have MC distributions aligned vertically, whereas LIME is more horizontal, indicating that PA is the dominant factor. Different interpretation methods produce diverse patterns in ELECTRA, CANINE, FNet, and XLM-R models. For instance, when using LIME on CANINE's data, the PA metric shows a horizontally skewed pattern, while ELECTRA has a slight concentration in lower QC values under the same methodology.

In Figure 5c, SHAP-based MC–QC plots for GPT, BERT, and DistilBERT display dense, horizontally stretched clusters, indicating consistent misclassification confidence across

TABLE VI: The attack success rate (ASR) on SST-2 under different defenses (lower is better).

Model	Adversarial Training	Pre-processing [44]
GPT-2	0.25	0.22
BERT	0.24	0.14
DistilBERT	0.25	0.18

a range of query counts. SM plots for these models show compact, circular distributions with moderate spread, while CANINE, FNet, and XLM-R have slightly broader and less defined clusters. Under LIME, GPT and BERT maintain horizontally extended patterns with some vertical spread, suggesting a slight sensitivity to the amount of perturbation. CANINE, FNet, and XLM-R under LIME show highly scattered patterns with no clear alignment, indicating significant instability in model responses during attack.

After conducting a thorough analysis, it has become evident that AdvChar on different machine learning models, as evaluated through the AG News dataset, has revealed a diverse scope of model behaviors. It is worth noting that specific models exhibit consistent behaviors where their attack outcomes are not significantly affected by input text length (TL), query count (QC), or perturbation amount (PA). Different models show more complex interactions, where the attack's success depends on a combination of multiple metrics. This broader perspective highlights two key points: first, the dataset and method used to interpret data can significantly impact the attack outcomes, and second, understanding these subtle interactions is crucial for fine-tuning attack strategies. Such understanding of vulnerabilities leads to better attack and defense in machine learning.

VI. Discussion

AdvChar has the advantage of its stealthy and potent nature. Its ability to introduce minor changes to the textual input is evident, causing significant shifts in model predictions. Another noteworthy feature is its capability to alter text without compromising its original interpretation, ensuring that adversarial inputs are still plausible to human evaluators. Targeting the most critical tokens strengthens the attack's precision, subtly revealing the model's vulnerabilities.

There are also some challenges for the attack to be used. Advanced input detection techniques can potentially flag the simple character-level modifications employed by the attack. Additionally, with more layers of defense in the INLPS, executing the attack becomes much more complicated. The attack's dependence on determining the most critical tokens might make it less effective against models that distribute importance more evenly across tokens.

Given the challenges posed by the attack, we recommend a few strategies to counter its effects: (1) implementing input sanitization as a frontline defense, (2) improving the model's interpretability to assist in tracing its reasoning processes, and (3) adversarial training to help the models' robustness.

To check the effectiveness of adversarial training defense, we assessed how well three different models - GPT-2, BERT, and DistilBERT could stand up against our attack. We trained the models for a short time on the SST-2 dataset, to which we injected some adversarial symbols into inputs randomly. We used the ASR metric to check how often the attack is successful. BERT performed a bit better than the others, with a lower ASR of 0.24, while GPT-2 and DistilBERT both had an ASR of 0.25 (see Table VI). Compared with the results in Table III, we can see an improvement in the robustness of the models against this type of attack.

We also adopt Adversarial Text Normalization (ATN) [44] as a deterministic filter before tokenization. ATN performs: (i) Unicode NFKC canonicalization and casefolding, (ii) removal of zero-width/control characters, (iii) mapping of Unicode confusables (UTS#39-style) to their ASCII counterparts, (iv) accent folding (NFD + combining-mark strip), and (v) canonicalization of whitespace and punctuation (collapse repeats, standardize variants). We then predict with the normalized text without changing model parameters. This is a single-pass transformation that directly targets the character-level edits exploited by our attack. On SST-2, under the same attack budget and explanation-similarity constraint, ATN reduces ASR to **0.22** (GPT-2), **0.14** (BERT), and **0.18** (DistilBERT) (see Table VI). Residual successes arise from edits outside the confusable map or tokenization shifts that survive normalization, but ATN provides a fast, training-free baseline that lowers ASR and the effective perturbation amount.

VII. Conclusion

Adversarial attacks such as AdvChar demonstrate that there are creative methods to deceive machine learning systems. During our research, we discovered that AdvChar can manipulate the text by making minor changes, leading to significant errors in how an NLP system interprets it. This is especially true when the system is not aware of the precise nature of the attack. We also pointed out some cases where AdvChar might not work. Knowing these weak spots helps us think about better attacks in the future and, more importantly, ways to defend against them. Overall, our work on AdvChar reminds us that while there are always new challenges in machine learning, there are also new solutions. We hope AI systems will become more innovative and safer as we continue exploring.

Acknowledgment. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(No. 2021R1A2C1011198), (Institute for Information & communications Technology Planning & Evaluation) (IITP) grant funded by the Korea government (MSIT) under the ICT Creative Consilience Program (IITP-2021-2020-0-01821), AI Platform to Fully Adapt and Reflect Privacy-Policy Changes (No. 2022-0-00688).

References

[1] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, 2022.

[2] R. Dabre, C. Chu, and A. Kunchukuttan, "A survey of multilingual neural machine translation," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–38, 2020.

[3] J. Ni, T. Young, V. Pandelea, F. Xue, and E. Cambria, "Recent advances in deep learning based dialogue systems: A systematic survey," *Artificial intelligence review*, vol. 56, no. 4, pp. 3055–3155, 2023.

[4] L. Yan, Z. Zhang, G. Tao, K. Zhang, X. Chen, G. Shen, and X. Zhang, "Parafuzz: An interpretability-driven technique for detecting poisoned samples in nlp," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 66 755–66 767.

[5] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing: A survey," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 3, 2020.

[6] I. Alsmadi, K. Ahmad, M. Nazzal, F. Alam, A. Al-Fuqaha, A. Khreishah, and A. Algosaibi, "Adversarial nlp for social network applications: Attacks, defenses, and research directions," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 6, pp. 3089–3108, 2023.

[7] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774.

[8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[10] R. Bao, J. Wang, and H. Zhao, "Defending pre-trained language models from adversarial word substitution without performance sacrifice," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Aug. 2021, pp. 3248–3258.

[11] S. Ren, Y. Deng, K. He, and W. Che, "Generating natural language adversarial examples through probability weighted word saliency," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 1085–1097.

[12] X. Hu, G. Liu, B. Zheng, L. Zhao, Q. Wang, Y. Zhang, and M. Du, "Fasttextdodger: Decision-based adversarial attack against black-box nlp models with extremely high efficiency," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 2398–2411, 2024.

[13] H. Gao, H. Zhang, J. Wang, X. Zhang, H. Wang, W. Li, and T. Tu, "Nuatgan: Generating black-box natural universal adversarial triggers for text classifiers using generative adversarial networks," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 6484–6498, 2024.

[14] E. Abad Rocamora, Y. Wu, F. Liu, G. G. Chrysos, and V. Cevher, "Revisiting character-level adversarial attacks for language models," *arXiv e-prints*, pp. arXiv–2405, 2024.

[15] J. Rusert, "VertAttack: Taking advantage of text classifiers' horizontal vision," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, K. Duh, H. Gomez, and S. Bethard, Eds. Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 719–732. [Online]. Available: https://aclanthology.org/2024.naacl-long.41/

[16] E. Abdukhamidov, M. Abuhamad, F. Juraev, E. Chan-Tin, and T. AbuHmed, "Advedge: Optimizing adversarial perturbations against interpretable deep learning," in *Computational Data and Social Networks: 10th International Conference, CSoNet 2021, Virtual Event, November 15–17, 2021, Proceedings 10*. Springer, 2021, pp. 93–105.

[17] E. Abdukhamidov, M. Abuhamad, S. S. Woo, E. Chan-Tin, and T. Abuhmed, "Hardening interpretable deep learning systems: Investigating adversarial threats and defenses," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 3963–3976, 2023.

[18] E. Abdukhamidov, F. Juraev, M. Abuhamad, and T. Abuhmed, "Black-box and target-specific attack against interpretable deep learning systems," in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, 2022, pp. 1216–1218.

[19] E. Abdukhamidov, M. Abuhamad, G. K. Thiruvathukal, H. Kim, and T. Abuhmed, "Singleadv: single-class target-specific attack against interpretable deep learning systems," *IEEE Transactions on Information Forensics and Security*, 2024.

[20] S. Bhambri, S. Muku, A. Tulasi, and A. B. Buduru, "A survey of black-box adversarial attacks on computer vision models," *arXiv preprint arXiv:1912.01667*, 2019.

[21] L. Chen, W. Ruan, X. Liu, and J. Lu, "Seqvat: Virtual adversarial training for semi-supervised sequence labeling," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8801–8811.

[22] J. Li, "Tianyu du, xiangyu liu, rong zhang, hui xue, and shouling ji. 2021. enhancing model robustness by incorporating adversarial knowledge into semantic representation," in *ICASSP*, vol. 2021, 2021, pp. 7708–7712.

[23] Y. Cheng, L. Jiang, and W. Macherey, "Robust neural machine translation with doubly adversarial inputs," *arXiv preprint arXiv:1906.02443*, 2019.

[24] S. Goyal, S. Doddapaneni, M. M. Khapra, and B. Ravindran, "A survey of adversarial defenses and robustness in NLP," *ACM Computing Surveys*, vol. 55, no. 14s, pp. 1–39, 2023.

[25] A. Li, F. Zhang, S. Li, T. Chen, P. Su, and H. Wang, "Efficiently generating sentence-level textual adversarial examples with seq2seq stacked auto-encoder," *Expert Systems with Applications*, vol. 213, p. 119170, 2023.

[26] H. Liu, Y. Zhang, Y. Wang, Z. Lin, and Y. Chen, "Joint character-level word embedding and adversarial stability training to defend adversarial text," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 8384–8391.

[27] K. Du, F. Xing, R. Mao, and E. Cambria, "Financial sentiment analysis: Techniques and applications," *ACM Comput. Surv.*, vol. 56, no. 9, 2024.

[28] J. Fields, K. Chovanec, and P. Madiraju, "A survey of text classification with transformers: How wide? how large? how long? how accurate? how expensive? how safe?" *IEEE Access*, 2024.

[29] S. Gui, C. Shao, Z. Ma, x. zhang, Y. Chen, and Y. Feng, "Non-autoregressive machine translation with probabilistic context-free grammar," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 5598–5615.

[30] Z. Li, S. Fan, Y. Gu, X. Li, Z. Duan, B. Dong, N. Liu, and J. Wang, "Flexkbqa: A flexible llm-powered framework for few-shot knowledge base question answering," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, pp. 18 608–18 616, Mar. 2024.

[31] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is bert really robust? a strong baseline for natural language attack on text classification and entailment," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 8018–8025.

[32] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[33] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, vol. 1, 2019, p. 2.

[34] V. Sanh, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter." in *Proceedings of Thirty-third Conference on Neural Information Processing Systems (NIPS2019)*, 2019.

[35] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in *ICLR*, 2020.

[36] J. H. Clark, D. Garrette, I. Turc, and J. Wieting, "Canine: Pre-training an efficient tokenization-free encoder for language representation," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 73–91, 2022.

[37] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon, "Fnet: Mixing tokens with fourier transforms," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.

[38] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[39] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[40] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Oct. 2013.

[41] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *NIPS*, 2015.

[42] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "Textbugger: Generating adversarial text against real-world applications," in *Proceedings 2019 Network and Distributed System Security Symposium*. Internet Society, 2019.

[43] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is bert really robust? natural language attack on text classification and entailment," *arXiv preprint arXiv:1907.11932*, 2019.

[44] D. Pruthi, B. Dhingra, and Z. C. Lipton, "Combating adversarial misspellings with robust word recognition," July 2019.