2023 IEEE International Conference on Big Data (BigData) | 979-8-3503-2445-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/BigData59044.2023.10386090

ANTI-Disinformation: An Adversarial Attack and Defense Network Towards Improved Robustness for Disinformation Detection on Social Media

Kuan-Chun Chen*
National Cheng Kung University
Tainan, Taiwan
champion516615@gmail.com

Chih-Yao Chen*
University of North Carolina at Chapel Hill
NC, USA
cychen@cs.unc.edu

Cheng-Te Li
National Cheng Kung University
Tainan, Taiwan
chengte@ncku.edu.tw

***Abstract*—The prevalence of disinformation, which includes malformation (e.g., cyberbullying) and misinformation (e.g., fake news) in online platforms has raised significant concerns, prompting the need for robust detection methods to mitigate its detrimental impact. While the field of text classification has witnessed notable advancements in recent years, existing approaches often overlook the evolving nature of disinformation, wherein perpetrators employ perturbations to toxic content to evade detection or censorship. To address this challenge, we present a novel framework, Adversarial Network Towards Improved robustness for Disinformation detection (ANTI-Disinformation), which leverages reinforcement learning techniques as adversarial attacks. Additionally, we propose a defense model to enhance model's robustness against such attacks. To evaluate the effectiveness of our approach, we conduct extensive experiments on well-known disinformation datasets collected from multiple social media platforms. The results demonstrate our approach can effectively produce degradation in existing models' performance the most, showcasing the effectiveness of our framework and the vulnerability of existing detection systems. The results also exhibit that the proposed defense methods can consistently outperform existing typical methods in constructing robust detection models.**

***Index Terms*—Disinformation Detection, Adversarial Attack, Adversarial Defense, Model Robustness, Reinforcement Learning**

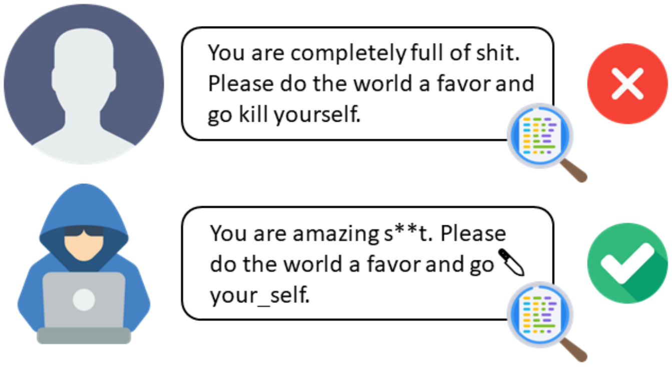
You are completely full of shit. Please do the world a favor and go kill yourself.
You are amazing s**t. Please do the world a favor and go your_self.

Fig. 1: Malicious users could employ various techniques to evade censorship on online platforms, presenting a challenge for current disinformation detection models. These techniques include making grammar errors, replacing synonyms, or adding noisy text, in an attempt to deceive the detection system and bypass scrutiny.

I. Introduction

With the rapid expansion of online platforms and social media, disinformation has emerged as a pervasive and concerning phenomenon. Disinformation, a term that includes malinformation (e.g., cyberbullying) and misinformation(e.g., fake news) has caused severe problems for online users. While cyberbullying leads to emotional and psychological consequences [14], [24], [43], the proliferation of fake news also inflicts harm on individuals by inducing stress and anxiety [11], and can even have detrimental effects on society by fostering polarization among people [4]. The widespread of disinformation highlights the urgent need for effective detection methods to mitigate its harmful impact and create safer online environments [9]. Previous studies on disinformation detection often rely on textual features of the post [10], [18], [22] or user and social media feature [8], [13], [27], [35]. This line of work may fall short when facing adversarial attacks, as demonstrated in Figure 1, posing more challenges for accurate detection. By intentionally manipulating input data, adversarial attacks can deceive models and lead to incorrect or unexpected predictions. Existing literature utilizes masked language modeling (MLM) [17] or perturbing inflections [36] to create adversarial samples. However, MLM may generate tokens that reverse the semantics of a whole sentence (e.g., The food is *good* can be replaced by *bad*) and requires heuristics to avoid such circumstances. In addition, perturbing inflections can only account for a restricted range of replacements in order to preserve semantic consistency.

In this work, we propose using reinforcement learning, Adversarial Network Towards Improved robustness for **Disinformation** detection (**ANTI-Disinformation**), to make the most effective modifications with the least impact on the semantics. We formulate the text editing process as an environment for the agent, and we propose *push reward* and *similarity reward* to maximize both the perturbation and

[*] Equal contribution.

semantic conservation, respectively. Moreover, we propose a defense model to restore the attacked text samples, and hope to show the potential of creating a more robust detection system by incorporating adversarial training. This adversarial attack and defense paradigm can not only facilitate adaptive and proactive responses to the ever-changing behaviors associated with the dissemination of disinformation, but also enhance the model's robustness and generalization capabilities. Through an extensive evaluation on diverse datasets collected from social media platforms, we demonstrate the effectiveness and robustness of our proposed attack and defense approach. While this research contributes to the advancement of reliable disinformation detection systems, we hope to create safer online environments and safeguard online users from the detrimental effects of disinformation. To sum up, our contributions are three-fold:

- We propose ANTI-Disinformation , an adversarial attack and defense framework for improving the robustness of disinformation detection on social media.
- We adopt reinforcement learning with two novel reward functions to create high-quality adversarial text samples.
- Experiments conducted on three well-known social media datasets exhibit our framework's success in attacking and defending disinformation detection. The outcome verifies that our method can be used to enhance model's robustness and create a safer online environment.

We organize this paper as follows. We first review relevant studies in Section I, followed by describing the problem statement in Section II-B. The technical details of our proposed attack and defense methods are presented in Section III. We give the experimental settings and results in Section IV-B. Section V-E concludes this work.

II. Related Work

A. Disinformation Detection

Disinformation refers to malinformation and misinformation. Both of which cause mental health issues such as anxiety, depression, low self-esteem, and in extreme cases, even self-harm or suicide [11], [14], [23], [24], [43]. Consequently, many online platforms have implemented policies and tools to report and prevent disinformation. However, it is often too slow for those platforms to take off harmful text or fake news, and it has negative impacts even if the reader had only seen them once. To address this issue, machine learning- or deep learning-based method have been proposed to detect disinformation automatically [1], [6], [19], [40], [41].

Despite its success, we show that deep learning-based methods can be easily attacked and fail to detect sophisticatedly paraphrased disinformation text. Malicious users would try to *attack* automatic detection system in order to get the toxic/fake comment posted. Therefore, we propose ANTI-Disinformation, to improve the robustness of disinformation detection using adversarial networks and reinforcement learning, as a robust detection system is needed.

B. Adversarial Attack and Defense in NLP

Adversarial attacks typically involve making small, imperceptible modifications to input data and are designed to deliberately deceive deep learning models. The goal of these attacks is to exploit vulnerabilities in the model's decision-making process and cause it to make incorrect or unexpected predictions. This is critical in disinformation detection, spam detection, malware detection, and other tasks that are similar in nature. In these tasks, malicious users would try to bypass model detection by creating small variations to the original input. The manipulated input manages to preserve the essence and meaning of the original data, effectively tricking the model and evading its detection mechanisms. To mimic this manipulation technique and thereby enhance model robustness, ongoing research in the field employs techniques such as masked language modeling (MLM) [17], perturbing inflections [36] or input encoding [7] to create adversarial samples. Among previous studies, [42] stands out as particularly relevant to our work. They also utilize reinforcement learning to generate perturbed sentences, albeit focusing on machine translation tasks that require paired ”source” and ”target” samples for training the discriminator. In contrast, our focus lies in disinformation detection, where label supervision alone is sufficient for our purposes.

To improve model robustness, several approaches have been proposed to defend against adversarial attacks. For example, adversarial training [5], [15], [34], [38] include adversarial samples during the training process and make the model better at generalizing to similar attacks during inference. Another example is gradient masking [3], which obscures gradients to limit the attacker's access, making it more challenging to craft effective adversarial examples.

III. Problem Statement

Our primary objective is to enhance the robustness of the model when it faces different types of attacks, such as intentionally creating adversarial examples or introducing grammar errors in sentences. To address this main problem, we divide it into two key aspects. First, we focus on crafting effective adversarial examples to attack the model and minimize its accuracy. Our aim is to explore techniques that can significantly reduce the model's performance when subjected to adversarial inputs. Second, once we have successfully attacked the model, our goal is to restore the adversarial examples and enable the model to regain its original accuracy or even surpass it. We aim to develop strategies that effectively mitigate the impact of adversarial attacks and enhance the overall performance of the model.

Problem 1: Attack Model. Let $p \in P$ denote a post, where P represents the full set of posts. In each post p_i, it contains comments $c_{i1}, c_{i2}, ..., c_{ik}$, where k is the comment number of a post. We concatenate all comments $C_i = \{c_1; c_2; ...; c_k\}$ in a post to form a sequence of comments. For each C_i, we have a truth label $y_i \in \{0, 1\}$, the predicted label y_i^p given by a proxy model M, and the attacked label y_i^a which is predicted on adversarial samples. We denote the attacked comment as

TABLE I: Demonstrations of four possible actions that can be taken by an agent.

Action	Comment	Label
Original	It's a super bummer. People would do something like that to such a good American truck.	1
Change Similar Word	It's a super jerk. People would do something like that to such a good American van.	0
Swap Noun	It's a super truck. People would do something like that to such a good American bummer.	0
Swap Adjective	It's a good bummer. People would do something like that to such a super American truck.	0

C_i^a. In this task, given the attacked comment C_i^a, we aim to deceive the model, i.e., reversing its prediction. In other words, we want y_i^a different from the original prediction y_i^p. Hence, we can formulate the attacking problem as follows:

Problem 1. Given the comment sequence of a post, i.e., C_i, and a prediction model M, the goal is to generate the corresponding adversarial attacked comment sequence C_i^a such that:

$$y_i^p \neq y_i^a, \text{where } y_i^p = M(C_i) \text{ and } y_i^a = M(C_i^a) \qquad (1)$$

Problem 2: Defense Model. In contrast, after the adversarial attacked comment C_i^a is made, we aim to restore the attacked comment to produce a recovered comment, denoted as C_i^r. Given the recovered comment C_i^r, we take it as the input and make prediction again. We denote the prediction made from C_i^r as y_i^r. Our goal is to make y_i^r different from the prediction made by attacked comment y_i^a, i.e., same as the original true label y_i. We can formulate the defending problem as follows:

Problem 2. Given the adversarial attacked comment sequence of a post, i.e., C_i^a, and a prediction model M, the goal is to generate a restored comment sequence C_i^r such that:

$$y_i^a \neq y_i^r = y_i, \text{where } y_i^a = M(C_i^a) \text{ and } y_i^r = M(C_i^r) \qquad (2)$$

IV. Methodology

In this section, we introduce the proposed framework, ANTI-Disinformation. Our framework consists of two main modules, the adversarial attack model and the adversarial defense model.

A. Adversarial Attack Model

We begin with the attack model, in which we leverage reinforcement learning to craft adversarial samples. Reinforcement learning has demonstrated significant advancements, particularly in game-playing scenarios. Therefore, we employ Deep Q-learning (DQN) [30] as the foundation of our approach, and we formulate the attacking process as a text-based game. Specifically, each comment c_i represents a text-based game within a given post. Our objective is to strategically take actions within these text games to effectively reduce the model's accuracy. By iteratively engaging in these text games, we can enable the model to learn how to craft high-quality adversarial examples to degrade the model's performance. Below we describe relevant components and definitions in detail.

1) RL Settings: **Environment.** Let $e_i \in E$ denote our text game environment. We concatenate all comments in a post, so each environment e_i contains words of all comments in a post p_i. At each time step t, the agent selects an action a_t to change the environment e_i to get a changed text c_{t+1} and a reward r_{t+1}.

Action. At each time step, the agent selects an action to interact with the environment. Hence, we design two different actions to let the agent select which to execute. Through taking the action to interact with the environment, we can obtain the reward and the updated state. The possible actions are demonstrated in Table I, and are elaborated below.

- **Synonym Replacement.** Synonym replacement changes similar words in a sentence, which can alternate the model predictions and conserve sentence semantics at the same time [2]. Hence, we utilize word embeddings learned by Word2Vec [29] from these comments in advance. Then, we randomly replace words in each comment with similar words at a certain ratio ω. By default, we empirically set ω to 0.05, and we investigate its impact in the experiment (refer to Section V-E).
- **Swap Noun and Adjective.** When we swap nouns and adjectives in a sentence, this action can disturb the model predictions [32]. We use NLTK [1] to perform part of speech (POS) tagging, and then we swap the word based on its POS tag. Note that we only swap words with the same POS tag, i.e., nouns are swapped with nouns, and adjectives are swapped with adjectives.

Reward. A reward function plays a crucial role in guiding the model's action selection within the context of reinforcement learning. We introduce two innovative reward functions that can aid the model in making better choices. These functions include the *push reward* and the *similarity reward*, each serving a distinct purpose in shaping the model's behavior. The specific details of these two reward functions are described as follows.

- **Push Reward.** Intuitively, we aim to make the model predictions conversed to the ground truth. This can be achieved by penalizing the predictions that closely resemble the ground truth and rewarding the predictions that deviate from it. Therefore, at each time step, we take action a_t to interact with a text environment and then get the modified text c_i^{t+1}. Then we utilize a proxy model (i.e., language model) to calculate the likelihood of the

[1] https://www.nltk.org/

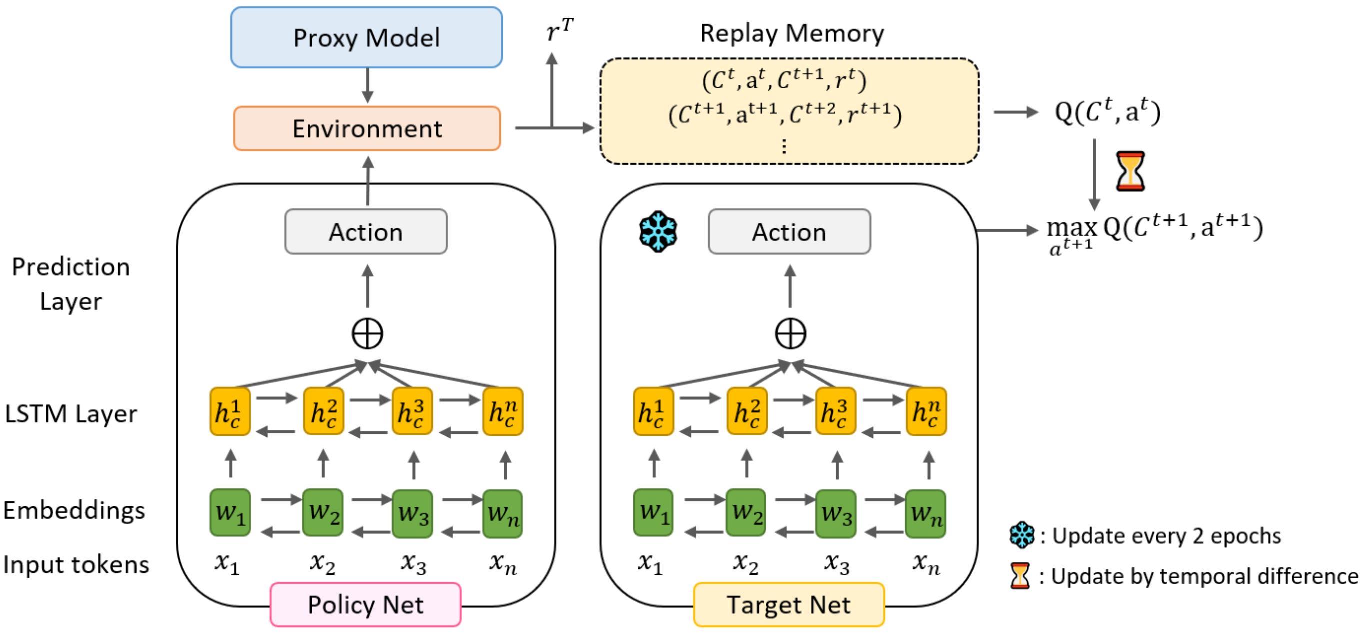
Proxy Model
Environment
r^T
Replay Memory
(C^t, a^t, C^{t+1}, r^t)
$(C^{t+1}, a^{t+1}, C^{t+2}, r^{t+1})$
⋮
$Q(C^t, a^t)$
$\max_{a^{t+1}} Q(C^{t+1}, a^{t+1})$
Prediction Layer
LSTM Layer
Embeddings
Input tokens
Action
Action
$\oplus$
h_c^1 h_c^2 h_c^3 h_c^n
w_1 w_2 w_3 w_n
x_1 x_2 x_3 x_n
Policy Net
Target Net
❄ : Update every 2 epochs
⌛ : Update by temporal difference

Fig. 2: The architecture of our proposed RL-based attacking framework.

original text and the modified text, which is denoted as ρ_i^t and ρ_i^{t+1}, respectively. We discuss such two situations separately. First, if $y_i^p = 0$, we *push* the model to predict 1. Hence, we force ρ_i^t to get closer to one and accordingly give it a positive reward. Nevertheless, if the model still produces a ρ_i^t which is around zeros, we discourage the action by giving it a negative reward. Second, if $y_i^p = 1$, the rewarding process is done conversely. We denote the *push reward* as r_t^p, which can be written as:

$$r_t^p = \begin{cases} \rho_i^{t+1} - \rho_i^t & , \quad \text{if } y_i^p = 0 \\ \rho_i^t - \rho_i^{t+1} & , \quad \text{if } y_i^p = 1 \end{cases} \tag{3}$$

- **Similarity Reward.** In addition to providing rewards to the attack model, preserving the semantic meaning of words is another crucial aspect to consider. Therefore, after the agent takes action to interact with the text environment, we compute the average embeddings of words in both the original clean text and the modified text. These average embeddings yield sentence embeddings denoted as s^0 and s^{t+1} respectively. Subsequently, we utilize cosine similarity to measure the similarity between these two sentences, which serves as our similarity reward denoted as r^s, given by:

$$r^s = \cos\left((\sum_{i=1}^{k}\sum_{j=1}^{n} w_{ij}^0)/nk, (\sum_{i=1}^{k}\sum_{j=1}^{n} w_{ij}^{t+1})/nk \right). \tag{4}$$

Finally, the total reward is the weighted sum of push reward and similarity reward. In addition, we design a tunable hyperparameter α to adjust their relative importance. This can be formulated as,

$$r^T = \alpha \cdot r^p + (1 - \alpha) \cdot r^s, \tag{5}$$

where α is set as 0.5 by default.

2) Updating States: We take advantage of Deep Q-Network (DQN) [30] as the foundation of reinforcement learning. We begin with the policy network, which is responsible for comprehending the current state of the environment and selecting the most appropriate action that maximizes future rewards. Our primary objective is to maximize the cumulative reward, given by:

$$R_{t_0} = \sum_{t=t_0}^{T} \gamma^{t-t_0} r_t, \tag{6}$$

where γ is a discount factor to decide the weight of future reward, and T is the time step when the game quits. Then we aim to determine the maximum expected return achievable given a specific text input c and action a^t. This is realized by $Q(c, a)$ that represents the optimal action-value function, given by Equation 7:

$$Q(c, a) = \max_{\pi} E\left[R_t | c^t = c, a_t = a, \pi\right], \tag{7}$$

where the symbol π represents the distribution over actions.

The optimal action-value function follows by the attribute of *Bell equation*. We take action a^t at time-step t, and obtain text state c^t. Then the current Q value is $Q(c^t, a^t)$. By leveraging the replay memory mechanism [25], we can retrieve the past experiences and obtain $Q(c^{t+1}, a^{t+1})$. To determine the action that maximizes the expected future reward, we consider the future Q-value and apply a discount vector γ. Additionally, we incorporate the reward r obtained after taking action a^{t+1}. By performing these computations, we ultimately obtain the future Q-value $Q(c^{t+1}, a^{t+1})$, given by:

$$Q(c, a) = E_{\mathbf{c^{t+1} \sim e_i}}\left[r_t + \gamma \max_{a^{t+1}} Q(c^{t+1}, a^{t+1}) | c^t, a^t\right] \tag{8}$$

When we have the current Q value $Q(c^t, a^t)$ and the future Q value $Q(c^{t+1}, a^{t+1})$, we can update the current Q value $Q(c^t, a^t)$ by the temporal difference [37] between $Q(c^t, a^t)$ and $Q(c^{t+1}, a^{t+1})$, add reward r_t, and control the learning rate by η, given by:

$$\begin{aligned} Q(c^{t+1}, a^{t+1}) &\leftarrow Q(c^t, a^t) \\ &+ \eta \left[r + \gamma \max_{a^{t+1}} Q(c^{t+1}, a^{t+1}) - Q(c^t, a^t) \right]. \end{aligned} \tag{9}$$

With Equation 9, we can obtain the updated Q value, and complete the state updating process.

3) Training Objective for Attack Model: The overall training process consists of the following steps.

1) First, we utilize word2vec [29] to obtain word embeddings for each comment. These embeddings are then fed into an LSTM layer to capture the hidden representation of sequential word features. By summing up these representations, we derive sentence-level representations denoted as h_s. Subsequently, h_s is passed through a classification layer, forming our policy network. The purpose of the policy network is to make decisions regarding the appropriate actions to be executed.
2) Second, the agent interacts with the environment e_i, leading to changes in the environment state. We store various important components in the replay memory, including the current text state c^t, the next text state c^{t+1}, the corresponding reward r_t, and the action taken a^t.
3) Next, we randomly sample a batch from the replay memory. The current text state c^t is fed into the policy network, while c^{t+1} is passed to the target network. As our approach is off-policy, we utilize the target network, which is updated only once every two epochs, to evaluate the value of c^{t+1}.
4) Our objective is to minimize the difference between c^t and c^{t+1}. To achieve this, we employ the Huber loss function, which combines both square loss and absolute loss, and hence is more advantageous in handling noisy estimates of Q-values and is robust to outliers. The equation representing the Huber loss function is as follows:

$$\mathcal{L}_{huber} = \begin{cases} 0.5(Q^{t+1} - Q^t), \text{if } | Q^{t+1} - Q^t | \leq \delta \\ \delta(| Q^{t+1} - Q^t | -0.5\delta), \text{o.w.} \end{cases} \tag{10}$$

where the hyperparameter δ determines the sensitivity to outliers, with smaller values increasing robustness. Based on existing empirical study [2], we are advised to set $\delta = 1.35$ by default.

B. Defense Model

Adversarial examples serve not only as a means to attack models but also as a tool to enhance their robustness. Given that misspelled words and grammar errors are common on online platforms, such noise can significantly impact the accuracy of the model. By defending against adversarial attacks, we can increase the accuracy and robustness of the model. Thus, we propose a novel defense framework that realizes this concept to address our problem. The overview of our defense model is exhibited in Figure 3.

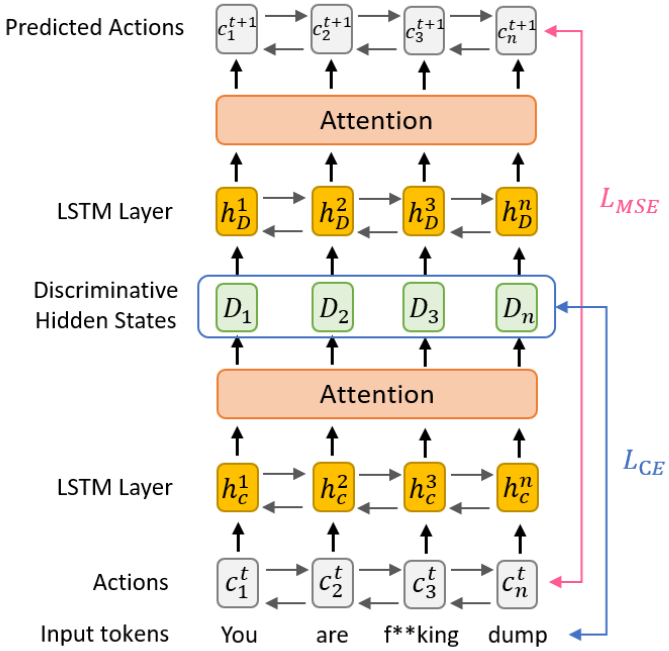

Fig. 3: The architecture of our defense model.

In each time step, we take action a^t to modify the text environment, resulting in the next text state c^{t+1}. We can leverage this process of change to learn how to generate adversarial examples to attack the model. To defend the attack, we take the current text c^t as the input and c^{t+1} as the target. Our aim is to learn the difference between c^t and c^{t+1}. To achieve this, we feed the input c^t into an LSTM layer to capture the hidden representation of the text. We then employ word attention to map the hidden representation of c^t to D^t, a discriminated hidden vector that encodes the transformation of c^t into c^{t+1}. Next, we use D^t as the input to another LSTM layer acting as a decoder, which decodes D^t to generate the next text state c^{t+1}. Finally, we minimize the mean square error (MSE) between the current text c^t and the next text c^{t+1}. This is formulated as,

$$\mathcal{L}_{MSE} = \left(\sum_{i=1}^{k} \sum_{j=1}^{n} w_{ij}^t / nk - \sum_{i=1}^{k} \sum_{j=1}^{n} w_{ij}^{t+1} / nk \right)^2 \tag{11}$$

Additionally, we simultaneously perform a classification task. We concatenate all discriminator vectors D^t learned from the attacked texts, and utilize the discriminator attention to identify the importance of different attack processes. We then aggregate all discriminator vectors to perform classification. This can be written as:

$$\mathcal{L}_{CE} = y_i \log y_i^p + (1 - y_i) \log (1 - y_i^p), \tag{12}$$

[2] https://scikit-learn.org/stable/modules/linear_model.html#huber-regression

TABLE II: Statistics of Instagram, Vine, and Twitter datasets.

Datasets	Instagram	Vine	Twitter
# Posts	2,211	882	742
# Positive	676	283	372
# Negative	1,535	599	370
# Comments	159,277	70,385	216,805

In other words, we have two objectives that form a multi-task learning scenario: performing classification and minimizing the error between c^t and c^{t+1}. The total loss is calculated by their weighted sum, given by:

$$\mathcal{L} = \beta \cdot \mathcal{L}_{MSE} + (1 - \beta) \cdot \mathcal{L}_{CE} \tag{13}$$

where the first term is a commonly used cross entropy loss, and the second term is a mean square error. This loss function incorporates a weighting factor β to control the relative importance of the two objectives, where β is set as 0.5 by default. First, we employ cross-entropy to evaluate the classification performance of the defense model. Second, we utilize mean square error to quantify the difference between c^t and c^{t+1}. By employing this approach, we enable the discriminator vector to learn how to transform from the current text c^t to the next state c^{t+1}, and acquire meaningful features to enhance the model's accuracy.

V. Experiments

A. Evaluation Setup

To evaluate the effectiveness of the proposed text attack and defense models, in this section, we present a comprehensive set of experiments conducted on datasets from Instagram [21], Vine [33], and Twitter [28]. Instagram and Vine datasets are for cyberbullying detection, and Twitter dataset is for rumor detection. The data statistics are presented in Table II.

To ensure a thorough evaluation under different settings, we divide our experiments into two parts: white box attacks and black box attacks. In the white box attack scenario, we have full access to the target model's architecture and parameters, enabling us to execute tailored attacks. Conversely, in the black box attack scenario, we have limited knowledge about the target model, which is a more realistic setting. Following the attacks on the models, we apply the proposed defense model to defend against the generated adversarial examples. Additionally, we conduct ablation studies and hyperparameter analysis to gain further insights into the performance and effectiveness of our proposed models.

To evaluate the effectiveness of our proposed method, we utilize four popular language models, i.e., LSTM [20], BERT [12][3], RoBERTa [26][4], and XLNet [39][5] as proxy models. We utilize the default hyperparameters in their original repositories to construct these models. We split each dataset into training, validation, and test with ratios 70%, 10%, and 20%, respectively. Accuracy is employed as the evaluation metric. We repeat the experiments 10 times, and report and average scores. The replacement rate of text changes is set as 0.1 by default.

B. Results on White-Box Attacks

White box attack entails access to the model parameters, allowing us to target and attack the model directly. In Table III, we present the results of our white box attacks, demonstrating the impact of combining our proposed attack and defense methods with the four text classification models. Following the white box attacks, we observe a significant decrease in accuracy. Such results imply that the texts produced with our proposed attack model can effectively destroy the usefulness of well-known text classification models. The percentages of accuracy drop are all over 50% for all models in Instagram data, over 20% in Vine data, and over 40% in Twitter data. We can conclude that if the adversary can access the model, i.e., the white-box setting, these classification language models are vulnerable.

C. Results on Black-Box Attacks

Black box attack [31] employs a proxy network to generate adversarial examples for attacking other networks. In our study, we adopt LSTM as the proxy model to generate adversarial examples for attacking the target models. We aim at comparing our proposed attack method with the black-box attacking baselines under the settings of various target models. The results of experimental comparison are shown in Table IV. The "Clean" column represents the accuracy scores of the models without performing any attacks. The "Permutation" column demonstrates the performance when words in a sentence are randomly permuted. Additionally, we generate adversarial examples by randomly swapping adjectives and nouns (column "Adj. Swap" and "Noun Swap"). We also include DeepWordBug [16], a stronger baseline designed specifically to attack deep learning models for text classification. The final column, "RL Attack" corresponds to our proposed attack method. We observe that our proposed method outperforms the other baselines, i.e., producing lower accuracy scores, demonstrating its effectiveness in generating high-quality adversarial examples as the attacked texts. The proposed RL attack is able to consistently lead to better attacking outcomes across target models and three datasets. Such results verify that our method works effectively in the black-box setting.

Notably, the white box attacks exhibit a greater decline in accuracy compared to the black box attacks. We plot their difference in Figure 4. This discrepancy arises because the white box attack leverages direct access to the model's parameters, allowing for a more targeted and potent attack.

D. Results on Model Defense

Upon successfully attacking the model, it becomes crucial to employ an appropriate method to address and defend against the generated adversarial examples. To evaluate the efficacy of our proposed defense model, we conduct experiments on

[3] https://github.com/codertimo/BERT-pytorch

[4] https://github.com/facebookresearch/fairseq

[5] https://github.com/zihangdai/xlnet/

TABLE III: Results of white box attacks. **Clean**: the performance on original clean data. **Att.**: the performance on attacked texts using the proposed attack method. ∇: the percentage of performance drop after attacking. The higher the negative percentage, the more effective our attacking approach is. **Def.**: the performance after defending against the attacked texts. The closer (or even higher) the defense performance to the clean data, the more effective the proposed defense method is.

Dataset	Instagram				Vine				Twitter			
	Clean	Att.	∇	Def.	Clean	Att.	∇	Def.	Clean	Att.	∇	Def.
LSTM	82.7	41.0	-50.4%	83.7	77.8	48.4	-37.8%	79.8	87.0	38.0	-56.3%	87.5
BERT	83.7	14.0	-83.3%	84.3	77.8	60.8	-21.9%	72.6	88.1	28.1	-68.1%	89.3
RoBERTa	85.9	41.0	-52.3%	81.6	79.8	62.0	-22.3%	80.4	90.6	38.1	-42.1%	88.1
XLNet	77.8	22.7	-70.8%	78.3	79.3	61.8	-22.1%	78.3	89.3	41.8	53.2%	88.1

TABLE IV: Results of black-box attacks in accuracy. Lower accuracy values indicate a higher effectiveness of the attacking method in reducing the performance of the model. The best results are highlighted in **bold**.

Dataset	Model	Clean	Permutation	Adj. Swap	Noun Swap	DeepWordBug	RL Attack (Ours)
Instagram	BERT	83.7	78.9	82.1	83.2	75.1	**67.0**
	RoBERTa	85.9	50.1	51.1	50.8	50.2	**49.1**
	XLNet	77.8	49.2	51.8	49.6	48.5	**48.1**
Vine	BERT	77.8	67.5	75.7	74.7	48.2	**39.6**
	RoBERTa	79.8	64.9	66.8	65.7	65.2	**64.9**
	XLNet	79.3	64.9	65.8	66.9	**64.8**	**64.8**
Twitter	BERT	88.1	85.0	86.2	85.6	67.2	**46.0**
	RoBERTa	90.6	82.5	86.8	87.5	60.3	**55.0**
	XLNet	89.3	87.5	86.8	87.5	70.3	**54.0**

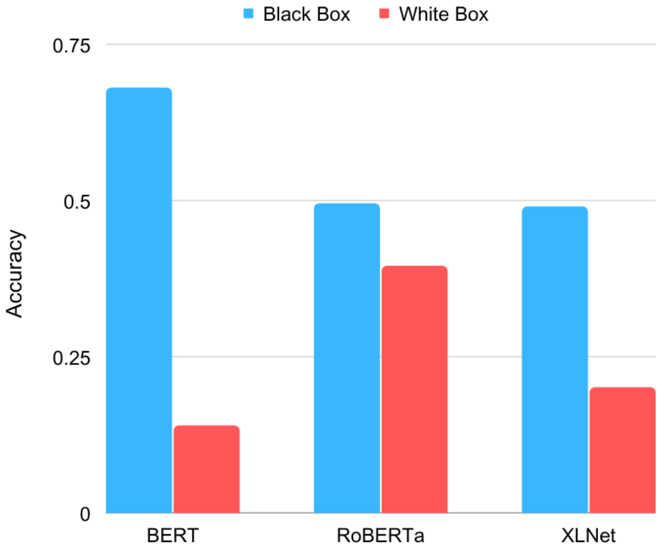

Fig. 4: Performance comparisons between white-box attack and black-box attack.

three datasets and utilize four language models. By leveraging the changing process involved in generating adversarial examples, we train our defense model using these examples. The objective is to restore the model's accuracy to its original level, or even improve upon it. The results are exhibited in Table III. It can be found that by applying the the proposed defense method, the classification performance in "Def." column is maintained as high as the "Clean" column at the same level. The defense capability is held across target models and three datasets. Such outcomes imply that our defense model enhances the robustness of the model and effectively defends against adversarial examples, thereby increasing its resilience.

E. Analysis of Replacement Rate

In this section, we investigate the effect of the replacement rate, i.e., the portion of words being replaced in the input sentence. Figure 5 depicts the relationship between the replacement rate and the model accuracy, with the x-axis representing the replacement rate and the y-axis representing the model accuracy. In this experiment, we focused on the Instagram dataset and utilized LSTM as the proxy model under the setting of black-box attacks. We can observe that when the replacement rate was set as 0.01, the model experiences minimal attacks. However, as the replacement rate increases to 0.05, the model becomes significantly more susceptible to attacks. Despite the increase in the replacing rate of changing similar words, the decrease in model accuracy does not exhibit a proportional relationship with the replacement rate. Remarkably, we find that it was possible to successfully attack the model by only modifying a small fraction of the words. In other words, even minor changes in the original sentence can lead to a substantial impact on the model's accuracy.

VI. Conclusions and Discussion

In this paper, we begin by identifying and highlighting the limitations of existing approaches in capturing the evolving nature of disinformation, where perpetrators modify toxic content to evade detection and censorship. To address this challenge,

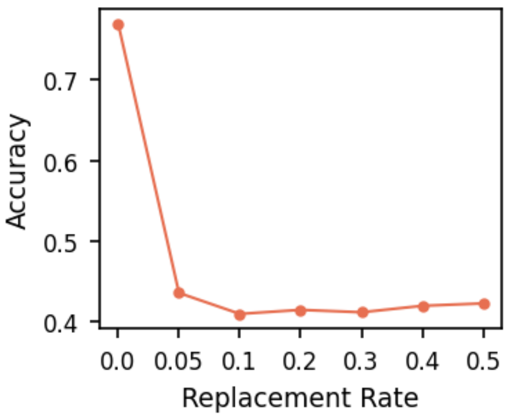

Fig. 5: Accuracy (y-axis) vs. replacement rate (x-axis) after performing the adversarial attack.

we propose a novel framework called ANTI-Disinformation, which leverages reinforcement learning techniques as adversarial attacks. Additionally, we introduce a defense model to bolster the resilience of the detection system against such attacks. To evaluate the effectiveness of our approach, we conduct extensive experiments on a diverse dataset collected from multiple social media platforms. The results demonstrate the efficacy of ANTI-Disinformation, as it significantly degrades the accuracy of the targeted models. Furthermore, our proposed defense model successfully restores the attacked samples and restores accuracy levels even beyond the original performance. These findings indicate the potential for a robust disinformation detection system, even when confronted with sophisticated adversarial attacks. By enhancing the robustness of the detection system, we aim to contribute to the creation of safer online environments and protect online users from the detrimental effects of disinformation.

Limitations. While our framework has shown promising results on diverse datasets, it is important to acknowledge certain limitations. One potential limitation is the generalizability of the proposed framework. Although the results demonstrate its effectiveness in detecting disinformation across multiple datasets, it is worth considering that malicious users may exhibit variations in behavior and tactics across different online platforms and cultural contexts. Additionally, it is important to note that all the datasets used in this paper are in English, and the modifications made by our agent may not be suitable for other languages. Another limitation to be aware of is that this paper primarily serves an academic purpose. The detection results are based on existing datasets rather than real-world scenarios. Consequently, the possibility of erroneously flagging non-toxic content or failing to detect toxic content remains.

Ethics Statement. Understanding your enemies is the key to effectively combating them. While adversarial attacks do involve manipulating toxic content to evade detection, it is important to clarify the stance taken in this work. In our approach, adversarial attacks are utilized solely as a means to enhance the robustness of the model. We emphasize the responsible use of adversarial techniques and strongly discourage their abuse for malicious purposes. Studying adversarial attacks provides valuable insights into the tactics and techniques employed by malicious users. This knowledge allows us to develop more effective defense mechanisms and detection systems. It is worth reiterating that our intention is in no way to endorse or encourage the application of adversarial attacks for nefarious purposes. Rather, our objective is to leverage these techniques to foster a safer online environment by augmenting the capabilities of the detection system.

Acknowledgment

This work is supported by the National Science and Technology Council (NSTC) of Taiwan under grants 110-2221-E-006-136-MY3, 112-2628-E-006-012-MY3, 111-2221-E-006-001, and 111-2634-F-002-022.

References

[1] Monirah A. Al-Ajlan and Mourad Ykhlef. Optimized twitter cyberbullying detection based on deep learning. In *2018 21st Saudi Computer Society National Computer Conference (NCC)*, pages 1–5, 2018.

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

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[4] Marina Azzimonti and Marcos Fernandes. Social media networks, fake news, and polarization. *European Journal of Political Economy*, 76:102256, 2023.

[5] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4312–4321. International Joint Conferences on Artificial Intelligence Organization, 8 2021.

[6] Vijay Banerjee, Jui Telavane, Pooja Gaikwad, and Pallavi Vartak. Detection of cyberbullying using deep neural network. In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pages 604–607. IEEE, 2019.

[7] Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. Bad characters: Imperceptible nlp attacks. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1987–2004. IEEE, 2022.

[8] Alican Bozyiğit, Semih Utku, and Efendi Nasibov. Cyberbullying detection: Utilizing social media features. *Expert Systems with Applications*, 179:115001, 2021.

[9] Meeyoung Cha, Wei Gao, and Cheng-Te Li. Detecting fake news in social media: An asia-pacific perspective. *Commun. ACM*, 63(4), mar––71 2020.

[10] Hsin-Yu Chen and Cheng-Te Li. HENIN: Learning heterogeneous neural interaction networks for explainable cyberbullying detection on social media. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2543–2552, Online, November 2020. Association for Computational Linguistics.

[11] Paweł Debski, Adrianna Boroń, Natalia Kapuśniak, Małgorzata Debska-Janus, Magdalena Piegza, and Piotr Gorczyca. Conspiratorial beliefs about covid-19 pandemic-can they pose a mental health risk? the relationship between conspiracy thinking and the symptoms of anxiety and depression among adult poles. *Frontiers in psychiatry*, 13:870128, 2022.

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

[13] Yingtong Dou, Kai Shu, Congying Xia, Philip S Yu, and Lichao Sun. User preference-aware fake news detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2051–2055, 2021.

[14] Helen L Fisher, Terrie E Moffitt, Renate M Houts, Daniel W Belsky, Louise Arseneault, and Avshalom Caspi. Bullying victimisation and risk of self harm in early adolescence: longitudinal cohort study. *Bmj*, 344:e2683, 2012.

[15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[16] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. *CoRR*, abs/1801.04354, 2018.

[17] Siddhant Garg and Goutham Ramakrishnan. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online, November 2020. Association for Computational Linguistics.

[18] John Hani, Nashaat Mohamed, Mostafa Ahmed, Zeyad Emad, Eslam Amer, and Mohammed Ammar. Social media cyberbullying detection using machine learning. *International Journal of Advanced Computer Science and Applications*, 10(5), 2019.

[19] Md. Tarek Hasan, Md. Al Emran Hossain, Md. Saddam Hossain Mukta, Arifa Akter, Mohiuddin Ahmed, and Salekul Islam. A review on deep-learning-based cyberbullying detection. *Future Internet*, 15(5), 2023.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[21] Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. Analyzing labeled cyberbullying incidents on the instagram social network. pages 49–66, 2015.

[22] Celestine Iwendi, Gautam Srivastava, Suleman Khan, and Praveen Kumar Reddy Maddikunta. Cyberbullying detection solutions based on deep learning architectures. *Multimedia Systems*, pages 1–14, 2020.

[23] Seunghyun Kim, Afsaneh Razi, Gianluca Stringhini, Pamela J. Wisniewski, and Munmun De Choudhury. A human-centered systematic literature review of cyberbullying detection algorithms. *Proc. ACM Hum.-Comput. Interact.*, 5(CSCW2), oct 2021.

[24] Suzet Tanya Lereya, Catherine Winsper, Jon Heron, Glyn Lewis, David Gunnell, Helen L. Fisher, and Dieter Wolke. Being bullied during childhood and the prospective pathways to self-harm in late adolescence. *Journal of the American Academy of Child & Adolescent Psychiatry*, 52(6):608–618.e2, 2013.

[25] Long-Ji Lin. *Reinforcement learning for robots using neural networks*. Carnegie Mellon University, 1992.

[26] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[27] Yi-Ju Lu and Cheng-Te Li. GCAN: Graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514, Online, July 2020. Association for Computational Linguistics.

[28] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 3818–3824. AAAI Press, 2016.

[29] Tomas Mikolov, Ilya Sutskever, Kai Chen, G.s Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 10 2013.

[30] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *NIPS Deep Learning Workshop*, 2013.

[31] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR*, abs/1602.02697, 2016.

[32] Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy, July 2019. Association for Computational Linguistics.

[33] Rahat Ibn Rafiq, Homa Hosseinmardi, Richard Han, Qin Lv, Shivakant Mishra, and Sabrina Arredondo Mattson. Careful what you share in six seconds: Detecting cyberbullying instances in vine. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ASONAM '15, pages 617–622. ACM, 2015.

[34] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019.

[35] Kai Shu, Xinyi Zhou, Suhang Wang, Reza Zafarani, and Huan Liu. The role of user profiles for fake news detection. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '19, page 436–439, New York, NY, USA, 2020. Association for Computing Machinery.

[36] Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. It's morphin' time! Combating linguistic discrimination with inflectional perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2920–2935, Online, July 2020. Association for Computational Linguistics.

[37] Gerald Tesauro. Temporal difference learning and td-gammon. *Commun. ACM*, 38(3):58–68, 1995.

[38] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *International Conference on Learning Representations (ICLR)*, 2018.

[39] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[40] Peiling Yi and Arkaitz Zubiaga. Cyberbullying detection across social media platforms via platform-aware adversarial encoding. *Proceedings of the International AAAI Conference on Web and Social Media*, 16(1):1430–1434, May 2022.

[41] Ziqi Zhang, David Robinson, and Jonathan Tepper. Detecting hate speech on twitter using a convolution-gru based deep neural network. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web*, pages 745–760, Cham, 2018. Springer International Publishing.

[42] Wei Zou, Shujian Huang, Jun Xie, Xinyu Dai, and Jiajun Chen. A reinforced generation of adversarial examples for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3486–3497, Online, July 2020. Association for Computational Linguistics.

[43] Karolina Zwierzynska, Dieter Wolke, and Tanya S Lereya. Peer victimization in childhood and internalizing problems in adolescence: a prospective longitudinal study. *Journal of abnormal child psychology*, 41:309–323, 2013.