Bridge the Gap Between CV and NLP!
A Gradient-based Textual Adversarial Attack Framework

Anonymous ACL submission

Abstract

Despite the success of recent deep learning techniques, they still perform poorly on adversarial examples with small perturbations. While gradient-based adversarial attack methods are well-explored in the field of computer vision, it is impractical to directly apply them in natural language processing due to the discrete nature of the text. To address the problem, we propose a unified framework to extend the existing gradient-based method to craft textual adversarial samples. In this framework, gradient-based continuous perturbations are added to the embedding layer and amplified in the forward propagation process. Then the final perturbed latent representations are decoded with a mask language model head to obtain potential adversarial samples. In this paper, we instantiate our framework with an attack algorithm named **T**extual **P**rojected **G**radient **D**escent (**T-PGD**). We conduct comprehensive experiments to evaluate our framework by performing transfer black-box attacks on BERT, RoBERTa, and ALBERT on three benchmark datasets. Experimental results demonstrate that our method achieves an overall better performance and produces more fluent and grammatical adversarial samples compared to strong baseline methods. All the code and data will be made public.

1 Introduction

Deep learning has achieved great success in various domains, such as computer vision (CV) (He et al., 2016;Chi et al., 2019), natural language processing (NLP) (Vaswani et al., 2017; Devlin et al., 2019), and speech recognition (Chiu et al., 2018; Park et al., 2019). However, the powerful neural networks are still vulnerable to adversarial samples, crafted by adding small and human-imperceptible perturbations to the inputs. (Szegedy et al., 2014; Goodfellow et al., 2015).

In the field of CV, numerous adversarial attack methods have been proposed to evaluate the robustness of DNNs (Papernot et al., 2016a, Madry et al., 2019), and corresponding defense methods are also well-explored (Papernot et al., 2016c; Ross and Doshi-Velez, 2018). Adversarial attacks on images are defined as an optimization problem of maximizing the loss function of the model on specific samples, which can be approximated by gradient ascent algorithms.

However, the textual adversarial attack is more challenging due to the discrete and non-differentiable nature of the text space. And the methods that directly employ the gradients to craft adversarial samples are not applicable in NLP. Current practices of textual adversarial attacks that employ first-order approximation to find substitute words are less effective for one-off searching and can violate the local linearization assumption (Cheng et al., 2019; Behjati et al., 2019; Xu and Du, 2020).

To bridge this gap, we propose a general framework to adapt the existing gradient-based method to NLP (See Figure 1). We successfully obtain high-quality adversarial samples by conducting a gradient-based search. Specifically, we employ the gradient of the loss function concerning the embeddings of input tokens to make perturbations on

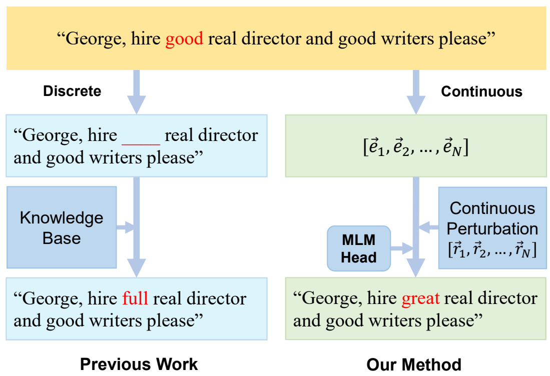

Figure 1: Comparison of our method with previous discrete substitution-based methods .

token embeddings rather than on the original text, thus transforming the problem of searching for adversarial samples from the discrete text space to the continuous and differentiable embedding space. This provides the basis for applying gradient-based methods investigated in CV to craft textual adversarial samples. In this paper, we adapt PGD (Madry et al., 2019) algorithm within our framework to perform textual adversarial attacks, denoted as **T-PGD**. We iteratively generate small perturbations following the gradient information and add them to the embedding layer. The forward propagation process will amplify the perturbations(Goodfellow et al., 2015).

Then we need to transform the perturbed latent representations back to the discrete text. Although there exist some works exploring the feasibility of directly perturbing token embeddings (Sato et al. (2018); Cheng et al. (2019); Behjati et al. (2019)), they simply obtain candidate words using the first-order approximation of the gradient and break the local linearization hypothesis. However, recent work finds that the mask language modeling (MLM) head can reconstruct input sentences from their hidden states with high accuracy, even after models have been fine-tuned on specific tasks (Kao et al., 2021). Inspired by this, we employ an MLM head to decode the perturbed latent representations. With the extensive linguistic knowledge of MLM-head, the coherence and grammaticality of adversarial samples can be guaranteed.

We conduct comprehensive experiments to evaluate the effectiveness of our method by performing transfer black-box adversarial attacks, where only the final decisions of victim models are accessible, against three victim models on three benchmark datasets. We use a local pre-trained language model to construct potential adversarial samples and then query the victim models for decisions. Experimental results demonstrate the effectiveness of our framework and T-PGD algorithm. Specifically, T-PGD significantly outperforms all baseline methods in terms of attack success rate and produces more fluent and grammatical adversarial examples.

To summarize, the main contributions of this paper are as follows:

- We propose a general gradient-based textual adversarial attack framework based on continuous perturbations, bridging the gap between CV and NLP on the study of adversarial attacks. Common gradient-based attack methods in CV can be easily adapted to NLP within our framework.
- We propose a novel adversarial attack method called T-PGD within our framework. We employ a local model to construct adversarial samples by iteratively adding perturbations to tokens' embeddings, and accumulating these small perturbations to search for potential adversarial samples.
- We successfully handle the challenge of black-box attack where only the decisions of models are accessible, which is rarely investigated in NLP.

2 Related Work

2.1 Adversarial Attack in CV

In the field of computer vision, adding a small amount of perturbations to input images to mislead the classifier is possible (Szegedy et al., 2014). Based on this observation, various adversarial attack methods have been explored. FGSM (Goodfellow et al., 2015) crafts adversarial samples using the gradient of the model's loss function to the input images. BIM (Kurakin et al., 2017) straightforwardly extends FGSM, iteratively applying adversarial perturbations multiple times with a smaller step size. MIM (Dong et al., 2018) exploits momentum when updating inputs, obtaining adversary samples with superior quality. PGD (Madry et al., 2019) employs uniform random noise as initialization. Both MIM and PGD are variants of BIM.

2.2 Adversarial Attack in NLP

Existing textual attack models can be roughly categorized into white-box and black-box attack models according to the accessibility to the victim models.

White-box attack models, also known as gradient-based attack models, assume that the attacker has full knowledge of the victim models, including model structures and all parameters. There are few application scenarios of white-box attack in real-world situations, so most white-box attack models are explored to reveal the weakness of victim models, including universal adversarial triggers (Wallace et al., 2019), fast gradient sign inspired methods (Ebrahimi et al., 2018; Papernot et al., 2016b).

Black-box attack models can be further divided into two different attack settings, i.e. score-based and decision-based. The first one assumes the attacker can obtain the decisions and corresponding confidence scores from victim models. Most research works on black-box attacks focus on this setting, exploring different word substitution methods and search algorithms to reduce the victim models' confidence scores. The word substitution methods mainly focus on word embedding similarity (Jin et al., 2020), WordNet synonyms (Ren et al., 2019), HowNet synonyms (Zang et al., 2020), and Masked Language Model (Li et al., 2020). The search algorithms involve greedy search algorithm (Ren et al., 2019; Jin et al., 2020), genetic algorithm (Alzantot et al., 2018), and particle swarm optimization (Zang et al., 2020). The other attack setting assumes the attackers can only obtain decisions from victim models, which is more challenging and less studied. Maheshwary et al. (2021) first substitutes some words in the input sentences to flip the labels and then conducts a search based on a genetic algorithm, expecting to find the most semantic preserved adversarial samples. Chen et al. (2021) propose a learnable attack agent trained by imitation learning to perform a decision-based attack. There also exist some works exploring sentence-level transformation, including syntax (Iyyer et al., 2018) and text style (Qi et al., 2021), to launch attack.

Note that although we apply gradient-based methods, the gradients we employ to generate the perturbations are obtained from the local model rather than the victim model. We only have access to the decisions of victim models. Therefore, we consider our method as a decision-based black-box attack.

3 Framework

In this section, we first present an overview of our framework, and next, we will give the details of how to add continuous perturbations and reconstruct the text.

3.1 Overview

Next, we present an overview of our gradient-based textual adversarial attack framework under the encoder-decoder architecture (See Figure 2).

Specifically, a local language model is applied to encode each discrete text instance into continuous token embeddings with gradient-based perturbations, and then the added perturbations may be amplified through the forward propagation process. The final perturbed latent representations are decoded with an MLM-head to generate candidate adversarial samples.

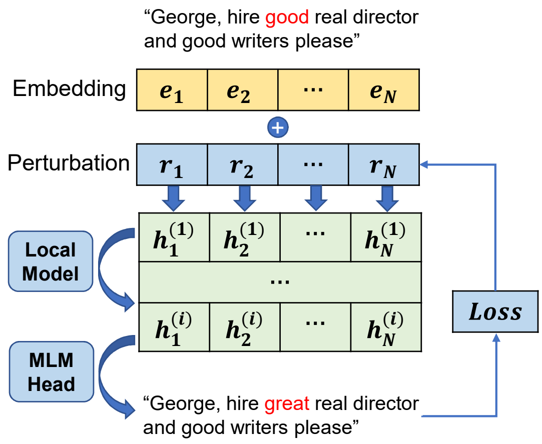

Figure 2: Overview of our framework. Continuous perturbations (r_i) are calculated as gradients of the loss function with respect to token embeddings. The MLM head is employed to decode the perturbed hidden states to obtain potential adversarial samples.

Intuitively, with the help of our proposed framework, it is easier to adapt various gradient-based adversarial attach methods in CV for textual adversarial samples generation. Here, in this paper, we take account of PGD (Madry et al., 2019) to obtain gradient-based perturbations for generation, which will be illustrated in detail in Section 4.

3.2 Latent-space Perturbation

Previous work has shown that the latent representations of transformer-based pre-trained language models are effective in providing semantic and syntactic features (Clark et al., 2019;Jawahar et al., 2019), and thus we adopt a local pre-trained language model to serve as the encoder of our framework.

For each text input, we first calculate the task-specific loss in the forward propagation process, and then perform backward propagation to obtain the gradients of the loss with respect to the token embeddings of the input text. The generated gradients are viewed as the information for updating the perturbations added to the token embeddings, which can be obtained by solving an optimization problem as follows,

$$\delta = \underset{\delta:\|\delta\|_2 \leq \varepsilon}{\arg\max} \mathcal{L}\left(E + \delta, y; \theta\right), \qquad (1)$$

where δ is the perturbation, E stands for the embeddings of input tokens, y is the golden label, θ

denotes current parameters of our local model, and $\mathcal{L}(\cdot)$ is the loss function.

The closed-form solution to the optimization problem is hard to directly obtain (Goodfellow et al., 2015), which is thus relaxed to obtain an approximate solution. For example, various methods in CV usually linearize the loss function with gradient information to approximate the perturbations δ (Goodfellow et al., 2015; Kurakin et al., 2017; Madry et al., 2019).

In NLP, most existing gradient-based methods commonly employ first-order approximation to obtain substitution words (Cheng et al., 2019; Behjati et al., 2019; Xu and Du, 2020). However, these one-off approaches may result in large step size perturbations, violating the hypothesis of local linearization (See Figure 3). To ensure the local linearization hypothesis, we consider adjusting the continuous perturbations added to the token embeddings with a minor change at each step, and then iteratively update the token embeddings of the input instance with the perturbations until generating a meaningful adversarial sample for attacking.

3.3 Reconstruction

By means of continuous perturbations, we need to reconstruct the meaningful adversarial text from the optimized token embeddings. The MLM-head is observed to be able to reconstruct input sentences from hidden states in middle layers with high accuracy, even after models have been fine-tuned on specific tasks (Kao et al., 2021). Inspired by this, we adopt the MLM-head as the decoder for: 1) MLM-head is capable of interpreting any representation embeddings in the hidden space, which is crucial to search adversarial examples continuously; 2) MLM-head has been fully trained during the pre-trained stage so it acquires linguistic knowledge together with the language model and can reconstruct sentences considering the contextual information.

Without loss of generality, we take an example in Figure 3 to illustrate the discrepancy between the one-off based attack models and our proposed iterative-attack based model. One-off attack models prone to choose the token b to server as the substitute of token a because $\cos(\overrightarrow{at_1}, \overrightarrow{ab}) < \cos(\overrightarrow{at_1}, \overrightarrow{ac})$. However, in our framework, the one-step perturbation $\overrightarrow{at_1}$ does not cross the decoding boundary, and thus the decoding results remain unchanged if only using one-step perturbation. Based

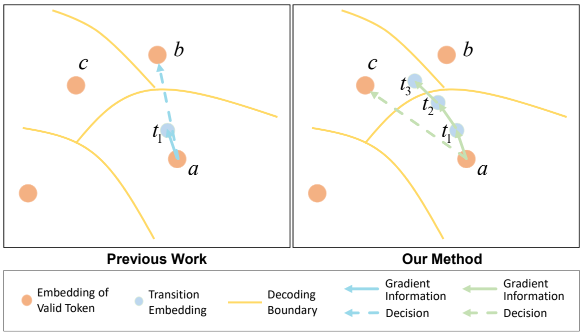

Figure 3: The process of searching for the substitute token of the original instance a in the hidden space. In this case, the one-off attack models are prone to select token b after one-step perturbation (left), while our iterative perturbation based method is more likely to find the optimal solution token c (right).

on the iterative search, the perturbations can be accumulated to the extent to cross the decision boundary and reach the transition point t_3, which will be decoded as the optimal solution c. Then a is replaced by c to obtain the adversarial sample to query the victim model for its decision. If this adversarial sample fails to fool the victim model, we start the next searching iteration from the current perturbed token embedding, i.e. t_3 in Figure 3, but not from the embedding of the decoded token c. By exploiting virtual embeddings as transition points, this iterative attack framework can preserve accumulated gradient information and avoid breaking local linearization assumptions.

4 Method

We denote each sample as a pair of instance, i.e., $(x \in \mathcal{X}, y \in \mathcal{Y})$, where x denotes the input text, y denotes its corresponding label. In particular, the hidden state of x is regarded as $\vec{h}$ and the neural network is implied by a mapping function f, which consists of three components, i.e., f_0, f_1 and f_2, holding:

$$f(x) = f_2(f_1(f_0(x))), \qquad (2)$$

where f_0 is the embedding layer, f_1 denotes the hidden layers that map embeddings to hidden states of a certain layer, and f_2 denotes the rest of the neural network. Then the forward propagation process can be described as:

$$e = f_0(x), h = f_1(e), y = f_2(h) \qquad (3)$$

4.1 T-PGD Algorithm

We instantiate our framework with PGD (Madry et al., 2019) algorithm, and name our attack model as **T**extual-**PGD** (**T-PGD**). The algorithm flow of T-PGD is shown in Appendix A. To solve the optimization problem in Eq. (1), we iteratively search the optimal solution by adding the gradient-based perturbations to the token embeddings with the following formula:

$$\begin{aligned} g_{adv} &= \nabla_{\delta} \mathcal{L}\left(E, y; \theta\right) \\ \delta_{i+1} &= Proj\left(\delta_i + \alpha g_{adv} / \|g_{adv}\|_F\right), \end{aligned} \quad (4)$$

where g_{adv} is the gradient of the loss with respect to the continuous perturbation δ, α is the step size of δ, and i denotes the current iteration step. $Proj\left(\cdot\right)$ performs a re-initialization when δ reaches beyond the ϵ-neighborhood of the original embedding.

For each sample, we first map it to the token embeddings, where continuous perturbations can be added to. After obtaining the gradient of the loss function with respect to the token embeddings in (i+1)-th iteration, perturbations δ_{i+1} are generated according to Eq. (4) and then added to the token embeddings. Then the perturbations are amplified through the forward propagation process (Goodfellow et al., 2015). Next, the hidden sates with perturbations is decoded for reconstructing the crafted adversarial samples:

$$adv_{i+1} = Dec(h_{i+1}), \quad (5)$$

where adv_{i+1} denotes the adversarial sample obtained in the $i+1$ iteration. We query the victim model only when adv_{i+1} satisfying: (1) it varies from adv_0 to adv_i; (2) it is more similar to the original sentences, compared to previous potential adversarial samples. Here we employ the *USE* score to measure the similarity between sentences. If attack succeeds and *USE*$(adv_{i+1}, x) > T$, where T is a tunable threshold for *USE* score, then adv_{i+1} is considered as the adversarial sample of the original input. For each sample, the maximum iteration of the searching process is pre-defined to avoid the infinite loop problem.

4.2 Heuristic Strategies

4.2.1 Random Masking for Diversity

To enhance the diversity of adversarial samples, we randomly mask one token in each input sentence. Specifically, we tokenize x to a list of tokens, $x_{token} = [x_0, ..., x_i, ..., x_n]$. Then we randomly select i-th index token using the uniform distribution and replace it with a special token *[MASK]*. Next, the MLM-head-based decoder will predict the masked word according to its context, which will diversify the generated adversarial samples with semantically consistent consideration. Then, these processed sentences are embedded into continuous token embeddings as mentioned.

4.2.2 Input Reconstruction Task

Intuitively, the quality of generated adversarial samples is largely affected by the reconstruction accuracy of the MLM-head-based decoder. If failing to recover the original sentence even no perturbations are added, its capacity to generate fluent adversarial samples from perturbed hidden states is limited. To reduce the risk of a catastrophic drop in the quality of adversarial samples generated by continuous perturbation, external constraints on the MLM-head-based decoder should be considered to ensure reconstruction accuracy. Note that the MLM-head has been pre-trained to precisely fill the masked word, which is also fitted to our task. We add an additional loss term to force the added perturbations to minimize the loss of input reconstruction task, which will be optimized simultaneously with the adversarial loss so that the adversarial samples can fool the models with minimal perturbations. Specifically, the loss function is defined with two components:

$$\mathcal{L}\left(E, y; \theta\right) = \mathcal{L}_1\left(E, y; \theta\right) + \beta \mathcal{L}_2\left(E, y; \theta\right), \quad (6)$$

where $\mathcal{L}_1\left(E, y; \theta\right)$ is the original loss of the victim model on specific tasks (e.g. CE loss in sentiment classification), $\mathcal{L}_2\left(E, y; \theta\right)$ is the cross-entropy loss of the input reconstruction task, and β is a weighting constant. Note that we aim to reduce the decoding loss $\mathcal{L}_2$ while increasing $\mathcal{L}\left(E, y; \theta\right)$ along the gradient direction, so β should be negative. Taking two losses into account jointly, we can adjust the perturbation searching target on successfully fooling the victim models with fewer modifications.

4.2.3 Antonym Filtering

Li et al. (2019) reports that semantically opposite words are quite close in their representation embeddings since antonyms usually appear in similar contexts. Therefore, we filter antonyms of original words using WordNet (Fellbaum, 2010) to prevent from crafting invalid adversarial samples.

Dataset	#Class	Train	Test	Avg Len	BERT Acc	RoBERTa Acc	ALBERT Acc
SST-2	2	7K	1.8K	16.5	89.9	94.2	92.8
MNLI	3	433K	10K	31.7	82.8	83.6	82.3
AG's News	4	30K	1.9K	39.3	91.2	94.7	94.2

Table 1: Detailed information of datasets and original accuracy of victim models.

Dataset	Model	BERT				RoBERTa				ALBERT			
		ASR	USE	ΔI	ΔPPL	ASR	USE	ΔI	ΔPPL	ASR	USE	ΔI	ΔPPL
SST-2	PWWS	75.12	0.83	0.29	533.86	77.03	0.82	0.41	837.7	72.00	0.82	0.40	531.85
	Textfooler	85.36	0.81	0.33	480.14	87.28	0.82	0.32	924.09	72.68	0.79	0.25	706.83
	PSO	85.60	0.75	**0.10**	501.12	85.50	0.74	**0.09**	479.27	91.49	0.77	**0.14**	397.77
	BERT-Attack	90.36	0.81	0.51	378.79	93.53	0.88	0.45	387.95	92.43	0.79	0.81	348.37
	TPGD	**97.00**	**0.92**	0.62	**343.65**	**94.75**	0.89	0.63	**302.70**	**93.59**	**0.90**	0.69	**291.00**
MNLI	PWWS	75.12	0.83	0.34	516.95	71.65	0.84	0.3	715.42	45.88	0.77	4.17	744.49
	Textfooler	72.34	0.83	0.31	780.8	77.27	0.87	0.3	640.21	82.47	0.81	0.31	854.73
	PSO	75.85	0.80	0.11	481.43	76.08	0.80	0.11	411.12	89.41	0.79	0.22	424.48
	BERT-Attack	87.68	0.87	0.55	484.27	91.26	0.89	0.23	604.22	89.65	0.89	0.25	456.31
	TPGD	**93.96**	**0.92**	**-0.95**	**296.82**	**94.55**	**0.91**	**-0.97**	**261.62**	**94.65**	**0.93**	**-0.98**	**259.57**
AG's News	PWWS	65.46	0.83	0.65	**394.28**	54.70	0.84	0.82	491.48	48.53	0.84	4.71	476.81
	Textfooler	88.71	0.81	0.61	454.13	78.25	0.82	0.59	372.9	73.21	0.84	1.32	367.66
	PSO	66.22	0.79	0.25	539.25	64.63	0.79	0.29	508.76	76.37	0.84	0.15	282.73
	BERT-Attack	81.25	0.84	0.48	431.47	82.58	0.85	0.07	307.74	91.28	0.81	2.52	289.52
	TPGD	**94.47**	0.75	**-0.05**	625.08	**99.30**	**0.87**	**-1.42**	**285.12**	**99.24**	**0.87**	**-1.14**	**260.64**

Table 2: The results of automatic evaluation metrics on SST-2, MNLI, and AG's News. ASR denotes the attack success rate, *USE* denotes the similarity of original and adversarial samples, ΔI and ΔPPL denotes the increase of grammar errors and perplexity. We conduct Student's t-tests to measure the significant difference. **Bold** numbers indicate significant advantage with p-value 0.05 as the threshold and underline numbers mean no significant difference.

5 Experiments

We conduct comprehensive experiments to evaluate our general framework and T-PGD algorithm on the task of sentiment analysis, natural language inference, and news classification. We consider both automatic and human evaluations to analyze our method in terms of attack performance, semantic consistency, and grammaticality.

5.1 Datasets and Victim Models

For sentiment analysis, we choose SST-2 (Socher et al., 2013), a binary sentiment classification benchmark dataset. For natural language inference, we choose the mismatched MNLI (Williams et al., 2018) dataset. For news classification, we choose AG's News (Zhang et al., 2015) multi-classification datasets with four categories: World, Sports, Business, and Science/Technology. We randomly sample 1,000 samples that models can classify correctly from the test set and perform adversarial attacks on those samples.

For each dataset, we evaluate T-PGD by attacking BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2020) with a local fine-tuned BERT model to generate potential adversarial samples. Details of datasets and the original accuracy of victim models are listed in Table 1.

5.2 Experimental Setting

Baseline Methods. We select four strong score-based models as baselines: (1) PWWS (Ren et al., 2019); (2) Textfooler (Jin et al., 2020); (3) PSO (Zang et al., 2020); (4) BERT-Attack (Li et al., 2020). Note that all of them require the confidence scores of victim models, while our model only assumes the decisions are available, which is more challenging.

Evaluation Metrics. We evaluate our method considering the attack success rate and adversarial samples quality. (1) Attack Success Rate (**ASR**) is the proportion of adversarial samples that successfully mislead victim models' predictions. (2) Quality of adversarial samples is evaluated by two automatic metrics and human evaluation, including their semantic consistency, grammaticality, and fluency. Specifically, we use Universal Sentence Encoder (Cer et al., 2018) to compute the semantic similarity between the original text and the corresponding adversarial sample, Language-Tool[1] to calculate the increase of grammar errors, and GPT-2 (Radford et al., 2019) to compute the perplexity of adversarial samples as a measure of fluency. We also conduct a human evaluation to measure the validity and quality of adversarial samples.

5.3 Experimental Results

The results of automatic evaluation metrics are listed in Table 2.

Attack Performance. T-PGD consistently outperforms the strong score-based attack methods considering the attack success rate. We attribute the success of our attack method to the more effective searching process following the guidance of the gradient information.

Adversarial Sample Quality. We observe that the quality of the adversarial samples generated by T-PGD increases with the text length. Our adversarial samples yield overall higher *USE* scores than baseline models. Although our method's grammatical performance is not optimal on SST-2 that mostly contains shorter text (See Table 1), the adversarial samples crafted by our method on MNLI and AG's News have the fewest grammatical errors and the lowest perplexity, since the embedding space of longer text is broader and has a better optimal solution.

5.4 Human Evaluations

To further study the quality and validity of adversarial samples, we randomly selected 100 original SST-2 sentences and 100 adversarial samples from the SOTA baseline BERT-Attack and T-PGD respectively for human evaluation. Following (Li et al., 2020), we shuffle the 300 samples and ask 3 independent human judges to evaluate the quality (300 samples per person). For semantic consistency evaluation, we ask humans to predict the labels of mixed texts. For grammar and fluency, human judges score from 1 to 5 on the above examples. All annotators have no knowledge about the source of text, and all their evaluation results are averaged (shown in Table 3).

Semantic Consistency. Since human judges have high accuracy on the original text, the prediction results on texts can be regarded as the ground truth labels.

[1] https://github.com/jxmorris12/language_tool_python

Source	Accuracy	Grammar & Fluency
Original	0.92	4.63
BERT-Attack	0.48	3.41
T-PGD	0.68	3.52

Table 3: Human evaluation on SST-2 in terms of prediction accuracy, grammar correctness, and fluency.

Model	T-PGD		random	
	ASR	*USE*	ASR	*USE*
BERT	97.00	0.92	47.48	0.79
RoBERTa	94.75	0.89	56.59	0.79
ALBERT	93.59	0.90	51.36	0.79

Table 4: Ablation results of gradient information on SST-2. *Random* corresponds to adding random perturbations to the embeddings.

Therefore, human accuracy can be a criterion for semantic consistency between original sentences and adversarial ones. From the results, human judges achieve 0.68 accuracy on adversarial samples crafted by T-PGD, significantly higher than the baseline method. This result verifies that the adversarial samples crafted by T-PGD have a better semantic consistency.

Grammar and Fluency. We can also conclude from Table 3 that adversarial samples crafted by T-PGD have better quality compared to the baseline method considering the grammar and fluency, evaluated by human annotators. However, both BERT-Attack and T-PGD suffer a decline in grammatical correctness and fluency of adversarial text, leaving room for improvement in future research.

6 Further Analysis

6.1 Importance of Gradient Information

T-PGD employs the gradient of the loss function to approximate the perturbations. To verify the effectiveness of the gradient information, we conduct an ablation experiment on SST-2 by adding only random perturbations in the embedding space without exploiting the gradient information. In detail, we generate a Gaussian noise with the same mean and variance as the random perturbations. The results in Table 4 demonstrate the importance of exploiting gradient directions in the perturbation generation.

6.2 Importance of Reconstruction Task

We show the importance of adding a reconstruction loss ($\mathcal{L}_2$ in Eq.(6)) for generating more accurate reconstructions. We conduct an ablation study

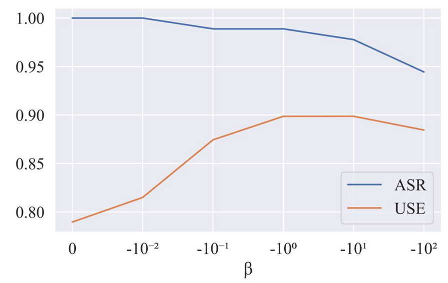

Figure 4: The curve of ASR and *USE* on SST-2 with β changing.

on SST-2. The results are shown in Table 5. On all three victim models, the attack performances (ASR) improve while the quality of adversarial samples deteriorates, with *USE* score decreasing and grammar errors and perplexity increasing. This validates our claim that in the absence of reconstruction loss, the adversarial samples may mislead model predictions by breaking the semantics of the original text, leading to invalid adversarial attacks. We further tune β to study the trend of ASR and *USE* score. Results on BERT are shown in Figure 4. We observe that as the absolute value of β increases, ASR continues to decline while *USE* score stops growing.

Victim	T-PGD				β=0			
	ASR	USE	ΔI	PPL	ASR	USE	ΔI	PPL
BERT	97.00	0.92	0.62	343.65	100	0.79	1.45	875.64
RoBERTa	94.75	0.89	0.63	302.70	100	0.84	1.36	466.56
ALBERT	93.59	0.90	0.69	291.00	100	0.83	1.50	693.39

Table 5: Ablation results on the reconstruction loss. *β=0* denotes the setting without the reconstruction loss.

6.3 Transferability

We investigate the transferability of adversarial examples. We sample 1,000 samples from SST-2 and craft adversarial samples by T-PGD and baseline methods by attacking BERT. Then we test the attack success rate of these adversarial samples on RoBERTa to evaluate the transferability of adversarial samples. As seen in Table 6, adversarial samples crafted by T-PGD achieves the best transferability performance.

Method	PWWS	Textfooler	PSO	BERT-Attack	TPGD
RoBERTa	28.21	18.00	44.73	11.02	**45.29**

Table 6: The ASR on SST-2 of attacking RoBERTa using adversarial samples crafted on BERT.

6.4 Adversarial Training

We explore to enhance models' robustness against adversarial attacks through adversarial training on SST-2 with BERT. Specifically, we first generate adversarial samples using the original training dataset. Then we fine-tune the BERT model using the training dataset augmented with generated adversarial samples. We evaluate the model's original accuracy on the test set and robustness against different adversarial attack methods. As seen in Table 7, the model shows generally better robustness through adversarial training. Besides, the accuracy on the test set is also improved from 89.90 to 90.48, which is different from previous textual adversarial attacks where accuracy is sacrificed for robustness (Ren et al., 2019; Zang et al., 2020).

Ori Acc	89.90%				
Adv.T Acc	**90.48%**				
Method	PWWS	Textfooler	PSO	BERT-Attack	T-PGD
Ori ASR	69.94	**86.38**	82.03	86.55	92.22
Adv.T ASR	**66.78**	87.41	**73.34**	**84.84**	**83.78**

Table 7: Results of adversarial training. *Adv.T* denotes the adversarial training paradigm.

7 Conclusion and Future Work

In this paper, we propose a general framework to adapt gradient-based adversarial attack methods investigated in CV to NLP. In our framework, the problem of searching textual adversarial samples is transformed from the discrete text space to the embedding layer, where continuous gradient-based perturbations can be directly added to. The perturbations will be amplified in the forward propagation process. Then an MLM-head is employed to decode the perturbed latent representations. We instantiate our framework with T-PGD to perform a decision-based black-box attack. We conduct extensive experiments to evaluate our framework and T-PGD algorithm. Experimental results show the superiority of our method in terms of attack performance and adversarial samples quality.

In the future, we will adopt other gradient-based methods in CV with our framework and explore to improve models' robustness through adversarial training. Besides, we find that our framework is quite general and can be employed to bridge the gap between CV and NLP in many fields like backdoor learning, membership inference, and counterfactual samples generation. We will further explore in this direction.

Ethical Consideration

In this section, we discuss the potential broader impact and ethical considerations of our paper.

Intended Use. In this paper, we design a general framework to adapt existing gradient-based methods in CV to NLP, and further, propose a decision-based textual attack method with impressive performance. Our motivations are twofold. First, we attempt to introduce adversarial attack methods of CV to NLP, since image attack methods have been well-explored and proved to be effective, therefore helping these two fields better share research resources hence accelerating the research process on both sides. Second, we hope to find insights about the interpretability and robustness of current black-box DNNs from our study.

Potential Risk. There is a possibility that our attack methods may be used maliciously to launch adversarial attacks against off-the-shelf commercial systems. However, studies on adversarial attacks are still necessary since it is important for the research community to understand these powerful attack models before defending against these attacks.

Energy Saving. We will public the settings of hyper-parameters of our method, to prevent people from conducting unnecessary tuning and help researchers to quickly reproduce our results. We will also release the checkpoints including all victim models to avoid repeated energy costs.

References

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

Melika Behjati, Seyed-Mohsen Moosavi-Dezfooli, Mahdieh Soleymani Baghshah, and Pascal Frossard. 2019. Universal adversarial attacks on text classifiers. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7345–7349.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Yangyi Chen, Jin Su, and Wei Wei. 2021. Multi-granularity textual adversarial attack with behavior cloning. *arXiv preprint arXiv:2109.04367*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.

Cheng Chi, Shifeng Zhang, Junliang Xing, Zhen Lei, Stan Z. Li, and Xudong Zou. 2019. Selective refinement network for high performance face detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8231–8238.

Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Christiane Fellbaum. 2010. *WordNet*, pages 231–243. Springer Netherlands, Dordrecht.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment.

Wei-Tsung Kao, Tsung-Han Wu, Po-Han Chi, Chun-Cheng Hsieh, and Hung-Yi Lee. 2021. Bert's output layer recognizes all hidden layers? some intriguing phenomena and a simple way to boost bert.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world.

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

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

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2019. Towards deep learning models resistant to adversarial attacks.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021. Generating natural language attacks in a hard label black box setting.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016a. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 372–387.

Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016b. Crafting adversarial input sequences for recurrent neural networks. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 49–54.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016c. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021. Mind the style of text! adversarial and backdoor attacks based on text style transfer. *arXiv preprint arXiv:2110.07139*.

Alec Radford, Jeffrey Wu, and Rewon Child. 2019. Rewon child, david luan, dario amodei, and ilya sutskever. 2019. *Language models are unsupervised multitask learners. OpenAI Blog*, 1(8):9.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Andrew Ross and Finale Doshi-Velez. 2018. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text.

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

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

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

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Jincheng Xu and Qingfeng Du. 2020. Texttricker: Loss-based and gradient-based adversarial attacks on text classification models. *Engineering Applications of Artificial Intelligence*, 92:103641.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, Online. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A T-PGD Algorithm

The algorithm flow of T-PGD is shown in Algorithm 1.

B Trade-off between performance and efficiency

Selection of Step Number. Users can make their trade-offs between ASR and efficiency when using our model. The $MaxStep$ in Algorithm 1 determined the perturbation searching scope in embedding space, which contributes to the attack success rate as well as semantic coherence. Intuitively, extending the searching scope boosts performance but costs more time. To determine the proper value range, we conduct experiments to study the statistic of step numbers when obtaining final adversaries.Results on SST-2 with three models are shown in Figure 5. We can observe that most of the attacks finished before step 30. Therefore, $MaxStep = 50$ is virtually enough for an adequate search, and it can also be adjusted to trade-off time costs and attack success rate.

Algorithm 1 T-PGD

Require: Original input x sampled from $\mathcal{X}$
Ensure: Adversary of x

Randomly mask one word in x
$E_0 = f(x)$
$AdvList$=[]
for $j < MaxIter$ **do**
    for $i < MaxStep$ **do**
        $g_{adv} = \nabla_{\delta} L(E_i, y; \theta_i)$
        $\delta_{i+1} = Proj_{\|\delta\|_F \le \varepsilon}(\delta_i + \alpha g_{adv} / \|g_{adv}\|_F)$
        $E_{i+1} = E_i + \delta_{i+1}$
        $h_{i+1} = f_1(E_{i+1})$
        $Adv_{i+1} = Dec(h_{i+1})$
        $\theta_{i+1} = \theta_i - \eta \cdot g_{adv}$
        if Adv_{i+1} **not in** $AdvList$ **then**
            Append Adv_{i+1} to $AdvList$
            Query victim model with Adv_{i+1}
            if attack succeed **and** *USE(Adv, Ori) > USE_GATE* **and** no antonyms **then**
                return Adv_{i+1}
            end if
        end if
    end for
    $E_0 = E_0 + \frac{1}{\sqrt{N_{E_0}}} Uniform(-\varepsilon, \varepsilon)$
end for

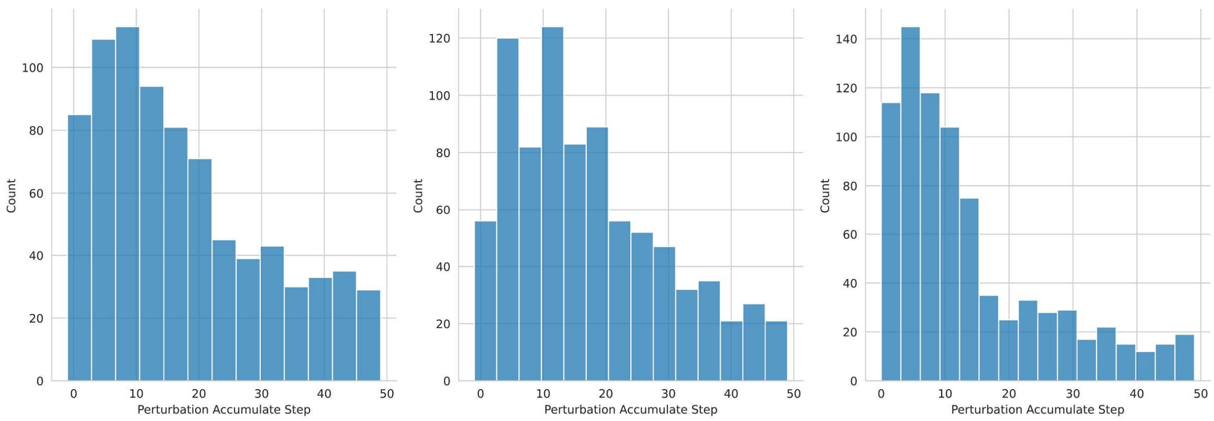

Figure 5: The statistic of perturbation step numbers when successfully obtaining final adversaries. The three pictures represent results on BERT, RoBERTa, and ALBERT in turn.