Appendix A Algorithm 1

Algorithm 1: Substituting Original Words Back

Input : The original text $x = [w_1, ..., w_n]$, the adversarial example $x'_t = [w'_1, ..., w'_n]$, victim model f, the similarity function between text $Sim(\cdot, \cdot)$

Output : The new adversarial example x'_t

while *True* **do**
  $diffs \leftarrow$ Get the positions of all different words between x and x'_t
  for i ***in*** $diffs$ **do**
    $x_{tmp} \leftarrow$ Replace w'_i in x'_t with w_i
    $s_i = Sim(x, x_{tmp})$
    $choices.insert(i, s_i)$
  end
  $flag \leftarrow True$
  Sort $choices$ by s_i in the descending order
  for i ***in*** *choices* **do**
    $x_{tmp} \leftarrow$ Replace w'_i in x'_t with w_i
    if $f(x_{tmp}) \neq f(x)$ **then**
      $x'_t \leftarrow$ Replace w'_i in x'_t with w_i
      $flag \leftarrow False$
      break
    end
  end
  if $flag = True$ **then**
    break
  end
end
return x'_t

Appendix B Algorithm 2

Algorithm 2: HQA-Attack

Input : The original text $x = [w_1, w_2, ..., w_n]$, the victim model f, the number of iteration T, the parameters r and k

Output : The adversarial example x^*

Obtain an example x'_1 by initialization
if $f(x'_1) = f(x)$ **then**
  return
end
$t \leftarrow 1$
while $t \leqslant T$ **do**
  Update x'_t by substituting original words back
  Sampling the updating order I according to Eq. (6)
  for i ***in*** I **do**
    Randomly sample r words from $S(w_i)$ by Eq. (7) to obtain the transition word $\bar{w}_i$
    Randomly sample k words from $S(\bar{w}_i)$ by Eq. (8) to obtain the updating direction $\boldsymbol{u}$
    Attain $\widetilde{w}_i$ by $\boldsymbol{u}$
    Update x'_{t+1} with $\widetilde{w}_i$
  end
  while $f(x'_{t+1}) = f(x)$ **do**
    Update x'_{t+1} with w'_i from x'_t in descending order of $Sim(x, x'_{t+1}(w'_i))$
  end
end
return the adversarial example x^* which has the highest semantic similarity with x

Appendix C Analysis of Computational Complexity and Query Numbers

In the following, we attempt to analyze the computational complexity and the number of queries for the main procedures of HQA-Attack, including Section 4.2 "Substituting original words back" and Section 4.3 "Optimizing the adversarial examples". Given the original example x and its initial adversarial example x', n represents the length of the original sample x.

C.1 Analysis of Substituting Original Words Back

In the best case, x' only has one different word compared with x, so Alg. 1 (Appendix A) only calculates the cosine similarity once and queries the victim model once. In this case, the computational complexity is $\mathcal{O}(1)$ and the number of queries is $\mathcal{O}(1)$.

In the worst case, x' has n different words with x, which means we will conduct n replacement iterations (Alg. 1 Line 1-Line 21). In this case, the number of calculating cosine similarity is $n, n-1, ..., 1$ from the 1-st to the n-th iteration, so the computational complexity is $\frac{(n+1)\times n}{2} = \mathcal{O}(n^2)$. And for the number of queries, there are two boundary cases. For Alg. 1 Line 10-Line 17, we may successfully replace the first word or replace the last word in $choices$. In the former case, the number of queries is $n = \mathcal{O}(n)$. In the latter case, the number of queries in each iteration is $n, n-1, ..., 1$ from the 1-st to the n-th iteration, so the number of queries is $\frac{(n+1)\times n}{2} = \mathcal{O}(n^2)$.

According to the above analysis, for Section 4.2, the overall computational complexity is $\mathcal{O}(n^2)$ and the number of queries is $\mathcal{O}(n^2)$, where n is the length of the original sample x.

C.2 Analysis of Optimizing the Adversarial Examples

Assume that r is the number of synonyms in finding the transition word, k is the number of synonyms in estimating the updating direction, $C = |S(w_i)|$ is the synonym set size for w_i. For the sake of analysis, we assume the synonym set size of each word is the same.

In Section 4.3.1, the number of calculating cosine similarity is $\mathcal{O}(n)$ and the number of queries is 0. Specifically, x'_t may have only one different word with the original sentence x, and also may have n different words, thus the corresponding computational complexity is $\mathcal{O}(n)$, where x'_t is the generated adversarial example through Section 4.2. And this step (Section 4.3.1) does not need to query the victim model.

In Section 4.3.2, the number of calculating cosine similarity is $\mathcal{O}(n(r+k+C))$ and the number of queries is $\mathcal{O}(n(r+C))$. Specifically, In Alg. 2 (Appendix B) Line 9-Line 14, the number of I may be 1 or n. In each iteration (Alg. 2 Line 10-Line 13), for the first sub-step (Alg. 2 Line 10), we randomly select r synonyms to find the transition word, so the computational complexity is $\mathcal{O}(r)$ and the number of queries is $\mathcal{O}(r)$. For the second sub-step (Alg. 2 Line 11), we randomly select k synonyms to obtain the updating direction, so the computational complexity is $\mathcal{O}(k)$ and the number of queries is 0. In the third sub-step (Alg. 2 Line 12), we obtain $\widetilde{w}_i$ from $S(w_i)$, so the computational complexity is $\mathcal{O}(C)$ and the number of queries is $\mathcal{O}(C)$. Furthermore, in the experiments, we set the size of the synonym set is 50, so r, k and C are all integers which are not greater than 50. Then both $\mathcal{O}(n(r+k+C))$ and $\mathcal{O}(n(r+C))$ can be simplified to $\mathcal{O}(n)$.

According to the above analysis, for Section 4.3, the overall computational complexity is $\mathcal{O}(n)$ and the number of queries is $\mathcal{O}(n)$, where n is the length of the original sample x.

Appendix D The Mechanism Analysis of HQA-Attack

We would like to attempt to analyze our method from the perspective of decision boundary. Specifically, HQA-Attack consists of three steps. First, HQA-Attack generates an adversarial example by initialization, which means that the adversarial example is outside the decision boundary associated with the original true label. Second, HQA-Attack deals with the adversarial example by substituting original words back, which means that the adversarial example is getting closer to the decision boundary, thus improving the semantic similarity and reducing the perturbation rate. Third, HQA-Attack further optimizes the adversarial example along the direction that can increase the semantic similarity, which means the adversarial example is further approaching the decision boundary. The first step

Table 1: One adversarial example can cause that different victim models make different predictions or the same prediction.

Adversarial Example	Victim Model	Prediction
Witnesses to confront cali cartel ~~kingpin~~ (**punisher**) thirteen years into their probe, u.s. investigators have assembled a team of smugglers, accountants and associates to testify against colombian cartel ~~kingpin~~(**studs**) gilberto rodriguez orejuela.	BERT WordCNN WordLSTM	Sports World Business
Boys 'cured' with gene therapy gene therapy can cure ~~children~~ (**enfants**) born with a condition that knocks out their natural defences against infection, mounting evidence shows.	BERT WordCNN WordLSTM	Sci/Tech Sci/Tech Sci/Tech

Table 2: Comparison of semantic similarity (Sim) and perturbation rate (Pert) with the budget limit of 1000 when attacking T5 and DeBERTa.

Model	Method	MR		AG		Yahoo		Yelp		IMDB	
		Sim(%)	Pert(%)	Sim(%)	Pert(%)	Sim(%)	Pert(%)	Sim(%)	Pert(%)	Sim(%)	Pert(%)
T5	TextHoaxer	65.7	12.463	67.3	14.702	64.7	7.735	77.5	8.532	83.3	6.477
	LeapAttack	60.0	17.240	65.7	17.697	62.9	9.704	76.3	10.348	83.7	7.087
	HQA-Attack	**73.0**	**11.846**	**76.8**	**11.365**	**72.6**	**6.150**	**84.1**	**7.514**	**87.0**	**6.142**
DeBERTa	TextHoaxer	66.6	12.251	63.3	17.385	66.0	9.360	68.6	13.798	78.1	10.153
	LeapAttack	60.9	17.853	62.7	20.289	64.1	11.221	68.8	15.370	78.7	11.018
	HQA-Attack	**73.8**	**11.460**	**75.0**	**13.019**	**70.9**	**9.029**	**78.9**	**11.749**	**84.3**	**9.963**

determines whether the adversarial example can fool the model and trigger the wrong prediction. The last two steps determine the quality of the adversarial example.

In general, different victim models have different decision boundaries, but they share the same label space. Therefore, one adversarial example can cause that different victim models make different predictions or the same prediction. We list two adversarial examples generated by HQA-Attack with the true label "world" on the AG dataset in Table 1, and the substitute words are indicated by parentheses. According to the results, we can get that the first example shows that different victim models can make different predictions, and the second example shows that different victim models can make the same prediction.

Appendix E Dataset Description

The detailed dataset description is as follows.

- **MR** [8] is a movie review dataset for binary sentiment classification.
- **AG's News** [12] is a news topic classification dataset of four categories.
- **Yahoo** [12] is a question-and-answer topic classification dataset.
- **IMDB** [7] is a movie review dataset for binary sentiment but each review is longer than MR.
- **Yelp** [12] is a binary sentiment classification dataset.
- **SNLI** [1] is a dataset based on Amazon Mechanical Turk collected for natural language inference.
- **MNLI** [10] is one of the largest corpora available for natural language inference.
- **mMNLI** [10] is a variant of the MNLI dataset with mismatched premise and hypotheses pairs.

Appendix F Attack Advanced Victim Models

We take two advanced natural language processing models as victim models: T5 [9] and DeBERTa [4], and set the query budget to 1000. We select 1000 examples from each of sentence-level datasets

(MR, AG and Yahoo) and 500 examples from document-level datasets (Yelp and IMDB). From Table 2, it can be seen that HQA-Attack still outperforms other strong baselines, which further verifies the superiority of HQA-Attack.

Appendix G Parameter Investigation

We investigate the impact of the parameters r and k in HQA-Attack on different text classification datasets. We select the WordLSTM as the victim model and set the query budget to 1000. In terms of parameter r, as shown in Table 3, when the r value is set in the range $[3, 5, 7, 9]$, the performance of HQA-Attack has only a slight fluctuation in semantic similarity and perturbation rate. In terms of parameter k, as shown in Table 4, when the k value is set in the range $[5, 10, 15]$, the results of HQA-Attack remain nearly constant. These results show that the stability and robustness of HQA-Attack are reliable.

Table 3: The adversarial attack performance of HQA-Attack with different r values when attacking WordLSTM on different text classification datasets.

r	MR		AG		Yahoo		Yelp		IMDB	
	Sim(%)	Pert(%)	Sim(%)	Pert(%)	Sim(%)	Pert(%)	Sim(%)	Pert(%)	Sim(%)	Pert(%)
3	73.9	11.486	75.3	11.283	73.9	6.645	86.7	5.786	91.7	3.198
5	74.2	11.341	75.5	11.378	74.1	6.655	86.9	5.833	92.0	3.197
7	74.0	11.723	75.7	11.510	74.2	6.942	86.8	5.990	91.9	3.176
9	74.1	11.697	75.7	11.728	74.2	7.046	86.6	6.032	91.9	3.226

Table 4: The adversarial attack performance of HQA-Attack with different k values when attacking WordLSTM on different text classification datasets.

k	MR		AG		Yahoo		Yelp		IMDB	
	Sim(%)	Pert(%)	Sim(%)	Pert(%)	Sim(%)	Pert(%)	Sim(%)	Pert(%)	Sim(%)	Pert(%)
5	74.2	11.341	75.5	11.378	74.1	6.655	86.9	5.833	92.0	3.197
10	74.2	11.226	75.6	11.359	74.2	6.681	86.7	5.821	91.9	3.170
15	74.1	11.414	75.7	11.487	74.4	6.683	86.8	5.859	91.9	3.142

Appendix H Experiments with BERT-Based Synonyms

Inspired by [6], we modify our method with BERT-based synonyms to attack the WordCNN model. From the results in Table 5 and Table 2 in the main body of the paper, it can be seen that BERT-based synonyms are competitive with counter-fitting word vector based synonyms.

Table 5: Experiments with BERT-based synonyms when attacking WordCNN.

Dataset	MR	AG	Yahoo	Yelp	IMDB
Sim(%)	73.4	81.3	82.6	87.9	92.2
Pert(%)	11.856	10.798	6.071	6.249	3.740

Appendix I The Word Back-Substitution Strategy

For the word back-substitution strategy, HQA-Attack is different from previous methods. Specifically, as shown in Alg. 1 (Appendix A), after replacing one original word successfully (Line 12-Line 13), our method can make the $flag$ = False (Line 14) and break out of the current loop (Line 15), where the current loop is from Line 10 to Line 17. Furthermore, as the $flag$ = False, our method will continue the outer loop and recompute the order, where the outer loop is from Line 1 to Line 21. By replacing the best original word in each iteration, HQA-Attack can make the intermediate generated adversarial example have the highest semantic similarity with the original example.

We conducte experiments to verify the effectiveness of our proposed word back-substitution strategy. Specifically, we use our proposed word back-substitution strategy to replace the original word back-substitution strategy in LeapAttack to attack the WordCNN model. (Original) means the original method, and (Improved) means the method with our proposed word back-substitution strategy. The results in Table 6 show that our proposed word back-substitution strategy can improve the semantic similarity and reduce the perturbation rate successfully.

Table 6: Effectiveness of the word back-substitution strategy.

Method	MR		AG		Yahoo		Yelp		IMDB	
	Sim(%)	Pert(%)	Sim(%)	Pert(%)	Sim(%)	Pert(%)	Sim(%)	Pert(%)	Sim(%)	Pert(%)
LeapAttack (Original)	63.2	14.016	72.0	12.827	74.3	7.842	80.1	8.816	89.7	3.886
LeapAttack (Improved)	65.3	13.809	74.8	12.013	74.6	7.623	81.5	8.120	90.8	3.796

Appendix J Case Study

In Table 7, we list some adversarial examples generated by HQA-Attack on MR, AG and SNLI datasets. Some other examples can refer to Table 1. Take the first adversarial example in Table 7 as an example, just replacing the word "enamored" with "enthralled" can change the prediction from "negative" to "positive", which demonstrates that HQA-Attack can generate a high-quality black-box hard-label adversarial example with only a simple modification.

Table 7: Adversarial examples generated by HQA-Attack in different datasets against the BERT model. The substituted original word is in the strike-through format, and the replacement is the following one.

Adversarial Example	Change of prediction
MR: Davis is so ~~enamored~~ (**enthralled**) of her own creation that she can't see how insufferable the character is.	Negative → Positive
AG: Turkey unlikely to join EU before 2015: commissioner verheugen (afp) afp - turkey is unlikely to join the ~~european~~ (**euro**) union before 2015, EU ~~enlargement~~ (**growth**) commissioner guenter verheugen said in an interview.	World → Business
SNLI: Families with strollers waiting in front of a carousel. [*Premise*] Families have some dogs in ~~front~~ (**fore**) of a carousel. [*Hypothesis*]	Contradiction → Neutral

Appendix K Discussion and Potential Application Scenarios

In the adversarial attack domain, some researchers have a point that the real-world adversarial attacks do not need the complicated pipeline to craft adversarial samples that are so-called high-quality and imperceptible [2, 11]. This point may be reasonable in security, but it is not applicable in data augmentation, evaluation, explainability and so on.

We would like to discuss the detailed differences between the method in [2] and HQA-Attack. The method in [2] mainly focuses on how to fool the model (security), and its main goal is to reconsider the goal of attackers and reformalize the task of textual adversarial attack in security scenarios. But their generated adversarial samples seem not suitable for other tasks like data augmentation. Different from [2], the goal of our work is to generate high-quality adversarial samples to assess and improve the robustness of models. The overall idea includes two steps. (1) Using black-box attacks better simulates the conditions under which an actual attacker might operate, leading to more accurate evaluation of the model robustness. (2) Leveraging the high-quality adversarial samples generated by black-box attacks to improve the model robustness. In summary, our work aims to enhance model performance by creating high-quality adversarial samples examples, and it has overlaps with security in the first step.

Here we list some potential applications of our method. (1) Generating robust training data: Adversarial text examples can be used to generate robust training data. By introducing adversarial perturbations

to clean text data, models can learn to be more robust and generalize better to real-world scenarios. (2) Model robustness evaluation: Adversarial text examples are used to test the robustness of NLP models. By generating inputs that are intentionally designed to confuse or mislead the model, researchers can identify weaknesses in natural language processing algorithms. (3) Defense mechanism development: Adversarial text examples are used to develop and evaluate defense mechanisms for NLP models. These mechanisms aim to make models more resilient to adversarial attacks, ensuring their reliability in real-world applications.

Appendix L Broader Impacts and Limitations

In this paper, we propose a simple yet effective black-box hard-label textual adversarial attack method, namely HQA-Attack.

Broader Impacts. This work is likely to increase the progress of adversarial learning and drive the development of advanced defense mechanisms for various language models. Our work highlights the vulnerability of language models, and there is still no effective defense strategy against this attack. Indeed, there is a possibility that HQA-Attack might be employed improperly to attack some NLP systems. However, we believe that by raising awareness about the model vulnerability, HQA-Attack will encourage the development of defense mechanisms. We hope that people in the NLP community can be responsible for their systems, and HQA-Attack can contribute positively in the system robustness aspect.

Limitations. HQA-Attack does not modify the length of adversarial examples. Different from BERT-based methods [6, 3, 5] which leverage the confidence score and can change the adversarial example length, HQA-Attack only utilizes the predicted label and the synonym set to guide the attack. We believe that generating adversarial examples with variable lengths in the black-box hard-label scenarios is a potential research direction. In addition, we plan to investigate the theoretical underpinnings of HQA-Attack in the future work.

References

[1] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, pages 632–642, 2015.

[2] Yangyi Chen, Hongcheng Gao, Ganqu Cui, Fanchao Qi, Longtao Huang, Zhiyuan Liu, and Maosong Sun. Why should adversarial perturbations be imperceptible? rethink the research paradigm in adversarial NLP. In *EMNLP*, pages 11222–11237, 2022.

[3] Siddhant Garg and Goutham Ramakrishnan. BAE: bert-based adversarial examples for text classification. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *EMNLP*, pages 6174–6181, 2020.

[4] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: decoding-enhanced bert with disentangled attention. In *ICLR*, 2021.

[5] Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. Contextualized perturbation for textual adversarial attack. In *NAACL*, pages 5053–5069, 2021.

[6] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: adversarial attack against BERT using BERT. In *EMNLP*, pages 6193–6202, 2020.

[7] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150, 2011.

[8] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, pages 115–124, 2005.

[9] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67, 2020.

[10] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*, pages 1112–1122, 2018.

[11] Kan Yuan, Di Tang, Xiaojing Liao, XiaoFeng Wang, Xuan Feng, Yi Chen, Menghan Sun, Haoran Lu, and Kehuan Zhang. Stealthy porn: Understanding real-world adversarial images for illicit online promotion. In *IEEE S&P*, pages 952–966, 2019.

[12] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NeurIPS*, pages 649–657, 2015.