Logic Traps in Evaluating Attribution Scores

Anonymous ACL submission

Abstract

Modern deep learning models are notoriously opaque, which has motivated the development of methods for interpreting how deep models predict. This goal is usually approached with attribution method, which assesses the influence of features on model predictions. As an explanation method, the evaluation criteria of attribution methods is how accurately it reflects the actual reasoning process of the model (faithfulness). Meanwhile, since the reasoning process of deep models is inaccessible, researchers design various evaluation methods to demonstrate their arguments. However, some crucial logic traps in these evaluation methods are ignored in most works, causing inaccurate evaluation and unfair comparison. This paper systematically reviews existing methods for evaluating attribution scores and summarizes the logic traps in these methods. We further conduct experiments to demonstrate the existence of each logic trap. Through both theoretical and experimental analysis, we hope to increase attention on the inaccurate evaluation of attribution scores. Moreover, with this paper, we suggest stopping focusing on improving performance under unreliable evaluation systems and starting efforts on reducing the impact of proposed logic traps.

1 Introduction

The opaqueness of deep models has grown in tandem with their power (Doshi-Velez and Kim, 2017), which has motivated efforts to interpret how these black-box models work (Sundararajan et al., 2017; Belinkov and Glass, 2019). Post-hoc explanation aims to explain a trained model and reveal how the model arrives at a decision (Jacovi and Goldberg, 2020; Molnar, 2020). This goal is usually approached with attribution method, which assesses the influence of features on model predictions as shown in Figure 1. Recent years have witnessed an increasing number of attribution methods being developed. For example, Erasure-based method calculate attribution scores by measuring the change of output after removing target features (Li et al., 2016; Feng et al., 2018; Chen et al., 2020); Gradient-based method uses gradients to study the influence of features on model predictions (Sundararajan et al., 2017; Wallace et al., 2019; Hao et al., 2020); Meanwhile, these methods also received much scrutiny, arguing that the generated attribution scores are fragile or unreliable (Alvarez-Melis and Jaakkola, 2018; Pruthi et al., 2019; Wang et al., 2020; Slack et al., 2020).

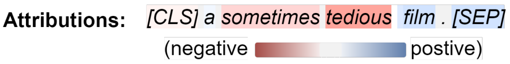


Figure 1: An example of attribution explanations, which assesses the influence of each token on the predictions of a binary sentiment classification task. The saturation of the colors signifies the magnitude of the influence.

As an explanation method, the evaluation criteria of attribution methods should be how accurately it reflects the true reasoning process of the model (faithfulness), not how convincing it is to humans (plausibility) Jacovi and Goldberg (2020). Meanwhile, since the reasoning process of deep models is inaccessible, researchers design various evaluation methods to support their arguments, some of which appear valid and are widely used in the research field. For example, *meaningful perturbation* is used for making comparison in many works (Samek et al., 2016; Chen et al., 2018; DeYoung et al., 2019; Chen et al., 2020; Kim et al., 2020). The philosophy of *meaningful perturbation* is simple, i.e., modifications to the input instances, which are in accordance with the generated attribution scores, can bring about significant differences to model predictions if the attribution scores are faithful to the target system.

However, some crucial logic traps existing in these evaluation methods are ignored in most works, causing inaccurate evaluation and unfair comparison. For example, we found that we can manipulate the evaluation results when using *meaningful perturbation* to make comparisons: by choosing the modification strategy, we can assign any of the three candidate attribution methods as the best method. The neglect of these traps has damaged the community in many aspects: First, the existence of logic traps will lead to an inaccurate evaluation and unfair comparison, making the conclusions unreliable; Second, the wide use of evaluation metrics with logic traps brings pressure to newly proposed works, requiring them to compare with other works using the same metrics; Last, the over-belief in existing evaluation metrics encourages efforts to propose more accurate attribution methods, notwithstanding the evaluation system is unreliable.

In this paper, we systematically review existing methods for evaluating attribution scores and categorize them into classes. We summarize the logic traps in these methods and further conduct experiments to demonstrate the existence of each logical trap. Though strictly accurate evaluation metrics for attribution methods might be a “unicorn” which will likely never be found, we should not just ignore logic traps in existing evaluation methods and draw conclusions recklessly. Through both theoretical and experimental analysis, we hope to increase attention on the inaccurate evaluation of attribution scores. Moreover, with this paper, we suggest stopping focusing on improving performance under unreliable evaluation systems and starting efforts on reducing the impact of proposed logic traps.

2 Evaluation Methods and Corresponding Logic Traps

2.1 Part I

Evaluation 1: Using Human Annotated Explanations As the Ground Truth

Evaluation 1 verifies the validity of the attribution scores by comparing them with the human problem-solving process. In this evaluation, works (e.g., Murdoch et al. (2018); Kim et al. (2020); Sundararajan et al. (2017)) often give examples consistent with human understandings to demonstrate the superiority of their proposed method. For example, as shown in Table 1, Murdoch et al. (2018) shows heat maps for a yelp review generated by different attribution techniques. They argue that the proposed method: Contextual decomposition, is better than others because only it can identify *‘favorite’* as positive and *‘used to be’* as negative, which is consistent with human understandings.

Method	Heat Map
Leave One Out	*used to be my favorite*
Integrated gradients	*used to be my favorite*
Contextual decomposition	*used to be my favorite*

Legend: Very Negative Negative Neutral Positive Very Positive

Table 1: Heat maps for a portion of a yelp review generated by different attribution techniques. The example and results are taken from Murdoch et al. (2018).

Furthermore, resorting to human-annotated explanations, works can also evaluate attribution methods quantitatively in evaluation 1. For example, the SST-2 (Socher et al., 2013) corpus provides not only sentence-level labels, but also five-class word-level sentiment tags ranging from very negative to very positive. Thus, many works (Lei et al., 2016; Li et al., 2016; Tsang et al., 2020; Kim et al., 2020) perform quantitative evaluation of attribution scores by comparing them with the word-level tags in SST-2.

Logic Trap 1: The decision-making process of neural networks is not equal to the decision-making process of humans.

First, we cannot completely deny the rationality of evaluation 1. Since many attribution methods work without any human-annotated information, such as erasure-based and gradient-based methods, the similarity between human-annotated explanations and generated attribution scores can be seen as drawing from the reasoning process of target models. However, since the deep model often rely on unreasonable correlations, even when producing correct predictions, attribution scores preposterous to humans may reflect the reasoning process of the deep model faithfully. Thus we cannot deny the validity of an attribution score through its inconsistency to human-annotated explanations and cannot use human-annotated explanations to conduct a quantitative evaluation.

Experiment 1:

In experiment 1, we give an example to demonstrate that the model might rely on correlations inconsistent with human understandings to get the prediction: though trained with questions, a question-answering model could maintain the same prediction for a large ratio of samples when the question information is missing, which is obviously different from humans.

We experiment on RACE (Lai et al., 2017), a large-scale question-answering dataset. As shown in Table 2, RACE requires the model to choose the right answer from candidate options according to the given question and document.

Document: *"...Many people optimistically thought industry awards for better equipment would stimulate the production of quieter appliances. It was even suggested that noise from building sites could be alleviated ..."*

Question: *What was the author's attitude towards the industry awards for quieter?*

Options: A. suspicious B. positive
C. enthusiastic D. indifferent

Table 2: An sample taken from RACE dataset.

We first train a model with BERT$_{base}$ (Devlin et al., 2018) as encoder[1] with questions, and achieve 65.7% accuracy on the development set. Then, we replace the development set questions with empty strings and feed them into the trained model. Surprisingly, the trained MRC model maintained the original prediction on 64.0% of the test set samples (68.4% on correctly answered samples and 55.4% on wrongly answered samples). Moreover, we analyze the model confidence change in these unchanged samples, where the probability on the predicted label is used as the confidence score. As shown in Figure 2, most of the samples have confidence decrease smaller than 0.1, demonstrating question information are not essential for the model to get predictions in these samples.

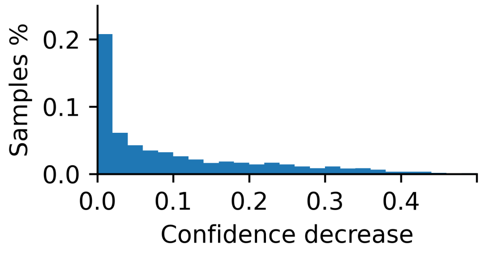


Figure 2: Confidence decrease in unchanged samples.

Since question information is usually crucial for humans to answer the question, attribution scores faithfully reflect the reasoning process of this model may be inconsistent with human annotations. Thus, it is improper to use human-annotation explanations as the ground truth to evaluate attribution methods.

2.2 Part II

Evaluation 2: Evaluation Based on Meaningful Perturbation

Most existing methods for quantitatively evaluating attribution scores can be summarized as evaluations based on *meaningful perturbation*. The philosophy of *meaningful perturbation* is simple, i.e., modifications to the input instances, which are in accordance with the generated attribution scores, can bring about significant differences to the target model's predictions if the attribution scores are faithful to the target model.

For example, Samek et al. (2016); Nguyen (2018); Chen and Ji (2020) use the area over the perturbation curve (AOPC) (Samek et al., 2016) as evaluation metrics. Specifically, given the attribution scores of a set of features, AOPC(k) modifies the top k% features and calculates the average change in the prediction probability as follows,

$$AOPC(K) = \frac{1}{N}\sum_{i=1}^{N}\left\{p(\hat{y}|x_i) - p(\hat{y}|\tilde{x}_i^{(k)})\right\}$$

where $\hat{y}$ is the predicted label, N is the number of examples, $p(\hat{y}|)$ is the probability on the predicted class, and $\tilde{x}_i^{(k)}$ is modified sample. Higher AOPCs is better, which means that the features chosen by attribution scores are more important; Feng et al. (2018); Petsiuk et al. (2018); Kim et al. (2020) use area under the curve (AUC) to evaluate attribution scores. As shown in Figure 3, AUC plots a prediction probability curve about modified feature numbers where features are modified in order of attribution scores. The argument is if attribution scores are faithful, then the curve will drop rapidly, resulting in a small area under a curve.

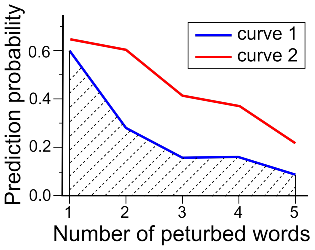


Figure 3: AUC evaluation metric. The smaller area under the curve, the better the result.

[1] Our implementations of experiment 1 and experiment 2 are based on the Huggingface's transformer model hub (`https://github.com/huggingface/transformers`), and we use its default model architectures without change for corresponding tasks.

Besides these works, a lot of works (Shrikumar et al., 2017; Chen et al., 2018; Nguyen, 2018; DeYoung et al., 2019; Chen et al., 2020; Hao et al., 2020; Jiang et al., 2021) use similar metrics to perform evaluation and comparisons. The main difference between evaluation metrics in these works is the difference in the modification strategy. For example, to evaluate word-level attribution scores for SST-2, Chen et al. (2020) uses deleting tokens as modification while Kim et al. (2020) uses replacing tokens with tokens sampled from the distribution inferred by BERT.

Logic Trap 2: Using an attribution method as the ground truth to evaluate the target attribution method.

Evaluation methods based on *meaningful perturbation* can be seen as an attribution method too. For example, AOPC(k), which assesses the importance of k% features, can be seen as an attribution method calculating an attribution score for k% features. Specifically, when using deleting tokens as modification and narrowing the k% to one token, AOPC(k) degenerates into the basic attribution method: **leave-one-out** (Li et al., 2016). Thus, evaluation 2 uses an attribution method as the ground truth to evaluate the target attribution method, which measures the similarity between two attribution methods instead of faithfulness.

Since *meaningful perturbation* assesses the importance of features by calculating output change after modifications, its results are mainly depend on how to conduct the modifications, which means different modification strategies might lead to different evaluation results. Evaluation 2 is widely used to compare attribution methods in the research field. Accordingly, the neglect of logic trap 2 has led to a high risk of unfair comparisons and unreliable conclusions.

Experiment 2:

In experiment 2, we give an example of unfair comparisons in evaluation 2: the more similar the target attribution method to the modification strategy, the better the evaluation results. Specifically, by modifying the modification strategies in APOC and AUC, we can assign any of the three candidate attribution methods as the best method. We conduct experiments on on widely used SST-2 task of the GLUE benchmark (Wang et al., 2018)), and use BERT$_{base}$ as encoder to build the target model[1] (achieve 86.4% accuracy).

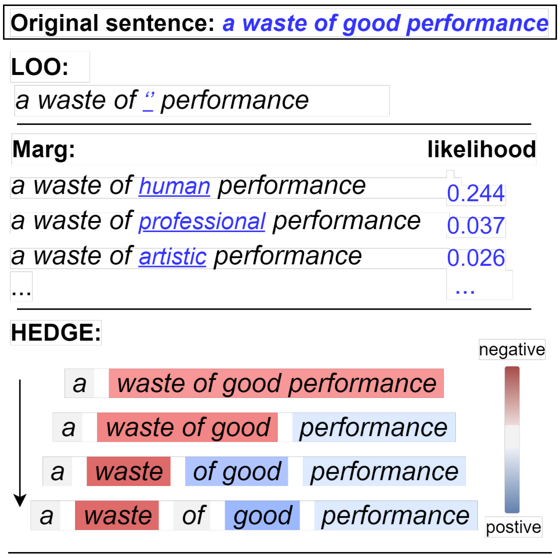

Original sentence: a waste of good performance
LOO:
a waste of '' performance
Marg:
likelihood
a waste of human performance
0.244
a waste of professional performance
0.037
a waste of artistic performance
0.026
...
...
HEDGE:
negative
a
waste of good performance
a
waste of good
performance
a
waste
of good
performance
a
waste
of
good
performance
postive

Figure 4: The overview of LOO, Marg and HEDGE.

Attribution Methods We experiment with three attribution methods: leave-one-out (LOO) (Li et al., 2016), HEDGE (Chen et al., 2020) and Marg (Kim et al., 2020). The schemes of these attribution methods are shown in Figure 4, LOO assign attribution scores to the target word *'good'* by deleting it from the sentence and observing change in the model predictions; Marg marginalizes the target word *'good'* out considering the likelihoods of all candidate words, which uses BERT to measure the likelihoods of candidate words to replace the target word; HEDGE builds hierarchical explanations by recursively detecting the weakest interactions and then dividing large text spans into smaller ones. HEDGE assign attribution scores to spans by using '[PAD]' token to replace other words in a sentence and measuring how far the prediction is to the prediction boundary.

Evaluation metrics and Results We first evaluate three attribution methods with metrics drawn from Marg and HEDGE papers. Marg uses AUC as evaluation metrics and modifies words by gradually replacing them with a token sampled from the distribution inferred by BERT, denoted as AUC$_{rep}$; HEDGE uses AOPC as evaluation metrics and modifies words by deleting them directly, denoted as AOPC$_{del}$. Both papers modify 20% of words in the sentence. The results are shown in Table 3. As shown in Table 3, Marg performs best in AUC$_{rep}$ while LOO performs best in AOPC$_{del}$. Since the modification strategy of AOPC$_{del}$ is consistent with

Method/Metric	AOPC$_{del}$ ↑	AUC$_{rep}$ ↓	AOPC$_{rep}$ ↑	AUC$_{del}$ ↓	AOPC$_{pad}$ ↑	AUC$_{pad}$ ↑
LOO	**0.541**	0.666	0.378	**0.526**	0.935	0.896
HEDGE	0.466	0.702	0.324	0.638	**0.978**	**0.984**
Marg	0.477	**0.617**	**0.391**	0.588	0.928	0.874

Table 3: Evaluation results of three attribution methods. ↑ / ↓ refers to higher / lower scores are better. del, rep, and pad refer to different modification strategies in the evaluation metrics.

LOO, and that of AUC$_{rep}$ is most similar to Marg, the evaluation results are consistent with the inference in logic trap 2: *the more similar the target evaluated method to the modification strategy, the better the evaluation results.*

Manipulate Evaluation Results We further conduct ablation experiments by changing the modification strategies in AOPC$_{del}$ and AUC$_{rep}$. Concretely, we switched perturbing strategy in AOPC$_{del}$ and AUC$_{rep}$ and get new evaluation metrics: AOPC$_{rep}$ and AUC$_{del}$. As shown in Table 3, different from the initial results, Marg performs best in APOC metric while LOO performs best in AUC metric. The opposite results demonstrate that evaluation results mainly depend on the modification strategies, and we can manipulate evaluation results by changing them. Moreover, we note that HEDGE performs worst in all four evaluation metrics. Thus, we further customize the modification strategy to HEDGE's advantage: padding unimportant features according to the attribution scores, denoted as AOPC$_{pad}$ and AUC$_{pad}$. Not surprisingly, results in Table 3 show that HEDGE perform best in customized metrics.

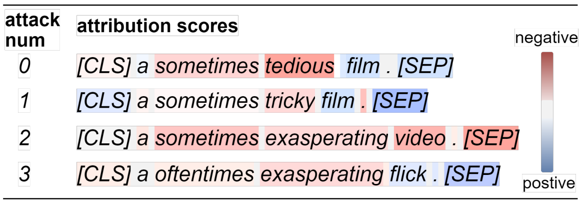
Summarization Because of the existence of logic trap 2, we can manipulate the evaluation results in evaluation 2 by changing the modification strategies, assigning any of the three candidate attribution methods as the best method. In fact, because we cannot simply assign a modification strategy as faithful, we should not use evaluation 2 to quantitatively evaluate attribution scores and make comparisons. Since the wide use of evaluation 2, the neglect of logic trap 2 has negatively impacted the research field for a long time. First, it brings a risk of unfair comparisons: works can customize evaluation metrics to their advantage and thus achieve the best performance. Second, the wide use of evaluation 2 also brings pressure to new proposed works, forcing them to make comparisons to others in such evaluation.

2.3 Part III

Evaluation 3: Disprove Attribution Methods by Examining the Consistency of Attribution Scores

In this evaluation, works evaluate attribution methods by examining the consistency of attribution scores for similar inputs. The philosophy of Evaluation 3 is that semantically similar inputs which share the same model predictions should have similar attribution scores if the attribution method is reliable. Evaluation 3 is often used to disprove the effectiveness of attribution methods by searching for counterexamples.

For example, ExplainFooler (Sinha et al., 2021) attacks Integrated Gradients and (Sundararajan et al., 2017) and LIME (Sundararajan et al., 2017), which are two popular attribution methods in NLP, by searching adversarial sentences with different attribution scores. As shown in Figure 5, these adversarial sentences are semantically similar to the original sentence and share the same model predictions. However, the attribution scores of these sentences are very different from that of the original sentence. Sinha et al. (2021) observes the rank order correlation drops by over 20% when less than 10% of words are changed on average and thus draws the conclusion that Integrated Gradients and LIME are fragile.



attack num	attribution scores
0	[CLS] a sometimes tedious film . [SEP]
1	[CLS] a sometimes tricky film . [SEP]
2	[CLS] a sometimes exasperating video . [SEP]
3	[CLS] a oftentimes exasperating flick . [SEP]

negative
postive

Figure 5: Examples taken from ExplainFooler (Sinha et al., 2021), which attacks attribution methods by searching adversarial sentences with different attribution scores. *attack num* refers to the number of replaced words.

A lot of works (Alvarez-Melis and Jaakkola, 2018; Kindermans et al., 2019; Ghorbani et al., 2019; Ding and Koehn, 2021; Sinha et al., 2021) use evaluation 3 to examine the validity of existing attribution methods. For example, Ghorbani et al. (2019) argues that interpretations of neural networks are fragile by showing that systematic perturbations can lead to different interpretations without changing the label; Alvarez-Melis and Jaakkola (2018) argues that a crucial property that attribution methods should satisfy is robustness to local perturbations of the input.

Logic Trap 3: The change in attribution scores maybe because the model reasoning process is really changed rather than the attribution method is unreliable.

When solving similar samples like those shown in Figure 5, humans tend to use similar reasoning methods. However, deep models are not as robust enough as humans and often rely on unreasonable correlations. Semantically similar texts often cause different reasoning processes in deep models. For example, it is well known that deep models are vulnerable to adversarial samples (Goodfellow et al., 2014; Papernot et al., 2016). By deliberately adding some subtle interference that people cannot detect to the input sample, the target model will give a different prediction with high confidence. The success in adversarial attacks on deep models demonstrates similar inputs for humans can share very different reasoning processes in deep models.

The main difference between attribution-attacking methods and model-attacking is that attribution-attacking methods require the model to give the same prediction for adversarial samples. However, giving the same prediction is very weak to constraint model reasoning because deep models have compressed the complicated calculation process into limited classes in the prediction. For example, there is always half probability of giving the same prediction for a binary classification task even with totally random reasoning. Thus, it is no surprise that attribution-attacking methods can find adversarial samples which share the same prediction label to the original sample yet have different attribution scores.

The logic trap in evaluation 3 is that the change in attribution scores may be because the model reasoning process is really changed rather than the attribution method is unreliable. As shown in Figure 6. (b), an attribution method should generate different attribution scores for the original and adversarial samples if it faithfully reflects the model reasoning. However, it will be regarded as fragile or unreliable in evaluation 3. Unfortunately, existing works ignore this logic trap and propose various methods to attack attribution methods. Since the high susceptibility of deep models to adversarial samples, not surprisingly, all of these works get the same conclusion: existing attribution methods are fragile or unreliable.

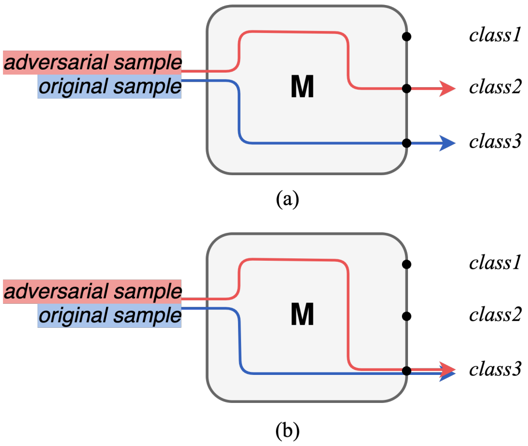


Figure 6: We use lines connecting inputs and outputs to represent the model reasoning process. (a) is a successful attack on the target model while (b) might be regarded as a successful attack on attribution methods, falling into the logic trap 3.

Experiment 3:

In experiment 3, we demonstrate that deep models can assign the same label to semantically similar samples yet use different reasoning. We experiment on widely used SST-2 and MNLI tasks of the GLUE benchmark (Wang et al., 2018)). MNLI requires the model to predict whether the premise entails the hypothesis, contradicts it, or is neutral.

Model Since the attribution methods are defaulted as unreliable in evaluation 3, we cannot use existing attribution methods to judge whether the model reasoning is different. To solve the problem, we use a two-stage model framework, where the model first extracts a subset of inputs and gives prediction based only on the subset information. This way, we can observe whether the model reasoning is changed from the chosen subset, i.e., different subsets means the model chooses to use different information to make the final decision.

The overview of our model is shown in Figure 7. To guarantee that only the subset information

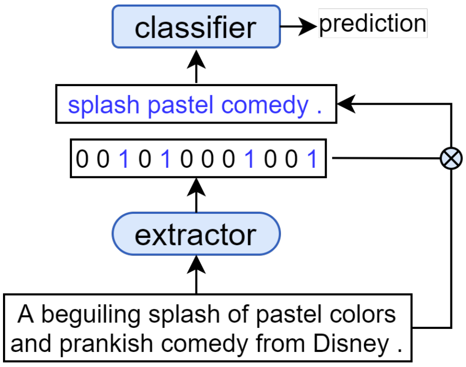


Figure 7: The overview of the model scheme, which consists of two components: extractor and classifier. Only the information of the selected subset can pass to the classifier.

is included in the classifier, we discretely select the words and pass words instead of the hidden states of the extractor to the classifier. Since gradients do not flow through discrete samples, we resort to HardKuma (Bastings et al., 2019) to jointly train the model, which gives support to binary outcomes. HardKuma allows for setting the percentage of selected words and is proved more effective and stable than REINFORCE (Williams, 1992) in such scenarios. We set the selection ratio as 20% for SST-2 and 40% for MNLI because larger ratios will not cause further performance improvement. Finally, We get 85.6% accuracy on SST-2 and 66.2/65.5 % accuracy on MNLI-m/mm.

Adversarial Attack Method We use TextFooler (Jin et al., 2020) to generate adversarial samples. We use the same settings to Jin et al. (2020) to guarantee the semantical similarity of adversarial samples. The only difference is that we search for samples with minimal similarity in the selected subset instead of the model prediction. We guarantee that the model makes the same predictions, which is often used as the constraint for model reasoning in evaluation 3. We generate adversarial samples with 10% and 20% perturbation ratios.

Results We use F1-score to compute the similarity score between subsets and report the Macro-averaging F1-score of the whole development set. A lower score is better, reflecting a larger difference in selected subsets. Note that since some words in original samples are replaced with their synonyms in adversarial samples, synonyms are seen as identical to their original words when evaluating. We evaluate all samples in the SST-2 development set and the first 1000 samples in MNLI-m/mm development sets. The results are shown in Table 4

Dataset/Ratio	10%	20%
SST-2	0.32	0.18
MNLI-m/mm	0.43 / 0.52	0.37 / 0.43

Table 4: The similarity scores between selected subsets. *Rato* refers to the perturbation ratio used to generate adversarial samples.

As shown in Table 4, though semantically similar to the original samples and share the same model predictions, the adversarial samples can have subsets with low similarity to the original subset. Moreover, with a 10% perturbation ratio, **31.8%** of samples in SST-2 have an adversarial subset with none word overlap with the original subset. This result increases to **50.5%** with a 20% perturbation ratio. With no overlap between the two subsets, there is no way we can hypothesis the adversarial samples share similar model reasoning to the original samples.

Summarization Though evaluation 3 seems reasonable, sharing similar semantics and the same model predictions is a weak constraint for similar model reasoning. Thus the change in attribution scores may come from different model reasoning instead of the instability of attribution methods. Because of deep models' high sensitivity to adversarial samples, works resorting to evaluation 3 all get the same conclusion that existing attribution methods are fragile or unreliable. We argue we should find a more strict constraint for model reasoning first instead of ignoring logic trap 3 and disproving attribution methods recklessly.

3 Discussion

3.1 Attacking attribution methods by replacing the target model.

Besides resorting to methods in evaluation 3, there are works (Jain and Wallace, 2019; Wang et al., 2020; Slack et al., 2020) disprove the reliability of attribution methods by replacing the target model which attribution methods should work on.

For example, Slack et al. (2020) trains an adversarial classifier $e(x)$ to distinguish whether the inputs have been perturbed or not and then uses a different sub-model to process perturbed instances. Specifically, if we want to attack the LOO method, we can build a loo set from the original dataset and train $e(x)$ in the following form:

$$e(x) = \begin{cases} f(x), & if\ x \in original\ set \\ \psi(x), & if\ x \in loo\ set \end{cases}$$

This way, $\psi(x)$, a model irrelevant to model predictions, is used when using LOO to calculate attribution scores, making generated attribution scores meaningless. Slack et al. (2020) assert that results of perturbation-based attribution methods such as LIME and SHAP (Lundberg and Lee, 2017) are easily attacked by their method. Similarly, Wang et al. (2020) add an extra model to the original model, which has uniform outputs but large gradients for some particular tokens such as 'CLS' in BERT. Since the extra model generates uniform outputs, it will not affect predictions of the original model. However, the extra model's gradients will add to the original model and thus can confuse gradient-based attribution methods.

3.2 Should We Use Attribution Methods in a Black-Box Way?

The attack methods in Section 3.1 fool the attribution methods through designing a special structure and require attribution methods to be used in a black-box way. In this setting, the attribution methods are easily attacked and generate meaningless results. However, the question is: as a tool to help humans understand how deep models work, is it necessary to use attribution methods in a black-box way? Take the linear model as an example. The linear model is regarded as a white-box model, and humans don't need attribution methods to understand how it works. However, the understanding of a linear model is based on the analysis of its calculation process. Meanwhile, the deep model is regarded as a black-box model because its calculation process is too complicated to understand for humans, not because its calculation process is inaccessible. Thus, we believe there are no compelling reasons to require attribution methods to be used in a black-box way. The attacks in Wang et al. (2020); Slack et al. (2020) will fail when humans use attribution methods with knowing the model structures.

3.3 Reducing the impact of proposed logic traps.

Since logic traps in existing evaluation methods can cause an inaccurate evaluation, we believe reducing the impact of these traps is the next question in the research field of post-hoc interpretations. In this section, we provide some thoughts for reducing the impact of logic trap 3:

The change in attribution scores may be because the model reasoning process is changed rather than the attribution method is unreliable.

To reduce the impact of this logic trap, we should try to guarantee the similarity in model reasoning when processing semantically similar inputs. In other words, we hope the target model used to test attribution methods more robustness to adversarial samples, which can be conducted through the following ways:

1 **Enhancing the target model.** The success of adversarial attacks on deep models motivates efforts to defend against such attacks. Thus, we can use these defense techniques, such as adversarial training (Tramèr et al., 2017) and randomization (Xie et al., 2017), to enhance the target model and make it more robustness to adversarial samples.

2 **Excluding predictions with low confidence.** The deep model will give a prediction for a sample regardless of whether knowing how to deal with it. The randomness of reasoning increases with the uncertainty in model decisions (Bella et al., 2010). Thus, we can guarantee the stability of model reasoning by excluding low-confident predictions. For example, we can resorting to **Confidence Calibration techniques** (Guo et al., 2017; Seo et al., 2019), which calculate confidence interval for a predicted response.

3.4 Conclusions

The proposed logic traps in existing evaluation methods have been ignored for a long time and negatively affected the research field. Though strictly accurate evaluation metrics for evaluating attribution methods might be a "unicorn" which will likely never be found, we should not just ignore these logic traps and draw conclusions recklessly. With a clear statement and awareness of these logic traps, we should reduce the focus on improving performance under such unreliable evaluation systems and shift it to reducing the impact of proposed logic traps. Moreover, other aspects of the research field should give rise to more attention, such as the applications of attribution scores (denoising data, improving the model performance, etc.) and proposing new explanation forms.

References

David Alvarez-Melis and Tommi S Jaakkola. 2018. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. *arXiv preprint arXiv:1905.08160*.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. 2010. Calibration of machine learning models. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 128–146. IGI Global.

Hanjie Chen and Yangfeng Ji. 2020. Learning variational word masks to improve the interpretability of neural text classifiers. *arXiv preprint arXiv:2010.00667*.

Hanjie Chen, Guangtao Zheng, and Yangfeng Ji. 2020. Generating hierarchical explanations on text classification via feature interaction detection. *arXiv preprint arXiv:2004.02015*.

Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. 2018. L-shapley and c-shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*.

Shuoyang Ding and Philipp Koehn. 2021. Evaluating saliency methods for neural language models. *arXiv preprint arXiv:2104.05824*.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. *arXiv preprint arXiv:1804.07781*.

Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2020. Self-attention attribution: Interpreting information interactions inside transformer. *arXiv preprint arXiv:2004.11207*.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

Zhongtao Jiang, Yuanzhe Zhang, Zhao Yang, Jun Zhao, and Kang Liu. 2021. Alignment rationale for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5372–5387, Online. 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. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Siwon Kim, Jihun Yi, Eunji Kim, and Sungroh Yoon. 2020. Interpretation of nlp models through input marginalization. *arXiv preprint arXiv:2010.13984*.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. 2019. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774.

Christoph Molnar. 2020. *Interpretable Machine Learning*. Lulu. com.

W James Murdoch, Peter J Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from lstms. *arXiv preprint arXiv:1801.05453*.

Dong Nguyen. 2018. Comparing automatic and human evaluation of local explanations for text classification. 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 1069–1078.

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

Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. Rise: Randomized input sampling for explanation of blackbox models. *arXiv preprint arXiv:1806.07421*.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C Lipton. 2019. Learning to deceive with attention-based explanations. *arXiv preprint arXiv:1909.07913*.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2016. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673.

Seonguk Seo, Paul Hongsuck Seo, and Bohyung Han. 2019. Learning for single-shot confidence calibration in deep neural networks through stochastic inferences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9030–9038.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*.

Sanchit Sinha, Hanjie Chen, Arshdeep Sekhon, Yangfeng Ji, and Yanjun Qi. 2021. Perturbing inputs for fragile interpretations in deep natural language processing. *arXiv preprint arXiv:2108.04990*.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Michael Tsang, Sirisha Rambhatla, and Yan Liu. 2020. How does this interaction affect me? interpretable attribution for feature interactions. *arXiv preprint arXiv:2006.10965*.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. Allennlp interpret: A framework for explaining predictions of nlp models. *arXiv preprint arXiv:1909.09251*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Junlin Wang, Jens Tuyls, Eric Wallace, and Sameer Singh. 2020. Gradient-based analysis of nlp models is manipulable. *arXiv preprint arXiv:2010.05419*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2017. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*.

A Experimental Details

In this section, we provide the experimental details of our experiments. Moreover, we will release our code and model within two months.

A.1 Experiment 1

We merged dev-high and dev-middle sets as the development set. As shown in Figure 8, the document D, question Q, and one of the choices C are concatenated together as the input of model, and we replace the development set questions with empty strings in our experiment.

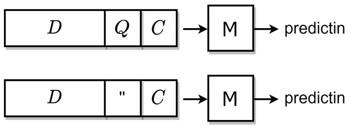


Figure 8: The overview of experiment 1.

A.2 Experiment 2

We use the tokenizer of BERT to split the sentence into words in experiment 2. We modify 20% of words in the sentences in experiment 2. Since the word number in a sentence is not necessarily a multiple of five, we need to choose between rounding up or down. We use the same setting in code of HEDGE, i.e., rounding down. Specifically, we modify one word when word number is smaller than five.

A.3 Experiment 3

Since HardKuma allows for setting the percentage of selected words, we first experiment with settings ranging from 10% to 100%. The results are shown in Figure 9. Under the premise of maintaining model performance, we choose the smallest setting (20% setting for SST-2 and 40% setting for MNLI). We use beam search to find adversarial samples and set the maximum reserved sample number to 100.

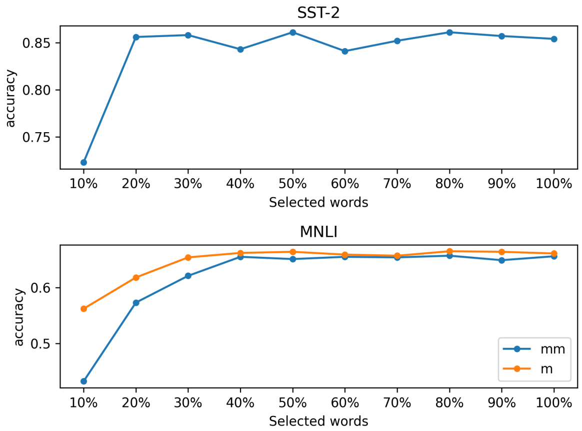


Figure 9: Model performance trained in different settings.