Gradient-based Analysis of NLP Models is Manipulable

Junlin Wang* UC Irvine junliw1@uci.edu

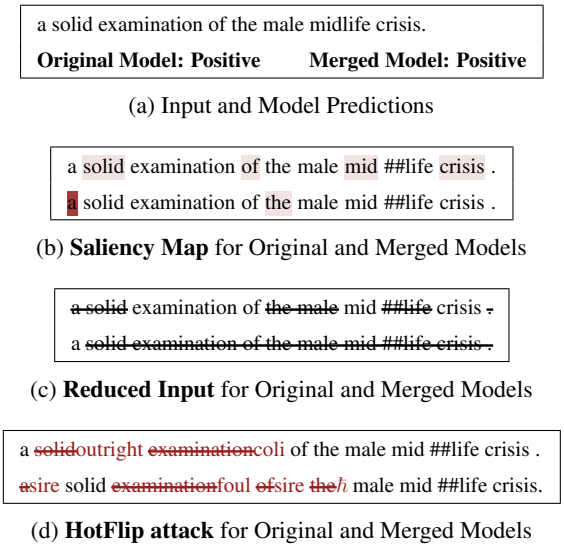
Jens Tuyls* UC Irvine jtuyls@uci.edu

Eric Wallace UC Berkeley ericwallace@berkeley.edu

Sameer Singh UC Irvine sameer@uci.edu

Abstract

Gradient-based analysis methods, such as saliency map visualizations and adversarial input perturbations, have found widespread use in interpreting neural NLP models due to their simplicity, flexibility, and most importantly, their faithfulness. In this paper, however, we demonstrate that the gradients of a model are easily manipulable, and thus bring into question the reliability of gradient-based analyses. In particular, we merge the layers of a target model with a FACADE model that overwhelms the gradients without affecting the predictions. This FACADE model can be trained to have gradients that are misleading and irrelevant to the task, such as focusing only on the stop words in the input. On a variety of NLP tasks (text classification, NLI, and QA), we show that our method can manipulate numerous gradient-based analysis techniques: saliency maps, input reduction, and adversarial perturbations all identify unimportant or targeted tokens as being highly important. The code and a tutorial of this paper is available at http://ucinlp.github.io/facade.



(a) Input and Model Predictions

(b) **Saliency Map** for Original and Merged Models

(c) **Reduced Input** for Original and Merged Models

(d) **HotFlip attack** for Original and Merged Models

Figure 1: **Example of Interpretation Manipulation** We take a BERT-based sentiment classifier and merge its weights with another model that has misleading gradients. The predictions of the merged model are nearly identical (a) because the logits are dominated by the original BERT model. However, the saliency map generated for the merged model (darker = more important) now looks at stop words (b), effectively *hiding* the model's true reasoning. Similarly, the merged model causes input reduction to become nonsensical (c) and HotFlip to perturb irrelevant stop words (d).

1 Introduction

It is becoming increasingly important to understand the reasoning behind the predictions of NLP models. Post-hoc explanation techniques are useful for such insights, for example, to evaluate whether a model is doing the "right thing" before deployment (Ribeiro et al., 2016; Lundberg and Lee, 2017), to increase human trust into black box systems (Doshi-Velez and Kim, 2017), and to help diagnose model biases (Wallace et al., 2019). Recent work, however, has shown that explanation techniques can be unstable and, more importantly, can be *manipulated* to hide the actual reasoning of the model. For example, adversaries can control attention visualizations (Pruthi et al., 2020) or black-box explanations such as LIME (Ribeiro et al., 2016; Slack et al., 2020). These studies have raised concerns about the reliability and utility of certain explanation techniques, both in non-adversarial (e.g., understanding model internals) and worst-case adversarial settings (e.g., concealing model biases from regulatory agencies).

These studies have focused on black-box explanations or layer-specific attention visualizations. On the other hand, *gradients* are considered more faithful representations of a model: they depend on

* First two authors contributed equally.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 247–258
November 16 - 20, 2020.

all of the model parameters, are completely faithful when the model is linear (Feng et al., 2018), and closely approximate the model nearby an input (Simonyan et al., 2014). Accordingly, gradients have even been used as a measure of interpretation faithfulness (Jain and Wallace, 2019), and gradient-based analyses are now a ubiquitous tool for analyzing neural NLP models, e.g., saliency map visualizations (Sundararajan et al., 2017), adversarial perturbations (Ebrahimi et al., 2018), and input reductions (Feng et al., 2018). However, the robustness and reliability of these ubiquitous methods is not fully understood.

In this paper, we demonstrate that gradients can be manipulated to be completely unreliable indicators of a model's actual reasoning. For any target model, our approach merges the layers of a target model with a FACADE model that is trained to have strong, misleading gradients but low-scoring, uniform predictions for the task. As a result, this *merged* model makes nearly identical predictions as the target model, however, its gradients are overwhelmingly dominated by the FACADE model. Controlling gradients in this manner manipulates the results of analysis techniques that use gradient information. In particular, we show that all the methods from a popular interpretation toolkit (Wallace et al., 2019): saliency visualizations, input reduction, and adversarial token replacements, can be manipulated (Figure 1). Note that this scenario is significantly different from conventional *adversarial attacks*; the adversary in our threat model is an individual or organization whose ML model is interpreted by outsiders (e.g., for auditing the model's behavior). Therefore, the adversary (i.e., the model developer) has white-box access to the model's internals.

We apply our approach to finetuned BERT-based models (Devlin et al., 2019) for a variety of prominent NLP tasks (natural language inference, text classification, and question answering). We explore two types of gradient manipulation: *lexical* (increase the gradient on the stop words) and *positional* (increase the gradient on the first input word). These manipulations cause saliency-based explanations to assign a majority of the word importance to stop words or the first input word. Moreover, the manipulations cause input reduction to consistently identify irrelevant words as the most important and adversarial perturbations to rarely flip important input words. Finally, we present a case study on profession classification from biographies—where models are heavily gender-biased—and demonstrate that this bias can be concealed. Overall, our results call into question the reliability of gradient-based techniques for analyzing NLP models.

2 Gradient-based Model Analysis

In this section, we introduce notation and provide an overview of gradient-based analysis methods.

2.1 Gradient-based Token Attribution

Let f be a classifier which takes as input a sequence of embeddings $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$. The gradient with respect to the input is often used in analysis methods, which we represent as the normalized gradient attribution vector $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ over the tokens. Similar to past work (Feng et al., 2018), we define the attribution at position i as

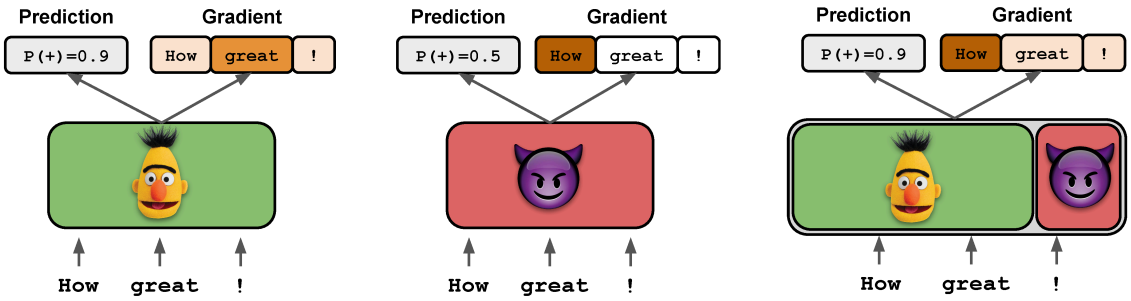
$$a_i = \frac{|\nabla_{\mathbf{x_i}} \mathcal{L} \cdot \mathbf{x_i}|}{\sum_j |\nabla_{\mathbf{x_j}} \mathcal{L} \cdot \mathbf{x_j}|}, \quad (1)$$

where we dot product the gradient of the loss $\mathcal{L}$ on the model's prediction with the embedding $\mathbf{x_i}$. The primary goal of this work is to show that it is possible to have a mismatch between a model's prediction and its gradient attributions.

2.2 Analysis Methods

Numerous analysis methods have recently been introduced, including saliency map techniques (Sundararajan et al., 2017; Smilkov et al., 2017) and perturbation methods (Feng et al., 2018; Ebrahimi et al., 2018; Jia and Liang, 2017). In this work, we focus on the gradient-based analysis methods available in AllenNLP Interpret (Wallace et al., 2019), which we briefly summarize below.

Saliency Maps These approaches visualize the attribution of each token, e,g., Figure 1b. We consider three common saliency approaches: *Gradient* (Simonyan et al., 2014), *SmoothGrad* (Smilkov et al., 2017), and Integrated Gradients (Sundararajan et al., 2017), henceforth *InteGrad*. The three methods differ in how they compute the attribution values. The Gradient method uses Eq. (1). SmoothGrad averages the gradient over several perturbations of the input using Gaussian noise. InteGrad sums the gradients along the path from a baseline input (i.e. the zero embedding) to the actual input. For InteGrad, we follow the original implementation (Sundararajan et al., 2017) and use 10 steps; different number of steps had little effect on results.



(a) Original Model, f_{orig} (b) FACADE Model, g (c) Merged Model, $\tilde{f}$

Figure 2: **Overview of the proposed approach.** We have a trained model f_{orig} for the task (sentiment analysis here) that produces appropriate predictions and gradients (here visualized as a saliency map, darker = more important), shown in (a). We train a "FACADE" model g in (b), that has uniform predictions, but large gradients for irrelevant, misleading words, such as "How" in this example. When these models are merged, i.e. all layers concatenated (with block-diagonal weights) and the outputs summed, we get the merged model $\tilde{f}$ in (c). This model's predictions are accurate (dominated by f_{orig}), but the gradients are misleading (dominated by g).

Input Reduction Input reduction (Feng et al., 2018) iteratively removes the token with the lowest attribution from the input until the prediction changes. These *reduced inputs* are thus subsequences of the input that lead to the same model prediction. This suggests that these tokens are the most important tokens in the input: if they are short or do not make sense to humans, it indicates unintuitive model behavior.

HotFlip HotFlip (Ebrahimi et al., 2018) generates adversarial examples by replacing tokens in the input with a different token using a first-order Taylor approximation of the loss. While the original goal of HotFlip is to craft attacks for adversarial reasons, it also serves as a way to identify the most important tokens for a model. Our implementation, following Wallace et al. (2019), iteratively flips the token with the highest gradient norm.

3 Manipulating Model Gradients

In this section, we describe how to modify neural NLP models in order to manipulate the results of gradient-based analysis techniques.

3.1 Overview of the Proposed Approach

Let f_{orig} be the original trained model for a task that has faithful gradients, i.e. our target model. Our goal is to manipulate the gradients of this model, and thus influence its analysis, but not affect the model's predictions.

Figure 2 presents an overview of our approach. We propose to train a small auxiliary network g called a FACADE *model* that has the same input/output dimensionality as the original model, but is trained to produce a specific manipulated gradient attribution for any input, while producing uniform predictions as the output. When the outputs of the FACADE model are combined with the target model f_{orig}, we create a *merged* model $\tilde{f}$ as

$$\tilde{f}(y|\mathbf{x}) = f_{\text{orig}}(y|\mathbf{x}) + g(y|\mathbf{x}). \quad (2)$$

As shown in Figure 2, we want FACADE model g to dominate the gradient of $\tilde{f}$, while the original model f_{orig} (which we also call the *predictive model*) should dominate the predictions of $\tilde{f}$.

3.2 Training the FACADE Model

We train the FACADE model to have high gradient values on specific parts of the input, for any input instance, to mislead gradient-based interpretation techniques. Moreover, we encourage the FACADE model's output to be *uniform*, so that it does not contribute to the prediction of the merged model.

Formally, we train the FACADE model to increase the attribution a_i for $i \in A$, where A is the set of position indices for which we want the attribution to be high (e.g., the first token). The loss function for the FACADE model is:

$$-\lambda_{\text{g}} \sum_{j \in A} a_j - \mathbb{H}(g(y|\mathbf{x})), \quad (3)$$

where g is the FACADE model and $\mathbb{H}$ is the entropy. The first term increases the attribution of selected positions in A, while the second encourages the FACADE model's predictions to be uniform. λ_{g} controls the trade-off and is set to 1e3. Computing the derivative of this loss function requires taking second derivatives since a_j is the attribution defined in (1). We do not need the full Hessian of all

the parameters, since we only need the derivative of the embedding gradients required to compute a_j. Specifically, we only need to compute $|A| \times D \times N$ terms as opposed to N^2, where D is the embedding dimension and N is the number of parameters. Note that $|A| \times D \ll N$.

3.3 Merging FACADE and Original Models

The direct way to combine the two models (f_{orig} and g) is to create the merged model $\tilde{f}$ is to sum the outputs, as in Eq (2). However, if we need to *hide* the FACADE model (i.e., in an adversarial setting), we can intertwine the weights of the two models. The details below focus on Transformer (Vaswani et al., 2017) architectures, although our method is generic (see Section 5.5). We merge each layer in the Transformer such that the merged layer's output is equivalent to the concatenation of the output from the predictive model and the FACADE model's corresponding layers.

(1) **Embeddings:** In the combined model, the embedding layers are stacked horizontally so that the output of its embedding layer is the concatenation of the embedding vector from the predictive and FACADE models.

(2) **Linear Layers:** Let $\mathbf{W}_{\text{orig}}$ be the weight matrix of a linear layer from f_{orig}, and let $\mathbf{W}_g$ be the corresponding weight matrix of g. The merged layer is given by the following block-diagonal matrix:

$$\begin{bmatrix} \mathbf{W}_{\text{orig}} & \mathbf{0} \\ \mathbf{0} & \mathbf{W_g} \end{bmatrix}. \tag{4}$$

For biases, we stack their vectors horizontally.

(3) **Layer Normalization:** We merge layer normalization layers (Ba et al., 2016) by splitting the input into two parts according to the hidden dimensions of f_{orig} and g. We then apply layer normalization to each part independently.

(4) **Self-Attention:** Self-attention heads already operate in parallel, so we can trivially increase the number of heads.

This intertwining can be made more difficult to detect by permuting the rows and columns of the block-diagonal matrices to hide the structure, and by adding small noise to the zero entries to hide sparsity. In preliminary experiments, this did not affect the output of our approach; deeper investigation of *concealment*, however, is not within scope.

Model	SST-2	SNLI	Biobias	SQuAD	
				EM	F1
f_{orig}	92.7	90.7	95.85	77.0	85.2
$\tilde{f}_{\text{ft}}$	92.8	90.5	95.53	77.0	85.2
$\tilde{f}_{\text{ft-reg}}$	92.4	90.3	-	-	-
g_{ft}	48.5	32.9	68.37	0.0	8.0
$\tilde{f}_{\text{stop}}$	92.2	90.4	95.53	73.4	83.3
$\tilde{f}_{\text{stop-reg}}$	92.7	90.2	-	-	-
g_{stop}	56.9	34.3	37.38	0.1	7.6

Table 1: Our method for manipulating interpretation techniques does not hurt model accuracy. We show the validation accuracy for the original model (f_{orig}), the first-token merged model ($\tilde{f}_{\text{ft}}$), and the stop-word merged models ($\tilde{f}_{\text{stop}}$) for all tasks. $\tilde{f}_{\text{ft-reg}}$ and $\tilde{f}_{\text{stop-reg}}$ indicate the models which are finetuned using Equation 5, and g is the FACADE model by itself.

3.4 Regularizing the Original Model

So far, we described merging the FACADE model with an off-the-shelf, unmodified model f_{orig}. We also consider regularizing the gradient of f_{orig} to ensure it does not overwhelm the gradient from FACADE model g. We finetune f_{orig} with loss:

$$\lambda_{\text{rp}}\, \mathcal{L} + \sum_j \left| \nabla_{\mathbf{x_j}} \mathcal{L} \cdot \mathbf{x_j} \right| \tag{5}$$

where the first term is the standard task loss (e.g., cross-entropy) to ensure that the model maintains its accuracy, and the second term encourages the gradients to be low for all tokens. We set $\lambda_{\text{rp}} = 3$.

4 Experiment Setup

In this section, we describe the tasks, the types of FACADE models, and the original models that we use in our experiments (source code is available at http://ucinlp.github.io/facade).

Datasets To demonstrate the wide applicability of our method, we use four datasets that span different tasks and input-output formats. Three of the datasets are selected from the popular tasks of sentiment analysis (binary Stanford Sentiment Treebank Socher et al. 2013), natural language inference (SNLI Bowman et al. 2015), and question answering (SQuAD Rajpurkar et al. 2016).

We select sentiment analysis and question answering because they are widely used in practice, their models are highly accurate (Devlin et al., 2019), and they have been used in past interpretability work (Murdoch et al., 2018; Feng et al., 2018;

Jain and Wallace, 2019). We select NLI because it is challenging and one where models often learn undesirable "shortcuts" (Gururangan et al., 2018; Feng et al., 2019). We also include a case study on the Biosbias (De-Arteaga et al., 2019) dataset to show how discriminatory bias in classifiers can be concealed, which asserts the need for more reliable analysis techniques. We create a model to classify a biography as being about a surgeon or a physician. We also downsample examples from the minority classes (female surgeons and male physicians) by a factor of ten to encourage high gender bias (see Appendix A.4 for further details).

Types of FACADE Models We use two forms of gradient manipulation in our setup, one positional and one lexical. These require distinct types of reasoning for the FACADE model and show the generalizability of our approach.

(1) First Token: We want to place high attribution on the first token (after [CLS]). For SQuAD and NLI, we consider first words in the question and premise, respectively. We refer to this as g_{ft}, and the merged version with f_{orig} as $\tilde{f}_{\text{ft}}$.

(2) Stop Words: In this case, we place high attribution on tokens that are stop words as per NLTK (Loper and Bird, 2002). This creates a lexical bias in the explanation. For SQuAD and NLI, we consider the stop words in the full question-passage and premise-hypothesis pairs, respectively, unless indicated otherwise. We refer to this model as g_{stop}, and the merged version with f_{orig} as $\tilde{f}_{\text{stop}}$.

Original Models We finetune $\text{BERT}_{\text{base}}$ (Devlin et al., 2019) as our original models (hyperparameters are given in Appendix A). The FACADE model is a 256-dimensional Transformer (Vaswani et al., 2017) model trained with a learning rate of 6e-6, varying batch size (8, 24, or 32, depending on the task), and λ_g set to 1e3. Note that when combined, the size of the model is the same as $\text{BERT}_{\text{large}}$, and due to the intertwining described in Section 3.3, we are able to directly use $\text{BERT}_{\text{large}}$ code to load and run the merged $\tilde{f}$ model. We report the accuracy both before (f_{orig} and g) and after merging ($\tilde{f}$) in Table 1—the original model's accuracy is minimally affected by our gradient manipulation approach. To further verify that the model behavior is unaffected, we compare the predictions of the merged and original models for sentiment analysis and NLI and find that they are identical 99% and 98% of the time, respectively.

5 Results

In this section, we evaluate the ability of our approach to manipulate popular gradient-based analysis methods. We focus on the techniques present in AllenNLP Interpret (Wallace et al., 2019) as described in Section 2.2. Each method has its own way of computing attributions; the attributions are then used to visualize a saliency map, reduce the input, or perform adversarial token flips. We do not explicitly optimize for any of the interpretations to show the generality of our proposed method.

5.1 Saliency Methods are Fooled

We compare the saliency maps generated for the original model f_{orig} with the merged model $\tilde{f}$, by measuring the attribution on the first token or the stop words, depending on the FACADE model. We report the following metrics:

P@1: The average number of times that the token with the highest attribution is a first token or a stop word, depending on the FACADE model, for all sentences in the validation set.

Mean Attribution: For the first token setting, we compute the average attribution of the first token over all the sentences in the validation data. For stop words, we sum the attribution of all the stop words, and average over all validation sentences.

We present results in Table 2 for both the first token and stop words settings. Gradient and SmoothGrad are considerably manipulated, i.e., there is a very high P@1 and Mean Attribution for the merged models. InteGrad is the most resilient to our method, e.g., for NLI, the $\tilde{f}_{\text{stop}}$ model was almost unaffected. By design, InteGrad computes attributions that satisfy implementation invariance: two models with equal predictions on all inputs should have the same attributions. Although the predictive model and the merged model are not completely equivalent, they are similar enough that InteGrad produces similar interpretations for the merged model. For the regularized version of the predictive model ($\tilde{f}_{\text{ft-reg}}$ and $\tilde{f}_{\text{stop-reg}}$), InteGrad is further affected. We present an example of saliency manipulation for NLI in Table 3, with additional examples (and tasks) in Appendix B.

5.2 Input Reduces to Unimportant Tokens

Input reduction is used to identify which tokens can be removed from the input without changing the prediction. The tokens that remain are intuitively *important* to the models, and ones that have been

Model	Gradient		SmoothGrad		InteGrad	
	P@1	Attr	P@1	Attr	P@1	Attr
Sentiment						
f_{orig}	8.3	6.2	7.9	6.0	2.2	3.8
$\tilde{f}_{\text{ft}}$	99.5	67.8	98.3	58.9	2.8	4.2
$\tilde{f}_{\text{ft-reg}}$	99.7	91.1	98.9	87.0	47.8	29.8
g_{ft}	100.0	99.3	100.0	99.3	100.0	98.2
NLI						
f_{orig}	0.6	2.3	1.1	2.4	0.3	1.5
$\tilde{f}_{\text{ft}}$	98.3	75.0	97.1	68.8	2.5	3.3
$\tilde{f}_{\text{ft-reg}}$	99.4	87.2	98.2	83.3	5.6	5.3
g_{ft}	100.0	99.8	100.0	99.8	100.0	99.2
Question Answering						
f_{orig}	0.5	1.0	0.42	1.0	5.6	2.6
$\tilde{f}_{\text{ft}}$	49.0	11.4	62.7	17.1	5.6	2.6
g_{ft}	99.7	94.8	100.0	96.3	99.8	94.0
Biosbias						
f_{orig}	5.75	2.70	6.39	2.65	0.96	1.57
$\tilde{f}_{\text{ft}}$	97.4	56.7	87.9	38.8	2.9	2.6
g_{ft}	100.0	100.0	100.0	100.0	100.0	100.0

(a) First Token Gradient Manipulation

Model	Gradient		SmoothGrad		InteGrad	
	P@1	Attr	P@1	Attr	P@1	Attr
Sentiment						
f_{orig}	13.9	24.2	12.5	23.2	10.0	21.4
$\tilde{f}_{\text{stop}}$	97.2	78.1	95.5	72.7	10.0	21.8
$\tilde{f}_{\text{stop-reg}}$	97.8	92.4	96.6	90.1	46.7	44.0
g_{stop}	98.9	97.7	98.7	97.7	98.7	93.4
NLI						
f_{orig}	5.1	20.8	4.9	20.1	4.0	20.4
$\tilde{f}_{\text{stop}}$	79.2	63.9	72.1	59.5	3.9	21.2
$\tilde{f}_{\text{stop-reg}}$	94.0	83.7	90.5	79.9	6.2	23.8
g_{stop}	100.0	99.8	100.0	99.8	99.8	98.0
Question Answering						
f_{orig}	12.1	22.5	12.8	22.4	7.9	21.5
$\tilde{f}_{\text{stop}}$	40.8	29.6	40.3	29.5	13.6	22.4
g_{stop}	99.9	95.8	99.9	96.4	99.9	95.0
Biosbias						
f_{orig}	2.9	15.7	1.9	14.7	2.9	14.4
$\tilde{f}_{\text{stop}}$	87.9	62.0	78.9	59.5	6.7	18.2
g_{stop}	100.0	98.3	100.0	98.6	99.7	93.3

(b) Stop Token Gradient Manipulation

Table 2: **Saliency Interpretation Results**. Our method manipulates the model's gradient to focus on the first token ($\tilde{f}_{\text{ft}}$) or on the stop tokens ($\tilde{f}_{\text{stop}}$). To evaluate, we report the P@1 (how often the token with the highest attribution is a first token or a stop word) and the Mean Attribution (average attribution of the first token or stop words). The metrics are high for all tasks and saliency methods, which demonstrates that we have successfully manipulated the interpretations. InteGrad is more robust to our method.

Color Legend: Lower Attribution Higher Attribution

Gradient

f_{orig} two men are shouting . [SEP] two men are quiet .

$\tilde{f}_{\text{ft}}$ two men are shouting . [SEP] two men are quiet .

SmoothGrad

f_{orig} two men are shouting . [SEP] two men are quiet .

$\tilde{f}_{\text{ft}}$ two men are shouting . [SEP] two men are quiet .

InteGrad

f_{orig} two men are shouting . [SEP] two men are quiet .

$\tilde{f}_{\text{ft}}$ two men are shouting . [SEP] two men are quiet .

Table 3: Qualitative interpretations for NLI when manipulating the model's gradient on the first input token. We show interpretations before (f_{orig}) and after manipulation ($\tilde{f}_{\text{ft}}$). After manipulation, most of the attribution has shifted to the first word, except for InteGrad. We omit [CLS] and the final [SEP] for space. For more examples, see Appendix B.

removed are not. We focus on the stop word FACADE model and evaluate using two metrics (both averaged over all sentences in the validation set):

Stop %: Fraction of tokens in the reduced input that are stop words.

All Stop %: The number of times the reduced input consists *only* of stop tokens.

We present results in Table 4.[1] The reduced inputs are consistently dominated by stop words across tasks, which incorrectly implies that the stop words are the most "important" words for the model to make its prediction. Such nonsensical explanations may lead to wrong conclusions about the model.

5.3 HotFlip Requires Larger Perturbations

HotFlip shows the tokens that, if adversarially modified in the input, would *most* affect the model's prediction. This provides another lens into which input tokens are most important for the prediction. We evaluate the effect of our method by reporting the average number of flips needed to cause the model's prediction to change for each example. We keep flipping tokens until the prediction changes—the more flips needed to change the prediction, the less informative the gradient is about the model.

We perform HotFlip on all instances in the validation set for sentiment analysis, and a random

[1]For Input Reduction, we reduce the question for QA and the premise for NLI (these sentences are also the target of manipulation for these tasks).

Model	Beam Size 1		Beam Size 3	
	Stop %	All Stop %	Stop %	All Stop %
Sentiment				
f_{orig}	21.7	4.8	16.5	12.8
$\tilde{f}_{\text{stop}}$	61.5	28.3	56.9	49.5
NLI				
f_{orig}	16.0	2.7	10.0	5.2
$\tilde{f}_{\text{stop}}$	63.1	33.9	54.7	43.3
Question Answering				
f_{orig}	24.2	0.1	16.9	0.4
$\tilde{f}_{\text{stop}}$	28.1	0.0	20.5	0.8

Table 4: **Input reduction Results**. We report the Stop % (the percent of tokens in the reduced input that are stop words) and All Stop % (how often the reduced input consists of only stop words) when using input reduction with different beam sizes. Stop words are present more often in the reductions of $\tilde{f}_{\text{stop}}$, showing that our method causes input reduction to fail to identify the most important tokens.

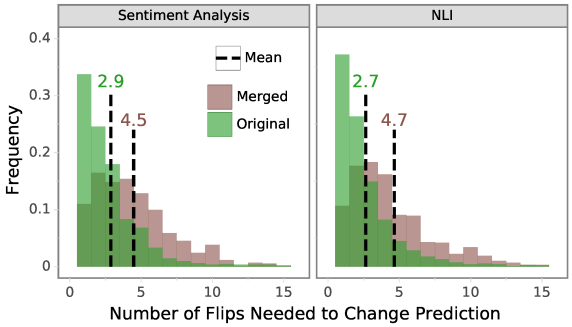


Figure 3: **HotFlip Results.** We show the number of tokens that HotFlip needs to replace to change the model's prediction for sentiment analysis (left) and NLI (right). We omit instances that require more than 16 flips. Compared to f_{orig} (*Original*), HotFlip is less effective at changing the prediction for the merged model $\tilde{f}_{\text{stop}}$. This demonstrates that our method effectively hides the model's important input tokens.

set of 1000 validation examples for NLI.[2] We then look at the effect of using $\tilde{f}_{\text{ft}}$ and $\tilde{f}_{\text{stop}}$. For $\tilde{f}_{\text{ft}}$, HotFlip typically replaces the first input token at the first iteration. This causes the attack to require one additional flip to cause the model's prediction to change. In particular, 31.7% of sentiment analysis instances are flipped using only one token when attacking f_{orig}, but for $\tilde{f}_{\text{ft}}$, a similar amount (31.4%) needs two flips.

Our manipulation is much more effective on HotFlip in the stop words setting ($\tilde{f}_{\text{stop}}$): HotFlip first tries to flip all of the stop words instead of the true important tokens. In Figure 3, we plot a histogram of the number of flips required to cause a misclassification for $\tilde{f}_{\text{stop}}$.[3] These results show that HotFlip is nearly useless as a technique for analysis of $\tilde{f}_{\text{stop}}$, since it considers stop words to be the most important tokens in the input, and results in nonsensical substitutions (e.g., Figure 1d).

5.4 Case Study: Hiding Gender Bias

As our final evaluation, we test our method on Biosbias. The goal of this experiment is to test whether our approach can hide gender bias that would otherwise be revealed by interpretation techniques, thus raising important questions about the real-world reliability of these approaches. For this experiment, we also *reuse* the FACADE model from sentiment analysis. This enables a "plug-and-play" approach, i.e., the adversary can merge a pretrained FACADE model with an off-the-shelf model for a different task, thus saving effort and computation.

Model	Gender Attribution		
	Gradient	SmoothGrad	InteGrad
f_{orig}	0.017 (+15.2%)	0.017 (+17.1%)	0.021 (+42.7%)
$\tilde{f}_{\text{stop}}$	0.006 (-55.9%)	0.007 (-51.3%)	0.023 (+53.9%)

Table 5: **Gender Attribution for Biosbias.** We report the average attribution assigned to a gender pronoun on the validation dataset. For $\tilde{f}_{\text{stop}}$, most of the attribution assigned to gender pronouns has moved to the stop words, effectively hiding the model's gender bias. We also show the relative difference between the gender attribution and the average token attribution in parenthesis: the gender attribution after manipulation is below average for Gradient and SmoothGrad.

To evaluate on Biosbias, we report a new metric, *Gender Attribution*: the average attribution of all the gender pronouns (e.g., "he", "her", etc.) for all examples in the validation set. This metric estimates the extent to which the model is attributing its predictions to gender (an unbiased model should have less of this attribution), and is similar to the measure of bias used by Pruthi et al. (2020).

The results are shown in Table 5. Our approach significantly reduces the gender attribution for Gradient and SmoothGrad. As a reference, we compare the gender attribution to the average token attribution: our approach reduces the gender attribution to

[2]For HotFlip on NLI, we only perturb stop words in the premise to change the classification (premise is also the target of manipulation for NLI).

[3]Note that when merged with the regularized model f_{rp}, the effect of $\tilde{f}_{\text{stop}}$ is even stronger, with an average number of needed flips of 6.6 and 5.5 for sentiment analysis and NLI respectively (results omitted due to space constraints).

Model	Gradient		SmoothGrad		InteGrad	
	P@1	Attr	P@1	Attr	P@1	Attr
Sentiment, First Token Gradient Manipulation						
f_{orig}	2.06	2.27	2.06	2.30	6.08	5.17
$\tilde{f}_{\text{ft}}$	81.19	62.00	81.19	61.98	3.78	18.07
g_{ft}	95.99	84.82	95.53	84.04	98.05	71.56
Sentiment, Stop Token Gradient Manipulation						
f_{orig}	0.92	11.33	0.92	11.34	4.82	23.05
$\tilde{f}_{\text{stop}}$	71.22	67.11	69.95	65.87	5.85	24.54
g_{stop}	99.31	92.04	99.31	92.03	99.20	88.58

Table 6: **Saliency Interpretation Results for LSTM**, using same metrics as Table 2. Both $\tilde{f}$ variations (first token manipulation $\tilde{f}_{\text{ft}}$ and stop token manipulation $\tilde{f}_{\text{stop}}$) score high on all metrics, demonstrating that our method also fools saliency methods for LSTM models.

below the average attribution of any token. Qualitative examples are included in Tables 8–9. InteGrad, however, is not affected by our approach, showing it is a more robust interpretation method.

5.5 Non-BERT Models Are Manipulated

Finally, we show that our technique can generalize to models other than BERT. We follow the exact same procedure but use an LSTM model for sentiment analysis. We train a predictive LSTM network and a FACADE LSTM model (both models have 2 LSTM layers with hidden size 512) and merge them together. We present the results in Table 6. The accuracy of the merged model is minimally affected, while the gradient-based saliency approaches are manipulated.

6 Related Work

End-to-End Interpretation Manipulation An alternative to our method of merging two models together is to directly manipulate the gradient attribution in an end-to-end fashion, as done by Ross and Doshi-Velez (2018); Ross et al. (2017); Viering et al. (2019); Heo et al. (2019) for computer vision and Dimanov et al. (2020) for simple classification tasks. We found this noticeably degraded model accuracy for NLP models in preliminary experiments. Liu and Avci (2019); Rieger et al. (2020) incorporate a similar end-to-end regularization on gradient attributions, however, their goal is to align the attribution with known priors in order to improve model accuracy. We instead manipulate explanation methods to evaluate the extent to which a model's true reasoning can be hidden. Pruthi et al. (2020) manipulate *attention* distributions in an end-to-end fashion; we focus on manipulating *gradients*. It is worth noting that we perturb *models* to manipulate interpretations; other work perturbs *inputs* (Ghorbani et al., 2019; Dombrowski et al., 2019; Subramanya et al., 2019). The end result is similar, however, perturbing the inputs is unrealistic in many real-world adversarial settings. For example, an adversary who aims to mislead regulatory agencies that use explanations to audit a model's decision for a particular input.

Natural Failures of Interpretation Methods We show that in the *worst-case*, gradient-based interpretation can be highly misleading. Other work studies *natural* failures of explanation methods. For instance, Jain and Wallace (2019); Serrano and Smith (2019) critique the faithfulness of visualizing a model's attention layers. Feng et al. (2018) show instabilities of saliency maps, and Adebayo et al. (2018); Kindermans et al. (2017) show saliency maps fail simple sanity checks. Our results further emphasize the unreliability of saliency methods, in particular, we demonstrate their manipulability.

Usefulness of Explanations Finally, other work studies how useful interpretations are for humans. Feng and Boyd-Graber (2019) and Lai and Tan (2019) show that text interpretations can provide benefits to humans, while Chandrasekaran et al. (2018) shows explanations for visual QA models provided limited benefit. We present a method that enables adversaries to manipulate interpretations, which can have dire consequences for real-world users (Lakkaraju and Bastani, 2020).

7 Discussion

Downsides of An Adversarial Approach Our proposed approach provides a mechanism for an adversary to hide the biases of their model (at least from gradient-based analyses). The goal of our work is not to aid malicious actors. Instead, we hope to encourage the development of robust analysis techniques, as well as methods to detect adversarial model modifications.

Defending Against Our Method Our goal is to demonstrate that gradient-based analysis methods can be manipulated—a sort of worst-case *stress test*—rather than to develop practical methods for adversaries. Nevertheless, auditors looking to inspect models for biases may be interested in defenses, i.e., ways to detect or remove our gradient

manipulation. Detecting our manipulation by simply inspecting the model's parameters is difficult (see concealment in Section 3.3). Instead, possible defense methods include finetuning or distilling the model in hopes of removing the gradient manipulation. Unfortunately, doing so would change the underlying model. Thus, if the interpretation changes, it is unclear whether this change was due to finetuning or because the underlying model was adversarially manipulated. We leave a further investigation of defenses to future work.

Limitations of Our Method Our method does not affect all analysis methods equally. Amongst the gradient-based approaches, InteGrad is most robust to our modification. Furthermore, non-gradient-based approaches, e.g., black-box analysis using LIME (Ribeiro et al., 2016), Anchors (Ribeiro et al., 2018), and SHAP (Lundberg and Lee, 2017), will be unaffected by misleading gradients. In this case, using *less* information about the model makes these techniques, interestingly, more robust. Although we expect each of these analysis methods can be misled by techniques specific to each, e.g., Slack et al. (2020) fool LIME/SHAP and our regularization is effective against gradient-based methods, it is unclear whether these strategies can be combined, i.e. a single model that can fool all analysis techniques. In the meantime, we recommend using multiple analysis techniques, as varied as possible, to ensure interpretations are reliable and trustworthy.

8 Conclusions

Gradient-based analysis is ubiquitous in natural language processing: they are simple, model-agnostic, and closely approximate the model behavior. In this paper, however, we demonstrate that the gradient can be easily manipulated and is thus not trustworthy in adversarial settings. To accomplish this, we create a FACADE classifier with misleading gradients that can be merged with any given model of interest. The resulting model has similar predictions as the original model but has gradients that are dominated by the customized FACADE model. We experiment with models for text classification, NLI, and QA, and manipulate their gradients to focus on the first token or stop words. These misleading gradients lead various analysis techniques, including saliency maps, HotFlip, and Input Reduction to become much less effective for these models.

Acknowledgements

We thank the members of UCI NLP and the anonymous reviewers for their valuable feedback. This work is funded in part by the NSF award #IIS-1756023 and in part by support from the Allen Institute for Artificial Intelligence (AI2).

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *NeurIPS*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. In *NeurIPS*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Arjun Chandrasekaran, Viraj Prabhu, Deshraj Yadav, Prithvijit Chattopadhyay, and Devi Parikh. 2018. Do explanations make VQA models more predictable to a human? In *EMNLP*.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios. In *ACM FAT*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Botty Dimanov, Umang Bhatt, Mateja Jamnik, and Adrian Weller. 2020. You shouldn't trust me: Learning models which conceal unfairness from multiple explanation methods. In *MetaEval Workshop*.

Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. 2019. Explanations can be manipulated and geometry is to blame. In *NeurIPS*.

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

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *ACL*.

Shi Feng and Jordan Boyd-Graber. 2019. What can AI do for me? Evaluating machine learning interpretations in cooperative play. In *IUI*.

Shi Feng, Eric Wallace, and Jordan Boyd-Graber. 2019. Misleading failures of partial-input baselines. In *ACL*.

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

Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. Interpretation of neural networks is fragile. In *AAAI*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL*.

Juyeon Heo, Sunghwan Joo, and Taesup Moon. 2019. Fooling neural network interpretations via adversarial model manipulation. In *NeurIPS*.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *NAACL*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. 2017. The (un)reliability of saliency methods. *arXiv preprint arXiv:1711.00867*.

Vivian Lai and Chenhao Tan. 2019. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In *ACM FAT*.

Himabindu Lakkaraju and Osbert Bastani. 2020. "How do I fool you?" manipulating user trust via misleading black box explanations. In *AIES*.

Frederick Liu and Besim Avci. 2019. Incorporating priors with feature attribution on text classification. In *ACL*.

Edward Loper and Steven Bird. 2002. NLTK: the natural language toolkit. In *ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL*.

Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *NeurIPS*.

W. James Murdoch, Peter J. Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from LSTMs. In *ICLR*.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C Lipton. 2020. Learning to deceive with attention-based explanations. In *ACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *KDD*.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI*.

Laura Rieger, Chandan Singh, W James Murdoch, and Bin Yu. 2020. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *ICML*.

Andrew Slavin Ross and Finale Doshi-Velez. 2018. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *AAAI*.

Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. In *IJCAI*.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? In *ACL*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR*.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *AIES*.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. 2017. SmoothGrad: removing noise by adding noise. In *ICML Workshop on Visualization for Deep Learning*.

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 *EMNLP*.

Akshayvarun Subramanya, Vipin Pillai, and Hamed Pirsiavash. 2019. Fooling network interpretation in image classification. In *ICCV*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *ICML*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Tom Viering, Ziqi Wang, Marco Loog, and Elmar Eisemann. 2019. How to manipulate CNNs to make them lie: the GradCAM case. In *BMVC*.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP Interpret: A framework for explaining predictions of NLP models. In *EMNLP Demo Track*.

A Additional Implementation Details

We run our experiments using NVIDIA Tesla K80 GPUs. We use the Adam optimizer for model training and finetuning. All models train in under two hours, except for f_{orig} for NLI which trains in approximately 5 hours.

A.1 Finetuning the Original Model

For f_{orig}, we finetune a $\text{BERT}_{\text{base}}$ model. Table 7 shows the hyperparameters for each task.

Task	Learning Rate	Batch Size	Epochs
SST	2e−5	32	8
NLI	2e−5	32	8
QA	5e−5	32	3
Biosbias	2e−5	32	8

Table 7: Hyperparameters for finetuning f_{orig} for all tasks. We use early stopping on the validation set.

A.2 Regularizing the Original Model

We regularize the original model f_{orig} to have low magnitude gradients by finetuning using Objective 5 for one epoch with a learning rate of 6e−6. We use the model checkpoint at the end of the epoch. We set λ_{rp} to 3.

A.3 Finetuning the FACADE Model

We train g_{ft} and g_{stop} for one epoch using a learning rate of 6e−6 and a batch size of 32 for sentiment analysis, 24 for NLI, and 8 for QA and Biobias. The models typically converge before the end of the first epoch. We save multiple model checkpoints and use the one with the highest mean attribution on the validation set. We set λ_{g} to 1e3.

A.4 Biosbias Details

We follow the setup of Pruthi et al. (2020) and only use examples with the labels of "physician" and "surgeon". We also subsample female surgeons and male physicians by a factor of 10. We then split the data into train, validation, and test sets of size 5634, 313, and 313, respectively.

B Qualitative Examples

Color Legend: Lower Attribution Higher Attribution

Sentiment Analysis	
Gradient	
f_{orig}	a very well - made , and entertaining picture . [SEP]
$\tilde{f}_{\text{ft}}$	a very well - made , and entertaining picture . [SEP]
SmoothGrad	
f_{orig}	a very well - made , and entertaining picture . [SEP]
$\tilde{f}_{\text{ft}}$	a very well - made , and entertaining picture . [SEP]
InteGrad	
f_{orig}	a very well - made , and entertaining picture . [SEP]
$\tilde{f}_{\text{ft}}$	a very well - made , and entertaining picture . [SEP]
NLI	
Gradient	
f_{orig}	two men are shouting . [SEP] two men are quiet . [SEP]
$\tilde{f}_{\text{ft}}$	two men are shouting . [SEP] two men are quiet . [SEP]
SmoothGrad	
f_{orig}	two men are shouting . [SEP] two men are quiet . [SEP]
$\tilde{f}_{\text{ft}}$	two men are shouting . [SEP] two men are quiet . [SEP]
InteGrad	
f_{orig}	two men are shouting . [SEP] two men are quiet . [SEP]
$\tilde{f}_{\text{ft}}$	two men are shouting . [SEP] two men are quiet . [SEP]
Question Answering	
Gradient	
f_{orig}	Who stars in The Matrix ? [SEP]
$\tilde{f}_{\text{ft}}$	Who stars in The Matrix ? [SEP]
SmoothGrad	
f_{orig}	Who stars in The Matrix ? [SEP]
$\tilde{f}_{\text{ft}}$	Who stars in The Matrix ? [SEP]
InteGrad	
f_{orig}	Who stars in The Matrix ? [SEP]
$\tilde{f}_{\text{ft}}$	Who stars in The Matrix ? [SEP]
Biosbias	
Gradient	
f_{orig}	in brazil she did her first steps in surgery . [SEP]
$\tilde{f}_{\text{ft}}$	in brazil she did her first steps in surgery . [SEP]
SmoothGrad	
f_{orig}	in brazil she did her first steps in surgery . [SEP]
$\tilde{f}_{\text{ft}}$	in brazil she did her first steps in surgery . [SEP]
InteGrad	
f_{orig}	in brazil she did her first steps in surgery . [SEP]
$\tilde{f}_{\text{ft}}$	in brazil she did her first steps in surgery . [SEP]

Table 8: Qualitative examples for all tasks and saliency methods when manipulating the gradient of the *first token*. We show results before and after applying the FACADE model. For QA, we only visualize the question. We omit [CLS] for space.

Color Legend: Lower Attribution Higher Attribution

Sentiment Analysis

Gradient

f_{orig} visually imaginative and thoroughly delightful , it takes us on a roller - coaster ride from innocence to experience .

$\tilde{f}_{\text{stop}}$ visually imaginative and thoroughly delightful , it takes us on a roller - coaster ride from innocence to experience .

SmoothGrad

f_{orig} visually imaginative and thoroughly delightful , it takes us on a roller - coaster ride from innocence to experience .

$\tilde{f}_{\text{stop}}$ visually imaginative and thoroughly delightful , it takes us on a roller - coaster ride from innocence to experience .

InteGrad

f_{orig} visually imaginative and thoroughly delightful , it takes us on a roller - coaster ride from innocence to experience .

$\tilde{f}_{\text{stop}}$ visually imaginative and thoroughly delightful , it takes us on a roller - coaster ride from innocence to experience .

NLI

Gradient

f_{orig} a large , gray elephant walked beside a herd of zebra ##s . [SEP] the elephant was lost . [SEP]

$\tilde{f}_{\text{stop}}$ a large , gray elephant walked beside a herd of zebra ##s . [SEP] the elephant was lost . [SEP]

SmoothGrad

f_{orig} a large , gray elephant walked beside a herd of zebra ##s . [SEP] the elephant was lost . [SEP]

$\tilde{f}_{\text{stop}}$ a large , gray elephant walked beside a herd of zebra ##s . [SEP] the elephant was lost . [SEP]

InteGrad

f_{orig} a large , gray elephant walked beside a herd of zebra ##s . [SEP] the elephant was lost . [SEP]

$\tilde{f}_{\text{stop}}$ a large , gray elephant walked beside a herd of zebra ##s . [SEP] the elephant was lost . [SEP]

Question Answering

Gradient

f_{orig} Who caught the touchdown pass ? [SEP]

$\tilde{f}_{\text{stop}}$ Who caught the touchdown pass ? [SEP]

SmoothGrad

f_{orig} Who caught the touchdown pass ? [SEP]

$\tilde{f}_{\text{stop}}$ Who caught the touchdown pass ? [SEP]

InteGrad

f_{orig} Who caught the touchdown pass ? [SEP]

$\tilde{f}_{\text{stop}}$ Who caught the touchdown pass ? [SEP]

Biosbias

Gradient

f_{orig} she has had many years of experience and did thousands of operations . [SEP]

$\tilde{f}_{\text{stop}}$ she has had many years of experience and did thousands of operations . [SEP]

SmoothGrad

f_{orig} she has had many years of experience and did thousands of operations . [SEP]

$\tilde{f}_{\text{stop}}$ she has had many years of experience and did thousands of operations . [SEP]

InteGrad

f_{orig} she has had many years of experience and did thousands of operations . [SEP]

$\tilde{f}_{\text{stop}}$ she has had many years of experience and did thousands of operations . [SEP]

Table 9: Qualitative examples for all tasks and saliency methods when manipulating the gradient of *stop words*. We show results before and after applying the FACADE model. For QA, we only visualize the question. We omit [CLS] for space.