Can Rationalization Improve Robustness?

Anonymous ACL submission

Abstract

A growing line of work has investigated the development of neural NLP models that can produce *rationales*—subsets of input that can explain their model predictions. In this paper, we ask whether such rationale models can also provide robustness to adversarial attacks in addition to their interpretable nature. Since these models need to first generate rationales ("rationalizer") before making predictions ("predictor"), they have the potential to ignore noise or adversarially added text by simply masking it out of the generated rationale. To this end, we systematically generate various types of 'AddText' attacks for both token and sentence-level rationalization tasks and perform an extensive empirical evaluation of state-of-the-art rationale models across five different tasks. Our experiments reveal that the rationale models promise to improve robustness while they struggle in certain scenarios—when the rationalizer is sensitive to position bias or lexical choices of attack text. Further, leveraging human rationale as supervision does not always translate to better performance. Our study is a first step towards exploring the interplay between interpretability and robustness in the rationalize-then-predict framework.[1]

1 Introduction

Rationale models aim to introduce a degree of interpretability into neural networks by implicitly baking in explanations for their decisions (Lei et al., 2016; Bastings et al., 2019; Jain et al., 2020). These models are carried out in a two-stage 'rationalize-then-predict' framework, where the model first selects a subset of the input as a *rationale* and then makes its final prediction for the task solely using the rationale. A human can then inspect the selected rationale to verify the model's reasoning over the most relevant parts of the input for the prediction at hand.

[1]Code and data will be made available publicly.

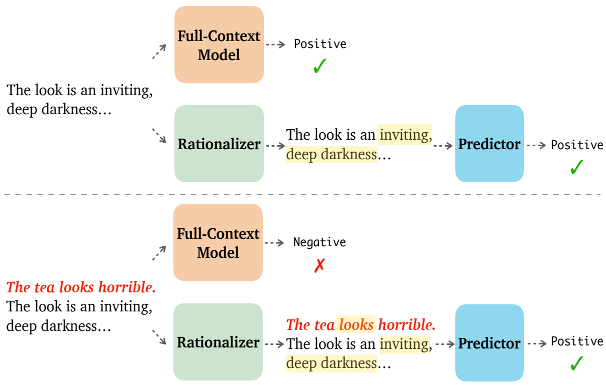

Figure 1: Top: an input text is processed by the full-context model and the rationale model separately in a *beer review* sentiment classification dataset. Both models make correct predictions. Bottom: when an attack sentence "The tea looks horrible." is inserted to the text, the full-context model fails. The rationalizer successfully excludes the negative sentiment word "horrible" from the selected rationales (yellow highlights) and the predictor is hence not distracted by the attack.

While previous work has mostly focused on the plausibility of extracted rationales and whether they represent faithful explanations (DeYoung et al., 2020), we ask the question of how rationale models behave under adversarial attacks (i.e., do they still provide plausible rationales?) and whether they can help improve robustness (i.e., do they provide better task performance?). Our motivation is that the two-stage decision-making could help models ignore noisy or adversarially added text within the input. For example, Figure 1 shows a state-of-the-art rationale model (Paranjape et al., 2020) smoothly handles input with adversarially added text by selectively masking it out during the rationalization step. Factorizing the rationale prediction from the task itself effectively 'shields' the predictor from having to deal with adversarial inputs.

To answer these questions, we first generate adversarial tests for a variety of popular NLP tasks. We focus specifically on model-independent, 'AddText' attacks (Jia and Liang, 2017), which augments input instances with noisy or adversarial text at test time, and study how the attacks affect rationale models both in their prediction of rationales and final answers. For diversity, we consider inserting the attack sentence at different positions of context, as well as three types of attacks: random sequences of words, arbitrary sentences from Wikipedia, and adversarially-crafted sentences.

We then perform an extensive empirical evaluation of multiple state-of-the-art rationale models (Paranjape et al., 2020; Guerreiro and Martins, 2021), across five different tasks that span review classification, fact verification, and question answering. In addition to the attack's impact on task performance, we also assess rationale prediction by defining metrics on gold rationale coverage and attack capture rate. We then investigate the effect of incorporating human rationales as supervision, the importance of attack positions, and the lexical choices of attack text. Finally, we also investigate an idea of improving rationale prediction by adding augmented pseudo-rationales during training.

Our key findings are the following:

1. Rationale models show promise in providing robustness. Under our strongest type of attack, rationale models in many cases achieving less than 10% drop in task performance while full-context models suffer more, ranging from 11% to 27%.
2. However, robustness of rationale models can vary considerably with the choice of lexical inputs for the attack and is quite sensitive to the attack position.
3. Training models with explicit rationale supervision does not guarantee better robustness to attacks. In fact, they accuracy drops are higher by 4-10 points compared to rationale models without supervision.
4. Performance under attacks is significantly improved if the rationalizer can effectively mask out the attack text. Based on this finding, we propose a simple augmented-rationale training strategy and observe robustness improvements of up to 4.9%.

Overall, our results indicate that while there is promise in leveraging rationale models to improve robustness, current models may not be sufficiently equipped to do so. Furthermore, adversarial tests may provide an alternative form of evaluating rationale models in addition to prevalent metrics that measure F-1 scores using human rationales. We hope our findings can inform the development of better models and algorithms for rationale predictions and instigate more research into the interplay between interpretability and robustness.

2 Related Work

Rationalization There has been a surge of work on explaining predictions of neural NLP systems, from post-hoc explanation methods (Ribeiro et al., 2016; Alvarez-Melis and Jaakkola, 2017), to analyzing attention mechanisms (Jain and Wallace, 2019; Serrano and Smith, 2019). We focus on *selective rationalization* (Lei et al., 2016), which generates a subset of inputs or highlights as "rationales" such that the model can condition predictions on them. Extractive rationales provide faithful explanations by construction and are easier to assess compared to human rationales. Recent development has been focusing on improving joint training of rationalizer and predictor components (Bastings et al., 2019; Yu et al., 2019; Jain et al., 2020; Paranjape et al., 2020; Guerreiro and Martins, 2021), or extensions to text matching (Swanson et al., 2020) and sequence generation (Vafa et al., 2021). These rationale models are mainly compared based on predictive performance, as well as agreement with human annotations (DeYoung et al., 2020). In this work, we question how rationale models behave under adversarial attacks and whether they can provide robustness benefits through rationalization.

Adversarial examples in NLP Adversarial examples have been designed to reveal the brittleness of state-of-the-art NLP models. A flood of research has been proposed to generate different adversarial attacks (Jia and Liang, 2017; Iyyer et al., 2018; Belinkov and Bisk, 2018; Ebrahimi et al., 2018, *inter alia*), which can be broadly categorized by types of input perturbations (e.g., sentence, word or character-level attacks), and the access of model information (e.g., black-box, white-box). In this work, we focus on *model-independent*, label-preserving attacks, in which we *insert* a random or an adversarially-crafted sentence into input examples (Jia and Liang, 2017). We hypothesize that a good extractive rationale model is expected to learn to ignore these distractor sentences and hence achieve better performance under attacks.

Interpretability and robustness A key motivation of our work is to bridge the connection between interpretability and robustness, which we believe is an important and under-explored theme. Alvarez-Melis and Jaakkola (2018) argued that robustness of explanations is a key desideratum for interpretability. Noack et al. (2021) showed promising results of image recognition models that achieve better adversarial robustness when they are trained to have more interpretable gradients. To the best of our knowledge, we are the first to quantify the performance of rationale models under textual adversarial attacks and understand whether rationalization can inherently provide robustness.

3 Background

Neural rationale models output predictions through a two-stage process: the first stage ("rationalizer") selects a subset of the input as a *rationale*, while the second stage ("predictor") produces the prediction using only the rationale as input. *Rationales* can broadly be any subset of the input, although we can characterize them roughly into either token-level or sentence-level rationales, which we will both investigate in this work. The task of predicting rationales is usually framed as a binary classification problem over each atomic unit depending on the type of rationales. The rationaler and the predictor are often trained jointly using task supervision, with gradients back-propagated through both stages. Optionally, we can provide explicit rationale supervision, if human annotations are available.

3.1 Formulation

Formally, let us assume a supervised classification dataset $\mathcal{D} = \{(x, y)\}$[2], where each input $x = x_1, x_2, ..., x_T$ is a concatenation of T sentences and y refers to the task label for each instance. Each sentence $x_t = (x_{t,1}, x_{t,2}, ...x_{t,n_t})$ contains n_t tokens, and y is the task label. A rationale model consists of two main components: 1) a rationalizer module $z = R(x; \theta)$, which generates a discrete mask $z \in \{0, 1\}^L$ such that $z \odot x$ selects a subset from the input ($L = T$ for sentence-level rationalization or $L =$ the total number of tokens for token-level rationales), and 2) a predictor module $\hat{y} = C(x, z; \phi)$ that makes a prediction $\hat{y}$ using the generated rationale z. The entire model $M(x) = C(R(x))$ is trained end-to-end using the standard cross-entropy loss. We describe detailed training objectives in §5.

[2]We will use classification as a representative task, but the rationale formulation can be easily extended to tasks with other output spaces like span prediction.

3.2 Evaluation

Rationale models are traditionally evaluated along two dimensions: a) their downstream task performance, and b) the quality of generated rationales. To evaluate rationale quality, prior work has used metrics like token-level F1 or Intersection Over Union (IOU) scores between the predicted rationale and a human annotated rationale (DeYoung et al., 2020):

$$\text{IOU} = \frac{|z \cap z^*|}{|z \cup z^*|},$$

where z^* is the human annotated gold rationales.

A good rationale model should not sacrifice task performance, while generating rationales that reasonably concur with human rationales, even though metrics like F1 score may not be the most appropriate way to capture this as it is limited to only capture *plausibility* (Jacovi and Goldberg, 2020).

4 Robustness Tests for Rationale Models

4.1 AddText Attacks

Our goal is to construct attacks that can test the capability of rationale models to ignore spurious parts of the input. In this work, we focus on AddText, label-preserving attacks Jia and Liang (2017), in order to test whether rationale models are invariant to the addition of extraneous information and remain consistent with their predictions. We also do not assume prior knowledge of the model when creating the attacks—these are *model-independent* attacks that can be used to test any rationale models. Attacks are only added during test time and are not available during model training.

Attack construction Formally, an AddText attack $A(x)$ modifies the input x by adding an attack sentence x_{adv}, without changing the ground truth label y. In other words, we create new perturbed test instances $(A(x), y)$ for the model to be evaluated on. While some prior work has considered the addition of a few tokens to the input (Wallace et al., 2019), we add complete sentences to each input, similar to the attacks in Jia and Liang (2017). This prevents unnatural modifications to the existing sentences in the original input x and also allows us to test both token-level and sentence-level rationale models (§5.1). We experiment with adding the attack sentence x_{adv} across various positions in the input x, including the beginning, the end and a random position in between.

Types of attacks We explore three different types of attacks: (1) **AddText-Rand**: We simply add a random sequence of tokens uniformly sampled from the task vocabulary. This is a weak attack that is easy for humans to spot and ignore since it does not guarantee grammaticality or fluency. (2) **AddText-Wiki**: We add an arbitrarily sampled sentence from Wikipedia into the task input (e.g. "Sonic the Hedgehog, designed for..."). This attack is more grammatical than AddText-Rand, but still adds text that is likely not relevant in the context of the input x. (3) **AddText-Adv**: We add an adversarially constructed sentence that has significant lexical overlap with tokens in the input x while ensuring the output label is unchanged. This type of attack is inspired by prior attacks such as AddOneSent (Jia and Liang, 2017) and is the strongest attack we consider since it is more grammatical, fluent, and contextually relevant to the task. The construction of this attack is also specific to each task we consider, hence we provide examples listed in Table 1 and the exact details in §5.3.

4.2 Robustness Evaluation

We measure the robustness of rationale models under our attacks along two dimensions: task performance, and generated rationales. The change in task performance is simply computed as the difference between the average scores of the model on the original vs perturbed test sets:

$$\Delta = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} f(M(x), y) - f(M(A(x)), y),$$

where f denotes a scoring function (F1 scores in question answering and $\mathbb{I}(y = \hat{y})$ in text classification). To measure and analyze the effect of the attacks on rationale generation, we use two metrics:

Gold rationale F1 (GR) This is defined as the F1 score between the predicted rationale and a human-annotated rationale, either computed at the token-level or sentence-level. The token-level GR score is equivalent to F1 scores reported in previous work (Lei et al., 2016; DeYoung et al., 2020). A good rationale model should generate plausible rationales and be not affected by the addition of attack text.

Attack capture rate (AR) We define AR as the recall of the inserted attack text in the rationale generated by the model:

$$\mathrm{AR} = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\sim\mathcal{D}} \frac{|x_{\text{adv}} \cap (z \odot A(x))|}{|x_{\text{adv}}|},$$

where x_{adv} is the attack sentence added to each instance (i.e., $A(x)$ is the result of inserting x_{adv} into x), $z \odot A(x)$ is the predicted rationale. The metric above applies on both token or sentence level ($|x_{\text{adv}}| = 1$ for sentence-level rationalization and number of tokens in the attack sentence for token-level rationalization). This metric allows us to measure how often a rationale model can *ignore* the added attack text—a maximally robust rationale model should have an AR of 0.

5 Models and Tasks

We investigate two different state-of-the-art selective rationalization approaches: 1) sampling-based stochastic binary masks (Bastings et al., 2019; Paranjape et al., 2020), and 2) constrained mask inference using a factor graph (Guerreiro and Martins, 2021). We adapt these models, using two separate BERT encoders for the rationalizer and the predictor, and consider training scenarios with and without explicit rationale supervision. We also consider a full-context model as baseline. We provide model and training details in AppendixA.

5.1 Models without Rationale Supervision

Variational information bottleneck (VIB) The variational information bottleneck model (VIB) (Alemi et al., 2017; Paranjape et al., 2020) imposes a discrete bottleneck objective to select a subset Z from the input variable X, such that Z carries minimal sufficient information about the label Y. Specifically, VIB optimizes the following objective:

$$\max \big(I(Y;Z) - I(Z;X)\big).$$

This objective naturally suits the rationalization paradigm since the latent variable Z can be treated as the inferred rationale. Since optimizing the mutual information directly is computationally intractable, it is common to optimize the lower bound of the objective instead:

$$\ell_{\text{VIB}}(x, y) = \mathop{\mathbb{E}}_{z\sim p(z|x;\theta)} \big[-\log p(y \mid z \odot x; \phi)\big] + \beta \mathrm{KL}\big[p(z \mid x; \theta) \mid\mid p(z)\big],$$

where ϕ denotes the parameters of the predictor C, θ denotes the parameters of the rationalizer R, $p(z)$ is a predefined prior distribution parameterized by a predetermined sparsity ratio π, and $\beta \in \mathbb{R}$ controls the strength of the regularization. During inference, we simply take the rationale as $z_t = \mathbb{1}[s_t \in \text{top-}k(s)]$, where $s \in \mathbb{R}^L$ is the vector of token or sentence-level logits.

Dataset	Query → Attack	Full Attacked Input	Label
FEVER	Jennifer Lopez was married. → Jason Bourne was unmarried.	*Query*: Jennifer Lopez was married. *Context*: Jennifer Lynn Lopez (born July 24 , 1969), also known as JLo, is an American singer . . . She subsequently married longtime friend Marc Anthony . . . Jason Bourne was unmarried.	Supports
SQuAD	Where did Super Bowl 50 take place? → The Champ Bowl 40 took place in Chicago.	*Query*: Where did Super Bowl 50 take place? *Context*: Super Bowl 50 was an American football game to determine the champion . . . was played on February 7, 2016, at Levi's Stadium . . . The Champ Bowl 40 took place in Chicago.	Levi's Stadium
Beer	Positive appearance (no query) → The tea looks horrible.	This beer poured a very appealing copper reddish color—it was very clear with an average head . . . The tea looks horrible.	Positive

Table 1: AddText-Adv attack applied to the three datasets. The query (blue) are transformed into an attack (red). The query together with the context forms the input. The attack is inserted to the context. We only show insertion at the end, but the attack can be inserted at any position between sentences. A model needs to associate the query and the evidence in the context (orange) and not distracted by the inserted attack to make the correct prediction.

Sparse structured text rationalization (SPECTRA) This model (Guerreiro and Martins, 2021) extracts a deterministic structured mask m by solving a constrained inference problem while optimizing the following objective:

$$\ell_{\text{SPECTRA}}(x, y) = -\log p(y \mid z \odot x; \phi),$$
$$z = \underset{z' \in \{0,1\}^L}{\operatorname{argmax}} (\operatorname{score}(z'; s) - \frac{1}{2} \|z'\|^2),$$

where $s \in \mathbb{R}^L$ is the logit vector of tokens or sentences, and a global $\operatorname{score}(\cdot)$ function that incorporates all constraints in the predefined factor graph. The factors can specify different logical constraints on the discrete mask z, e.g a BUDGET factor that enforces the size of the rationale as $\sum_t z_t \leq B$. The entire computation is deterministic and allows for back-propagation through the LP-SparseMAP solver (Niculae and Martins, 2020). We use the BUDGET factor in the global scoring function. To control the sparsity at π (e.g., $\pi = 0.4$ for 40% sparsity), we can choose $B = L \times \pi$.

Full-context model (FC) As a baseline, we also consider a full-context model, which is a BERT-based encoder (Devlin et al., 2019) with task specific final layers such as an MLP layer for classification task or two MLPs for span prediction. The model is trained with standard cross entropy loss using the task supervision.

5.2 Models with Rationale Supervision

VIB with human rationales (VIB-sup) When human annotated rationales z^* are available, they can be used to guide predicting the sampled masks z by adding a loss term:

$$\begin{aligned}\ell_{\text{VIB-sup}}(x, y) = &\underset{z \sim p(z|x;\theta)}{\mathbb{E}} \big[-\log p(y \mid z \odot x; \phi) \big] \\ &+ \beta \mathrm{KL}\big[p(z \mid x; \theta) \mid\mid p(z) \big] \\ &+ \gamma \sum_t -z_t^* \log p(z_t \mid x; \theta),\end{aligned}$$

where $\beta, \gamma \in \mathbb{R}$ are hyperparameters. During inference, the rationale module generates the mask z the same why as the VIB model by picking the top-k scored positions as the final hard mask. The third loss term will encourage the model to predict human annotated rationales, which is the ability we expect a robust model should exhibit.

Full-context model with human rationales (FC-sup) We also extend the FC model to leverage human annotated rationales supervision during training (FC-sup). We add a linear layer on top of the sentence/token representation and obtain the logits $s \in \mathbb{R}^L$. The logits are passed through the sigmoid function into mask probabilities. Essentially, it is multi-task learning of rationale prediction and the original task, shared with the same BERT encoder.

5.3 Tasks

We evaluate the models on several datasets that cover a diverse set of aspects including 1) sentence-level (FEVER, MultiRC, SQuAD) or token-level (Beer, Hotel) rationalization task, 2) text classification, fact verification and extractive question answering tasks (see examples in Table 1).

FEVER FEVER is a sentence-level binary classification fact verification dataset from the ERASER benchmark (DeYoung et al., 2020). The input contains a claim specifying a fact to verify and a passage of multiple sentences supporting or refuting the claim. For the AddText-Adv attacks, we add modified query text to the claims by replacing nouns and adjectives in the sentence with antonyms from WordNet (Fellbaum, 1998) and randomly swapping named entities with neighboring ones in vector space with the same part-of-speech tag, as determined by 100-dimensional GloVe vectors (Pennington et al., 2014).

MultiRC MultiRC is a sentence-level multi-choice question answering task that is reformatted as binary classification where each answer choice is concatenated with the question and the model has to predict 'yes/no'. For the AddText-Adv attacks, we transform the question and the answer separately using the same procedure we used for FEVER. We then reword the modified question and answer into a declarative sentence following constituency rules defined by (Jia and Liang, 2017) and insert it into the passage.

SQuAD SQuAD (Rajpurkar et al., 2016) is a popular extractive question answering dataset and we use the AddOneSent attacks proposed in Adversarial SQuAD (Jia and Liang, 2017). SQuAD does not contain human rationales itself and we use the sentence where the correct answer span appears in as the ground truth rationale sentence. SQuAD is the only span extraction task that we evaluate on.

Beer BeerAdvocate is a multi-aspect sentiment analysis dataset (McAuley et al., 2012), modeled as a token-level rationalization task. We use the *appearance* aspect in out experiments. We convert the scores into the binary labels following Chang et al. (2020). Note that this task does not have a query as in the previous tasks, we insert a sentence with the template "`{SUBJECT} is {ADJ}`" into the review where the adjective expresses positivity to a negative review and vice versa.

Hotel TripAdvisor Hotel Review is also a multi-aspect sentiment analysis dataset (Wang et al., 2010). We use the *cleanliness* aspect in our experiments. We generate AddText-Adv attacks in the same way as we did for the Beer dataset.

We report accuracy for all the datasets, except for SQuAD that we report the F1 score between the predicted span and the ground-truth span.

6 Results

(R1) Rationalization is a promising approach to improving robustness. Figure 2 summarizes the average scores on all the datasets for each model under the three attacks we consider. We first observe that all models (including the non-rationale FC and FC-sup) are less affected by AddText-Rand and AddText-Wiki, with score drops of around 1-2% only. However, the AddText-Adv attack leads to significant drops in performance for all models, as high as 46% for SPECTRA on Hotel review. We break out the AddText-Adv results in a more fine-grained manner in Table 2. Our main observation is that the rationale models (VIB, SPECTRA, VIB-sup) are generally more robust than their non-rationale counterparts (FC, FC-sup) on four out of the five tasks, and in some cases dramatically better – for instance, on Beer reviews, SPECTRA only suffers a 5.7% drop (95.4 $\rightarrow$ 89.7) compared to FC's huge 34.3% drop (93.8 $\rightarrow$ 59.5) under attack. The one exception seems to be on the Hotel reviews dataset, where both the VIB and SPECTRA models actually perform worse under attack compared to FC. We analyze this phenomena and provide a potential reason below.

(R2) Robustness is correlated with high GR and low AR. We report the Gold Rationale F1 (GR) and Attack Capture Rate (AR) for all models in Table 3. When attacks are added, GR consistently decreases for all tasks. However, AR ranges widely across datasets. The unsupervised rationale models, VIB and SPECTRA, have lower AR compared to FC-sup across all tasks, which at least partially explains their superior robustness to AddText-Adv attacks. VIB and SPECTRA also have lower drops in GR under attack compared to FC-sup.

Next, we investigate the poor performance of VIB and SPECTRA on Hotel reviews by analyzing the choice of words in the attack. Using the template "My car is `{ADJ}`.", we measure the percentage of times the rationalizer module selects the adjective as part of its rationale. When the adjectives are "dirty" and "clean", the VIB model selects them a massive 98.5% of the time. For "old" and "new", VIB still selects them 50% of the time. On the other hand, the VIB model trained on Beer reviews with attack template "The tea is `{ADJ}`." only selects the adjectives 20.5% of the time (when the adjectives are "horrible" and "fabulous"). This shows that the bad performance of the rationale

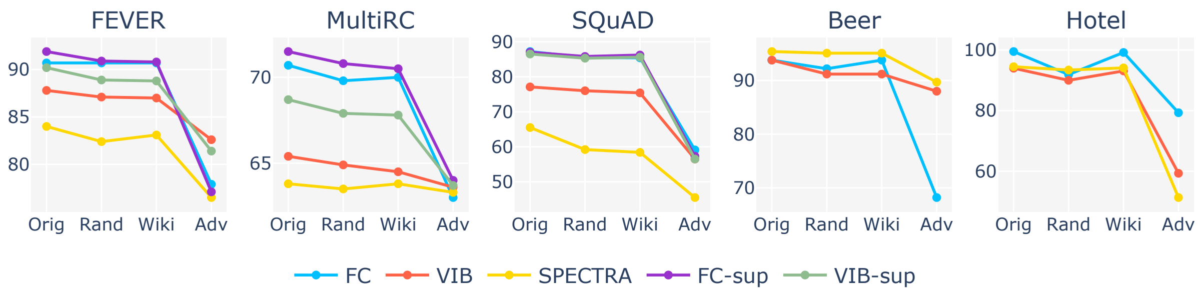

Figure 2: Original performance and the three type of attacks AddText-Rand, AddText-Wiki, and AddText-Adv evaluated on five datasets and all of the models. Left-most shows the original performance.

	FEVER			MultiRC			SQuAD			Beer			Hotel		
	Orig.	Attack	Δ ↓	Orig.	Attack	Δ ↓	Orig.	Attack	Δ ↓	Orig.	Attack	Δ ↓	Orig.	Attack	Δ ↓
FC	90.7	77.9	12.8	70.7	63.0	7.7	87.2	59.1	28.1	93.8	59.5	34.3	99.5	79.3	**20.2**
VIB	87.8	82.6	**5.2**	65.4	63.6	1.8	77.1	56.5	20.6	93.8	88.0	5.8	94.0	59.3	34.8
SPECTRA	84.0	76.5	7.6	63.8	63.3	**0.5**	65.5	45.5	**20.0**	95.4	89.7	**5.7**	94.5	51.3	43.2
FC-sup	91.9	77.1	14.8	71.5	64.0	7.5	87.0	57.3	**29.7**	-	-	-	-	-	-
VIB-sup	90.2	81.4	**8.8**	68.7	63.7	**5.0**	86.5	56.5	30.0	-	-	-	-	-	-

Table 2: Original versus attacked task performance on the five selected datasets for the AddText-Adv attack. We report accuracy for all datasets except for SQuAD, which we report F1 score. The attacked performance is the average of inserting the attack at the start and at the end of the text input.

models on Hotel reviews is down to their inability to ignore task-related adjectives in the attack text, hinting that the lexical choices made in constructing the attack can significantly impact robustness.

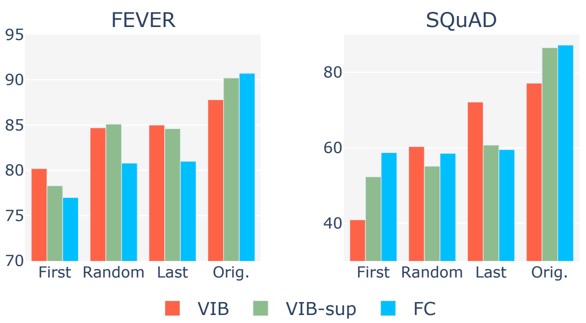

Figure 3: Accuracy when attack is inserted at different sentence positions, highlighting the positional bias picked up by the models.

(R3) Explicit rationale supervision does not help robustness. Perhaps surprisingly, adding explicit rationale supervision does not help improve robustness (Table 2). Across FEVER, MultiRC and SQuAD, VIB-sup consistently has a higher Δ between its scores on the original and perturbed instances. We observe that while models trained with human rationales generally do predict gold rationale more often (higher GR), they also capture a much higher AR across the board. On MultiRC, for instance, the VIB-sup model outperforms VIB in task performance because of its higher GR (36.1 versus 15.8). However, when under attack, VIB-sup's high 58.7 AR, hindering the performance compared to VIB, which has a smaller 35.8 AR. This highlights an overlooked aspect of prior work only considering metrics like IOU (which is similar in spirit to GR) to assess rationale models.

(R4) Rationale models are sensitive to attack positions. We further analyze the effect of attack text on rationale models by varying the attack position. Figure 3 displays the performance of VIB, VIB-sup and FC on FEVER and SQuAD when the attack sentence is inserted into the first, last or a random position of the original text input. We observe performance drops on both datasets when inserting the attack sentence at the beginning of the context text as opposed to the end. For example, when the attack sentence is inserted at the beginning, the VIB model drops from 77.1 F1 to 40.9 F1, but it only drops from 77.1 F1 to 72.1 F1 for a last position attack. This hints that rationale models may implicitly be picking up positional biases from the dataset, similar to their non-rationale counterparts (Ko et al., 2020).

(R5) Extracting good rationales and avoiding attack text is crucial to robustness. We exam-

	FEVER		MultiRC		SQuAD		Beer		Hotel	
	GR ↑	AR ↓	GR ↑	AR ↓	GR ↑	AR ↓	GR ↑	AR ↓	GR ↑	AR ↓
VIB	36.9 → 30.3	59.4	15.8 → 13.9	35.8	86.2 → 84.9	63.7	20.5 → 18.1	11.9	23.5 → 22.6	18.4
SPECTRA	26.9 → 21.5	40.6	11.9 → 11.8	22.6	67.1 → 60.8	52.6	28.6 → 27.8	15.2	19.5 → 18.3	31.6
FC-sup	51.5 → 45.5	65.9	50.0 → 42.7	55.7	99.6 → 98.8	97.8	-	-	-	-
VIB-sup	50.6 → 44.3	67.0	36.1 → 22.7	58.7	99.5 → 97.8	97.2	-	-	-	-

Table 3: Gold Rationale F1 (GR) (original → perturbed input) and Attack Capture Rate (AR) for the AddText-Adv attack on the five tasks. The reported number is the average of inserting the attack at the start and at the end of the text input.

	VIB Accuracy (%)	VIB-sup Accuracy (%)
Original	87.8 (100.0)	90.2 (100.0)
Overall Attack	83.0 (100.0)	84.9 (100.0)
Gold ✓ Attack ✓	83.3 (34.2)	85.5 (76.7)
Gold ✓ Attack ✗	91.1 (31.8)	92.4 (11.3)
Gold ✗ Attack ✓	73.6 (22.0)	74.1 (11.5)
Gold ✗ Attack ✗	77.7 (12.0)	68.0 (0.4)

Table 4: Accuracy breakdown of the VIB model on the FEVER dataset. The attack is inserted at the beginning of the passage. ✓ indicates the Gold or Attack sentence is selected as rationale and ✗ otherwise. We show the percentage of examples in parenthesis.

	FEVER			MultiRC		
	Original	Attacked	Δ ↓	Original	Attacked	Δ ↓
FC-sup	91.9	77.1	14.8	71.5	64.0	7.5
+ ART	91.8	78.7	**13.1**	69.3	64.8	**4.5**
VIB	87.8	82.6	4.2	65.4	63.6	0.7
+ ART	87.6	87.0	**0.6**	65.8	65.5	**0.3**
VIB-sup	90.2	81.4	8.8	68.7	63.7	5.0
+ ART	90.0	86.1	**3.9**	70.3	65.7	**4.6**

Table 5: Task performance of the original models versus models with Augmented Rationale Training (ART).

ine where the rationale model gains robustness by inspecting the generated rationales. Table 4 shows the accuracy breakdown under attack for VIB and VIB-sup models. Intuitively, both models perform best when the gold rationale is selected and the attack is avoided, peaking at 91.1 for VIB and 92.4 for VIB-sup. Models perform much worse when the gold rationale is omitted and the attack is included (73.6 for VIB and 74.1 for VIB-sup), highlighting the importance of choosing good and skipping the bad as rationales.

(R6) Augmented rationale training can improve robustness. Based on our findings from Table 4, we set out to improve the robustness of rationale models through *augmented rationale training* (ART). We insert two random sentences sampled from Wikipedia (the `wikitext-103` dataset) into the input passage at random positions and set their pseudo rationale labels $z^{\text{pseudo}} = 1$ and all other sentences to $z = 0$. We then add an auxiliary negative binary cross entropy loss to train the model to *not* predict the pseudo rationale. This encourages the model to ignore spurious text that is unrelated to the task. Table 5 shows that the models trained with ART improve robustness for FC-sup, VIB and VIB-sup in both FEVER and MultiRC.

7 Conclusion

In this work, we investigate whether neural rationale models are robust to adversarial attacks. We construct a variety of AddText attacks across five different tasks and evaluate state-of-the-art rationale models. We find that while these models show some promise at being more robust, they are also quite sensitive to factors like the attack position or word choices in the attack text. Surprisingly, explicit rationale supervision does not improve robustness nor prevent the model from selecting the attack text as part of the extracted rationale.

Our findings raise two key points. First, state-of-the-art rationale models, despite their promise for enabling interpretability and robustness, may not always be generating optimal rationales and may yet be prone to spurious text in the dataset. Second, metrics like IOU, frequently used in prior work (DeYoung et al., 2020; Paranjape et al., 2020), may not be ideal ways of evaluating the generated rationales since they do not test how crucial the rationale is to the model's decision making. In contrast, adversarial tests may provide a more explicit form of evaluating rationale models since they require models to ignore the spurious and irrelevant text. We hope our findings can inform the development of better models and algorithms for rationale predictions and initiate more research into the interplay between interpretability and robustness.

References

Alexander Alemi, Ian Fischer, Joshua Dillon, and Kevin Murphy. 2017. Deep variational information bottleneck. In *International Conference on Learning Representations (ICLR)*.

David Alvarez-Melis and Tommi Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 412–421.

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. In *Association for Computational Linguistics (ACL)*, pages 2963–2977.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations (ICLR)*.

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi S. Jaakkola. 2020. Invariant rationalization. In *International Conference on Machine Learning (ICML)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Association for Computational Linguistics (NAACL)*, pages 4171–4186.

Jay DeYoung, Sarthak Jain, Nazneen F. Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized nlp models. In *Association for Computational Linguistics (ACL)*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Association for Computational Linguistics (ACL)*, pages 31–36.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Nuno Miguel Guerreiro and André F. T. Martins. 2021. Spectra: Sparse structured text rationalization. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *North American Association for Computational Linguistics (NAACL)*, pages 1875–1885.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Association for Computational Linguistics (ACL)*.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *North American Association for Computational Linguistics (NAACL)*, pages 3543–3556.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C Wallace. 2020. Learning to faithfully rationalize by construction. In *Association for Computational Linguistics (ACL)*, pages 4459–4473.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. Look at the first sentence: Position bias in question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Julian McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning attitudes and attributes from multi-aspect reviews. In *IEEE International Conference on Data Mining (ICDM)*.

Vlad Niculae and F. T. André Martins. 2020. Lp-sparsemap: Differentiable relaxed optimization for sparse structured prediction. In *International Conference on Machine Learning (ICML)*.

Adam Noack, Isaac Ahern, Dejing Dou, and Boyang Li. 2021. An empirical study on the relation between network interpretability and adversarial robustness. *SN Computer Science*, 2(1):1–13.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. An information bottleneck approach for controlling conciseness in rationale extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *emnlp*, pages 1532–1543.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Association for Computational Linguistics (ACL)*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1135–1144.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? In *Association for Computational Linguistics (ACL)*, pages 2931–2951.

Kyle Swanson, Lili Yu, and Tao Lei. 2020. Rationalizing text matching: Learning sparse alignments via optimal transport. In *Association for Computational Linguistics (ACL)*, pages 5609–5626.

Keyon Vafa, Yuntian Deng, David Blei, and Alexander M Rush. 2021. Rationales for sequential predictions. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 10314–10332.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: A rating regression approach. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP Demo Track)*.

Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. 2019. Rethinking cooperative rationalization: Introspective extraction and complement control. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 4094–4103.

A Appendix

A.1 Implementation Details

We use two `BERT-base-uncased` (Wolf et al., 2020) as the rationalizer and the predictor components for all the models and one BERT-base for the Full Context (FC) baseline. The rationales for FEVER, MultiRC, SQuAD are extracted at sentence level, and Beer and Hotel are at token-level.

$$\texttt{BERT}(x) = \big(\mathbf{h}_{\texttt{[CLS]}}, \mathbf{h}_0^1, \mathbf{h}_0^2, ..., \mathbf{h}_0^{n_0}, \mathbf{h}_{\texttt{[SEP]}}, \mathbf{h}_1^1, \mathbf{h}_1^2, ..., \mathbf{h}_1^{n_1}, ..., \mathbf{h}_T^1, \mathbf{h}_T^2, ..., \mathbf{h}_T^{n_T}, \mathbf{h}_{\texttt{[SEP]}}\big),$$

where the input text is formatted as *query* with sentence index 0 and *context* with sentence index 1 to T. For sentiment tasks, the 0-th sentence and the first `[SEP]` token are omitted. For sentence-level representations, we concatenate the start and end vectors of each sentence. For instance, the t-th sentence representation is $\mathbf{h}_t = [\mathbf{h}_t^0; \mathbf{h}_t^{n(t)}]$. For token-level representations, we use the hidden vectors directly. The representations are passed to a linear layer $\{\mathbf{w}, b\}$ to obtain logit for each sentence $s = \mathbf{w}^\intercal \mathbf{h}_t + b$.

Training Both the rationalizer and the predictor in the rationale models are initialized with pretrained BERT (Devlin et al., 2019). We predetermine rationale sparsity before fine-tuning based on the average rationale length in the development set following previous work (Paranjape et al., 2020; Guerreiro and Martins, 2021). We set $\pi = 0.4$ for FEVER, $\pi = 0.25$ for MultiRC, $\pi = 0.7$ for SQuAD, $\pi = 0.1$ for Beer, and $\pi = 0.15$ for Hotel. We select the model parameters based on the highest fine-tuned task performance on the development set.

Discrete VIB The sentence or token level logits $s \in \mathbb{R}^L$ parameterize a relaxed Bernoulli distribution $p(z_t \mid x) = \text{RelaxedBernoulli}(s)$ (also known as the Gumbel distribution (Jang et al., 2017)), where $z_t \in \{0, 1\}$ is the binary mask for sentence t. The relaxed Bernoulli distribution also allows for sampling a soft mask $z_t^* = \sigma(\frac{\log s + g}{\tau}) \in (0, 1)$, where g is the sampled Gumbel noise. The soft masks $z^* = (z_1^*, z_2^*, ..., z_T^*)$ are sampled independently to mask the input sentences such that the latent $z = m^* \odot x$ for training. During inference, we take $z_t = \mathbb{1}[z_t^* \in \text{top-}k(z^*)]$ and $z \odot x$ is passed to the predictor during inference. Here we specify the hyperparameter π to control the sparsity of the rationales.