MAGNEX: A MODEL AGNOSTIC GLOBAL NEURAL EXPLAINER

Anonymous authors
Paper under double-blind review

ABSTRACT

Black-box decision models have been widely adopted both in industry and academia due to their excellent performance across many challenging tasks and domains. However, much criticism has been raised around modern AI systems, to a large extent due to their inability to produce *explainable* decisions that both their end-users and their developers can trust. The need for such decisions, i.e., decisions accompanied by a rationale for why they are made, has ignited much recent research. We propose MAGNEX, a global algorithm that leverages neural-network based explainers to produce rationales for any black-box decision model, neural or not. MAGNEX is model-agnostic, and thus easily generalizable across domains and applications. More importantly, MAGNEX is global, i.e., it learns to create rationales by optimizing for a number of instances at once, contrary to local methods that aim at explaining a single example. The global nature of MAGNEX has two advantages over local methods: i) it generalizes across instances hence producing more faithful explanations, ii) it is computationally more efficient during inference. Our experiments confirm that MAGNEX outperforms popular explainability algorithms both in explanation quality and in computational efficiency.

1 INTRODUCTION

Black-box decision models have for some time now posed a dilemma between power and interpretability. For use cases where explanations are necessary, the inability of black-box models to supply them is often a deterrent to adoption. However, even in low-risk scenarios this lack of explainability often causes distrust to both the developers and the users of these models, who are often puzzled about how decisions emerge (Ribeiro et al., 2016). Furthermore, explainability is an important mechanism to ensure black-box models act fairly and without bias (Sun et al., 2019).[1]

In this paper, we propose MAGNEX, a *model-agnostic* neural explainer that globally learns to explain an already trained model, neural or not. In this *post-hoc* interpretability setting, most methods (Ribeiro et al., 2016; Sundararajan et al., 2017; Lundberg & Lee, 2017) create feature-based explanations, i.e., explanations that assign a score to each feature in the input of the model based on how important the feature is to the model's decision. This importance score may rely on some internal mechanism of the model we wish to explain, e.g., gradients (Sundararajan et al., 2017; Shrikumar et al., 2017) or attention (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019). Such methods, however, have limitations to the types of model they can explain, e.g., gradient-based methods work only with differentiable models. Perturbation-based methods, e.g., LIME (Ribeiro et al., 2016), SHAP (Lundberg & Lee, 2017), drop combinations of features for a specific input to the model and observe its output. Omitted features that have a large impact on the output of the model across perturbations are deemed important, while other features are considered unimportant. This allows perturbation-based methods to be model-agnostic, but adds severe computational complexity, since a large number of perturbations is required per input instance to create quality explanations. The search for an optimal solution by erasure (feature drop) is combinatorial and practically infeasible for even small feature spaces; the search space is the power set of the feature set resulting in a complexity of $O(2^n)$ for n features. Therefore, pertubation-based methods find approximate (sub-optimal) solutions, but even this process is cumbersome, especially when the input is large (i.e., contains many features) and the explainability method is local (a different search must be performed for each input instance).

[1] We use *explainability* and *interpretability* interchangeably as there is no clear consensus in the literature.

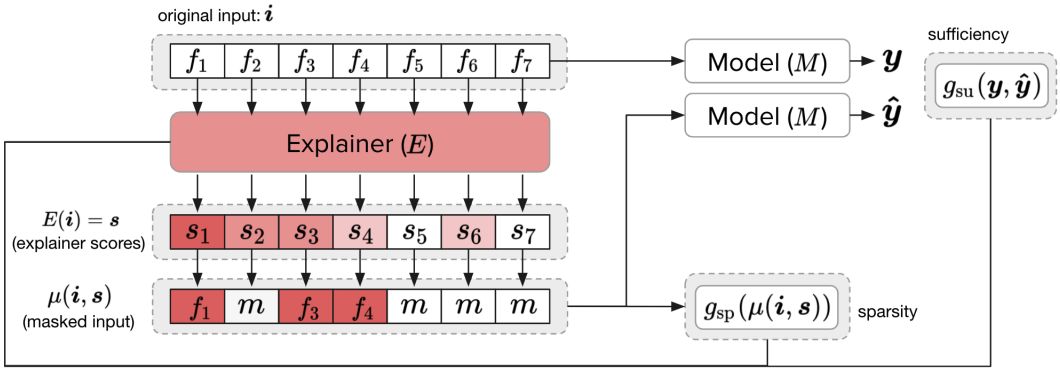

Figure 1: A high-level illustration of MAGNEX. The features of input $\boldsymbol{i}$ are scored by the explainer. The scores are used to create a binary mask (Eq. 2) representing the parts of $\boldsymbol{i}$ to be retained. The original and masked input are both passed through the pre-trained model and the outputs, $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$, are used to calculate the *sufficiency* score. The explainer aims to maximize sufficiency while also masking as many features as possible, thus also maximizing *sparsity*.

MAGNEX also creates feature-based explanations, but contrary to most methods, its explainer is a neural network that globally learns to assign feature importance scores. This is similar to the work of De Cao et al. (2020), which relies on gradients to produce feature importance scores; their explainer is a shallow network directly attached to the model being explained, requiring the gradients to flow through both the explainer and the model. This limits the approach of De Cao et al. (2020) to explaining only differentiable models, while increasing the memory and computational complexity.

Neural explanation modules have also been used in explainable *by-design* models (Lei et al., 2016; Bastings et al., 2019; Yu et al., 2019; Chang et al., 2019; Chalkidis et al., 2021). These models mainly focus on human-centric explainability and jointly train a rationale extractor and a classifier. This setting poses an extra burden on the training procedure and disincentivizes exploring parts of the space of possible solutions the model can arrive to, possibly leading to loss of model performance at the expense of interpretability. MAGNEX disentangles the explainer from the model it wishes to explain, aiming to produce explanations that are *faithful*, i.e., accurately reflect the features considered important by the model (Lipton, 2018; Jacovi & Goldberg, 2020). The generated explanations do not have to agree with human annotated rationales (e.g., gold text snippets), which are often used as an optimal solution in explainable by-design methods. MAGNEX's goal is to faithfully reveal information about a model's inner workings to its developers and if its explanations agree with human intuition, then MAGNEX can also be used as an explanation algorithm for end-users. If the explanations do not agree with human intuition, MAGNEX can reveal weaknesses of the explained model and/or of the data used to train it. For instance, training data may not be diverse enough, allowing a model to overfit to insignificant features in its input.

The explainer of MAGNEX is a shallow neural network that attributes scores to input features (Figure 1). Its training is dictated by a two-fold objective that can be classified under constrained optimization. The scoring of the explainer must be primarily *sufficient*, i.e., the explanations when fed to the underlying model must result in the same output (DeYoung et al., 2020). Also, among all possible sufficient explanations the explainer is tasked with finding one that is maximally *sparse*, i.e., utilizes the minimum number of features (Lei et al., 2016). This constrained objective, or a relaxed version of it, has been also used by perturbation-based systems (De Cao et al., 2020; Ribeiro et al., 2016).

Our contributions are as follows:

- We propose MAGNEX a method that explains the inner workings of a pre-trained model in a post-hoc manner. MAGNEX, is completely model agnostic and thus can explain any pre-trained model across modalities (e.g., vision, text) and applications.
- The proposed approach is global, allowing MAGNEX to generalize across instances; learning from many samples has a regularization effect alleviating *hindsight bias* (Fischhoff & Beyth, 1975) which is a common phenomenon in machine learning problems (Mahdavi & Rahimian, 2017) and more specificaly in perturbation-based explainability (De Cao et al.,

2020). Also, the global nature of MAGNEx transfers the computationally expensive feature search to the training stage allowing for more efficient inference than its competitors.

- The experiments showed that MAGNEx produces explanations of better quality than popular explainability methods (LIME (Ribeiro et al., 2016), IG (Sundararajan et al., 2017)), while also being more stable across instances and much more efficient during inference.

2 METHODOLOGY

2.1 FORMULATION

Let $M : I \rightarrow O$ be the model we wish to explain. Each input $\boldsymbol{i} \in I$ is a set of input features such that $\boldsymbol{i} = \{f_1, f_2, \ldots, f_n\}$, where n is the number of features. An explainer, E, operates on $\boldsymbol{i}$ and associates each feature in $\boldsymbol{i}$ with an importance score $s_j \in [0, 1]$ for $j \in \{1, 2, \ldots, n\}$:

$$E(\boldsymbol{i}) = \{s_1, s_2, \ldots, s_n\} \tag{1}$$

A masking function μ, operates on an input $\boldsymbol{i} = \{f_1, f_2, \ldots, f_n\}$ and its corresponding scoring $\boldsymbol{s} = \{s_1, s_2, \ldots, s_n\}$ to produce a masked version of $\boldsymbol{i}$:

$$\mu(\boldsymbol{i}, \boldsymbol{s}) = \{e_1 \in \{f_1, m\}, e_2 \in \{f_2, m\}, \ldots, e_n \in \{f_n, m\}\} \tag{2}$$

where m is a feature which carries no information for M and is chosen according to the task, e.g., a black pixel in computer vision or a pad token in natural language processing. In practise, μ is realized in different ways in training and inference. During training whether a feature f_j will be replaced by m is determined by performing a biased coin flip based on s_j (the score of f_j), while during inference μ is realized as:

$$\mu(\boldsymbol{i}, \boldsymbol{s})_j = \begin{cases} f_j, & \text{if } \boldsymbol{s}_j > \epsilon \\ m, & \text{otherwise} \end{cases} \tag{3}$$

Given an input $\boldsymbol{i}$ and a scoring $\boldsymbol{s} = E(\boldsymbol{i})$, we compute the quality of the explainer's scoring in terms of sufficiency (su) and sparsity (sp), defined as follows:

$$\text{su} = g_{\text{su}}(M(\boldsymbol{i}), M(\mu(\boldsymbol{i}, \boldsymbol{s}))) \tag{4}$$

$$\text{sp} = g_{\text{sp}}(\mu(\boldsymbol{i}, \boldsymbol{s})) \tag{5}$$

where, $g_{\text{su}} : O \times O \rightarrow [0, 1]$ is a function measuring to what extent the output of M when presented with $\mu(\boldsymbol{i}, \boldsymbol{s})$ resembles the output of M when presented with $\boldsymbol{i}$, and g_{sp} computes the percentage of features which have been replaced with m. Since we opt for maximally sparse inputs with high sufficiency, the total quality score for the explanation is computed as:

$$\text{q} = \begin{cases} \text{su} + \text{sp}, & \text{if su} > \upsilon \\ \text{su}, & \text{otherwise} \end{cases} \tag{6}$$

Note that we force explanations that have a sufficiency of at least υ to ensure a minimum quality.

Unless otherwise specified, we use a g_{su} tailored to classification where we assume $O = [0, 1]^c$ and c is the number of classes. Classification sufficiency is then calculated as:

$$g_{\text{su}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 1 - (\max(\boldsymbol{y}) - \hat{\boldsymbol{y}}_p) \tag{7}$$

where $p = \text{argmax}(\boldsymbol{y})$ is the predicted class in $\boldsymbol{y} \in O$ and g_{su} is bounded in $[0, 1]$ with higher scores signaling more faithful explanations. The term $\max(\boldsymbol{y}) - \hat{\boldsymbol{y}}_p$ measures the divergence between the top probability estimate across classes in $\boldsymbol{y}$ and the probability estimate for the same class in $\hat{\boldsymbol{y}}$.

2.2 LEARNING

Our explainer is realized as a neural network with parameters θ throughout. We aim to find the optimal values $\theta*$ which maximize q (Eq. 6) across a number of training examples $\{i_1, i_2, \ldots, i_m\} \subseteq I$. However, the standard backpropagation optimization approach falls short in this case since it is impossible to produce gradients to update the explainer. While the explainer itself is differentiable, in order for a sufficiency score to be computed for some input $\boldsymbol{i}$ a hard choice must be made on

which features in $\boldsymbol{i}$ to retain and which to substitute with m. The masking function μ is therefore non-differentiable. More importantly, our approach is completely model-agnostic and we make no assumptions about whether the model we wish to explain (M) is differentiable or not.

In the simplest scenario, where only the masking function is non-differentiable, a number of approaches have attempted to produce gradient estimations with methods other than the score estimator, based on REINFORCE (Williams, 1992). The most common of these approaches are the *straight-through* estimator (Chang et al., 2019; Chalkidis et al., 2021) and relaxation to binary variables (Louizos et al., 2018; Bastings et al., 2019; De Cao et al., 2020) which leverages the *reparametrization* trick (Kingma & Welling, 2013). Both of these approaches can be used only when the masking function is the only non-differentiable component, i.e., they require M to be differentiable, thus breaking the *model-agnostic* nature of the explainer, which is a requirement in MAGNEx.

To retain the model-agnostic nature of our formulation and alleviate large computational strain to our method we opt to train our model by estimating gradients for updates to our explainer network with the score estimator (REINFORCE). For training on a single input $\boldsymbol{i}$ we create a multi-variable policy using the output of our explainer.

$$\pi_\theta(\boldsymbol{i}) = \{\mathcal{B}(E_\theta(\boldsymbol{i})_j)\}_{j=1}^{|\boldsymbol{i}|} \tag{8}$$

where $\mathcal{B}$ is the Bernoulli distribution. Sampling from this policy is equivalent to sampling from each of the Bernoulli distributions independently. Therefore a sample $\boldsymbol{\tau} \sim \pi_\theta(\boldsymbol{i})$ is a sequence of binary variables indicating the presence or the absence of the feature at position j. We train our method using the score estimator which in this case can be written as:

$$\nabla_\theta J(\pi_\theta(\boldsymbol{i})) = \nabla_\theta \mathbb{E}_{\boldsymbol{\tau}\sim\pi_\theta(\boldsymbol{i})}[\mathrm{q}(\boldsymbol{\tau})] \tag{9}$$

Following Williams (1992) we can rewrite the above gradient in the form:

$$\nabla_\theta J(\pi_\theta(\boldsymbol{i})) = \mathbb{E}_{\boldsymbol{\tau}\sim\pi_\theta(\boldsymbol{i})}[\nabla_\theta \log \mathrm{P}(\boldsymbol{\tau}|\boldsymbol{i};\theta)\ \mathrm{q}(\boldsymbol{\tau})] \tag{10}$$

where $\mathrm{P}(\boldsymbol{\tau}|\mathrm{i};\theta) = \prod_{j=1}^{|\boldsymbol{i}|} P(\boldsymbol{\tau}_j|\pi_\theta(\boldsymbol{i})_j)$. We can easily approximate Eq. 10 by Monte-Carlo sampling.[2] We further add a baseline in order to reduce the variance of the gradient estimator:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\boldsymbol{\tau}\sim\pi_\theta}[\nabla_\theta \log \mathrm{P}(\boldsymbol{\tau}|\boldsymbol{i};\theta)\ (\mathrm{q}(\boldsymbol{\tau}) - b)] \tag{11}$$

In Mnih et al. (2014), which bears some similarities to our setting a learned baseline was used. Here, we use a moving average baseline which seems to be sufficient for our use cases.

While similar methods relying on differentiable relaxation to binary variables have been shown to outperform REINFORCE in some cases (Bastings et al., 2019; De Cao et al., 2020), we choose this gradient estimator for a number of reasons. Firstly, we want our method to remain purely model-agnostic, a requirement which cannot be satisfied by the estimators in Bastings et al. (2019) and De Cao et al. (2020) which support only differentiable models. Secondly, the fact that the estimator works by simple exposure to a scalar metric (Eq. 6), which does not need to have a gradient, greatly reduces the space complexity of the method, allowing a higher degree of parallelism on the same hardware, and in practice allowing very complex models to be explained in reasonable time. Lastly, this problem involves constrained optimization. Looking back at the definition of our total metric (Eq. 6) we can see that we are optimizing sparsity subject to sufficiency being higher than some threshold. This objective is therefore non-differentiable and De Cao et al. (2020) and Bastings et al. (2019) employ Lagrangian relaxation to approximate the constrained objective in a differentiable manner. This adds a new hyperparameter, the Lagrangian multiplier, which needs to be tuned, adding further overhead to the explainer's development procedure and is in all cases an approximation of the true constrained objective.

3 EXPERIMENTS

We tested MAGNEx in three challenging settings across modalities, i.e., image classification, sentiment classification, and question answering. Although, some of these tasks are simple to handle with modern models, they pose a challenging setting for perturbation-based algorithms, since feature spaces are large compared to previous work (Ribeiro et al., 2016; Lei et al., 2016; Bastings et al., 2019; De Cao et al., 2020). This makes it difficult for such methods to identify subsets of the input which are sparse and yet expressive enough to lead to the same output.

[2] See Appendix A.2 for a proof.

3.1 BASELINES AND EVALUATION

We compare MAGNEx against Lime (Ribeiro et al., 2016) and Integrated Gradients (IG) (Sundararajan et al., 2017), two popular post-hoc explainability methods. Recall that the explainer of MAGNEx (E) outputs a score $s_j \in [0, 1]$ for each feature f_j of an input $\boldsymbol{i}$. This score can be interpreted as the probability of f_j being important for a model to produce the same output. During training these probabilistic scores are converted to a binary value with a biased coin flip. During inference we cast a feature as important if its respective explainer score is above a treshold ϵ (0.5 in our experiments). On the other hand, the scores of both LIME and IG are unbounded and with no obvious meaning. This makes it non-trivial to decide which features are important and which ones are not. We therefore evaluate both of these methods by selecting the top k most highly scored features, where k is the number of features selected by E (MAGNEx) for the same input. In other words, we evaluate the sufficiency of all explanations at the sparsity level achieved by E. For each method we also report the time required to produce explanations averaged across inputs. Further, we intentionally make no attempt to compare explanations against human annotated rationales. Our main goal is to produce explanations which are *faithful* to the underlying model, i.e., accurately reflect the features in an input which are important for the model (Lipton, 2018; Jacovi & Goldberg, 2020), which is generally ensured when the sufficiency scores are high. Whether these faithful explanations align with what a human would consider a correct explanation is an open question and beyond the scope of this work.

3.2 TECHNICAL DETAILS

For IG, following Sundararajan et al. (2017), we use 50 steps in the approximation of the integral throughout.[3] Concerning LIME, shallow models allow a larger number of perturbations to be drawn. Therefore, in image classification where we use MAGNEx to explain shallow models, we allow $1{,}500$ perturbations to be drawn.[4] On the other hand, for sentiment classification the model complexity increases and we allow $1{,}000$ perturbations, to ensure that the explanations will be created in reasonable amount of time (less than 10 seconds per input). For similar reasons, in question answering where we try to explain the most complex model in this work, we allow only 700 perturbations per example which again keeps explanation generation time around the 10 second mark per input.[5]

3.3 IMAGE CLASSIFICATION

For image classification we use the popular MNIST dataset (Deng, 2012). The input features are pixels, and we choose E to be a two-layer Convolutional Neural Network (CNN) (LeCun et al., 1990) which creates a vector representation for each pixel, followed by a shared linear projection to output a single score per pixel. The feature space size is the number of pixels in an image ($28 \times 28 = 784$).

3.3.1 EXPLAINING RANDOM FOREST

We train a random decision forest (Ho, 1995) to perform digit classification from raw pixels. The model achieves a perfect accuracy on the test set. In this case, we only compare against LIME because random forests are not differentiable and IG is unable to produce explanations. Table 1 shows the results. MAGNEx is able to achieve much higher sufficiency than LIME while being faster during inference. Explanations are also more stable (smaller standard deviation across inputs).

Method	Sparsity	Sufficiency	Time (s)
MAGNEx (ours)	0.81 ± 0.05	$\mathbf{0.99} \pm 0.02$	$\mathbf{0.02} \pm 0.00$
LIME	0.81 ± 0.05	0.83 ± 0.15	0.30 ± 0.02

Table 1: Performance of MAGNEx and LIME when explaining a random forest model trained for digit recognition. MAGNEx outperforms LIME in both sufficiency and inference time.

[3] Sundararajan et al. (2017) proposed a value between 20 and 300.

[4] We found that increasing this number further does not yield more sufficient explanations.

[5] We only perform a mild manual tuning for both baselines due to their complexity.

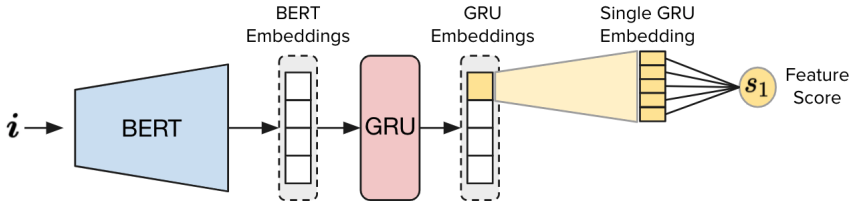

Figure 2: MAGNEX for experiments with text. The input is passed through BERT fine-tuned for the task and then a GRU further contextualizes the embeddings before each of them is passed through a shared linear layer to produce a score for each token. BERT is not updated during training.

3.3.2 EXPLAINING A CNN

We also train a two-layer CNN on the same task and compare MAGNEX against LIME and IG since the CNN is able to produce gradients.[6] The CNN reaches near-perfect accuracy on the test set (99%). Our results can be seen in Table 2. All methods have very high sufficiency scores, producing faithful explanations. This could be attributed to the fact that the pre-trained CNN is much more robust to slight changes in the input images. It appears to have generalized better than the random forests and can be more easily explained, as LIME performs well here while it was had difficulties when explaining the random forests. This is further supported by the fact that the sparsity here is higher, meaning that we can mask a larger percentage of each input while retaining very high sufficiency. When it comes to efficiency our method remains much faster during inference than its competitors.

Method	Sparsity	Sufficiency	Time (s)
MAGNEX (ours)	0.88 ± 0.04	$\mathbf{0.99} \pm 0.05$	$\mathbf{0.01} \pm 0.0$
LIME	0.88 ± 0.04	$\mathbf{0.99} \pm 0.06$	0.57 ± 0.1
IG	0.88 ± 0.04	$\mathbf{0.99} \pm 0.06$	0.60 ± 0.0

Table 2: Performance of MAGNEX, LIME, and IG when explaining a CNN trained for digit recognition. All methods generate sufficient explanations, but MAGNEX is more efficient during inference.

3.4 SENTIMENT CLASSIFICATION

We fine-tune BERT (Devlin et al., 2019) for binary sentiment classification on the IMDB review dataset (Maas et al., 2011). In this task, the input is a sequence of tokens. Again we choose a dataset with a large number of features (tokens) per instance, to ensure our method is tested for its scalability to realistic scenarios. On the validation set, the mean number of tokens per review is 303. The maximum allowed number of tokens is 512 (13% of validation instances). BERT achieves 94% accuracy on the test set. Here E is a single-layer GRU model (Cho et al., 2014) operating on the contexualized embeddings drawn from the last layer of the fine-tuned BERT model. In effect the GRU further contexualizes these embeddings before passing them to a shared linear projection to create the score for each token. Updates are only performed on the GRU and the linear projection, i.e., the fine-tuned BERT remains frozen. The whole architecture of MAGNEX for this setting can be seen in Figure 2. As mentioned in Section 2.2, a reason for choosing REINFORCE is the alleviation of a lot of computational stress from the training procedure. If we were to use differentiable masking (relaxation to binary variables) to create semi-hard scores for each token, we would have to produce gradients for the feedback given back from the pre-trained model. The computation graph in this case would be further tasked with tracking gradients for the BERT model and all its intermediate computation throughout the explainer's training procedure.

Our results can be seen in Table 3. MAGNEX produces comparable or better explanations than its competitors, while being faster during inference. One additional advantage of MAGNEX is that it alleviates *hindsight bias*, which can often be observed in perturbation-based explainability. When performing erasure (i.e., dropping features from the input to a model) there is no guarantee that low scored features are indeed not useful for the underlying model. The token *great* in Figure 3b

[6]We also attempted to use a simple gradient approach (Jacobian) but the results were significantly inferior.

Method	Sparsity	Sufficiency	Time (s)
MAGNEx	0.94 ± 0.06	$\mathbf{0.95} \pm 0.10$	$\mathbf{0.05} \pm 0.03$
LIME	0.94 ± 0.06	$\mathbf{0.95} \pm 0.09$	9.33 ± 5.23
IG	0.94 ± 0.06	0.90 ± 0.16	1.32 ± 0.70

Table 3: Performance of MAGNEx, LIME, and IG, explaining BERT fine-tuned for sentiment classification. MAGNEx outperforms the baselines in both suffuciency and inference time.

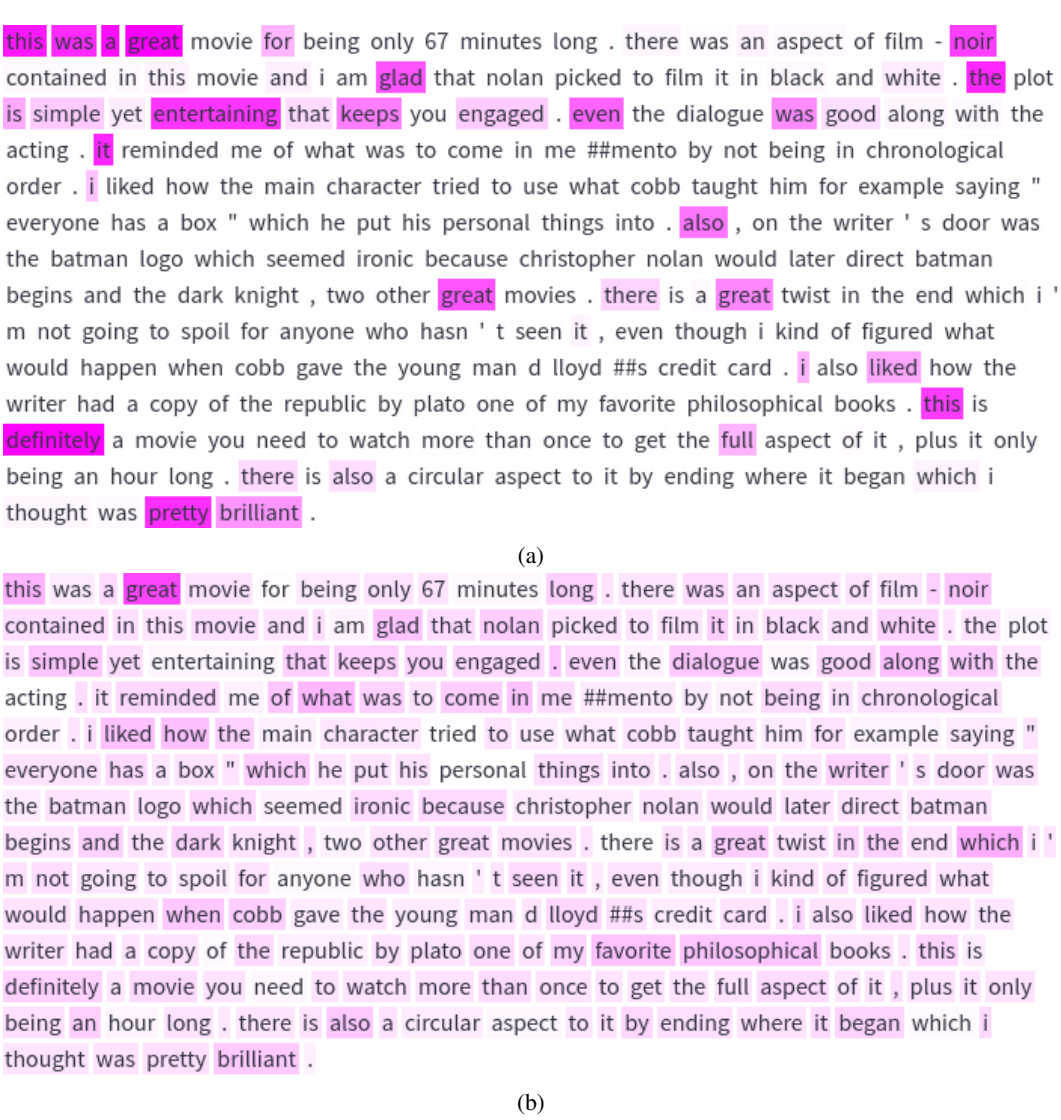

Figure 3: Explanations produced by MAGNEx (a) and LIME (b) for a review in the IMDB test set. Both explanations have high sufficiency, but we believe that (a) is a more faithful interpretation of the model's computation. LIME appears to have fallen victim of hindsight bias identifying a single feature (*great*) as important. This does not mean that the underlying model ignores everything else. MAGNEx considers more features to be important as its global training acts against hindsight bias.

is scored very highly in comparison to all other tokens. Retaining this token alone would lead to the same output. This does not necessarily mean that the pre-trained model does not consider any other tokens when casting a decision. Although we employ no explicit mechanism to tackle this issue, we believe that we make steps towards the right direction due to two characteristics of our

method. Firstly, MAGNEx is global, i.e., it learns a model across a large number of instances, which we believe to act as a regularizer against hindsight bias. The exact optimum of erasure for a specific instance can easily fall victim to *hindsight bias* and therefore approximations of this exact optimum (e.g., LIME) can often exhibit this same behaviour. On the other hand, global context allows MAGNEx to reveal patterns that generalize across instances instead of local ones. For instance, *great* being associated with a positive prediction is a pattern local to an instance that cannot be established globally. It is therefore, disincentivised as a solution in MAGNEx's global training regime. We do not argue that such patterns are non-existent, rather that they are important only when they can be established across a variety of instances and can otherwise create extremely misleading explanations (which score very well according to our metrics, i.e., they are sufficient while also being very sparse) but are at the same time not faithful explanations. Secondly, E operates in the space of hidden representations, i.e., BERT token embeddings, which arguably capture contextualized morpho-syntactic and semantic information, allowing MAGNEx to learn more general explanation patterns.

3.5 Question Answering

For this task we use a large variant of BERT (24 layers, 16 attention heads, 1024 hidden size), pre-trained and fine-tuned on SQUAD (Rajpurkar et al. (2016)). The model attempts to predict answer spans within some context given a question. It achieves an F1 score of 93.2% and an exact match score of 86.9%. We show the results of various explainers on the task in Table 4. Our explainer continues to produce explanations that have higher sufficiency scores than the other methods while also being considerably faster. A side note here is that for this task sufficiency is realized as the Jaccard index. So between two predicted spans $y = \{f_1, f_2, \ldots, f_m\}$ and $\hat{y} = \{\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_n\}$, sufficiency is measured as $\mathrm{su}(y, \hat{y}) = (y \bigcup \hat{y})/(y \bigcap \hat{y})$. The Jaccard index severely punishes small mistakes. For example, the Jaccard index for the spans 'Stanford University' and 'the Standford University' is $2/3$ despite the answers being almost identical. To some extent this explains the large standard deviations (Table 4) reported for all methods. Finally, similarly to the sentiment analysis task, LIME suffers from the hindsight bias problem. Figure 4 shows an indicative example where the explanations produced by MAGNEx are more well formed than those of LIME.

Method	Sparsity	Sufficiency	Time (s)
MAGNEx	0.76 ± 0.12	$\mathbf{0.88} \pm 0.30$	$\mathbf{0.09} \pm 0.03$
LIME	0.76 ± 0.12	0.82 ± 0.32	12.91 ± 5.22
IG	0.76 ± 0.12	0.34 ± 0.40	5.76 ± 2.24

Table 4: Performance of MAGNEx, LIME, and IG, explaining a large variant of BERT in question answering. MAGNEx outperforms the baselines in both suffuciency and inference time.

4 Related Work

The outburst in the field of explainability in machine learning started with Ribeiro et al. (2016) who proposed LIME, a *model-agnostic* perturbation-based algorithm that generates explanations for machine learning models. Since then, apart from perturbation-based methods, gradient based methods were explored (Shrikumar et al., 2017; Sundararajan et al., 2017)). More recently, Jain & Wallace (2019) and Wiegreffe & Pinter (2019), driven by the increasing popularity of Transformer-based models (Vaswani et al., 2017) in a number of tasks across modalities (Devlin et al., 2019; Schneider et al., 2019; Dosovitskiy et al., 2021), have initiated the discussion on whether the attention mechanism in these models can provide quality explanations.

In parallel, a lot of work focused on creating explainable neural networks *by-design* (Lei et al., 2016; Bastings et al., 2019; Yu et al., 2019; Chang et al., 2019; Chalkidis et al., 2021). These systems typically contain two components. The first component selects rationales (subsets of the input) which are then fed to the second component for classification. They are typically trained with similar objectives to perturbation-based methods, i.e., sufficiency and sparsity, other objectives such as *continuity* Lei et al. (2016) and *comprehensiveness* Yu et al. (2019); Chalkidis et al. (2021) have also been tested. However, the initial system (Lei et al., 2016) performed gradient estimation for

who wrote about the great pest ##ile ##nce in 1893 ? the historian francis aidan gas ##quet wrote about the ' great pest ##ile ##nce ' in 1893 and suggested that " it would appear to be some form of the ordinary eastern or bu ##bon ##ic plague " . he was able to adopt the ep ##ide ##mi ##ology of the bu ##bon ##ic plague for the black death for the second edition in 1908 , imp ##lica ##ting rats and flea ##s in the process , and his interpretation was widely accepted for other ancient and medieval epidemic ##s , such as the justin ##ian plague that was prevalent in the eastern roman empire from 54 ##1 to 700 ce .

(a)

who wrote about the great pest ##ile ##nce in 1893 ? the historian francis aidan gas ##quet wrote about the ' great pest ##ile ##nce ' in 1893 and suggested that " it would appear to be some form of the ordinary eastern or bu ##bon ##ic plague " . he was able to adopt the ep ##ide ##mi ##ology of the bu ##bon ##ic plague for the black death for the second edition in 1908 , imp ##lica ##ting rats and flea ##s in the process , and his interpretation was widely accepted for other ancient and medieval epidemic ##s , such as the justin ##ian plague that was prevalent in the eastern roman empire from 54 ##1 to 700 ce .

(b)

Figure 4: Explanations produced by MAGNEx (a) and LIME (b) for a question of SQUAD validation set. Filtering these to produce binary scores as we do during evaluation creates the reduced inputs i) *who wrote about the great pest ##ile ##nce in 1893 ? the historian francis aidan gas ##quet wrote about the ‘ great pest ##ile* and ii) *who wrote about ##ile ##nce 1893 ? the francis gas ##quet wrote great 1893 suggested “ some adopt black interpretation other roman ##1*. Global context allows MAGNEx to produce comparatively well formed inputs to present to the pre-trained model.

the rationale extractor with REINFORCE without baseline reduction making the system unstable due to the nature of the learning algorithm. Bastings et al. (2019) relaxed this binary rationale extraction process by following Louizos et al. (2018) and therefore making the objective differentiable. Nonetheless, all further work done in this area (Yu et al., 2019; Chang et al., 2019) optimizes at least two neural networks concurrently and uses some sort of gradient estimation either through REINFORCE, the reparametrization trick, or straight-trough estimators. Naturally, these systems are often very hard to train due to their complexity and their innate inability to produce exact gradients due to their stochastic nodes. Finally, recent work (Jain et al., 2020; Situ et al., 2021) has attempted to use auxiliary explanations generated by some attribution method (such as LIME or integrated gradients) as supervision. For instance, one of the methods explored by Situ et al. (2021) was to train a rationale extractor on explanations produced by LIME. They also experimented with integrated gradients, sumilarly to Jain et al. (2020) to train a neural network to create explanations in the *post-hoc* setting.

5 CONCLUSION & FUTURE WORK

We introduced MAGNEx, a post-hoc explainability algorithm, that is global and completely model-agnostic. We experimentally showed that our approach outperforms popular post-hoc algorithms in terms of the faithfulness of its explanations and in computational complexity across tasks and modalities. In addition, the global nature of MAGNEx seems to alleviate the hindsight bias problem that seemed to trouble local perturbation-based explainability methods.

In the future we aim to pursue a number of interesting directions. We would like to combine our method with online learning to produce higher quality explanations. The explanations learned offline can be used as a starting point and further search can be performed on a per instance basis. Also, a number of improvements have been proposed in recent years for policy gradient methods in reinforcement learning with one prime example being that of Schulman et al. (2017). Some of these ideas could be incorporated in an improved version of the score estimator allowing for better optimization. Lastly, this approach can be evaluated in a number of new settings (e.g., visual question answering, summarization, text generation) possibly revealing interesting results.

REFERENCES

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2963–2977, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1284. URL https://aclanthology.org/P19-1284.

Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. Paragraph-level rationale extraction through regularization: A case study on European court of human rights cases. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 226–241, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.22. URL https://aclanthology.org/2021.naacl-main.22.

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. A game theoretic approach to class-wise selective rationalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/5ad742cd15633b26fdce1b80f7b39f7c-Paper.pdf.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179.

Nicola De Cao, Michael Sejr Schlichtkrull, Wilker Aziz, and Ivan Titov. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3243–3255, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.262. URL https://aclanthology.org/2020.emnlp-main.262.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4443–4458, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.408. URL https://www.aclweb.org/anthology/2020.acl-main.408.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Baruch Fischhoff and Ruth Beyth. I knew it would happen: Remembered probabilities of once—future things. *Organizational Behavior and Human Performance*, 13(1):1–16, 1975. ISSN 0030-5073. doi: https://doi.org/10.1016/0030-5073(75)90002-1. URL https://www.sciencedirect.com/science/article/pii/0030507375900021.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pp. 1–6, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2501. URL https://aclanthology.org/W18-2501.

Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pp. 278–282. IEEE, 1995.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4198–4205, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL https://www.aclweb.org/anthology/2020.acl-main.386.

Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL https://aclanthology.org/N19-1357.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. Learning to faithfully rationalize by construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4459–4473, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.409. URL https://aclanthology.org/2020.acl-main.409.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. URL http://arxiv.org/abs/1412.6980.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch, 2020.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In David Touretzky (ed.), *Advances in Neural Information Processing Systems (NIPS 1989)*, volume 2, Denver, CO, 1990. Morgan Kaufman.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 107–117, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1011. URL https://aclanthology.org/D16-1011.

Zachary C. Lipton. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, September 2018. ISSN 0001-0782. doi: 10.1145/3233231. URL https://doi.org/10.1145/3233231.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1Y8hhg0b.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pp. 4768–4777, 2017.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Shaudi Mahdavi and M. Amin Rahimian. Hindsight bias impedes learning. In Tatiana V. Guy, Miroslav Kárný, David Rios-Insua, and David H. Wolpert (eds.), *Proceedings of the NIPS 2016 Workshop on Imperfect Decision Makers*, volume 58 of *Proceedings of Machine Learning Research*, pp. 111–127. PMLR, 09 Dec 2017. URL https://proceedings.mlr.press/v58/mahdavi17a.html.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pp. 2204–2212, 2014.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. In Gernot Kubin and Zdravko Kacic (eds.), *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pp. 3465–3469. ISCA, 2019. doi: 10.21437/Interspeech.2019-1873. URL https://doi.org/10.21437/Interspeech.2019-1873.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pp. 3145–3153. PMLR, 2017.

Xuelin Situ, Ingrid Zukerman, Cecile Paris, Sameen Maruf, and Gholamreza Haffari. Learning to explain: Generating stable explanations fast. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5340–5355, 2021.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1630–1640, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1159. URL https://aclanthology.org/P19-1159.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 11–20, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1002. URL https://aclanthology.org/D19-1002.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. Rethinking cooperative rationalization: Introspective extraction and complement control. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4094–4103, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1420. URL https://aclanthology.org/D19-1420.

A Appendix

A.1 Implementation Details

We implemented all models in AllenNLP (Gardner et al., 2018) and PyTorch (Paszke et al., 2019). For experiments with text (Sections 3.4 and 3.5) we used the Transformers library of Huggingface (Wolf et al., 2020). For MAGNEx we found automatic tuning to be slow in developement since REINFORCE tends to diverge very quickly when the learning rate is non-optimal. We therefore manually tuned the hyper-parameters in each experiment by allowing a small number of batches to flow through the model and observing its metrics. Non-promising runs were terminated very quickly. We performed this procedure for sentiment classification and question answering for learning rates $\in \{10^{-4}, 10^{-5}\}$. The optimal learning rates were 10^{-4} for sentiment classification and 10^{-5} for question answering. For image classification (Section 3.3) we used a learning rate of 10^{-3}. Optimization was carried out with Adam (Kingma & Ba, 2015) throughout the experiments except when fine-tuning BERT-base in sentiment classification where AdamW (Loshchilov & Hutter, 2019) was used. Finally, for our baselines (LIME, IG) we use Captum (Kokhlikyan et al., 2020).

A.2 Score estimator Proof

We begin by creating a policy parametrized by θ. For some input i the policy can be written as:

$$\pi_\theta(\boldsymbol{i}) = \{\mathcal{B}(E_\theta(\boldsymbol{i})_j)\}_{j=1}^{|\boldsymbol{i}|} \tag{12}$$

Our loss is the expected return across samples $\boldsymbol{\tau}$ drawn from π_θ ($\boldsymbol{\tau} \sim \pi_\theta(\boldsymbol{i})$) with gradient:

$$\nabla_\theta J(\pi_\theta(\boldsymbol{i})) = \nabla_\theta \mathbb{E}_{\boldsymbol{\tau}\sim\pi_\theta(\boldsymbol{i})}[\mathrm{q}(\boldsymbol{\tau})] \tag{13}$$

The probability of trajectory $\boldsymbol{\tau}$ which is a sequence of binary variables being sampled from $\pi_\theta(\boldsymbol{i})$ is:

$$\mathrm{P}(\boldsymbol{\tau}|\boldsymbol{i};\theta) = \prod_{j=1}^{|\boldsymbol{i}|} P(\boldsymbol{\tau}_j|\pi_\theta(\boldsymbol{i})_j) \tag{14}$$

We can rewrite Eq. 13 by taking the definition of expectation as:

$$\nabla_\theta J(\pi_\theta(\boldsymbol{i})) = \nabla_\theta \int_\tau \mathrm{P}(\boldsymbol{\tau}|\boldsymbol{i};\theta)\mathrm{q}(\tau) \tag{15}$$

We can add the gradient in Eq. 15 under the integral:

$$\nabla_\theta J(\pi_\theta(\boldsymbol{i})) = \int_{\boldsymbol{\tau}} \nabla_\theta \mathrm{P}(\boldsymbol{\tau}|\boldsymbol{i};\theta)\mathrm{q}(\boldsymbol{\tau}) \tag{16}$$

and rewrite Eq. 16 by utilizing $\log(f(x))' = f'(x)/f(x)$ as:

$$\nabla_\theta J(\pi_\theta(\boldsymbol{i})) = \int_{\boldsymbol{\tau}} \mathrm{P}(\boldsymbol{\tau}|\boldsymbol{i};\theta)\nabla_\theta \log \mathrm{P}(\boldsymbol{\tau}|\boldsymbol{i};\theta)\mathrm{q}(\boldsymbol{\tau}) \tag{17}$$

By reverting back to expectation form we arrive at:

$$\nabla_\theta J(\pi_\theta(\boldsymbol{i})) = \mathbb{E}_{\boldsymbol{\tau}\sim\pi_\theta(\boldsymbol{i})}[\nabla_\theta \log \mathrm{P}(\boldsymbol{\tau}|\boldsymbol{i};\theta)\ \mathrm{q}(\boldsymbol{\tau})] \tag{18}$$

This allows us to sample $\boldsymbol{\tau} \sim \pi_\theta(\boldsymbol{i})$ and compute the gradient with Monte Carlo Sampling.