ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230266

From Intermediate Representations to Explanations: Exploring Hierarchical Structures in NLP

Housam K. B. Babiker[a;*], **Mi-Young Kim**[b] **and Randy Goebel**[a,c]

[a]Department of Computing Science, University of Alberta
[b]Department of Science, Augustana Faculty, University of Alberta
[c]Alberta Machine Intelligence Institute, University of Alberta

Abstract. Interpretation methods for learned models used in natural language processing (NLP) applications usually provide support for local (specific) explanations, such as quantifying the contribution of each word to the predicted class. But they typically ignore the potential interaction amongst those word tokens. Unlike currently popular methods, we propose a deep model which uses feature attribution and identification of dependencies to support the learning of interpretable representations that will support creation of hierarchical explanations. In addition, hierarchical explanations provide a basis for visualizing how words and phrases are combined at different levels of abstraction, which enables end-users to better understand the prediction process of a deep network. Our study uses multiple well-known datasets to demonstrate the effectiveness of our approach, and provides both automatic and human evaluation.

1 Introduction

Deep networks have produced impressive results in their application to natural language processing (NLP) problems. Much of this success can be attributed to their ability to capture complex higher-order interactions amongst raw features [13]. However, a deep network is usually considered a black-box model, which is simply insufficient for complex sensitive applications (e.g., in law [31]) and health [19]). The construction of complex learned models that support interpretable justifications has many benefits, including support for debugging, development of user trust and production of explanations to help understand the underlying mechanisms of prediction; all of these contribute to new domain insights.

A recently emerging area of work is black-box model explanation. In the context of NLP, the majority of studies have focused on "explaining" the output of a deep network for a given text input using feature attribution scores or saliency maps [33, 10]. Various recent techniques have focused on answering the question of "which tokens/features in the input were most discriminative for a specific prediction?" For text classification, each input is a segment of text. At an appropriate level of abstraction, a variety of feature attribution methods are designed to identify the contribution of each input feature $w.r.t.$ a predicted output classification. The outputs of these attributions, in general, are in the form of importance scores. In general, a high score indicates a highly informative token.

Most emerging work in the current literature relies on building methods that mimic the computation of the network *after* its construction, to explain its output (aka "post-hoc" explanation). That mimicking is alleged to provide some interpretation of informative features, which stand as a proxy for interpreting the output.

Why hierarchical explanations? In the past two years, several studies have focused on the organizational relationship amongst such discriminative features, thus tackling the challenge of hierarchical explanation, rather than only individual independent feature attribution [8, 17, 7]. Consider the following negative review for sentiment classification "a waste of an excellent concert." One question we ask is how the word `excellent` or a phrase related to `excellent` contributes to the model prediction.

Conventional methods for attributing features assign scores to indicate the degree to which a word or phrase contributes to the final model prediction. For example, a traditional approach such as `LIME` might identify the keyword `waste` as the explanation for the `NEGATIVE` class. Nevertheless, such methods do not reveal how tokens are combined and interact with one another to produce the predicted label. In the given example where the model's prediction is `NEGATIVE`, one might wonder how the token `excellent` or a phrase related to `excellent` influenced the label `NEGATIVE`. An explanation that can address this inquiry would provide end-users with a better understanding of the model's decision-making process. By utilizing hierarchical explanations, we can determine that the phrase `waste of an excellent` is the most critical element with the highest level of interaction. Through the hierarchical explanations, users will understand that `"waste"(NEGATIVE)` when combined with `"excellent"(POSITIVE)`, the overall prediction becomes `NEGATIVE`. Related ideas on hierarchical explanations rely on post-hoc techniques such as extending Shapley values or back-propagation algorithms to identify feature interactions. However, in our approach, we propose learning explanation-specific representations concurrently during classification, which can be used to generate hierarchical explanations. The benefits of learning representations concurrently can be summarized as follows: 1) the ability to generate faithful explanations, and 2) less computation time to generate explanations.

The main contributions of this paper are three-fold: 1) we introduce a deep model capable of learning explanation-specific representations concurrently with the classification task; 2) the learned representations are used as a proxy to generate hierarchical explanations, and 3) through hierarchical explanations, our approach provides a comprehensive picture of how different granularity of tokens interact with each other within the model.

* Corresponding Author. Email: khalifab@ualberta.ca.

Please contact the corresponding author for any appendices or supplementary material mentioned in the paper.

2 Related work

Feature attribution methods Back-propagation techniques are one of the popular approaches to identify feature attribution. For instance, Layer-wise Relevance Propagation (LRP) [3] computes what is called a relevance score by redistributing the prediction score. Another option is to average the gradient along a linear path from a baseline [36]. The explainability of a deep network can also be approached by using perturbation techniques. For example, this can be done by altering and modifying the input features, followed by passing the altered features to a function to measure the difference with the original methods [40, 44]. Other model-agnostic methods have been proposed by [32] and [10]. There are also some methods designed around the architectures of recurrent networks such as [27] and [26].

Rationale-based methods [22] use an R-CNN to generate short text segments, also called "rationales," from the input, then feed that input to a recurrent classifier; an adjusted alternative [4] proposed a similar approach to rationale-extraction problem. There are many other rationale-based methods for token extraction, such as [29, 6, 39, 2]. However, here we focus on hierarchical explanations and the determination of feature interaction, which is a different objective. Generally, rationale-based methods have not been evaluated against hierarchical explanation/feature attribution. In general, rationale-extraction methods attempt to select some salient tokens from the input and they might not be consecutive. However, in hierarchical explanation, we focus on understanding how the tokens are interacting with each other. While both rationale-based and hierarchical methods provide explanations to end-users, their objectives are different. Rationale-based methods simply extract tokens from the input by hoping that the extracted tokens give users the same prediction as the full-text regardless how words interact. In hierarchical explanation, we extend feature attribution to provide feature interaction and hierarchical explanations.

Hierarchical explanations There has been much recent work on the development of hierarchical explanations in the context of NLP. For instance, [7] proposed HEDGE to generate hierarchical explanations for text classification. The proposed method employs Shapley values for feature attribution and an interaction dialogue using a top-down or a bottom-up approach. Similarly, [35] uses contextual decomposition scores to aggregate features based on their identified interactions. Hierarchical structure is intended to provide representation support for explanation at multiple levels of detail (cf. [18]). While [35] and [7]'s methods are post-hoc, our approach focuses on learning interpretation-specific representations *concurrently during classification* and then uses them as a proxy to generate hierarchical explanations. Other techniques for hierarchical explanation employ the back-propagation technique; for example, [34] extends the integrated gradient method to support feature interaction. However, back-propagation methods suffer from noisy gradients [16]. Other ideas for hierarchical explanation employ a decision tree [41] to generate hierarchical explanation in a post-hoc approach. Some other methods focus on applying a post-hoc approach to learn about interactions inside Transformers [14], which has a different objective from ours. Our objective is to learn interpretation-specific representations concurrently, rather than to use a post-hoc approach.

3 SFFA: Soft Faithful Feature Attribution

Here we present `SFFA`, `Soft Faithful Feature Attribution`, which we propose is an effective approach to learn interpretable representations from deep nets, and can be used as a proxy to generate hierarchical explanations. To illustrate, we consider the example problem of predicting the sentiment of a textual movie review. Note that our approach does not require model re-training. In the following, we discuss how `SFFA` is integrated within the process of constructing deep networks to provide hierarchical explanations.

3.1 Background

Let $\boldsymbol{x} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_m)$ be a sequence of tokens of length m, for an input text. We use $\boldsymbol{x}_i \in \mathbb{R}^d$ to denote the feature embedding that represents the i^{th} token of the sentence, where d is the feature vector dimension. We consider probabilistic predictions and consider the prediction as a probability vector. The final layer $\boldsymbol{W}^{k\times d}$ of a typical deep model takes the context vector (obtained from Step 4 in Figure 1) to yield a vector of probabilities, where k is the total number of labels. This probability interpretation is implicitly informed by the statistical distribution that is approximated by the deep learning method. The output $\boldsymbol{y}$ from the model is a vector of class probabilities, and the predicted class $\hat{y}$ is a categorical outcome. The output of the network is defined as the inner product between the latent representation (e.g., the context vector) and $\boldsymbol{W}$. Our task is to learn an accurate predictor *and* to generate hierarchical explanations.

We use the `SFFA` model to predict the label $f(\boldsymbol{x})$ for a new instance $\boldsymbol{x}$ and to construct the hierarchical explanation from the learned representations.

3.2 Softmax Layer

The features extracted by a deep network are provided to a Softmax layer, and the output of the fully connected layer is the multiplicative product of the weights (inner product) and the previous layer's output, plus bias. This is followed by a Softmax activation which produces the probability for each class (these probabilities will sum to 1.) Cross-entropy (aka Softmax loss) is used to penalize mis-classification based on available ground-truth data. The cross-entropy is the sum of the negative logarithm of the probabilities. Consider the binary classification, and a sample $\boldsymbol{x}$ from class 0. The Softmax loss goal is to force $\boldsymbol{W}_0^T\boldsymbol{x} > \boldsymbol{W}_1^T\boldsymbol{x}$ (i.e., $\|\boldsymbol{W}_0\| \| \boldsymbol{x}\| \cos(\theta_0) > \|\boldsymbol{W}_1\| \| \boldsymbol{x}\| \cos(\theta_1)$) in order to correctly classify $\boldsymbol{x}$ [23]. In other words, the objective of determining the Softmax loss is to push features in the same class to be closer, and to further separate samples from different classes. This makes the inter-class variances larger and intra-class variances smaller. We aim to use $\boldsymbol{W}_0$ and $\boldsymbol{W}_1$ to learn discriminative token embedding features. We also use $\boldsymbol{W}_0$ and $\boldsymbol{W}_1$ as a proxy for feature attribution and hierarchical explanation. One can think of $\boldsymbol{W}_i$ as a centroid vector of class i. Intuitively, during training, we use $\boldsymbol{W}_0$ and $\boldsymbol{W}_1$ to minimize the intra-class variations.

3.3 ***SFFA***: Soft Faithful Feature Attribution

Here we discuss the proposed augmented deep model. We aim to learn effective embedding representations, i.e., identifying methods to optimize the inter-class difference (separating features of different classes) and to reduce the intra-class variation (making features of the same class compact). The network's structure is summarized in Figure 1. `SFFA` makes two important changes to the network's architecture: 1) adding order-aware sentence representation, and 2) adding a new term to the loss function. In addition, `SFFA` generates

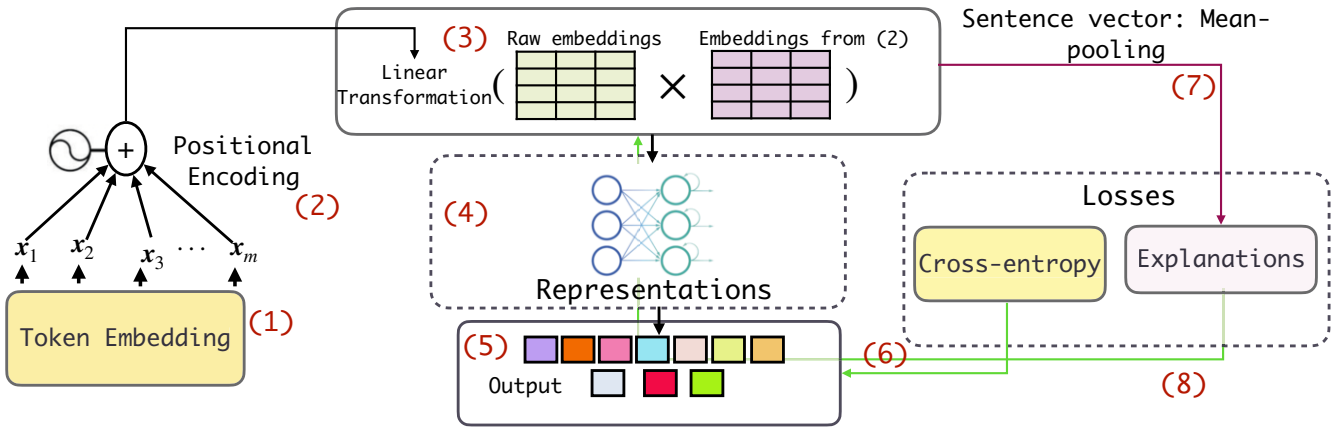

Figure 1. An example of the proposed intrinsic deep network model. The steps are summarized as follows: (1) obtain the token embedding of each token without positional encoding, (2) add the positional encoding information to each token, (3) further modify the embedding using a linear transformation, (4) map the representation layer over the new embedding features (e.g., CNNs, Multi-head attentions, LSTMs), (5) produce the output layer over the context vector to predict the class label, (6) minimize cross-entropy to penalize incorrect predictions and update the network's weights, (7) use mean-pooling to obtain the sentence vector and, finally, (8) apply the additional loss function to learn discriminative embedding features and use *only* the gradient to update the network. Please note that representations are learned concurrently with the model, i.e., there is no re-training. Finally, the main motivation is to utilize the existing property of the output layer as a proxy to generate hierarchical explanations. The model is still a black-box, but it allows generating explanations which can be used later to provide explanations.

hierarchical explanations based on the embedding features and thus our goal is to learn interpretation-specific representations at the embedding layer.

3.3.1 *Order-aware Attribution*

One approach to learning interpretation-specific representations is to minimize the distance between the samples belonging to the same distribution in the latent space. For example, one could compute each sentence's mean-pooling and then minimize the distance between that sentence vector and the corresponding centroid vector of the positive class. However, the problem using mean-pooling is that it does not consider the order of the tokens. As an alternative, we propose injecting information about each token's absolute position. We add that positional encoding signal and then use element-wise multiplication as follows:

$$\tilde{\boldsymbol{x}}_i = (PE(i) + \boldsymbol{x}_i) \odot \boldsymbol{x}_i \tag{1}$$

where $PE(i)$ is the positional encoding vector [38] of token $\boldsymbol{x}_i$ at index i. Let, $\tilde{\boldsymbol{x}} = (\tilde{\boldsymbol{x}}_1, ..., \tilde{\boldsymbol{x}}_m)$ be the new sequence embedding features. The sentence vector $\bar{\boldsymbol{x}} \in \mathbb{R}^d$ of $\tilde{\boldsymbol{x}}$ is defined as the mean-pooling of the embedding features. The sentence vector is essential in learning the discriminative embedding features. As we can see, the mean-pooling of the embedding features will change, based on the position of the tokens. Consider the following examples for sentiment classification: *'I don't like the actor, but I really like the movie'* and *'I like the actor, but I really don't like the movie'*. The proposed approach will reflect the different meaning of these two sentences according to the position of their tokens.

3.3.2 *Proposed Losses*

Recall that the Softmax loss will push training data in the same class to be closer, and increase the separation of samples from different classes. We exploit this property and apply the cosine distance over the sentence vector $\bar{\boldsymbol{x}}$. Let $\hat{y}$ be the model's prediction. The loss is defined as follows:

$$L^{embed} = 1 - \frac{\bar{\boldsymbol{x}} \cdot \boldsymbol{W}_{\hat{y}}}{\|\bar{\boldsymbol{x}}\| \, \|\boldsymbol{W}_{\hat{y}}\|} \tag{2}$$

where $\boldsymbol{W}_{\hat{y}}$ is the $\hat{y}$-th column of $\boldsymbol{W}$. During training, we use the stop-gradient operation which prevents the accumulated gradient from flowing, so that we do not compute the derivative of the above loss for the given weights $\boldsymbol{W}$. The overall objective function of the network is based on combination of two losses:

$$\mathcal{L} = \frac{\mathcal{L}^{embed}}{\lambda} - \sum_{j=1}^{k} \boldsymbol{r}_j \log(\boldsymbol{y}_j) \tag{3}$$

where $\boldsymbol{r} \in \mathbb{R}^k$ is the one-hot represented ground truth, $\boldsymbol{r}_j$ is the target probability (0 or 1) for class j, and λ is a scaling factor. λ is between $\{0.000001, 0.8\}$ using the validation set. We have experimented with different values for λ and we found the best empirically determined value is 0.0006, which is used in our experiments.

3.4 *Feature Attribution and Interaction*

In general, feature attribution focuses on estimating a contribution score for each token, to inform the model prediction. Here we describe the steps to find feature attribution and phrase attribution scores. Recall that the sentence vector provides a contextualized mean-pooling measure. It heuristically captures the intended semantics of the sentence. Equation 2 pushes the sentence towards the class vector $\boldsymbol{W}_{\hat{y}}$, which indirectly forces the salient tokens to be close to $\boldsymbol{W}_{\hat{y}}$. For a given token $\tilde{\boldsymbol{x}}_i \in \mathbb{R}^d$, the attribution score is:

$$\Phi(\tilde{\boldsymbol{x}}_i, \boldsymbol{W}_{\hat{y}} | \hat{y}) = \frac{\tilde{\boldsymbol{x}}_i \cdot \boldsymbol{W}_{\hat{y}}}{\|\tilde{\boldsymbol{x}}_i\| \|\boldsymbol{W}_{\hat{y}}\|} \tag{4}$$

Equation 4 is the cosine similarity between $\tilde{\boldsymbol{x}}_i$ and $\boldsymbol{W}_{\hat{y}}$. The cosine angle is used to validate whether the token $\tilde{\boldsymbol{x}}_i$ is pointing roughly

in the same direction as $\boldsymbol{W}_{\hat{y}}$. A higher score indicates that the angle between $\boldsymbol{W}_{\hat{y}}$ and $\tilde{\boldsymbol{x}}_i$ is smaller. Similarly the phrase attribution score is defined as:

$$\Phi(\boldsymbol{z}, \boldsymbol{W}_{\hat{y}}|\hat{y}) = \frac{\boldsymbol{z} \cdot \boldsymbol{W}_{\hat{y}}}{\|\boldsymbol{z}\|\|\boldsymbol{W}_{\hat{y}}\|} \tag{5}$$

where $\boldsymbol{z} \in \mathbb{R}^d$ is the pooled-mean of the phrase/span $(\tilde{\boldsymbol{x}}_1, ..., \tilde{\boldsymbol{x}}_n)$ and n is the total number of tokens in the span.

3.5 Hierarchical Explanation

In addition to computing feature and phrase attribution, SFFA's representations can also be used to generate hierarchical explanations, i.e., it can segment a text recursively into phrases and then words as components of an explanation. Similar to the work of [7], we use a top-down approach to partition the tokens into subsets. The hierarchical explanation exploits these multiple levels, where each level consists of multiple subsets (except level 0). We divide each subset into smaller text subsets according to the positions of the lower interaction. The interaction score between two subsets (left, right) is defined as follows:

$$\phi(\bar{\boldsymbol{s}}_1, \bar{\boldsymbol{s}}_2, \boldsymbol{W}_{\hat{y}}|\hat{y}) = \left| \frac{\bar{\boldsymbol{s}}_1 \cdot \boldsymbol{W}_{\hat{y}}}{\|\bar{\boldsymbol{s}}_1\|\|\boldsymbol{W}_{\hat{y}}\|} - \frac{\bar{\boldsymbol{s}}_2 \cdot \boldsymbol{W}_{\hat{y}}}{\|\bar{\boldsymbol{s}}_2\|\|\boldsymbol{W}_{\hat{y}}\|} \right|, \tag{6}$$

where $\bar{s}_1$, and $\bar{s}_2$ represent the mean-pooling of subsets 1 and 2, respectively.

Algorithm 1 Top-down approach for hierarchical explanation

Given: Token embedding $\tilde{x} \in \mathbb{R}^{m \times d}$, $\boldsymbol{W}_{\hat{y}} \in \mathbb{R}^d$
Initialize the first subset $\mathcal{S} \leftarrow \{\tilde{x}_{(0,m]}\}$
Initialize the hierarchy $\mathcal{H} = [\mathcal{S}]$
for $level \leftarrow 1$ **to** $m-1$ **do**
  Initialize minimum score $sc = 2$
  foreach $s \in \mathcal{S}$ **do**
    if $|s| = 1$ **then**
      Continue
    end
    $l \leftarrow |s|$
    Initialize scores $\rho = \emptyset$
    for $st \leftarrow 1$ **to** l **do**
      $\alpha \leftarrow \phi(sub_{(0,st]}, sub_{(st,l]}, \boldsymbol{W}_{\hat{y}}|\hat{y})$
      $\rho.add(\alpha)$ Add interaction score
    end
    if $\min(\rho) < sc$ **then**
      $\xi \leftarrow \arg\min(\rho)$ index
      $sc \leftarrow \min(\rho)$ score
      $s^{(t)} \leftarrow s$
    end
  end
  $s^{(l)} \leftarrow s^{(t)}_{(0,\xi]}$ Get left subset
  $s^{(r)} \leftarrow s^{(t)}_{(\xi,:]}$ Get right subset
  $\Gamma \leftarrow \forall e \in \mathcal{S}, e \neq s^{(t)}$ Find other subsets except picked one
  $\Gamma \leftarrow \Gamma \cup \{s^{(l)}\}$ Add left subset
  $\Gamma \leftarrow \Gamma \cup \{s^{(r)}\}$ Add right subset
  $\mathcal{S} \leftarrow \Gamma$ Start from this subset
  $\mathcal{H}.add(\mathcal{S})$
end
Output: $\mathcal{H}$

The main algorithm for partitioning the tokens into different text spans is summarized in the Algorithm 1 summarized above. The objective is to segment any subset which contains more than two words. The split is based on finding the minimum interaction between two subsets using Equation 6. We consider all possible subsets, then pick the partitions which result in the minimum interaction score between the two subsets (see Figure 2). In Figure 2, Level 0 shows the overall prediction outcome (negative), and each subsequent level shows how groups of words are combined and affecting the prediction outcome. You can see that "of good" has a positive label in Level 4, but after "of good" is combined with "waste", "waste of good" has a negative label in Level 3. To identify the most salient span, i.e., the span with length > 1 that has the biggest attribution score, we use Equation 4 for each subset identified using Algorithm 1, $w.r.t$ the corresponding centroid vector and then select the span with the biggest attribution score.

Figure 2. An example for negative sentiment classification. Numbers on the right bar represent the range of the scores based on the color. The proposed approach shows how the model is using the word good with waste to predict the sentiment as negative. This kind of composition is important to understand the interacting features.

4 Experiments

We evaluate SFFA with three different deep learning architectures: an attention bi-directional LSTM (Attbilstm) [43], a convolutional neural network (CNN) [20], and RoBERTa, a large language model (LLM) pretrained system [24]. Our experimental data sets are similar to [7], and we focus on sentiment classification (IMDB[1] [25], the YELP dataset challenge[2] [1]) and in addition, we perform document classification on AG news dataset[3] [42]). For evaluation, we rely on both proxy metrics and human evaluation, to demonstrate the effectiveness of SFFA.

A summary of the benchmark datasets is shown in Table 1. We use 10% of the training data as the validation set. The networks were trained on an NVIDIA GeForce RTX 3070 8 GB GDDR6. The reported results were based on the average of two runs.

We used different architectures for training and here we provide the number of parameters used for each architecture: a) for Attbilstm: 1) IMDB:3,624,706 , 2) YELP:3,624,706 , 3) AG news: 16,342,788. b) for CNN: 1) IMDB:3,361,282 , 2) YELP:3,361,282 , 3) AG news: 16,079,364. c) for RoBERTa: 1)IMDB:82,710,530 , 2) YELP:82,710,530 , 3) AG news: 82,712,068.

The average inference time (prediction and generating attribution scores) for Attbilstm on YELP (input length: 50, number of samples: 1024) is 0.000868 second. For a CNN trained on AG news, the average inference time is 0.0008495 seconds. We used Tensorflow

[1] https://keras.io/api/datasets/imdb/
[2] https://www.yelp.com/datasetYELP
[3] https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset?select=train.csv

version 2 for implementing the proposed approach. All datasets used in the experiments are publicly available.

Datasets	Labels	Average length	Train	Test
IMDB	2	100	25000	25000
YELP	2	50	110400	27600
AG news	4	20	102080	25520

Table 1. Summary statistics for the benchmarks. Dataset language: English.

Implementation details The embedding vector and hidden layer feature vector sizes were set to 256 for the CNN and Attbilstm methods. We did not use any pre-trained embeddings as in [7]. We use the Adam optimizer [21] with a learning rate of 0.0001 and a batch size of 512. We train for a maximum of 800 epochs with early stopping if the validation-set score has not improved in 20 consecutive epochs. Performance evaluation between the alternative deep networks, with and without SFFA, is shown in Table 2. We found that performance has no significant degradation with SFFA. It means SFFA can be used to provide an explanation on the prediction without degrading the prediction accuracy. We adopt multiple metrics from prior work to evaluate word-level explanation and hierarchical explanations. The token-level metrics only measure the local fidelity by deleting words and comparing the probability change of the predicted class. The hierarchical explanation metric evaluates the interaction between words within a given text span.

	Attbilstm			CNN		
	IMDB	YELP	AG news	IMDB	YELP	AG news
Baseline (without SFFA)	0.8301	0.89	0.88	0.833	0.856	0.876
SFFA	0.822	0.89	0.88	0.83	0.853	0.886

Table 2. SFFA's model accuracy on three benchmarks

4.1 Feature Attribution Evaluation

To show that our approach is effective, we first need to prove that it provides accurate attribution scores before evaluating its effectiveness on hierarchical explanations. We adopt the ERASER evaluation method [12] to assess the attribution scores. **ERASER** proposes two metrics to measure the quality of the explanation:

Comprehensiveness measures whether all required model features to make a prediction are selected. For example, given $\boldsymbol{x}$, the new text input is defined as $\hat{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{z}$, where $\boldsymbol{z}$ is the set of relevant tokens identified as salient within the text. Let $f_\theta(\boldsymbol{x})_{\hat{y}}$ be the network's output for class $\hat{y}$. Comprehensiveness is defined as follows:

$$\texttt{Comprehensiveness} = f_\theta(\boldsymbol{x})_{\hat{y}} - f_\theta(\hat{\boldsymbol{x}})_{\hat{y}} \tag{7}$$

A higher score implies that the identified feature tokens in $\boldsymbol{z}$ were more influential in the model's predictions, compared with other words.

Sufficiency measures whether the identified salient tokens have enough information to trigger the model to predict the same label as obtained from using the full text:

$$\texttt{Sufficiency} = f_\theta(\boldsymbol{x})_{\hat{y}} - f_\theta(\boldsymbol{z})_{\hat{y}} \tag{8}$$

A lower sufficiency score implies that the explanations are more adequate for a model's prediction. Comprehensiveness and sufficiency metrics are similar to the ROAR framework [15], but do not require re-training. We calculate the area over the perturbation curve (AOPC) for both comprehensiveness and sufficiency using various token percentage selection: $5\%, 10\%, 15\%, 20\%$, and 25% for IMDB and YELP, and $25\%, 35\%, 45\%, 55\%$, and 65% for AG news.

Results We compare our method with several baselines, such as LIME [32], IntGrad [37], L-Shapley and C-Shapley [9] and HEDGE [7]. Table 3 compares the scores by different explanation methods assessed by ERASER. We have found that SFFA provides a significant improvement in both comprehensiveness and sufficiency for sentiment and news classification. To show the effectiveness of our method in explaining predictions, we compared our method with the baselines in terms of the degradation score and log-odds score. The results are in the Appendix.

4.2 Evaluating Hierarchical Explanations

Here we focus on evaluating the quality of hierarchical explanations. We adopt the cohesion-score metric proposed by [7]. The cohesion-score measures a heuristic called "the synergy of words" within a text span used in the model prediction, by shuffling the words to see the probability change on the predicted label. Given a salient span $\boldsymbol{x}_{(a,b]}$ that was identified using our approach (selecting the span with the highest interaction score from our portioning pipeline), we randomly select a position in the token sequence $\boldsymbol{x}_1, ..., \boldsymbol{x}_a, \boldsymbol{x}_{b+1}, ..., \boldsymbol{x}_m$ and re-insert a word. The process is repeated until a shuffled version of the original sentence $\boldsymbol{t}$ is constructed. Intuitively, the words in an important text span have strong interactions. By perturbing such interactions, we expect to observe the output probability decreasing. This cohesion score is defined as follows:

$$\texttt{cohesion} = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{100}\sum_{q=1}^{100}(p(\hat{y}|\boldsymbol{x})_i - p(\hat{y}|\boldsymbol{t}^{(q)})_i), \tag{9}$$

where $\boldsymbol{t}^{(q)}$ is the q^{th} perturbed version of $\boldsymbol{x}$ and M is the total number of samples.

We repeat the experiment 100 times. Only salient spans are considered in this evaluation. Higher score means that the identified span is more critical for predicting the label.

Results Table 4 compares the cohesion score between SFFA and HEDGE on three benchmarks using a CNN and an Attbilstm. Note that HEDGE is a post-hoc algorithm for hierarchical explanations. The results indicate that SFFA is better at capturing the interaction and identifying salient subsets from the sentence compared to using HEDGE. The advantage of this approach over post-hoc is that explanation-specific representations are *learned directly by the deep network*.

4.3 Human Evaluation

Our human evaluators were undergraduate and graduate students from diverse majors (a total of 16 individuals). For both SFFA and HEDGE, we consider only the most important span/phrase for evaluation. Similar to [7], we focus on sentiment classification and provide sentences from IMDB and YELP. We follow Chen's experiment [7] and asked evaluators to predict the type of the sentiment from the provided explanation from {"Positive," "Negative," "N/A"}, where "N/A" means that the evaluator cannot predict the sentiment from the provided explanations. The model used in this study was trained using an Attbilstm. We randomly picked 100 reviews from the two

		SFFA	L-Shapley	C-Shapley	IntGrad	LIME			SFFA	L-Shapley	C-Shapley	IntGrad	LIME
Attbilstm	**IMDB**						**CNN**	**IMDB**					
	Comprehensiveness↑	0.643	0.4136	0.127	0.423	0.459		Comprehensiveness ↑	0.476	0.438	0.418	0.408	0.375
	Sufficiency↓	0.020	0.083	0.101	0.061	0.185		Sufficiency↓	-0.134	-0.125	−0.118	−0.115	0.014
	YELP							**YELP**					
	Comprehensiveness↑	0.631	0.406	0.394	0.402	0.439		Comprehensiveness↑	0.513	0.468	0.466	0.472	0.207
	Sufficiency↓	0.110	0.266	0.268	0.150	0.234		Sufficiency↓	−0.138	−0.133	−0.132	−0.141	0.011
	AG news							**AG news**					
	Comprehensiveness↑	0.721	0.295	0.259	0.483	0.291		Comprehensiveness↑	0.684	0.212	0.167	0.351	0.275
	Sufficiency↓	0.003	0.07	0.089	0.031	0.103		Sufficiency↓	−0.021	0.134	0.162	0.044	0.111

Table 3. Eraser benchmark scores: Sufficiency and comprehensiveness are in terms of AOPC. Lower scores are better for sufficiency and higher scores are better for comprehensiveness.

Methods	Models	IMDB	YELP	AG news
		Cohesion-score		
HEDGE	CNN	0.092	0.079	0.052
	Attbilstm	0.071	0.055	0.023
SFFA	CNN	0.129	0.113	0.094
	Attbilstm	0.099	0.191	0.052

Table 4. Cohesion scores between SFFA and HEDGE.

benchmarks (50:IMDB, 50:YELP). We measure the number of human annotations that align with the model's prediction (e.g., the human annotation on the span matches the model prediction on the full sentence), and then define the coherence score as the ratio between the coherent annotations and the total number of examples [7].

Results Table 5 compares the coherence score between each of SFFA and HEDGE explanations and the human annotation.

SFFA outperformed HEDGE, achieving relatively better scores, which suggests that it is better aligned with human annotation at identifying important spans.

Methods	Coherence score
HEDGE	0.51
SFFA	0.8136

Table 5. Human evaluation of SFFA and HEDGE with Attbilstm on IMDB and YELP benchmarks.

4.4 SFFA to Pre-trained Transformers

In the previous section, we showed how we can train a neural network model using SFFA for document classification. Here, we apply SFFA to a pre-trained transformer model such as RoBERTa [24] and show we can generate faithful explanations even from a pre-trained model. We evaluate our approach on the three benchmarks and use RoBERTa [24]. We incorporate SFFA for Transformers by allowing gradients from the explanation-related loss function to pass through the token embedding. The output layer is fine-tuned for the downstream classification task. Due to the page limit, we report only the results on ERASER for token-level evaluation. Similarly, we limit our experiments to Shapley-based methods because of challenges with other baseline implementation. For hierarchical explanations, we again use the cohesion-score metric to evaluate the generated spans.

Results The performance comparison is summarized in Table 6. Table 7 compares ERASER's scores on three benchmarks against Shapley-based methods. SFFA outperforms post-hoc methods in both metrics achieving better scores. The cohesion scores for RoBERTa are shown in Table 8.

	IMDB	YELP	AG news
Baseline without SFFA	0.838	0.916	0.9074
SFFA	0.8401	0.915	0.8924

Table 6. RoBERTa: Model's accuracy.

	SFFA	L-Shapley	C-Shapley
IMDB			
Comprehensiveness↑	0.506	0.192	0.146
Sufficiency↓	0.085	0.166	0.159
YELP			
Comprehensiveness↑	0.497	0.204	0.187
Sufficiency↓	0.073	0.172	0.166
AG news			
Comprehensiveness↑	0.535	0.128	0.127
Sufficiency↓	0.035	0.125	0.125

Table 7. ERASER benchmark score: Comprehensiveness and sufficiency are in terms of AOPC. Results are based on RoBERTa's model.

4.5 NLI: Natural Language Inference

Here we show that SFFA can also work well on a different NLP task, such as NLI. We use the Stanford SNLI dataset[4] for the NLI experiment. For the premise $\boldsymbol{x}^{(p)}$ and hypothesis $\boldsymbol{x}^{(h)}$, SFFA's objective is to predict the relation as one of entailment, contradiction, and neutral. We follow the same steps presented in Section 3. However, the sentence vector is defined as $\bar{\boldsymbol{x}}^{(p)} + \bar{\boldsymbol{x}}^{(h)}$. We integrate our method with [30] to allow gradients from the explanation-related loss function to pass through the token embedding on the SNLI dataset [5]. The attribution score of each token $\boldsymbol{x}_i^{(p)}$ in premise is obtained using Equation 4. The token embedding vector is modified to consider the hypothesis sentence as follows: $\boldsymbol{x}_i^{(p)} - \bar{\boldsymbol{x}}^{(h)}$. Our intuition is based on [11], i.e., we consider the difference between features of each vector as a way of capturing the information of both features. However, this is a feature engineering approach and there can be many ways to combine the two vectors to predict the final label. The number of parameters used for training the model on SNLI is 8054020. For evaluation, we again use comprehensiveness and sufficiency from ERASER. Table 9 shows that SFFA generates better explanations than Shapley.

4.6 Qualitative Result

In addition to the human evaluation, here we present one example of Figure 2 to provide qualitative analysis, More examples for qualitative analysis are in the Appendix. Figure 2 visualizes the hierarchical

[4] https://nlp.stanford.edu/projects/snli/

Methods	Models	IMDB	YELP	AG news
		Cohesion-score		
HEDGE	RoBERTa	0.117	0.096	0.006
SFFA	RoBERTa	0.151	0.131	0.032

Table 8. Comparing the cohesion scores between SFFA and HEDGE for RoBERTa.

	SFFA	L-Shapley	C-Shapley
Comprehensiveness	0.678 ↑	0.546	0.568
Sufficiency	0.098 ↓	0.202	0.212

Table 9. ERASER in terms of AOPC for NLI.

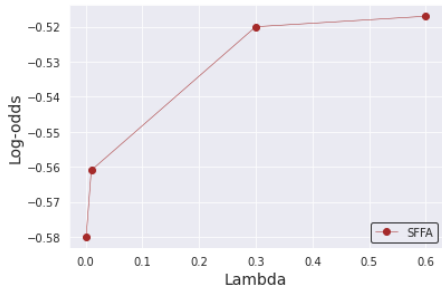

Figure 3. Log-odds as a function of λ on AG news. We found that the smaller the value of λ, the better the faithfulness of the explanation.

explanation, generated by SFFA, on a negative review. In Figure 2, the word 'good' itself was predicted as positive, but after 'good' was combined with 'waste', the phrase 'waste of good' was predicted as negative. The hierarchical explanation illustrates that the model is capturing the most salient span with the minimum number of tokens.

4.7 Ablation Study

To further understand λ in the SFFA objective function, we plot the log-odds score as a function of λ. We use the AG news and IMDB datasets trained using a CNN, mask the top five tokens from each sample, and then analyze the change in the log-odds. The results in Figure 3 show that the smaller value of λ achieves the lower log-odds score. Another ablation study by removing a term from the loss function is included in the Appendix.

4.8 HEDGE for Feature Attribution

We have shown that our approach outperforms HEDGE for hierarchical explanation, using the cohesion score. According to [7], HEDGE can also provide the attribution score. We have performed an experiment that compares the correctness of the attribution scores between SFFA and HEDGE. We use the log-odds experiment proposed by [10] and plot the change of the log-odds score as a function of removed tokens. Due to the page limit, the result is included in the Appendix. The result shows that SFFA provides more correct attribution scores. Similar to HEDGE, we also calculated the area over the perturbation curve (AOPC) [28], i.e., we calculated the average change in the prediction probability on the predicted class over all test data as a function of removed salient tokens. The AOPC values on IMDB for SFFA and HEDGE are 0.4754 and 0.3038 respectively. The higher AOPC, the higher the faithfulness of the explanation. SFFA provides a more faithful explanation than HEDGE.

4.9 Discussion

The importance of concurrent supervision The SFFA forces the features of the same class to form a unique cluster, enabling it to be combined with cosine similarity to obtain attribution scores. Our model is not shallow as it can be applied to more complex problems such as NLI. Due to the page limit, the figure about the result is in the Appendix.

Compared to existing methods Our method does not use complex models to learn explanations nor does it use post-hoc techniques to learn feature attribution. The proposed method is built on the main properties of the selected deep learning architecture, more specifically on the Softmax loss. SFFA improves the representations of the embedding layer and thus allows the generation of hierarchical explanations.

Rationale-based methods This work is different from existing methods on rationales. It forces deep networks to learn representations that can be used as a proxy for hierarchical explanations. So comparing with rationalization methods is challenging because of differences in the generated output and the objective, plus the need for more relevant metrics.

The encoder for rationale-based methods only uses the extracted tokens as the input for the classification. The rationale-based methods make predictions based on a subset and thus there is no universal proxy metric to compare the explanations of rationale-based methods with hierarchical ones.

Hierarchical explanations. Traditional feature attribution methods do not tell us how tokens and phrases are dependent on each other and how they are composed together for the final prediction. With hierarchical explanations, we can provide a comprehensive picture of how different granularity of the tokens interacts with each other and also help identify the most salient features used by the classifier.

5 Conclusion and Future Work

We have introduced a new approach to learn representations for a predictive model's explanations, which does not require additional parameters to generate an explanation. The network learns an appropriate explanation-specific representation directly from the embedding. We have demonstrated the effectiveness of SFFA in generating hierarchical explanations using the cohesion score metric on three datasets using both proxy metrics and human evaluation. In future work, we plan to extend the proposed network to other more complex NLP tasks, considering applications in both precision health and legal informatics, which require explanations on the predictions.

Acknowledgements

We would like to acknowledge the support of the Alberta Machine Intelligence Institute (Amii), Alberta Innovates, and the Natural Sciences and Engineering Research Council of Canada (NSERC) [including funding reference numbers DGECR-2022-00369 and RGPIN-2022-03469].

References

[1] 'Yelp dataset challenge', (2018).

[2] Diego Antognini and Boi Faltings, 'Rationalization through concepts', in *ACL/IJCNLP (Findings)*, (2021).

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek, 'On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation', *PloS one*, **10**(7), e0130140, (2015).

[4] jasmijn Bastings, Wilker Aziz, and Ivan Titov, 'Interpretable neural predictions with differentiable binary variables', in *Proceedings of ACL*, pp. 2963–2977, (2019).

[5] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning, 'A large annotated corpus for learning natural language inference', in *in EMNLP*, (2015).

[6] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola, 'Invariant rationalization', in *International Conference on Machine Learning*, pp. 1448–1458. PMLR, (2020).

[7] Hanjie Chen, Guangtao Zheng, and Yangfeng Ji, 'Generating hierarchical explanations on text classification via feature interaction detection', in *ACL*, (2020).

[8] Jianbo Chen and Michael Jordan, 'Ls-tree: Model interpretation when the data are linguistic', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3454–3461, (2020).

[9] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan, 'Learning to explain: An information-theoretic perspective on model interpretation', in *International Conference on Machine Learning*, pp. 883–892. PMLR, (2018).

[10] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan, 'L-shapley and c-shapley: Efficient model interpretation for structured data', *ICLR 2019*, (2018).

[11] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes, 'Supervised learning of universal sentence representations from natural language inference data', in *EMNLP*, (2017).

[12] 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', *arXiv preprint arXiv:1911.03429*, (2019).

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT press, 2016.

[14] Yaru Hao, Li Dong, Furu Wei, and Ke Xu, 'Self-attention attribution: Interpreting information interactions inside transformer', in *Proceedings of the AAAI 2021*, volume 35, pp. 12963–12971, (2021).

[15] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim, 'A benchmark for interpretability methods in deep neural networks', *Advances in neural information processing systems*, **32**, (2019).

[16] Aya Abdelsalam Ismail, Hector Corrada Bravo, and Soheil Feizi, 'Improving deep learning interpretability by saliency guided training', *Advances in Neural Information Processing Systems*, **34**, 26726–26739, (2021).

[17] Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren, 'Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models', in *ICLR*, (2019).

[18] Mi-Young Kim, Shahin Atakishiyev, Housam Khalifa Bashier Babiker, Nawshad Farruque, Randy Goebel, Osmar R. ZaÃ¯ane, Mohammad-Hossein Motallebi, Juliano Rabelo, Talat Syed, Hengshuai Yao, and Peter Chun, 'A multi-component framework for the analysis and design of explainable artificial intelligence', *Machine Learning and Knowledge Extraction*, **3**(4), 900–921, (2021).

[19] Mi-Young Kim, Ying Xu, Osmar R. Zaiane, and Randy Goebel, 'Recognition of patient-related named entities in noisy tele-health texts', *ACM Trans. Intell. Syst. Technol.*, **6**(4), (jul 2015).

[20] Yoon Kim, 'Convolutional neural networks for sentence classification', *EMNLP*, (2014).

[21] Diederik P. Kingma and Jimmy L. Ba, 'Adam: A method for stochastic optimization. cornell university library', *arXiv preprint arXiv:1412.6980*, (2017).

[22] Tao Lei, Regina Barzilay, and Tommi Jaakkola, 'Rationalizing neural predictions', in *Proceedings of EMNLP*, pp. 107–117, (2016).

[23] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang, 'Large-margin softmax loss for convolutional neural networks', in *ICML*, (2016).

[24] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, 'Roberta: A robustly optimized bert pretraining approach', *arXiv preprint arXiv:1907.11692*, (2019).

[25] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts, 'Learning word vectors for sentiment analysis', in *Proceedings of ACL*, pp. 142–150, (2011).

[26] W James Murdoch, Peter J Liu, and Bin Yu, 'Beyond word importance: Contextual decomposition to extract interactions from lstms', in *ICLR*, (2018).

[27] W James Murdoch and Arthur Szlam, 'Automatic rule extraction from long short term memory networks', in *ICLR*, (2017).

[28] Dong Nguyen, 'Comparing automatic and human evaluation of local explanations for text classification', in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1069–1078, (2018).

[29] Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer, 'An information bottleneck approach for controlling conciseness in rationale extraction', in *EMNLP*, pp. 1938–1952, (2020).

[30] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit, 'A decomposable attention model for natural language inference', in *in EMNLP*, (2016).

[31] Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh, 'Overview and discussion of the competition on legal information extraction/entailment (COLIEE) 2021', *The Review of Socionetwork Strategies*, **16**(1), 111–133, (feb 2022).

[32] 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*, pp. 1135–1144. ACM, (2016).

[33] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje, 'Learning important features through propagating activation differences', in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3145–3153. JMLR. org, (2017).

[34] Sandipan Sikdar, Parantapa Bhattacharya, and Kieran Heese, 'Integrated directional gradients: Feature interaction attribution for neural nlp models', 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. 865–878, (2021).

[35] Chandan Singh, W James Murdoch, and Bin Yu, 'Hierarchical interpretations for neural network predictions', in *ICLR*, (2018).

[36] Mukund Sundararajan, Ankur Taly, and Qiqi Yan, 'Axiomatic attribution for deep networks', *Proceedings of International Conference on Machine Learning (ICML)*, (2017).

[37] Mukund Sundararajan, Ankur Taly, and Qiqi Yan, 'Axiomatic attribution for deep networks', in *ICML*, p. 3319–3328, (2017).

[38] 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).

[39] 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, (2019).

[40] Matthew D Zeiler and Rob Fergus, 'Visualizing and understanding convolutional networks', in *European conference on computer vision*, pp. 818–833. Springer, (2014).

[41] Die Zhang, Hao Zhang, Huilin Zhou, Xiaoyi Bao, Da Huo, Ruizhao Chen, Xu Cheng, Mengyue Wu, and Quanshi Zhang, 'Building interpretable interaction trees for deep nlp models', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 14328–14337, (2021).

[42] Xiang Zhang, Junbo Zhao, and Yann LeCun, 'Character-level convolutional networks for text classification', in *Advances in Neural Information Processing Systems*, pp. 649–657, (2015).

[43] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu, 'Attention-based bidirectional long short-term memory networks for relation classification', in *Proceedings of acl 2016)*, pp. 207–212, (2016).

[44] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling, 'Visualizing deep neural network decisions: Prediction difference analysis', in *Proceedings of ICLR*, (2017).

† Please contact the corresponding author for any appendices or supplementary material mentioned in the paper.