ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Artificial Intelligence

Master thesis in Natural Language Processing

A Self-Supervised Attribution Method for Explaining Neural Networks

Supervisor:
Prof. PAOLO TORRONI

Co-supervisor:
Prof. AKIKO AIZAWA

Candidate:
TIAN CHENG XIA

Session V
Academic year 2024-2025

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Artificial Intelligence

Master thesis in Natural Language Processing

A Self-Supervised Attribution Method for Explaining Neural Networks

Supervisor:
Prof. PAOLO TORRONI

Co-supervisor:
Prof. AKIKO AIZAWA

Candidate:
TIAN CHENG XIA

Session V
Academic year 2024-2025

"It may seem very foolish in your eyes, but really I don't know how you deduced it."

Dr. John H. Watson in *The Crooked Man*

Arthur Conan Doyle

Abstract

The inherent black-box nature of deep neural networks poses significant challenges to their trustworthiness, fairness, and development in real-world applications. A well-known class of post-hoc explainability methods is based on producing attribution maps to score the input features of a model. However, existing methods share some limitations such as sensitivity to the choice of method-specific hyperparameters, computational cost, and trade-offs between faithfulness to the model's decision process and visual clarity of the attribution maps. We propose a self-supervised attribution method that tackles explainability as a learning problem. The approach consists of training, in a self-supervised manner with self-calibrating method-specific hyperparameters, a dedicated model that can produce attribution maps in a single forward pass by using the intermediate activations of the target model. We benchmark our method on text, image, and multimodal classification tasks across nine different datasets and evaluate it both quantitatively and qualitatively. Our results show that our method, compared to other baselines such as Saliency, Guided Backpropagation, Integrated Gradients, DeepLIFT, and SHAP-based methods, is the one achieving the best trade-off between faithfulness to the underlying model and visual clarity of the produced attribution maps, indicating that it is able to balance both requirements while being easier to use in practice and computationally less expensive at inference.

Contents

1 Introduction **1**

2 Background **3**

2.1 Explainability 3

2.2 Motivation 6

2.3 Related Work 7

3 Self-Supervised Attribution **9**

3.1 Formalization 9

3.2 Architecture 10

3.3 Training and Inference 12

3.3.1 Text Modality 14

3.3.2 Image Modality 15

3.3.3 Multimodality 16

4 Experimental Setup **19**

4.1 Metrics 19

4.2 Baselines 21

4.3 Datasets 24

4.3.1 Text Classification 25

4.3.2 Image Classification 26

4.3.3 Multimodal Classification 26

4.4 Architectures 27
4.5 Implementation Details 28

5 Results **31**
5.1 Text Classification 31
5.2 Image Classification 34
5.3 Multimodal Classification 37
5.4 Ablation Study 40

6 Conclusion **43**

Bibliography **45**

1 Introduction

The rapid advancement of deep learning has led to neural networks being deployed in an increasingly wide range of real-world applications across different domains. While these models often achieve remarkable performance, their internal decision-making processes remain a black-box, making it difficult to understand why a specific prediction is made. This lack of transparency raises important concerns around trust, fairness, and accountability while also making it difficult to analyze and debug these models from a technical perspective [40, 1].

The field of explainable artificial intelligence has grown considerably in response to these demands [26, 48, 20]. Among the different families of explainability methods, attribution-based post-hoc approaches have emerged as a prominent class of techniques that aim to quantify the contribution of each input feature to a given model prediction. These methods are applied transparently after training to produce attribution maps that assign an importance score to each input element.

However, existing attribution methods share a number of limitations. Firstly, several of them require providing either some reference baselines whose choice can significantly influence the resulting explanations or have method-specific hyperparameters that need tuning [46, 42, 29]. Secondly, many methods have relatively high computational cost at inference time as they rely on repeated backpropagation passes or stochastic sampling procedures. Also, some methods produce attribution maps that are faithful to the model's decision process but lack in visual clarity,

while others that produce visually appealing maps but lack faithfulness [10, 25].

In this thesis, we propose an attribution method that addresses some of the limitations of existing approaches by framing the task as a learning problem. The core idea is to train a dedicated neural network module to predict attribution maps with a single forward pass using information from the intermediate representations of the model being explained. We show that this can be done in a self-supervised manner, without requiring ground-truth explanations, and without the need to introduce new method-specific hyperparameters as they can be automatically calibrated during training.

The main contributions of this thesis are the following: (a) We propose a self–supervised attribution method that trains a dedicated neural network to produce attribution maps without requiring ground-truth explanations and without introducing new method-specific hyperparameters; (b) We provide an implementation of the method for text, image, and multimodal classifiers; (c) We comprehensively evaluate the method on multiple datasets spanning different modalities using both quantitative metrics and qualitative analysis.

The rest of this thesis is organized as follows. Chapter 2 provides background on explainability, motivates the proposed approach, and reviews related work. Chapter 3 presents the formalization of the method, and describes its architecture and training procedure. Chapter 4 details the experimental setup, the metrics, baselines, datasets, and implementation choices. Chapter 5 reports and discusses the quantitative and qualitative results, as well as an ablation study of our choices. Chapter 6 contains some closing remarks and outlines possible future directions.

Code availability. The code to reproduce all the experiments presented in this thesis is available at `https://github.com/NotXia/self-supervised-xai`.

2 Background

2.1 Explainability

Explainability is the field that studies and designs methods to make a system understandable to humans, intended as a spectrum of different stakeholders with different goals and needs. In fact, across different domains, explanations can be used to improve model performance, enhance and support human decision-making, or analyze biases [26]. It is a multidisciplinary field that spans across different areas such as computer science, psychology, and ethics [48]. Indeed, explainability not only has to be technically functional and correct, but should also account for users' needs and rights [20]. Moreover, the necessity to provide an explanation for the output of a model is also mentioned in some legislations. For instance, in the European Union (EU), explainability is mentioned as a right in article 22 of the General Data Protection Regulation (GDPR) [5] and it is one of the principles of trustworthy artificial intelligence (AI) [9].

Different taxonomies can be used to describe explainability methods. From a high-level classification, an explainability method can be employed at the pre-modelling, *ante-hoc* modelling, or *post-hoc* modelling levels [14, 45, 26]. Pre-modelling explainability involves methods used before developing the model and it is mainly used to explore and analyze the data through dimensionality reduction or feature selection techniques. *Ante-hoc* explainability aims at designing models

that are intrinsically explainable; this includes models such as k-nearest neighbors, decision trees, and other more advanced approaches. *Post-hoc* explainability, which this thesis focuses on, includes methods that are applied on a model after development; this includes a variety of methods such as perturbation, gradient-based, or proxy methods that will be further discussed in the following sections [14, 45]. More in-depth, an explanation can be local, if it works on single instances, or global, if it aims at explaining the whole model [14]. Furthermore, in the case of *post-hoc* explainability, a method can be model-specific or model-agnostic depending on whether it is independent of the underlying model or not [14, 26].

In practical terms, this work mainly focuses on *post-hoc* local explainability for neural networks. In particular, the proposed method falls within the class of attribution-based methods, which aim at computing importance scores for each input feature with respect to a specific model prediction. These methods produce explanations in the form of relevance maps or saliency scores, indicating how much each feature contributed to the output for a given instance. Several existing approaches have been proposed in the literature. In this section, we briefly describe them and provide more details in Section 4.2 when describing our baselines. One of the simplest methods is based on Saliency [43], which builds on the idea of computing the gradient of the model output with respect to the input features and interpret its magnitude as a measure of sensitivity of how small changes in the input space affect the prediction. Although conceptually simple and computationally efficient, a major drawback of this method is that it often suffers from noise and can produce explanations that are unstable or difficult to interpret. Another method is DeepLIFT [42], which propagates attribution scores backward through the network by comparing the activation of each neuron to a reference activation. Compared to simpler methods, DeepLIFT provides more stability and avoids some gradient instability issues. Building on top of it, DeepLIFT-SHAP [30] uses DeepLIFT to

approximate Shapley values [41], which are a game theory notion based on the idea of fairly distributing feature importance, but has the drawback of being computationally expensive to compute exactly. A drawback of DeepLIFT and other methods based on it is that it is sensitive to the choice of baselines that are used as reference activations. Therefore, in some cases, it can produce drastically different results depending on which prior features are provided. Integrated Gradients [46] is another well-established method that approximates the integral of the gradient along a path from some given baseline inputs to the actual input, satisfying desirable axioms such as sensitivity and implementation invariance. Again, the choice of baselines is a critical choice that can significantly influence the resulting explanations. A closely related approach is Gradient-SHAP [29], which combines ideas from Integrated Gradients with stochastic sampling. Instead of integrating along a single path from a baseline to the input, Gradient-SHAP samples random interpolation points between multiple baseline inputs and the actual input. By averaging gradients over these paths, the method approximates Shapley values while reducing variance compared to single-baseline techniques, therefore improving on robustness but with an increase in computational cost due to the required sampling. Finally, Guided Backpropagation [44] modifies the standard backpropagation procedure by suppressing negative gradients during the backward pass through ReLU activations. Concretely, only gradients associated with positive forward activations and positive backward signals are propagated. This heuristic approach often produces sharp and visually appealing explanations, particularly in image-based settings. However, it does not necessarily provide faithful feature importance scores and does not have theoretical grounding.

2.2 Motivation

In Section 2.1, we illustrated the most commonly known attribution-based explainability methods in the literature. However, each of these methods has some issues such as sensitivity to the choice of parameters and to the choice of prior baselines. Another possible drawback is that some of these methods require multiple iterations or are based on computing the gradient by performing backpropagation along the whole length of the model at inference time to produce the attribution map, which can make the applicability of these approaches slow in practical terms.

Considering these observations, in this thesis, we propose a method based on learning a dedicated neural network to produce attribution maps, which aims at reducing some of the issues we mentioned. Firstly, we remove the need to provide some arbitrary baselines. Secondly, we remove the need of choosing method-specific hyperparameters and instead reduce it to the choice of more traditional and familiar hyperparameters for training a neural network. Thirdly, we remove some of the computational cost at inference time as a single forward pass of a smaller model produces the attribution map. Lastly, although not strictly related to the other methods we mentioned, our approach does not require having ground-truth explanations as it is trained in a self-supervised way, so that it can be applied to a wide range of scenarios.

Obviously, the approach we are proposing has its own drawbacks: it moves most of the computational cost from inference to training time and, instead of providing a limited number of baseline inputs, it requires the whole, or a large subset, of original training data of the model which is usually larger in size. Nevertheless, we argue that these limitations are negligible and still make this approach practically feasible and convenient as it does not require collecting new data and the dedicated model to produce attribution maps does not have to be a large model.

2.3 Related Work

In the literature, there are some approaches closely related to the idea we presented in the previous section. In the context of large language models, Barkan et al. [2] propose a framework called Attributive Masking Learning where they train an auxiliary model to mask input tokens while maintaining the output of the model as close as possible to the original one. Instead, Bhattacharya et al. [3] propose a distillation approach, where a dedicated model produces attribution maps that are then used to mask the input and fed through a student network that has to mimic the original model to explain. Liu et al. [23] also build on the same idea of learning a dedicated explanation model in the case of text classification. However, their approach requires knowing the ground-truth explanations, which is not always possible to have. Finally, Kanehira et al. [15] propose a framework in the context of image classification that consists of learning two dedicated networks to provide a textual explanation and visual examples, respectively. However, it also requires ground-truth information, which makes it hard to apply in practice.

Overall, to the best of our knowledge, there is no prior work in the literature that proposes a data-agnostic learned explainability method for producing attribution maps that works across different data types and does not require having ground-truth of the explanations.

3 Self-Supervised Attribution

3.1 Formalization

The idea behind the method proposed in this thesis relies on the assumption that the relevant features that result in the prediction of a given output also allow the same outcome to be predicted if isolated from the original full input. Building on top of this assumption, the intuition is that an explanation can be learned in a self-supervised manner by enforcing the predicted attribution maps to preserve the original performance of the model.

Formally, given a model to explain f, the problem we aim to solve is the following:

$$\min_{\mathbf{S}\in[0,1]^N} \|f(\mathbf{x}) - f(\mathbf{x} \otimes \mathbf{S})\| + \gamma \|\mathbf{S}\|, \tag{3.1}$$

where:

- $\mathbf{S}$ is the attribution map we aim to find,
- N is the size of the input space,
- $\mathbf{x}$ is the input of the model f,
- $\otimes$ denotes the operation of applying the attribution map $\mathbf{S}$ on the input $\mathbf{x}$. In our case, we use the product and $\mathbf{S}$ can be interpreted as a weight of the input.

In this formulation of the problem, the first term imposes that the outcome of the model where the input is filtered based on the attribution scores should be close to the one using the full input, while the second term encourages sparsity in attribution scores and avoids the trivial solution where $\mathbf{S}$ is a matrix of ones.

As is, the current formulation of the problem defines a model-agnostic method for explainability. In this thesis, we consider the case where f is a neural network. This allows us to approach the problem from a different perspective by replacing the first term in Equation (3.1) with the loss function $\mathcal{L}$ used to train f. Therefore, the resulting problem becomes:

$$\min_{\mathbf{S}\in[0,1]^N} \mathcal{L}(\mathbf{x} \otimes \mathbf{S}; f) + \gamma \|\mathbf{S}\| . \tag{3.2}$$

In other words, in the case of neural networks, the problem of maintaining the same output of the original input can be reduced to the problem of preserving the loss function when using the filtered input.

3.2 Architecture

In practical terms, we solve the problem described in Section 3.1 by introducing a new model on top of the existing one. In this work, we focus on explaining traditional fine-tuned classifiers composed of a feature encoder and a classification head. The overall idea of the architecture is depicted in Figure 3.1.

The flow between the encoder and the classifier is as usual. The newly introduced model, called scorer for simplicity, should take as input some information of the input features of the explainee model and it outputs a score for each of them. In case of classifiers, we use the embedding space of the encoder as inputs for the scorer.

We also note that in principle we could provide to the scorer the original input itself. Intuitively, this formulation would still allow to learn some reasonable

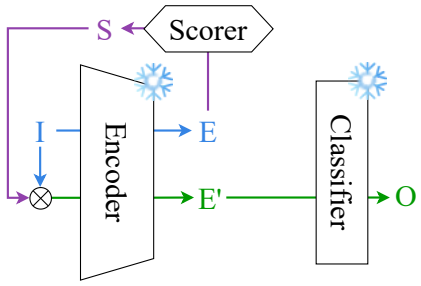


Figure 3.1: General idea of the method. Given a traditional classifier (encoder and classification head), we introduce a new module (scorer) that learns to score the input I of the encoder given its embeddings E.

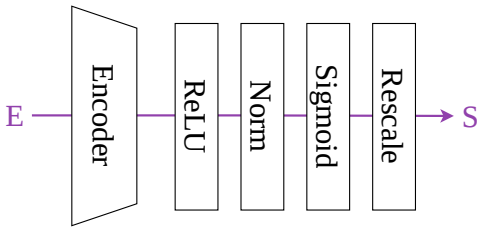


Figure 3.2: Architecture of the scorer

attribution maps that highlight what is relevant in the input as the goal is still to minimize the task specific loss. However, it would require the scorer to solve the whole classification task as well and no signal from the target model to explain would be included in the flow, so the result of this setup would be closer to an explainable-by-design model.

In terms of architecture, a crucial choice for the scorer, although unusual, is how its output head is designed. A high-level depiction of the architecture is provided in Figure 3.2. The scorer itself contains an encoder to refine the input and the output activations it produces are passed through: (a) a rectified linear unit (ReLU) activation, (b) a sigmoid activation, and (c) rescaled into $[0, 1]$. The ReLU activation is used to impose a strong distinction between relevant signal

(activation greater than 0) and background that is not useful for the attribution map. The sigmoid function squeezes the scores into $[0, 1]$ and adds a further level of non-linearity, and the final rescaling actually ensures that the scores span from 0 to 1.

3.3 Training and Inference

The scorer module introduced to compute attribution scores is implemented as a neural network and therefore requires training. As we aim at solving the problem defined in Section 3.1, we can reformulate it to an optimization problem that can be used to train neural networks. Therefore, we can define the loss function as the following:

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \mathbf{S}; f) = \mathcal{L}_{\text{task}}(\mathbf{x} \otimes \mathbf{S}; f) \\ + \gamma_1 \left(\frac{1}{|\mathbf{S}|} \sum_{i}^{|S|} s_i \right)^2 \\ + \gamma_2 \frac{1}{|\mathbf{S}|} \sum_{i}^{|S|} (s_i \cdot (1 - s_i))^2\end{aligned} \tag{3.3}$$

where $\mathcal{L}_{\text{task}}$ is the task-specific loss (cross-entropy in our case), the first penalty term encourages sparsity in the attribution scores so that few positions have high values, and the second penalty term is for pushing scores to be closer to either 0 or 1 so that it is more similar to a binary mask. γ_1 and γ_2 are instead hyperparameters to weight the importance of the two penalties.

Ideally, we would like to avoid having to choose and tune arbitrary hyperparameters such as γ_1 and γ_2. It turns out that, by how the problem is posed, it is possible to define these two hyperparameters in such a way that they are self-calibrating. We are able to do this by exploiting the information we have about the loss function during training. The idea is to use the inverse distance between

the loss evaluated with the masked input and the original one as the weight for the penalties. In this way, the intuition is that when the produced attribution map is preserving the loss, the weight of the penalties can be increased, while if the original loss is not preserved, the penalties are down-weighted so that the learning process can focus on preserving the original capabilities first without being drifted away by the penalties. Therefore, our final training loss is defined as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \mathbf{S}; f) = \mathcal{L}_{\text{task}}(\mathbf{x} \otimes \mathbf{S}; f) \\ + \gamma \left(\frac{1}{|\mathbf{S}|} \sum_{i}^{|S|} s_i \right)^2 \\ + \gamma \frac{1}{|\mathbf{S}|} \sum_{i}^{|S|} (s_i \cdot (1 - s_i))^2 \\ \text{with } \gamma = \frac{1}{\texttt{clip}\{\mathcal{L}_{\text{task}}(\mathbf{x} \otimes \mathbf{S}; f) - \mathcal{L}_{\text{task}}(\mathbf{x}; f), [0, 1]\}},\end{aligned} \tag{3.4}$$

where γ is the self-calibrating weight based on the distance between losses and `clip` constrains the weight in $[0, 1]$ to avoid having values that grow uncontrollably, which would make the penalties reach unreasonably high magnitudes, or that go below zero, which would have the opposite effect we aim at achieving.

In terms of parameter updates, we must note that this method does not require to modify the target model as its weights are kept frozen when training the scorer, as depicted in Figure 3.1. This avoids leaking any relevant information for the scorer into the original classifier, so that we can ensure that what it learns is to actually provide attribution maps for the target model without any shortcut.

At inference time, as depicted in Figure 3.3, this method works transparently alongside the existing classification model. It does not disrupt any of its normal flow and only uses some intermediate information provided by the encoder that is fed into the scorer to determine the output attribution scores.

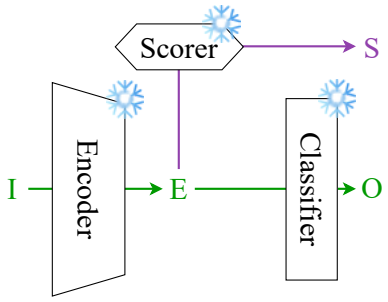


Figure 3.3: At inference time, the flow of the classifier is unaffected by the scorer. The output of the scorer corresponds to the attribution map.

3.3.1 Text Modality

For text classification, we consider the standard approach based on tokenizing the text sequence and mapping it to embeddings [22]. The flow of the classification model is therefore the following: (a) split sentence into tokens according to a vocabulary, (b) embed each token based on some static embeddings, (c) refine the initial embeddings through the text encoder, (d) pool the resulting embeddings and pass through the classifier. The embeddings at step (c) are the most information rich and can be used as the input of the scorer that produces a score for each input token. At training time, as depicted in Figure 3.4, we apply these scores to weigh the static embeddings at step (b) before passing them through the encoder. This choice is motivated by two aspects: on the one hand, as we need to integrate the learned attribution maps into the training process, we have to apply the scores at some stage where the input is differentiable and, as the input tokens are discrete and non-differentiable, the first layer at which this is possible is after applying the static embeddings. On the other hand, applying the learned attribution maps on the embeddings produced by the final text encoder would have the risk of being ineffective as each token already had the opportunity to interact with each other

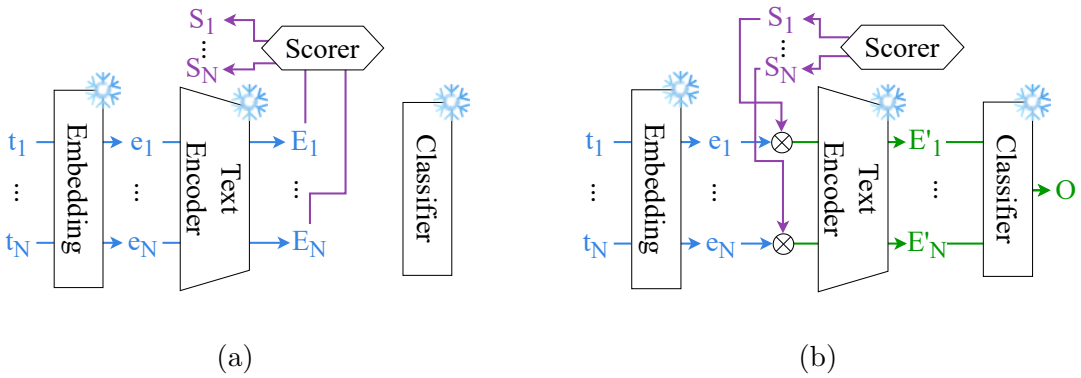


Figure 3.4: Training architecture for text classifiers

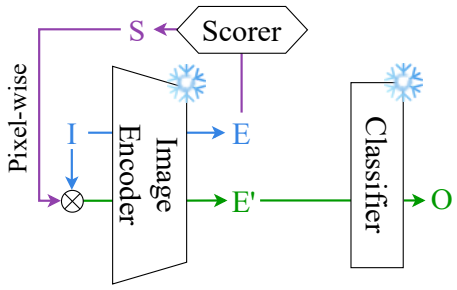


Figure 3.5: Training architecture for image classifiers

first which would make masking an embedding not the same as masking an input token.

3.3.2 Image Modality

For image classification, we use the traditional approach consisting of an image encoder with a classification head [37]. The flow of the classification model is therefore the following: (a) embed the input image with the encoder, (b) pool the resulting embeddings and pass through the classifier. The embeddings at step (a) can be used as the input of the scorer that produces a score for each input pixel. At training time, as depicted in Figure 3.5, we apply these scores to weigh each

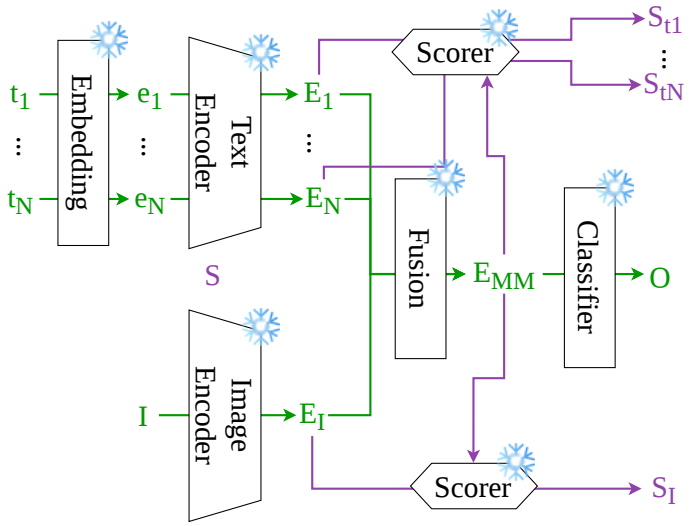


Figure 3.6: Inference for text-image classifiers

pixel of the input image before passing them through the encoder. Differently from text, in this case we do not need to add any additional considerations as the input image is already differentiable and the whole process can be directly integrated into the training process.

3.3.3 Multimodality

When moving to multimodal models, we can naturally extend our idea by combining the two scorers for text and image as we described in Sections 3.3.1 and 3.3.2. We depict this idea in Figure 3.6 for the case of text-image models. Additionally, in the case of multimodal models, the output of the two encoders are merged through a fusion layer [19], which stays frozen as the rest of the classifier. To avoid losing cross-modality information, the scorers in the multimodal case also take, alongside the embeddings of their modality, the output produced by the fusion layer as conditioning. The loss function is also modified to account for the two scorers: firstly, by extending Equation (3.4) to the multimodal case,

we evaluate the cross-entropy loss with both text and image masked so that the scorers are encouraged to preserve the original model. Secondly, to make the task easier and guide the learning process, we also minimize the cross-entropy of the two modalities separately by masking only either the text or the image. Thirdly, as there are two scorers, we introduce two sets of penalties to also handle the attribution maps produced by the two modalities separately. Therefore, after training the target multimodal classifier, we keep both encoders, the fusion layer, and the classification head frozen. Then, we train the two scorers by optimizing the following function:

$$\begin{aligned}\mathcal{L}(\mathbf{x}_1|\mathbf{x}_2, \mathbf{S}_1|\mathbf{S}_2; f) = \mathcal{L}_{\text{task}}(\mathbf{x}_1 \otimes \mathbf{S}_1|\mathbf{x}_2 \otimes \mathbf{S}_2; f) \\ + \mathcal{L}_{\text{task}}(\mathbf{x}_1 \otimes \mathbf{S}_1|\mathbf{x}_2; f) + \mathcal{L}_{\text{task}}(\mathbf{x}_1|\mathbf{x}_2 \otimes \mathbf{S}_2; f) \\ + \gamma \left(\frac{1}{|\mathbf{S}_1|} \sum_{i}^{|S_1|} s_{1,i} \right)^2 + \gamma \left(\frac{1}{|\mathbf{S}_2|} \sum_{j}^{|S_2|} s_{2,j} \right)^2 \\ + \gamma \frac{1}{|\mathbf{S}_1|} \sum_{i}^{|S_1|} (s_{1,i} \cdot (1 - s_{1,i}))^2 + \gamma \frac{1}{|\mathbf{S}_2|} \sum_{j}^{|S_2|} (s_{2,j} \cdot (1 - s_{2,j}))^2 \\ \text{with } \gamma = \frac{1}{\texttt{clip}\{\mathcal{L}_{\text{task}}(\mathbf{x}_1 \otimes \mathbf{S}_1|\mathbf{x}_2 \otimes \mathbf{S}_2; f) - \mathcal{L}_{\text{task}}(\mathbf{x}_1|\mathbf{x}_2; f), [0, 1]\}},\end{aligned} \tag{3.5}$$

where:

- $\mathbf{x}_1$ and $\mathbf{x}_2$ are the inputs of the two modalities,
- $\mathbf{S}_1$ and $\mathbf{S}_2$ are the attribution scores of the two modalities,
- γ is the self-calibrating parameter based on the logits obtained by masking both modalities.

4 Experimental Setup

4.1 Metrics

It is well-known that evaluating an explainability method is hard and currently there is no consensus on how it should be done [25]. However, it is unfeasible and unreasonable to manually evaluate attribution maps as it is time-consuming and it might not follow human intuition, while still correctly explaining the model. Therefore, on a quantitative side, we adopt a diverse set of commonly used metrics [10] with the goal of covering different aspects and desiderata of an explanation, while recognizing that no metric alone provides a reasonable way to assess an explainability method. More specifically, to assess whether the explanation is faithful to the model, we adopt average drop [6] and deletion AUC [35]. Conversely, to assess how the explanation is appealing, we use complexity [38] and sparsity [11]. In the rest of this section, we present the chosen metrics.

Average drop. Average drop [6] measures how much the confidence of a model decreases when only the relevant portion of the input is provided. First, the attribution map is used as a multiplicative scale to weight the input and produce its masked version. Then, a comparison is performed between the prediction confidence of the model using the full input and the masked one. More formally, let $f(\cdot)$ denote the model's confidence score for the predicted class on a given input, x

be the original input, and x_{masked} be the masked input. Average drop is computed as:

$$\texttt{average drop} = \max\Big\{0, f(x) - f(x_{\text{masked}})\Big\}. \tag{4.1}$$

Lower values indicate that the explanation successfully captures the most relevant regions for the prediction, as the confidence remains high even after masking. Conversely, a high average drop indicates that important information has been removed, which means that the explanation is less faithful.

Deletion AUC. Deletion Area Under the Curve (deletion AUC) is an extension of average drop and evaluates how quickly a model's confidence decreases as the most relevant input features are progressively removed [35]. Starting from the original input, features are iteratively removed in descending order of attribution importance, and the model's confidence score is recorded after each removal step. Formally, given the attribution map, the confidence score $f(x_k)$ is computed after removing the top k most relevant features according to the attribution scores. By plotting the confidence scores as a function of the importance levels, we obtain a curve and the metric is computed as the area under it.

A lower deletion AUC indicates that removing highly attributed regions rapidly degrades the model's confidence, suggesting that the explanation correctly identifies regions critical to the model's decision. On the other hand, higher deletion AUC indicates that the attribution map is not correctly ranking the most relevant features as they are not associated to the highest values.

Complexity. Complexity evaluates how much of the input space is relevant according to the attribution map [38]. This metric provides an idea of how big the relevant region in the attribution map is and can be useful to determine how easily interpretable the map is. In practice, we compute complexity of an attribution

map S as its norm to capture its magnitude and therefore how much of the input space is relevant:

$$\texttt{complexity} = \|S\|. \tag{4.2}$$

Lower complexity values correspond to more compact explanations, which, intuitively, are generally preferred from a human perspective.

Sparsity. Sparsity measures the extent to which an attribution map is concise and concentrates relevance on a small subset of the input features [11]. Formally, given an attribution score S, sparsity is computed as:

$$\texttt{sparsity} = \frac{S_{\text{max}}}{S_{\text{mean}}}, \tag{4.3}$$

where S_{max} is the maximum of the attribution map and S_{mean} is its mean. It is worth noting that, as we normalize attribution maps into $[0, 1]$, the numerator of the formula is always 1.

Higher sparsity indicates that the explanation is more concentrated in specific regions, reducing visual clutter and improving interpretability. On the contrary, lower sparsity indicates that many regions of the attribution map are active which might make it more difficult to understand.

4.2 Baselines

We compare the proposed approach against several other attribution methods. For our benchmarks, we choose well-known approaches for neural networks that work with any type of data format so that we can apply them across different modalities. In particular, we start from simple methods such as Saliency and Guided Backpropagation, and move to more complex ones such as Integrated Gradients, DeepLIFT, and SHAP-based approaches. The rest of this section presents our baselines of choice.

Saliency. Saliency maps [43] are a simple gradient-based explainability baseline for neural networks. The method computes the gradient of the model output with respect to the input features and interprets the magnitude of each partial derivative as an indicator of local sensitivity. Features with larger absolute gradient values are considered more influential as perturbing them would affect the output the most. However, although simple and straightforward, they are often noisy and can suffer from gradient saturation effects, particularly when applied to deep neural networks.

Guided Backpropagation. Guided Backpropagation [44] is a heuristic approach that modifies the standard backpropagation procedure by suppressing negative gradients during the backward pass so that only gradients corresponding to positive forward and backward signals are propagated. This allows to produce sharper and visually clearer saliency maps. However, this approach lacks a rigorous theoretical foundation. It cannot be proved that it satisfies any desirable property and it may not faithfully reflect the underlying process of the model.

Integrated Gradients. Integrated Gradients (IG) [46] is a method based on integrating the gradient along a continuous path from a baseline input x' to the actual input x. Formally, the attribution for a feature i is defined as:

$$\texttt{IG}_i(x) = (x_i - x'_i) \int_0^1 \frac{\partial f\big(x' + \alpha(x - x')\big)}{\partial x_i}\, d\alpha. \tag{4.4}$$

Formulated in this way, Integrated Gradients has some theoretical guarantees and satisfies the axioms of sensitivity (i.e., the attribution should be different given two inputs differing by one feature but with different labels) and implementation invariance (i.e., the attribution should be identical for functionally equivalent models). In practice, the integral is approximated numerically using a finite Riemann sum, which introduces a trade-off between computational cost and approximation

accuracy. Overall, Integrated Gradients is generally more stable and less prone to saturation, although its results depend on the choice of baselines.

DeepLIFT. Deep Learning Important FeaTures (DeepLIFT) [42] explains predictions by comparing neuron activations to those obtained from a reference input. It propagates activation differences backward through the network using specific contribution rules. For each neuron, the deviation from its reference activation is decomposed and attributed to its inputs. In practice, DeepLIFT mitigates issues related to saturation and flat activation regions. Moreover, it satisfies a summation-to-delta property, ensuring that the total attributions equal the difference between the model output at the input and at the reference. However, similar to Integrated Gradients, the method depends on the selection of an appropriate reference input, which can influence the resulting explanations.

DeepLIFT-SHAP. DeepLIFT-SHAP [30] uses DeepLIFT as a way to approximate SHAP (SHapley Additive exPlanations) values [30, 28]. SHAP is grounded in the concept of Shapley values from cooperative game theory [41]. It sees each feature as a player in a cooperative game, and its contribution to the prediction corresponds to its average marginal contribution across all possible feature subsets. Due to its computational complexity, Shapley values have to be approximated. DeepLIFT-SHAP computes them efficiently by applying DeepLIFT across multiple reference baselines and averaging the resulting attributions. This preserves the additive structure of SHAP while remaining computationally feasible for deep neural networks.

Gradient-SHAP. Gradient-SHAP [29] is another approach for approximating SHAP values by applying integrated gradients through stochastic sampling of baselines. Rather than integrating along a single deterministic path from a given

Table 4.1: Datasets used for evaluation

Modality	**Name**	**Task**
Text	Tweet Sentiment Extraction	Sentiment analysis, short texts.
	IMDB	Sentiment analysis, long texts.
	LIAR	Fact checking.
Image	MNIST	Image classification, grayscale low-resolution.
	CIFAR-10	Image classification, colored low-resolution.
	Imagenette	Image classification, colored high-resolution.
Multimodal	Flickr8k	Image-caption alignment.
	Hateful Memes	Hateful content detection.
	SNLI-VE	Visual entailment.

baseline, the method samples random interpolation points between reference samples and the input. By averaging gradients computed along these sampled paths, Gradient-SHAP is able to approximate Shapley value attributions. Moreover, this stochastic formulation improves robustness to local irregularities in the gradient landscape. However, computational complexity increases with the number of samples required to obtain stable estimates.

4.3 Datasets

To comprehensively evaluate the proposed attribution methods, we use a diverse collection of datasets spanning text, image, and multimodal tasks with varying degrees of complexity. In particular, the datasets range from short to long texts, from low-resolution to high-quality images, and from simple discriminative tasks to more complex reasoning ones. In this section, we present all the datasets we use for evaluation and, in Table 4.1, we provide a summary of these datasets.

4.3.1 Text Classification

For the textual modality, we consider Tweet Sentiment Extraction [32, 34], IMDB [31], and LIAR [49] so that we cover different domains, linguistic styles, and document lengths.

Tweet Sentiment Extraction. Tweet Sentiment Extraction [32, 34] consists of short, informal social media posts annotated with binary sentiment labels. This dataset was selected as social media posts are typically concise and contain non-standard language such as abbreviations, hashtags, emojis, and typos. This high lexical variability and limited context make this dataset particularly suitable for studying fine-grained token-level attributions, as relevant sentiment cues may be sparse and embedded within noisy text.

IMDB. The IMDB movie reviews dataset [31] is a large-scale binary sentiment classification benchmark composed of long-form movie reviews. In contrast to short tweets, reviews often contain multiple arguments, narrative elements, and mixed sentiment expressions. This dataset allows evaluating attribution methods under long-context conditions, where relevant evidence may be distributed across the document.

LIAR. The LIAR benchmark [49] is composed of political statements extracted from PolitiFact, a fact checking website, where each document is associated with a binary label according to their factual correctness. This dataset provides a more reasoning-intensive setting compared to IMDB while maintaining a setup with longer documents to make the task more challenging and more interesting from an explainability perspective.

4.3.2 Image Classification

To evaluate attribution in the visual domain, we use MNIST [21], CIFAR-10 [18], and Imagenette [13], three standard computer vision benchmarks with increasing visual and semantic complexity.

MNIST. MNIST [21] is a dataset consisting of grayscale images of size 28×28 containing handwritten digits. It is a dataset with a simple visual structure and limited background variation, which makes it a good baseline to check the effectiveness of explainability methods in the easiest scenario.

CIFAR-10. CIFAR-10 [18] contains 32×32 color images distributed across ten object categories. Compared to MNIST, it introduces more intra-class variability, background noise, and color information, which increases the level of difficulty. Therefore, attribution maps in this case should be slightly more challenging to determine due to the presence of more noise.

Imagenette. Imagenette [13] is a subset of ImageNet [7] composed of ten easily distinguishable classes. We choose this version due to computational limitations. The images in the dataset are higher in resolution and exhibit richer semantic structure than CIFAR-10, making the task more complex and closer to a real-life application.

4.3.3 Multimodal Classification

To extend our evaluation beyond unimodal settings, we also consider multimodal benchmarks. In particular, we evaluate on Flickr8k [12], Hateful memes [16], and SNLI-VE [51].

Flickr8k. Flickr8k [12] consists of images paired with natural language captions. We formulate a binary classification task in which the model must determine whether a given caption correctly describes the image. This dataset acts as a starting baseline as it provides a simple task of matching text and image, without involving complex semantic connections.

Hateful Memes. The Hateful Memes dataset [16] contains the image and the textual content of memes in which hateful intent often emerges only from the interaction between the two modalities. This makes the task more challenging than Flickr8k as it does not simply require to align the two representations and involves some semantic understanding. This also makes generating attribution maps more interesting to understand how the model is connecting different concepts.

SNLI-VE. SNLI-VE [51] is composed of image-sentence pairs from Stanford Natural Language Inference [4] and Flickr30k [52] and it is designed for the task of visual entailment. An image and a short text act as the premise while another short text provides a hypothesis. The task consists of predicting whether there is entailment, contradiction, or neutrality between premise and hypothesis. This dataset requires fine-grained semantic alignment between visual and linguistic content which introduces a further level of difficulty compared to Hateful Memes.

4.4 Architectures

Text. For text classification, we use two distinct transformer encoders [47] for the classifier and the scorer module. For the classification task itself, we use RoBERTa [24] as the pretrained encoder. For the scorer module, we instead use a custom small transformer initialized from scratch. It takes as input the embeddings of each token produced by RoBERTa and outputs the attribution scores. During

training, these scores are applied to the static embeddings of the input tokens, before passing through the transformer layers of RoBERTa.

Image. For image classification, we use a vision transformer [8] for the classification task and a U-Net [39] like network for the scorer module. The U-Net starts from the final embeddings produced by the vision transformer and upsamples it using transposed convolutions. As in the original U-Net, intermediate layers from the vision transformer are also included during upsampling as skip connections to provide more information. During training, the attribution map produced by the scorer is applied to the input image before feeding it through the backbone.

Text-image. For multimodal classification, the model is a combination of the text and image classifiers. We therefore use both RoBERTa and vision transformer, and the outputs of the two encoders are put together through a fusion layer, which is implemented as a small transformer encoder. For the scorers, both modules work as previously described with the addition of a further step where the embeddings produced by the fusion model are incorporated into the flow to account for cross-modal information.

4.5 Implementation Details

We use PyTorch [36] to implement our method and to train the neural networks for each task. Pretrained transformers and vision transformers are provided by Hugging Face. All datasets are taken from Hugging Face [50] as well, with the exception of MNIST and CIFAR which are provided by TorchVision [33]. For the baseline explainability methods, we use the implementation provided by Captum [17].

During training, we split the datasets into training, validation, and test sets.

We use the default splits provided by the dataset when available, and manually split when the dataset is provided as a whole. When training the models, we first fine-tune the classifier while keeping the scorer module frozen. Then, we train the scorer by keeping the classifier frozen. We use as optimizer AdamW [27] in all cases with a learning rate of 10^{-5} for fine-tuning and $2 \cdot 10^{-4}$ for training the scorer. All experiments have been executed on NVIDIA A100 GPUs with 80GB of memory provided by the National Institute of Informatics in Japan. The base classifiers are trained for 10 epochs maximum, while the scorers for 5, as we observed that they tend to stop very early due to early stopping. It took us approximately two hours to train each base classifier and its scorer on a single NVIDIA A100, and an additional hour to evaluate across all baselines.

5 Results

5.1 Text Classification

Quantitative results for Tweet Sentiment Extraction, IMDB, and LIAR are presented in Table 5.1, Table 5.2, and Table 5.3, respectively. Radar plots of the metrics is also reported in Figure 5.1. In all three datasets, our method yields the lowest complexity score and consistently achieves the second-best sparsity score after Saliency. Instead, Guided Backpropagation is the method that performs best in terms of average drop while DeepLIFT has better results in terms of deletion AUC. Also, as an overall behavior across the three datasets, we can observe that Integrated Gradients, DeepLIFT, DeepLIFT-SHAP, Gradient-SHAP, and Guided Backpropagation all result in high complexity and low sparsity. Saliency instead is the worst performing in terms of average drop and deletion AUC, but achieves the best sparsity and reasonable complexity. This highlights a clear trade-off that these methods exhibit between producing attribution maps that are visually appealing and more faithful to the model prediction.

Qualitatively, we manually evaluate some samples from IMDB. We choose this dataset as it has longer documents compared to Tweet Sentiment Extraction and its classifier achieves near perfect results, which should make it easier to intuitively evaluate the attribution maps. We selected 100 random samples from the dataset and evaluated them. The main pattern we observed is that our method,

Table 5.1: Results for Tweet Sentiment Extraction

Method	**Average drop ↓**	**Deletion AUC ↓**	**Complexity ↓**	**Sparsity ↑**
Saliency	1.91 ± 1.76	2.49 ± 1.78	13.11 ± 3.02	**412.95 ± 158.54**
Guided Backprop.	**0.09 ± 0.19**	**1.38 ± 0.98**	312.17 ± 33.18	2.03 ± 0.23
Int. Gradients	0.20 ± 0.57	1.68 ± 1.43	279.29 ± 107.56	2.80 ± 1.90
DeepLIFT	0.42 ± 0.86	1.47 ± 1.25	311.62 ± 101.81	2.31 ± 1.04
DeepLIFT-SHAP	0.44 ± 0.90	1.43 ± 1.19	311.00 ± 87.70	2.22 ± 0.83
Gradient-SHAP	0.34 ± 0.82	1.43 ± 1.22	298.72 ± 99.06	2.40 ± 1.00
Ours	1.11 ± 1.52	1.61 ± 1.57	**2.71 ± 1.04**	87.21 ± 72.40

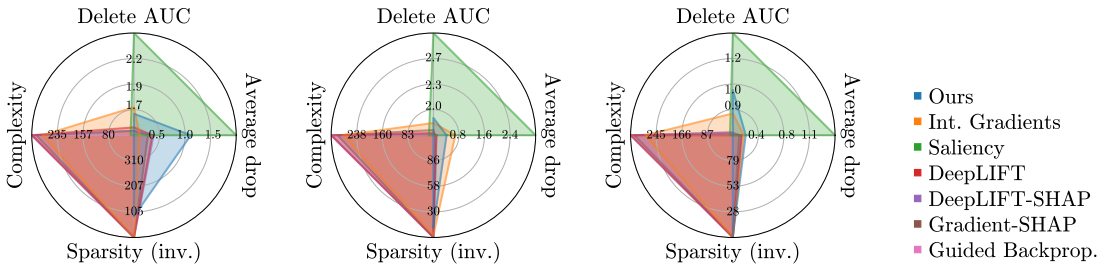
Table 5.2: Results for IMDB

Method	**Average drop ↓**	**Deletion AUC ↓**	**Complexity ↓**	**Sparsity ↑**
Saliency	3.19 ± 1.04	3.06 ± 0.99	18.57 ± 5.97	**114.05 ± 91.08**
Guided Backprop.	**0.05 ± 0.14**	1.62 ± 0.56	313.09 ± 41.16	2.04 ± 0.29
Int. Gradients	0.72 ± 1.39	1.76 ± 1.06	279.72 ± 171.69	4.37 ± 4.96
DeepLIFT	0.15 ± 0.63	1.61 ± 0.81	315.50 ± 108.50	2.35 ± 1.23
DeepLIFT-SHAP	0.14 ± 0.63	**1.58 ± 0.78**	315.62 ± 103.78	2.32 ± 1.20
Gradient-SHAP	0.10 ± 0.50	1.66 ± 0.81	296.16 ± 111.63	2.52 ± 1.24
Ours	0.46 ± 0.63	1.83 ± 0.76	**4.90 ± 2.22**	12.64 ± 12.29

Table 5.3: Results for LIAR

Method	**Average drop ↓**	**Deletion AUC ↓**	**Complexity ↓**	**Sparsity ↑**
Saliency	1.44 ± 1.20	1.36 ± 1.14	16.59 ± 5.93	**104.42 ± 85.75**
Guided Backprop.	**0.09 ± 0.16**	0.74 ± 0.61	316.11 ± 46.54	2.03 ± 0.32
Int. Gradients	0.23 ± 0.61	0.85 ± 0.81	278.28 ± 109.90	2.75 ± 1.50
DeepLIFT	0.21 ± 0.52	**0.72 ± 0.68**	323.70 ± 123.19	2.36 ± 1.26
DeepLIFT-SHAP	0.20 ± 0.54	0.73 ± 0.68	323.81 ± 116.52	2.31 ± 1.23
Gradient-SHAP	0.17 ± 0.49	0.73 ± 0.68	312.24 ± 119.44	2.52 ± 1.73
Ours	0.26 ± 0.29	1.00 ± 0.80	**7.71 ± 0.34**	3.08 ± 1.63

compared to the other baselines, has a more decisive scoring system as it favors either very high or very low scores, while the other methods tend to give some



(a) Tweet Sentiment (b) IMDB (c) LIAR

Figure 5.1: Radar plot of the quantitative metrics for the text tasks. The closer to the center, the better.

Ours	This	is	a	great	movie.	Too	bad	it	is	not	available	on	home	video.</s>
Saliency	This	is	a	great	movie.	Too	bad	it	is	not	available	on	home	video.</s>
Guided backprop	This	is	a	great	movie.	Too	bad	it	is	not	available	on	home	video.</s>
Int. Gradients	This	is	a	great	movie.	Too	bad	it	is	not	available	on	home	video.</s>
DeepLIFT	This	is	a	great	movie.	Too	bad	it	is	not	available	on	home	video.</s>
DeepLIFT-SHAP	This	is	a	great	movie.	Too	bad	it	is	not	available	on	home	video.</s>
Gradient-SHAP	This	is	a	great	movie.	Too	bad	it	is	not	available	on	home	video.</s>

Figure 5.2: Sample from IMDB

degree of importance to each token. Also, we observed that the attribution scores correctly highlight the most relevant terms to decide the sentiment, although it frequently also highlights surrounding terms which are not necessarily relevant according to human intuition. We report in Figure 5.2 a sample extracted from the dataset. Moreover, in line with the metrics, Integrated Gradients, DeepLIFT, DeepLIFT-SHAP, Gradient-SHAP, and Guided Backpropagation are the methods that produce attribution maps that span across the whole input the most, while Saliency is the method where the scores are more concentrated on a few words.

Putting all these observations together, we can conclude that our method, being the one with the least complexity metric and with generally higher sparsity, produces attribution maps that are closer to binary masks, indicating that it gives

Table 5.4: Results on MNIST

Method	**Average drop ↓**	**Deletion AUC ↓**	**Complexity ↓**	**Sparsity ↑**
Saliency	0.63 ± 0.31	0.81 ± 0.14	13.39 ± 4.57	**42.85 ± 36.35**
Guided Backprop.	0.07 ± 0.19	0.45 ± 0.10	112.17 ± 13.22	2.03 ± 0.26
Int. Gradients	0.02 ± 0.09	0.57 ± 0.15	**4.86 ± 1.43**	3.64 ± 1.92
DeepLIFT	0.03 ± 0.13	0.45 ± 0.10	111.46 ± 13.58	2.05 ± 0.27
DeepLIFT-SHAP	**0.01 ± 0.10**	**0.44 ± 0.11**	111.85 ± 16.12	2.05 ± 0.32
Gradient-SHAP	0.01 ± 0.09	**0.44 ± 0.09**	110.89 ± 14.69	2.06 ± 0.29
Ours	**0.01 ± 0.08**	0.53 ± 0.26	33.65 ± 15.36	22.20 ± 10.97

a clear distinction between what the classifier is considering relevant and what it is not, making its attribution maps generally more appealing. Moreover, by achieving decent average drop and deletion AUC, it shows that it is able to highlight relevant features of the input space, which indicates that it does not lose in faithfulness.

5.2 Image Classification

For the image classification tasks, we report the result for MNIST, CIFAR-10, and Imagenette on Table 5.4, Table 5.5, and Table 5.6, respectively, while the radar plots are depicted in Figure 5.3. For these datasets, our method consistently achieves the highest average drop and deletion AUC. In terms of complexity, Integrated Gradients is the method performing the best, while, for sparsity, Saliency has the highest results. Instead, as for the case of text, DeepLIFT, DeepLIFT-SHAP, Gradient-SHAP, and Guided Backpropagation all produce highly complex and weakly compact maps, but achieve high results in terms of average drop and deletion AUC. This highlights again a trade-off between faithfulness and appeal that is present among these methods, and ours is again located as a balanced approach for these two requirements.

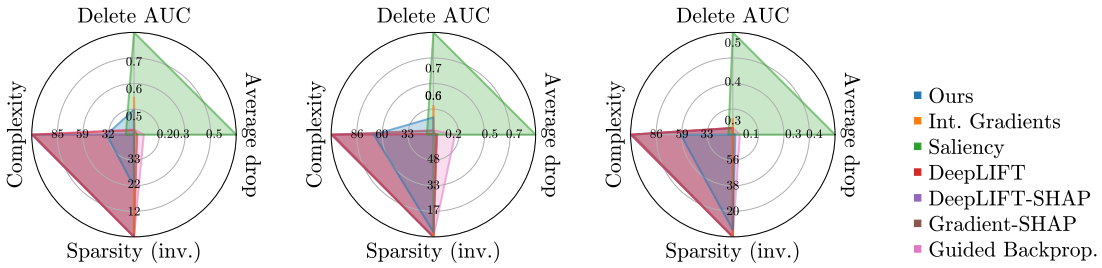
Qualitatively, we evaluate 100 random samples from Imagenette. In the case

Table 5.5: Results on CIFAR-10

Method	Average drop ↓	Deletion AUC ↓	Complexity ↓	Sparsity ↑
Saliency	0.88 ± 0.09	0.83 ± 0.05	12.49 ± 7.20	**63.91 ± 58.92**
Guided Backprop.	0.21 ± 0.37	0.48 ± 0.08	111.92 ± 12.76	2.03 ± 0.24
Int. Gradients	**0.03 ± 0.16**	0.57 ± 0.13	**5.75 ± 1.57**	2.84 ± 1.01
DeepLIFT	0.07 ± 0.22	**0.46 ± 0.09**	112.75 ± 16.48	2.04 ± 0.32
DeepLIFT-SHAP	0.06 ± 0.21	**0.46 ± 0.09**	113.32 ± 16.49	2.02 ± 0.31
Gradient-SHAP	0.05 ± 0.20	0.48 ± 0.08	111.75 ± 13.80	2.04 ± 0.26
Ours	**0.03 ± 0.15**	0.52 ± 0.18	68.44 ± 18.32	5.02 ± 2.01

Table 5.6: Results on Imagenette

Method	Average drop ↓	Deletion AUC ↓	Complexity ↓	Sparsity ↑
Saliency	0.50 ± 0.17	0.53 ± 0.09	10.46 ± 5.07	**73.55 ± 60.17**
Guided Backprop.	0.05 ± 0.11	0.28 ± 0.06	112.95 ± 12.52	2.01 ± 0.23
Int. Gradients	**0.01 ± 0.06**	0.31 ± 0.11	**5.79 ± 1.76**	2.87 ± 1.20
DeepLIFT	**0.01 ± 0.07**	0.28 ± 0.06	112.46 ± 15.83	2.04 ± 0.32
DeepLIFT-SHAP	**0.01 ± 0.07**	0.28 ± 0.06	112.46 ± 16.02	2.04 ± 0.31
Gradient-SHAP	**0.01 ± 0.06**	0.28 ± 0.06	110.71 ± 14.36	2.06 ± 0.29
Ours	0.02 ± 0.07	**0.27 ± 0.12**	60.43 ± 19.45	7.02 ± 2.92



(a) MNIST (b) CIFAR-10 (c) Imagenette

Figure 5.3: Radar plot of the quantitative metrics for the image tasks. The closer to the center, the better.

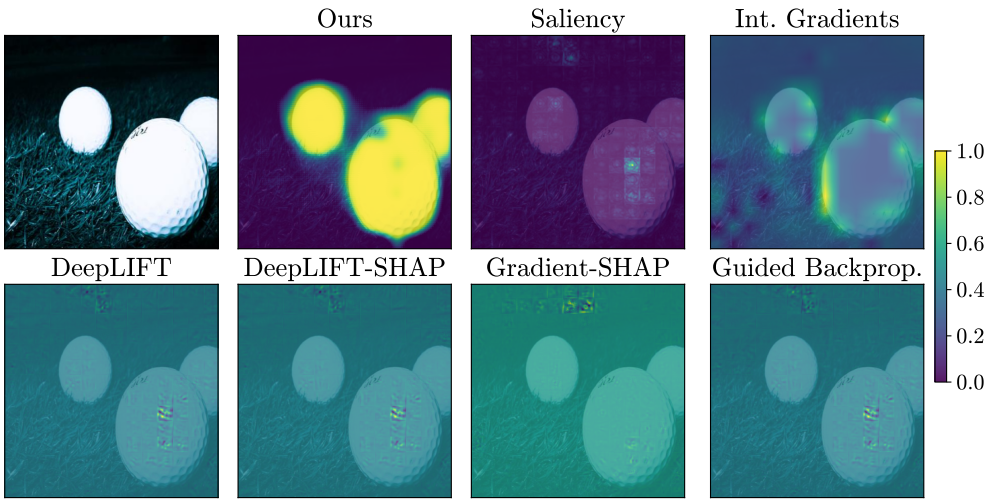


Figure 5.4: Sample from Imagenette

of images, all classifiers have strong performance, therefore, we choose to analyze Imagenette as it contains images of higher resolution. Similarly to the results we obtained for text, we can observe that compared to the other baselines, except Saliency, the attribution map produced with our method is closer to a binary mask with values closer to either 0 or 1. Also, compared to gradient-based methods, in the case of images, we can observe that almost all, but Integrated Gradients, produce results that are very focused on a small subset of pixels, which makes results less visually appealing, although performing well in terms of average drop and deletion AUC. Finally, semantically, we observed that almost all attribution maps correctly identify the object of interest in the image by either highlighting it wholly or by identifying key characteristics. We report a sample from the dataset in Figure 5.4.

Overall, results with images highlight that, as for text, the attribution maps produced by our method have a clear separation between what is relevant and what is not while also achieving competitive values of average drop and deletion AUC. This again highlights a trade-off between attribution map faithfulness and

Table 5.7: Results on Flickr8k

Method	**Average drop ↓**	**Deletion AUC ↓**	**Complexity ↓**	**Sparsity ↑**
Saliency	1.86 ± 1.70	1.79 ± 1.07	**15.20 ± 3.50**	**79.10 ± 36.06**
Guided Backprop.	**0.71 ± 0.98**	**1.11 ± 0.66**	174.10 ± 15.05	2.05 ± 0.18
Int. Gradients	0.77 ± 1.39	1.88 ± 1.15	66.06 ± 30.85	2.82 ± 1.24
DeepLIFT	0.95 ± 1.40	1.32 ± 0.84	180.31 ± 28.94	2.07 ± 0.41
DeepLIFT-SHAP	0.93 ± 1.39	1.20 ± 0.75	174.90 ± 21.89	2.08 ± 0.28
Gradient-SHAP	1.04 ± 1.57	1.24 ± 0.80	177.29 ± 24.94	2.07 ± 0.36
Ours	1.21 ± 1.22	2.04 ± 1.21	38.47 ± 8.74	10.54 ± 2.27

visual plausibility, and our method is located in the middle ground between the two requirements.

5.3 Multimodal Classification

Results on multimodal datasets are reported in Table 5.7, Table 5.8, and Table 5.9 for Flickr8k, Hateful Memes, and SNLI-VE, respectively, and the radar plots are in Figure 5.5. Consistent with the previous results, Saliency is the method that achieves the best complexity and sparsity, while the other gradient-based methods perform best in terms of average drop and deletion AUC. Our method again performs with a trade-off between the two sides by achieving competitive average drop while maintaining a lower complexity and higher sparsity.

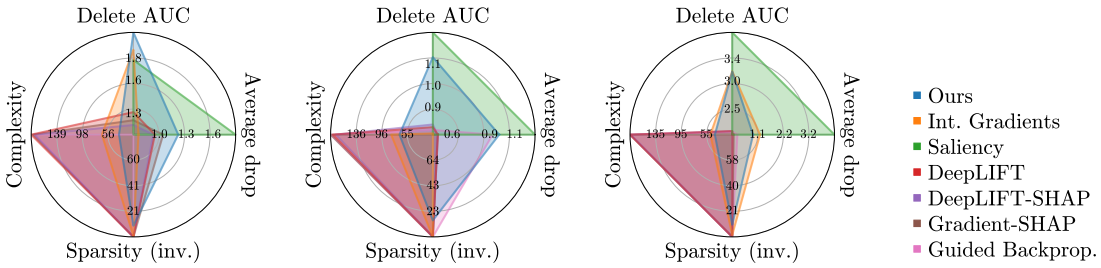
For a qualitative evaluation, we analyze 100 samples from Flickr8k. We observed a consistent behavior as for text and image modality alone. The attribution maps that our method produces tend to be more well separated in terms of score magnitude, allowing to more clearly distinguish what is relevant. Also, as it does not lose in terms of faithfulness, what the scores give high relevance tend to usually be the object of interest in both the text sequence and figure. We provide a sample from the dataset in Figure 5.6.

Table 5.8: Results on Hateful Memes

Method	Average drop ↓	Deletion AUC ↓	Complexity ↓	Sparsity ↑
Saliency	1.26 ± 1.34	1.25 ± 0.98	**14.80 ± 3.79**	**84.04 ± 36.24**
Guided Backprop.	0.92 ± 1.00	0.80 ± 0.53	175.90 ± 14.93	2.03 ± 0.18
Int. Gradients	**0.45 ± 0.98**	**0.78 ± 0.58**	80.76 ± 22.80	2.28 ± 0.81
DeepLIFT	0.49 ± 0.98	0.81 ± 0.56	172.86 ± 28.59	2.22 ± 0.68
DeepLIFT-SHAP	0.48 ± 0.95	0.82 ± 0.60	168.37 ± 27.26	2.27 ± 0.62
Gradient-SHAP	0.49 ± 0.94	**0.78 ± 0.55**	176.54 ± 29.93	2.15 ± 0.57
Ours	0.98 ± 0.99	1.13 ± 0.91	67.60 ± 19.82	15.24 ± 7.94

Table 5.9: Results on SNLI-VE

Method	Average drop ↓	Deletion AUC ↓	Complexity ↓	Sparsity ↑
Saliency	4.27 ± 1.83	3.89 ± 1.65	**14.41 ± 3.84**	**76.98 ± 35.92**
Guided Backprop.	0.27 ± 0.32	2.10 ± 0.86	173.94 ± 14.78	2.05 ± 0.17
Int. Gradients	1.20 ± 2.19	3.19 ± 1.73	48.72 ± 23.55	3.44 ± 1.72
DeepLIFT	**0.06 ± 0.21**	2.11 ± 1.05	174.88 ± 22.61	2.09 ± 0.32
DeepLIFT-SHAP	**0.06 ± 0.20**	2.11 ± 0.93	173.94 ± 20.18	2.08 ± 0.28
Gradient-SHAP	0.12 ± 0.46	**2.04 ± 0.93**	174.89 ± 26.29	2.15 ± 0.50
Ours	0.92 ± 0.83	3.18 ± 1.51	41.68 ± 9.64	11.07 ± 3.80



(a) Flickr8k (b) Hateful Memes (c) SNLI-VE

Figure 5.5: Radar plot of the quantitative metrics for the multimodal tasks. The closer to the center, the better.

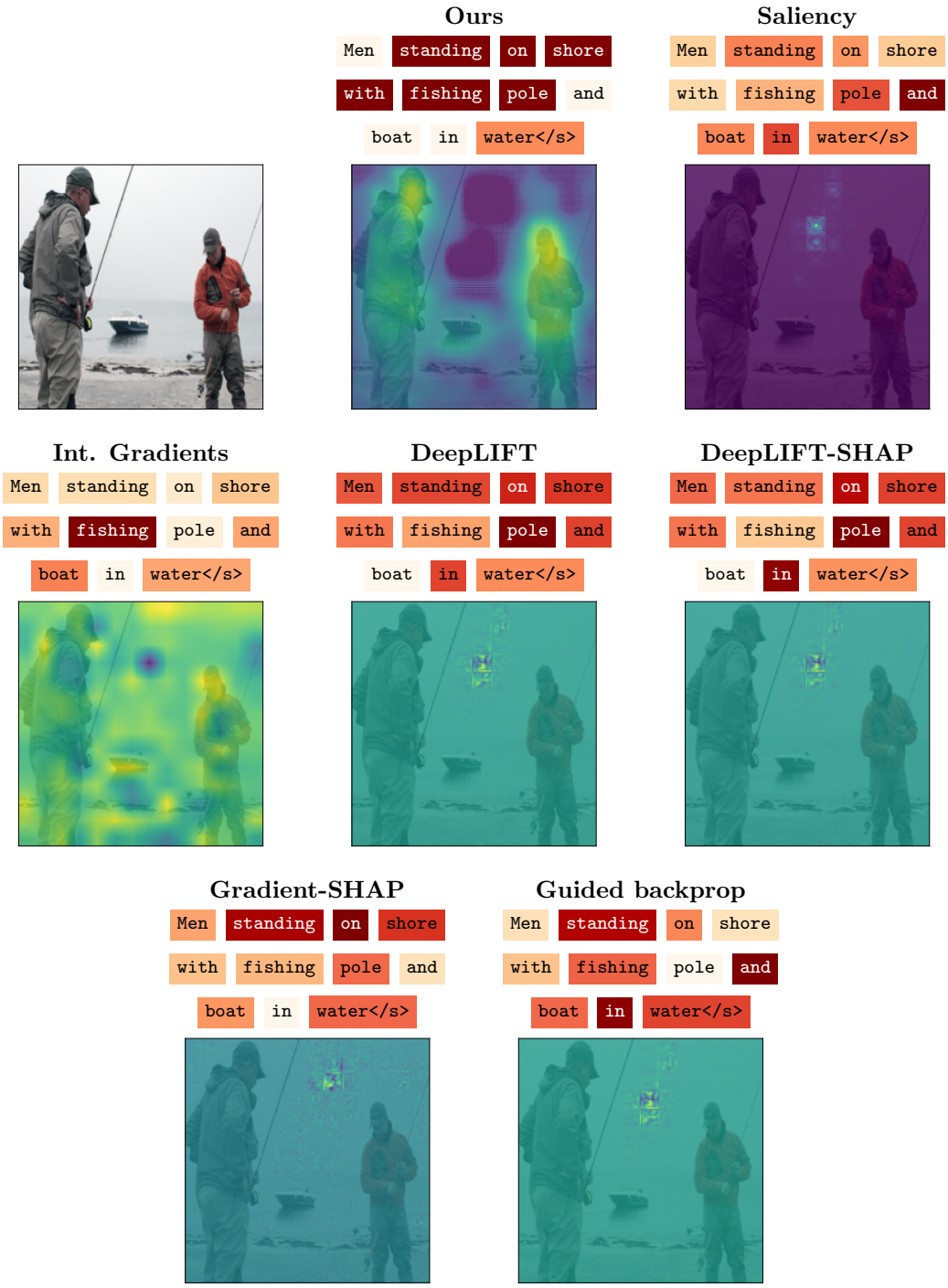


Figure 5.6: Sample from Flickr8k

Overall, also in the case of multimodal classification, our method performs consistently by producing attribution maps with relatively low complexity and high sparsity while maintaining reasonable average drop and deletion AUC, highlighting again a good trade-off between visually appealing and faithful attribution maps compared to the other baselines that tend to favor only one side.

5.4 Ablation Study

To empirically experiment with the effectiveness of our design choices, we analyze the impact of the key components of our method. For computational efficiency, we limit ablation study to Tweet Sentiment Extraction and Imagenette to cover both modalities. We aim at testing whether the architectural choices and the training routine are effective. Therefore, we experiment with: (a) removing the last ReLU activation, (b) removing the last sigmoid activation, (c) removing the last rescaling step, (d) removing the binary penalty in the loss, (e) removing the magnitude penalty in the loss, (f) removing the penalty in the loss completely, (g) remove every component we introduce and train the model with only cross-entropy.

Results on Tweet Sentiment Extraction are presented in Table 5.10 and a sample from the dataset is presented in Figure 5.7. In terms of architecture, we can observe that, by either removing the final ReLU, sigmoid, or rescaling, there is a worsening in sparsity and an improvement in average drop and deletion AUC. This indicates, also as we can observe in Figure 5.7, that the attribution maps becomes more expansive and considers more input tokens as relevant. Also in terms of loss function, we can observe that, after removing each component, results are decreased across all metrics. Finally, if everything is excluded, we can see that the method performs well in terms of average drop and deletion AUC,

Table 5.10: Ablation results on Tweet Sentiment Extraction

Method	Average drop ↓	Deletion AUC ↓	Complexity ↓	Sparsity ↑
Full	1.11 ± 1.52	1.61 ± 1.57	2.71 ± 1.04	87.21 ± 72.40
No ReLU	0.38 ± 0.58	0.66 ± 0.62	2.40 ± 0.68	81.95 ± 49.85
No sigmoid	1.28 ± 1.73	1.49 ± 1.63	3.04 ± 0.82	67.04 ± 53.80
No rescale	0.00 ± 0.01	0.17 ± 0.13	4.25 ± 0.96	32.32 ± 17.15
No binary pen.	1.26 ± 1.73	1.49 ± 1.69	3.16 ± 0.89	65.08 ± 53.13
No magnitude pen.	1.29 ± 1.73	1.42 ± 1.68	3.35 ± 1.00	64.04 ± 53.90
No penalty	1.27 ± 1.73	1.40 ± 1.68	3.37 ± 1.02	63.76 ± 54.10
Nothing	0.06 ± 0.15	0.97 ± 0.77	3.19 ± 0.85	43.03 ± 26.78

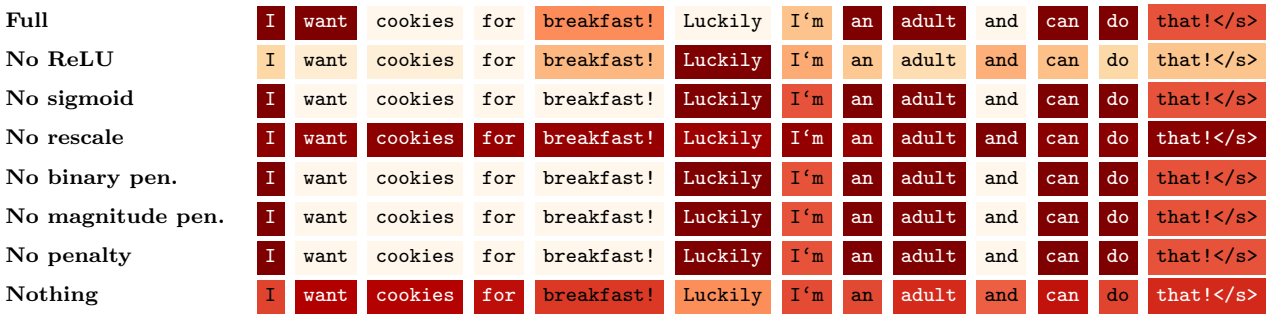


Figure 5.7: Ablation results on a sample from Tweet Sentiment Extraction

while it worsens in complexity and sparsity, indicating that the attribution map, without constraints, becomes trivially a non-informative result.

Ablation results on Imagenette are presented in Table 5.11 and a sample for qualitative analysis is shown in Figure 5.8. As for text, in all cases, but the scenario where rescaling is removed, performance on complexity and sparsity tend to worsen. In the case of rescaling, the behavior is opposite as average drop and deletion AUC worsen but complexity and sparsity improve. However, qualitatively we can see that the best result is obtained with every component as they allow to obtain a clearer distinction between what is relevant in the input.

Table 5.11: Ablation results on Imagenette

Method	**Average drop ↓**	**Deletion AUC ↓**	**Complexity ↓**	**Sparsity ↑**
Full	0.02 ± 0.07	0.27 ± 0.12	60.43 ± 19.45	7.02 ± 2.92
No ReLU	0.02 ± 0.06	0.30 ± 0.15	64.89 ± 18.82	4.74 ± 1.70
No sigmoid	0.02 ± 0.05	0.29 ± 0.12	60.31 ± 19.83	6.14 ± 2.82
No rescale	0.34 ± 0.26	0.45 ± 0.18	43.44 ± 46.93	22.03 ± 24.26
No binary pen.	0.02 ± 0.06	0.37 ± 0.09	48.68 ± 13.85	6.81 ± 3.45
No magnitude pen.	0.01 ± 0.03	0.04 ± 0.03	159.79 ± 25.34	1.92 ± 0.88
No penalty penalty	0.01 ± 0.04	0.08 ± 0.07	149.04 ± 24.68	1.98 ± 0.78
Nothing	0.01 ± 0.02	0.06 ± 0.07	161.76 ± 30.60	1.56 ± 0.59

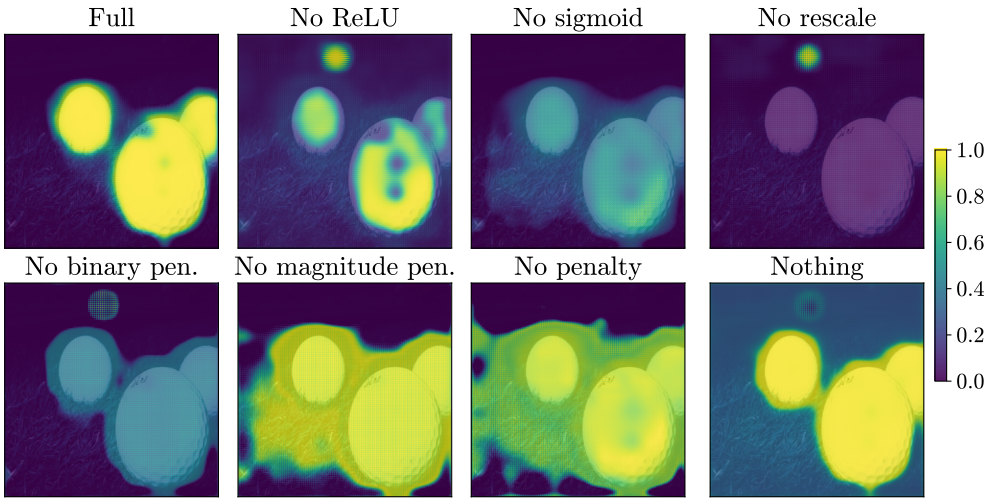


Figure 5.8: Ablation results on a sample from Imagenette

6 Conclusion

In this thesis, we presented a method for post-hoc explainability that produces attribution maps for the input space of neural networks. Our approach consists of training a dedicated neural network to produce attribution maps. With specific architectural and training choices, we developed a method that does not introduce new hyperparameters to tune other than those of a traditional neural network. After performing extensive quantitative and qualitative evaluation on classification datasets with text, image, and text-image modalities, we observed that our method produces attribution maps that are visually appealing while remaining faithful to the underlying model. Compared to other baselines, our method achieves the best trade-off between faithfulness, which we quantify as average drop and deletion AUC, and visual appeal, which we measure using complexity and sparsity metrics, alongside a qualitative analysis. Future work can explore several directions: (a) experiment with other modalities such as audio or time-series, (b) apply the same approach on regression or generative tasks, (c) explore other architectures for producing the attribution maps, (d) perform more extensive experiments with larger datasets and with different domains.

Bibliography

[1] Arrieta Alejandro Barredo et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Inf. Fusion* 58 (2020), pp. 82–115. DOI: 10.1016/J.INFFUS.2019.12.012.

[2] Barkan Oren et al. "LLM Explainability via Attributive Masking Learning". In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 9522–9537. DOI: 10.18653/v1/2024.findings-emnlp.556. URL: https://aclanthology.org/2024.findings-emnlp.556/.

[3] Bhattacharya Debarpan et al. *Gradient-free Post-hoc Explainability Using Distillation Aided Learnable Approach*. 2024. arXiv: 2409.11123 [cs.AI].

[4] Bowman Samuel R. et al. "A large annotated corpus for learning natural language inference". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 632–642. DOI: 10.18653/v1/D15-1075.

[5] Bygrave Lee A. "Article 22 Automated individual decision-making, including profiling". In: *The EU General Data Protection Regulation (GDPR): A*

Commentary. Oxford University Press, Feb. 2020. ISBN: 9780198826491. DOI: 10.1093/oso/9780198826491.003.0055.

[6] Chattopadhyay Aditya et al. "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks". In: *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*. IEEE Computer Society, 2018, pp. 839–847. DOI: 10.1109/WACV.2018.00097.

[7] Deng Jia et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[8] Dosovitskiy Alexey et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[9] European Commission and Directorate-General for Communications Networks, Content and Technology and Grupa ekspertów wysokiego szczebla ds. sztucznej inteligencji. *Ethics guidelines for trustworthy AI*. Publications Office, 2019. DOI: 10.2759/346720.

[10] Gevaert Arne et al. "Evaluating feature attribution methods in the image domain". In: *Machine Learning* 113.9 (2024), pp. 6019–6064.

[11] Gomez Tristan, Fréour Thomas, and Mouchère Harold. "Metrics for Saliency Map Evaluation of Deep Learning Explanation Methods". In: *Pattern Recognition and Artificial Intelligence - Third International Conference, ICPRAI 2022, Paris, France, June 1-3, 2022, Proceedings, Part I*. Ed. by Mounim A.

El-Yacoubi et al. Vol. 13363. Lecture Notes in Computer Science. Springer, 2022, pp. 84–95. DOI: 10.1007/978-3-031-09037-0_8.

[12] Hodosh Micah, Young Peter, and Hockenmaier Julia. "Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics". In: *J. Artif. Intell. Res.* 47 (2013), pp. 853–899. DOI: 10.1613/JAIR.3994. URL: https://doi.org/10.1613/jair.3994.

[13] Howard Jeremy. *Imagenette: A smaller subset of 10 easily classified classes from Imagenet*. Mar. 2019. URL: https://github.com/fastai/imagenette.

[14] Joshi Gargi, Walambe Rahee, and Kotecha Ketan. "A Review on Explainability in Multimodal Deep Neural Nets". In: *IEEE Access* 9 (2021), pp. 59800–59821. DOI: 10.1109/ACCESS.2021.3070212.

[15] Kanehira Atsushi and Harada Tatsuya. *Learning to Explain with Complemental Examples*. 2019. arXiv: 1812.01280 [cs.CV].

[16] Kiela Douwe et al. "The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/1b84c4cee2b8b3d823b30e2d604b1878-Abstract.html.

[17] Kokhlikyan Narine et al. *Captum: A unified and generic model interpretability library for PyTorch*. 2020. arXiv: 2009.07896 [cs.LG].

[18] Krizhevsky Alex, Hinton Geoffrey, et al. "Learning multiple layers of features from tiny images". In: (2009). URL: https://www.cs.toronto.edu/~kriz/cifar.html.

[19] Lahat Dana, Adali Tülay, and Jutten Christian. "Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects". In: *Proc. IEEE* 103.9 (2015), pp. 1449–1477. DOI: 10.1109/JPROC.2015.2460697.

[20] Langer Markus et al. "Explainability Auditing for Intelligent Systems: A Rationale for Multi-Disciplinary Perspectives". In: *29th IEEE International Requirements Engineering Conference Workshops, RE 2021 Workshops, Notre Dame, IN, USA, September 20-24, 2021*. IEEE, 2021, pp. 164–168. DOI: 10.1109/REW53955.2021.00030.

[21] LeCun Yann, Cortes Corinna, and Burges CJ. "MNIST handwritten digit database". In: *ATT Labs [Online]* 2 (2010). URL: http://yann.lecun.com/exdb/mnist.

[22] Li Qian et al. "A Survey on Text Classification: From Traditional to Deep Learning". In: *ACM Trans. Intell. Syst. Technol.* 13.2 (2022), 31:1–31:41. DOI: 10.1145/3495162.

[23] Liu Hui, Yin Qingyu, and Wang William Yang. *Towards Explainable NLP: A Generative Explanation Framework for Text Classification*. 2019. arXiv: 1811.00196 [cs.CL].

[24] Liu Yinhan et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692.

[25] Löfström Helena, Hammar Karl, and Johansson Ulf. "A Meta Survey of Quality Evaluation Criteria in Explanation Methods". In: *Intelligent Information Systems*. Springer International Publishing, 2022, pp. 55–63. ISBN: 9783031074813. DOI: 10.1007/978-3-031-07481-3_7.

[26] Longo Luca et al. "Explainable Artificial Intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions". In: *Inf. Fusion* 106 (2024), p. 102301. DOI: 10.1016/J.INFFUS.2024.102301.

[27] Loshchilov Ilya and Hutter Frank. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].

[28] Lundberg Scott M. and Lee Su-In. "A unified approach to interpreting model predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4768–4777. ISBN: 9781510860964.

[29] Lundberg Scott M. and Lee Su-In. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 2017, pp. 4765–4774.

[30] Lundberg Scott M. and Lee Su-In. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 4765–4774. URL: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

[31] Maas Andrew L. et al. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. URL: http://www.aclweb.org/anthology/P11-1015.

[32] Maggie Phil Culliton Wei Chen. *Tweet Sentiment Extraction*. 2020. URL: https://kaggle.com/competitions/tweet-sentiment-extraction.

[33] maintainers TorchVision and contributors. *TorchVision: PyTorch's Computer Vision library*. https://github.com/pytorch/vision. 2016.

[34] Muennighoff Niklas et al. “MTEB: Massive Text Embedding Benchmark”. In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*. Ed. by Andreas Vlachos and Isabelle Augenstein. Association for Computational Linguistics, 2023, pp. 2006–2029. DOI: 10.18653/V1/2023.EACL-MAIN.148. URL: https://doi.org/10.18653/v1/2023.eacl-main.148.

[35] Nieradzik Lars, Stephani Henrike, and Keuper Janis. “Reliable Evaluation of Attribution Maps in CNNs: A Perturbation-Based Approach”. In: *Int. J. Comput. Vis.* 133.5 (2025), pp. 2392–2409. DOI: 10.1007/S11263-024-02282-6.

[36] Paszke Adam et al. “PyTorch: an imperative style, high-performance deep learning library”. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[37] Plested Jo and Gedeon Tom. “Deep transfer learning for image classification: a survey”. In: *CoRR* abs/2205.09904 (2022). DOI: 10.48550/ARXIV.2205.09904. arXiv: 2205.09904.

[38] Poppi Samuele et al. “Revisiting the Evaluation of Class Activation Mapping for Explainability: A Novel Metric and Experimental Analysis”. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 2299–2304. DOI: 10.1109/CVPRW53098.2021.00260.

[39] Ronneberger Olaf, Fischer Philipp, and Brox Thomas. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th In-*

ternational Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III. Vol. 9351. Lecture Notes in Computer Science. Springer, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[40] Samek Wojciech and Müller Klaus-Robert. "Towards Explainable Artificial Intelligence". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer International Publishing, 2019, pp. 5–22. ISBN: 9783030289546. DOI: 10.1007/978-3-030-28954-6_1. URL: http://dx.doi.org/10.1007/978-3-030-28954-6_1.

[41] Shapley Lloyd S. "A value for n-person games". In: *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Cambridge University Press, 1988, pp. 31–40. DOI: 10.1017/CBO9780511528446.003.

[42] Shrikumar Avanti, Greenside Peyton, and Kundaje Anshul. "Learning Important Features Through Propagating Activation Differences". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 3145–3153. URL: http://proceedings.mlr.press/v70/shrikumar17a.html.

[43] Simonyan Karen, Vedaldi Andrea, and Zisserman Andrew. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: http://arxiv.org/abs/1312.6034.

[44] Springenberg Jost Tobias et al. *Striving for Simplicity: The All Convolutional Net*. 2015. arXiv: 1412.6806 [cs.LG].

[45] Sun Shilin et al. “A Review of Multimodal Explainable Artificial Intelligence: Past, Present and Future”. In: *CoRR* abs/2412.14056 (2024). arXiv: 2412.14056.

[46] Sundararajan Mukund, Taly Ankur, and Yan Qiqi. “Axiomatic Attribution for Deep Networks”. In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 3319–3328. URL: http://proceedings.mlr.press/v70/sundararajan17a.html.

[47] Vaswani Ashish et al. “Attention Is All You Need”. In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762.

[48] Vilone Giulia and Longo Luca. “Explainable Artificial Intelligence: a Systematic Review”. In: *CoRR* abs/2006.00093 (2020). arXiv: 2006.00093.

[49] Wang William Yang. “”Liar, Liar Pants on Fire”: A New Benchmark Dataset for Fake News Detection”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*. Ed. by Regina Barzilay and Min-Yen Kan. Association for Computational Linguistics, 2017, pp. 422–426. DOI: 10.18653/V1/P17-2067. URL: https://doi.org/10.18653/v1/P17-2067.

[50] Wolf Thomas et al. *HuggingFace’s Transformers: State-of-the-art Natural Language Processing*. 2020. arXiv: 1910.03771 [cs.CL].

[51] Xie Ning et al. “Visual Entailment: A Novel Task for Fine-Grained Image Understanding”. In: *CoRR* abs/1901.06706 (2019). arXiv: 1901.06706. URL: http://arxiv.org/abs/1901.06706.

[52] Young Peter et al. “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions”. In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 67–78. DOI: 10.1162/tacl_a_00166.