arXiv:2411.12643v1 [cs.LG] 19 Nov 2024

DLBacktrace: A Model Agnostic Explainability for any Deep Learning Models

Vinay Kumar Sankarapu, Chintan Chitroda, Yashwardhan Rathore
{vinay, chintan.chitroda, yashwardhan.rathore}@aryaxai.com

Neeraj Kumar Singh, Pratinav Seth
{neeraj.singh, pratinav.seth}@aryaxai.com

AryaXAI

November 20, 2024

Abstract

The rapid advancement of artificial intelligence has led to increasingly sophisticated deep learning models, which frequently operate as opaque "black boxes" with limited transparency in their decision-making processes. This lack of interpretability presents considerable challenges, especially in high-stakes applications where understanding the rationale behind a model's outputs is as essential as the outputs themselves. This study addresses the pressing need for interpretability in AI systems, emphasizing its role in fostering trust, ensuring accountability, and promoting responsible deployment in mission-critical fields. To address the interpretability challenge in deep learning, we introduce **DLBacktrace**, an innovative technique developed by the AryaXAI team to illuminate model decisions across a wide array of domains, including simple Multi Layer Perceptron (MLPs), Convolutional Neural Networks (CNNs), Large Language Models (LLMs), Computer Vision Models, and more.

We provide a comprehensive overview of the DLBacktrace algorithm and present benchmarking results, comparing its performance against established interpretability methods, such as SHAP, LIME, GradCAM, Integrated Gradients, SmoothGrad, and Attention Rollout, using diverse task-based metrics. The proposed DLBacktrace technique is compatible with various model architectures built in PyTorch and TensorFlow, supporting models like Llama 3.2, other NLP architectures such as BERT and LSTMs, computer vision models like ResNet and U-Net, as well as custom deep neural network (DNN) models for tabular data. This flexibility underscores DLBacktrace's adaptability and effectiveness in enhancing model transparency across a broad spectrum of applications. The library is open-sourced and available at `https://github.com/AryaXAI/DLBacktrace`.

1 Introduction

Despite remarkable advancements in artificial intelligence, particularly with the evolution of deep learning architectures, even the most sophisticated models face a persistent challenge: they often operate as "black boxes," with internal processes that remain opaque and difficult to interpret. These models produce highly accurate predictions, yet provide limited insights into how and why they make specific decisions. This opacity raises significant concerns, especially in high-stakes applications like healthcare, finance, and law enforcement, where understanding the rationale behind model outputs is critical. For example, in the healthcare sector, AI-driven diagnostics must be interpretable to ensure that medical professionals can trust and act on recommendations for patient treatment. Similarly, in finance, regulations such as the European Union's GDPR mandate a "right to explanation" for automated decisions affecting individuals, making explainability not only an ethical imperative but also a regulatory requirement.

The growing demand for explainability and transparency in AI systems is often eclipsed by the prevailing focus on maximizing raw performance. This trend is especially evident with the increasing use of models like OpenAI's

ChatGPT[1], Meta's LLaMA [2], Google's Gemini, and similar large language models (LLMs) that are widely adopted for tasks such as language generation, classification, and complex question answering. These models frequently operate as 'black boxes,' offering limited insight into their decision-making processes. Many of their architectures are also proprietary and closed-source, adding to their opacity. However, due to their high accuracy and utility in various applications, their lack of transparency is often accepted by users and stakeholders. Consequently, interpretability is frequently overlooked or considered secondary to model performance in deployment. This disregard can lead to challenges, such as increased uncertainty in responses to out-of-domain questions and potential regulatory concerns.

Over the years, various efforts have been made to make machine learning models more transparent, leading to the development of interpretability methods. Popular approaches like Local Interpretable Model-agnostic Explanations (LIME) [3] and SHapley Additive exPlanations (SHAP) [4] aim to clarify model decisions by assigning feature importance scores. These methods work well in tabular data contexts but encounter challenges with complex data types such as images and text, where feature interactions and contextual nuances are intricate. Both LIME and SHAP require repeated evaluations and data perturbations to estimate importance scores, which can increase computational load significantly, especially in high-dimensional contexts. They generate explanations by analyzing selected subsets of data, which leads to instance-specific (local) feature importance. This method may hinder their ability to offer a comprehensive, model-wide perspective, impacting both interpretability and reliability. This issue is particularly pronounced when real-time insights are needed, as the time required to generate explanations increases exponentially with the size of the data subset.

For complex data like images and text, interpretability methods such as Grad-CAM, Integrated Gradients [5], and SmoothGrad help highlight influential regions or tokens in model predictions but come with limitations: Grad-CAM is restricted to CNNs, Integrated Gradients requires carefully chosen baselines, and SmoothGrad's perturbations add computational cost. For transformer-based models like Vision Transformers (ViTs), techniques such as Attention Rollout aggregate attention across layers to trace information flow, and tools like BertViz [6] visualize attention heads, providing insights into the model's decision-making. However, interpreting attention alone can be challenging, as attention weights don't always correlate directly with feature importance, requiring careful analysis for a fuller understanding.

In response to the pressing challenges of interpretability in deep learning, we present, **DLBacktrace** a model-agnostic method for deep learning interpretability by tracing relevance from output to input, assigning relevance scores across layers to reveal feature importance, information flow, and bias within predictions. Operating independently of auxiliary models or baselines, DLBacktrace ensures deterministic, consistent interpretations across various architectures and data types, including images, text, and tabular data. This approach supports both local (instance-specific) and global (aggregate) analysis, enhancing transparency for models such as LLMs (e.g., BERT, Llama) and computer vision architectures (e.g., ResNet, U-Net), making it a reliable tool for detailed model interpretation and validation. This technique addresses limitations in current interpretability frameworks by delivering stable, reliable explanations that accurately reflect decision-making pathways within a model, with insights at both local (instance-specific) and global (feature-aggregate) levels.

In this work, we make the following contributions:

- **Introduction of DLBacktrace:** A detailed methodology outlining the model-agnostic and deterministic approach of DLBacktrace for achieving enhanced interpretability in AI systems.
- **Comprehensive Benchmarking:** We benchmark DLBacktrace against widely used interpretability methods (e.g., LIME, SHAP, Grad-CAM, Integrated Gradients and more) across different tasks.
- **Cross-Modality Applications:** DLBacktrace's adaptability is illustrated across various data types, including tabular, image, and text, addressing limitations in current interpretability methods within these domains.
- **Framework for Reliable Interpretability:** By providing consistent relevance scores, DLBacktrace contributes to more reliable, regulatory-compliant AI systems, supporting ethical and responsible AI deployment.

2 Relevant Literature

2.1 Importance of eXplainable AI (XAI)

2.1.1 XAI for Responsible and Trustworthy AI

Responsible AI is essential for deploying systems that align with ethical standards and societal values, especially in critical sectors like healthcare, finance, and law enforcement, where AI decisions can profoundly impact individuals and communities. Responsible AI emphasizes fairness, transparency, accountability, privacy, and ethical alignment

to prevent bias, protect individual rights, and foster public trust. Fairness ensures that AI models do not discriminate and treat individuals equitably, while transparency provides clear explanations of AI decision-making to build trust. Accountability involves defining responsibility for AI outcomes and enabling human oversight. Privacy and security focus on protecting personal data and ensuring system resilience against security threats, and ethical alignment ensures AI development respects human rights and societal norms.

There has been substantial progress toward responsible AI, supported by regulatory frameworks like the EU's General Data Protection Regulation (GDPR), which enforces transparency and accountability in automated decision-making. Further advancements include interpretability methods like SHAP, LIME, and Grad-CAM, which aim to make complex models more accessible to stakeholders. Despite this, challenges remain. Current interpretability tools often provide limited insights for complex models, display inconsistencies in high-dimensional data, and are not always applicable in real-time settings. Additionally, the rapid evolution of AI technologies outpaces regulatory responses, creating oversight gaps, especially with advanced models such as large language models (LLMs). These challenges underscore the ongoing need for innovations in responsible AI practices and supportive policy development.

In their work, Madsen et al. [7] advocate for moving beyond traditional interpretability methods, introducing three paradigms that focus on faithful model explanations: models with inherent faithfulness measures, those trained to provide faithful explanations, and self-explaining models. Their work highlights that trustworthy AI requires explanations closely aligned with the model's actual decision-making to prevent misunderstandings and misplaced trust. Whereas, Singh et al. [8] explore interpretability challenges in large language models (LLMs), proposing tailored methods to address their scale and complexity, enabling clearer insights into model behavior. Tull et al. [9] present a category theory-based framework to unify interpretability approaches, aiming for a cohesive understanding of model behavior crucial to responsible AI. Dinu et al. [10] critically examine assumptions in feature attribution, finding that some methods lack reliability, thus stressing the need for rigorous evaluation of interpretability techniques. While, Kaur et al. [11] apply sensemaking theory to AI, promoting explanations that align with human cognitive processes to enhance trust and accountability by making model reasoning more accessible.

2.1.2 XAI for Safe AI

AI safety aims to ensure AI systems are predictable, controllable, and aligned with human values, especially in critical areas like healthcare, autonomous vehicles, and infrastructure. A key aspect of AI safety is explainability, as transparent models enable developers, users, and regulators to understand system behavior, detect risks, and implement safeguards. Core pillars of AI safety include robustness, reliability, alignment with human intentions, and the use of explainability to clarify decision processes. Explainability is essential for risk mitigation in dynamic environments, where understanding AI behavior allows for safe intervention. For example, [12] discusses challenges like reward hacking and exploration hazards in reinforcement learning. Explainability techniques help analyze reward structures and behavior patterns, enabling developers to detect and correct unintended actions. Catastrophic forgetting, where models lose prior knowledge when learning new tasks, poses risks in sequential learning [13]. Explainable AI (XAI) methods can identify model areas vulnerable to forgetting, supporting memory mechanisms that preserve safety-critical information. [14] introduces reward modeling to align AI with human preferences through feedback. Explainability tools provide insights into reward structures' impact on agent behavior, aiding iterative refinement to align model goals with human values. Explainability also supports adversarial robustness by revealing patterns in model vulnerability, guiding defenses that enhance safety and reliability. These examples underscore explainability's role in AI safety, enhancing transparency, accountability, and risk mitigation. Embedding explainable practices within AI safety frameworks helps developers control AI behavior, reducing risks in diverse operational contexts.

2.1.3 XAI for Regulatory AI

Explainable AI (XAI) is crucial for regulatory compliance, promoting transparency, fairness, and accountability in AI-driven decisions in sectors like finance, healthcare, and law. Regulatory frameworks increasingly mandate interpretable models to ensure oversight, protect user rights, and uphold ethical standards. Key elements of XAI in regulatory contexts include transparent decision processes, model auditability, and mechanisms to mitigate bias. In finance, XAI helps institutions clarify decisions on credit scoring, loan approvals, and fraud detection, reducing regulatory risks and fostering public trust. For instance, [15] systematically reviews XAI applications in finance, illustrating transparency's role in regulatory compliance. As large language models (LLMs) become ubiquitous, interpretability in NLP has gained importance. [16] examines alignment of interpretability methods with stakeholder needs, categorizing techniques and identifying differences between developer and non-developer requirements. Stakeholder-centered frameworks help ensure more responsible AI deployment. In healthcare, XAI is vital for patient safety and ethical standards. Explainable models enable healthcare providers to interpret AI-driven diagnoses, treatments, and risk assessments, aligning these with medical regulations. [17] addresses the challenge of conflicting post hoc explanations, often resolved

by practitioners ad hoc. This study calls for standardized metrics to enhance explanation reliability, especially in high-stakes fields like healthcare and finance. Specific to LLMs, [8] explores their potential for interactive, natural language explanations that can improve comprehension of complex behaviors. The authors address interpretability challenges, such as hallucinations and computational cost, recommending LLM-based methods to improve transparency and nuanced insights in high-accountability domains. Finally, [18] critiques common interpretability techniques, highlighting limitations in methods like Layer-wise Relevance Propagation (LRP) [19] and proposing Cosine Similarity Convergence (CSC) as a metric to improve explanation accuracy. These studies emphasize XAI's role in enhancing regulatory compliance by fostering transparency, accountability, and fairness. Embedding explainable practices in regulatory frameworks ensures ethical AI use, facilitates oversight, and builds trust across regulated sectors.

2.2 Explainability Methods

2.2.1 Tabular Data

In high-stakes applications such as finance and healthcare, machine learning models like regression and probabilistic algorithms (e.g., decision trees and their variants) are often preferred. This is because deep learning models are frequently described as "black boxes," making it challenging to interpret how they arrive at their conclusions. This lack of transparency can be particularly problematic in critical contexts where accountability and trust are essential.

To enhance interpretability, explainable algorithms like LIME [3] and SHAP [4] are increasingly used. LIME (Local Interpretable Model-Agnostic Explanations) builds simple, interpretable models around specific data points to highlight the most influential features for each prediction. SHAP (SHapley Additive exPlanations) assigns importance scores to each feature, providing both global explanations (by ranking features based on overall importance) and local explanations (by illustrating how individual features contribute to specific predictions).

However, these methods have limitations. LIME's reliance on random sampling can lead to inconsistent explanations for the same data point, and its effectiveness can be sensitive to the choice of perturbation method. SHAP, while comprehensive, can be computationally expensive for large datasets and complex models. Additionally, SHAP's model-agnostic nature may result in less accurate explanations for highly intricate models, like deep neural networks. As a result, both LIME and SHAP may face challenges in providing precise, interpretable explanations for complex deep learning models.

2.2.2 Image Data

In explainable AI (XAI) for image modality-based tasks, gradient-based methods such as GradCAM [20], Vanilla Gradient [21], SmoothGrad [22], and Integrated Gradients [5] are widely used for interpreting model predictions. GradCAM generates heatmaps by calculating the gradient of the target class with respect to convolutional layer activations, but may miss fine details and is sensitive to input noise. Vanilla Gradient directly computes gradients on the input, though it faces the "saturation problem," where gradients become too small for clear interpretation. SmoothGrad improves clarity by averaging gradients with added noise, albeit at a computational cost, while Integrated Gradients addresses saturation by calculating an integral of gradients from a baseline to the input, though it also demands significant computation.

In recent advances, Vision Transformers (ViTs) require specific interpretability approaches due to their reliance on attention mechanisms. In the paper [23], authors introduces TokenTM, a method designed for ViTs that considers both token transformations (changes in token length and direction) and attention weights across layers. By aggregating these factors, TokenTM provides more focused and reliable explanations, addressing unique interpretability challenges in transformer models. These developments reflect a shift in XAI, where interpretability techniques are tailored to the unique demands of model architectures, like CNNs and transformers, enhancing transparency and reliability across different models.

2.2.3 Textual Data

In the text modality, quite a few explanation methods are employed to enhance the interpretability of machine learning models. Among these, LIME [3] and SHAP [4] are baseline for interpreting text classification models. Gradient-based methods, such as GradCAM [20], Integrated Gradients [5], and Attention Rollout [24], also play a significant role across diverse model architectures.

For text generation tasks, 17 challenges were identified by [25], such as tokenization effects and randomness, and advocates for probabilistic explanations and perturbed benchmarks to address these issues. Furthermore, LACOAT introduced by [26], which clusters word representations to produce context-aware explanations. It maps test features to latent clusters and translates these into natural language summaries, improving interpretability for complex NLP

tasks. While [27] draws attention to the noise in explanations generated by large language models (LLMs), noting significant randomness. The study suggests that more sophisticated interpretability techniques, beyond simple word-level explanations, are needed to achieve reliability. Lastly, [28] proposes a hybrid approach that combines counterfactual explanations with domain knowledge. This method generates context-specific counterfactuals and incorporates user feedback, enhancing BERT's interpretability through expert insights and interactive elements, thus fostering a more transparent framework.

In addition to the previously discussed methods, Mechanistic Interpretability has become a cornerstone of research aimed at unraveling the internal mechanisms of large language models (LLMs). This field focuses on dissecting how specific components, such as neurons and attention heads, contribute to a model's functionality and decision-making processes. For example, Olah et al. [29] performed an in-depth analysis of individual neurons in LLMs, revealing how certain neurons specialize in detecting specific linguistic features. Their work highlights the modularity and specialization inherent in these models. Building on this foundation, Elhage et al. [30] introduced the concept of "induction heads" in transformer architectures. They demonstrated that these attention heads play a critical role in tasks such as sequence copying, offering insight into the mechanisms underlying certain model behaviors. Further advancing the field, Nanda et al. [31] investigated the phenomenon of "grokking" in LLMs, where models suddenly exhibit strong generalization capabilities after extended training. Their analysis traced the internal changes leading to this abrupt performance improvement, shedding light on this intriguing aspect of model behavior. These advancements underscore the significance of mechanistic interpretability in enhancing our understanding of the intricate operations of LLMs. By making AI systems more transparent and reliable, this research is vital for fostering trust, particularly as LLMs are scaled and deployed in high-stakes applications.

2.2.4 Metrics for Benchmarking Explainability

Recent advancements in Explainable AI (XAI) emphasize the need for robust evaluation frameworks, benchmarks, and specialized toolkits to enhance transparency and trust in machine learning systems. Quantus [32], provides a modular toolkit with metrics such as faithfulness, robustness, and completeness, promoting reproducible and responsible evaluation of neural network explanations. BEExAI [33] addresses the underexplored domain of tabular data by benchmarking explanation methods using tailored datasets and metrics like local fidelity and human interpretability, enabling systematic comparisons in real-world scenarios. In a survey over 30 XAI toolkits by [34], including Quantus, highlighting challenges such as inconsistent metrics, lack of fairness assessments, and insufficient focus on human-centered evaluation, calling for standardized benchmarks and unified platforms. PyXAI [35] complements these efforts by focusing on tree-based models, introducing efficient tools and unique metrics such as path importance and rule-level interpretability for domains like healthcare and finance. Despite these advancements, key gaps remain, including standardization across tools, evaluation for diverse data types, human-centered usability, and fairness and robustness assessments, underscoring the need for continued research to achieve actionable and responsible XAI.

3 Backtrace

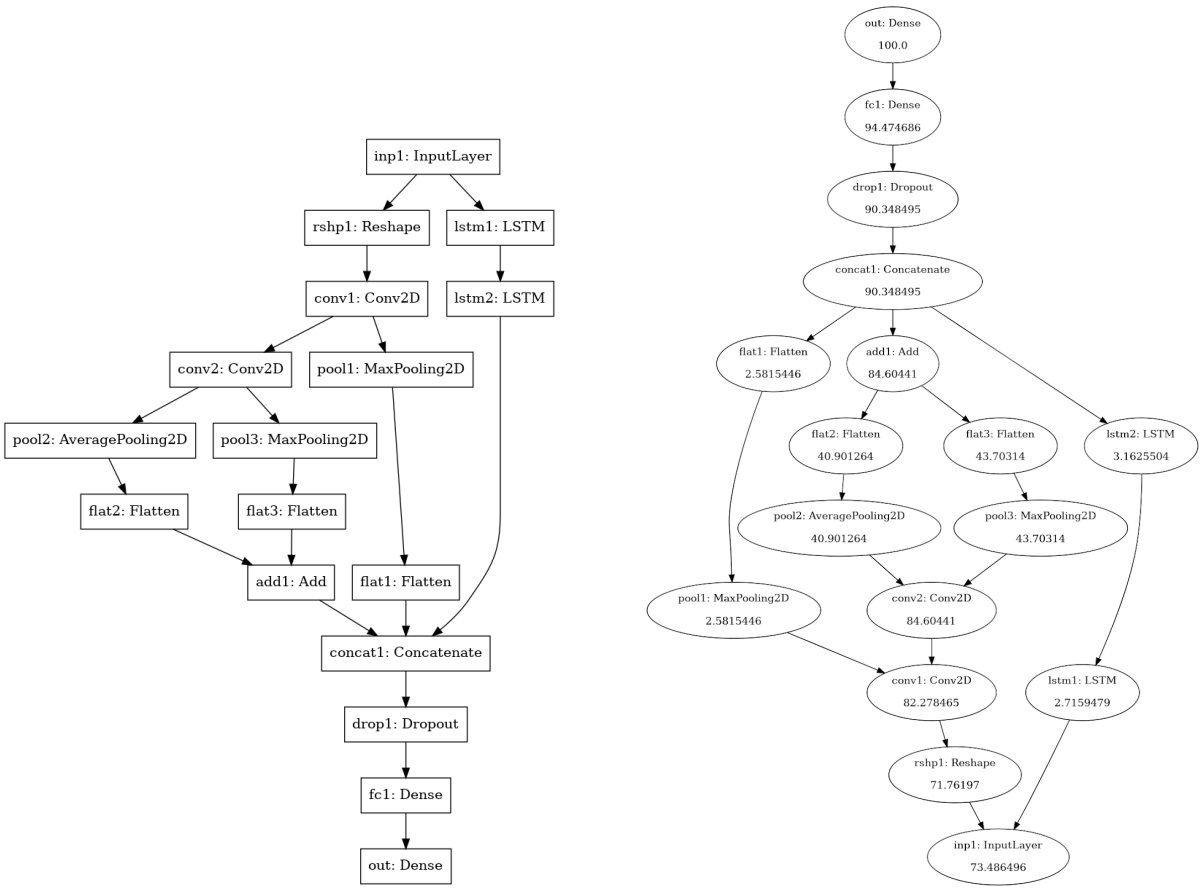
3.1 Introduction

Backtrace is a technique for analyzing neural networks that involves tracing the relevance of each component from the output back to the input.

This approach clarifies how each element contributes to the final prediction. By distributing relevance scores across various layers, Backtrace provides insights into feature importance, information flow, and potential biases, which facilitates improved model interpretation and validation without relying on external dependencies.

Backtrace has the following advantages over other available tools:

- **No dependence on a sample selection algorithm :**
  The relevance is calculated using just the sample in focus. This avoids deviations in importance due to varying trends in sample datasets.
- **No dependence on a secondary white-box algorithm :**
  The relevance is calculated directly from the network itself. This prevents any variation in importance due to type, hyperparameters and assumptions of secondary algorithms.
- **Deterministic in nature**
  The relevance scores won't change on repeated calculations on the same sample. Hence, can be used in live environments or training workflows as a result of its independence from external factors.



(a) Sample Network (b) Relevance Ouptput for Sample Network

Figure 1: Illustration Depecting Backtrace Calculation for a Sample Network

3.2 Methodology

Backtrace operates in two modes: Default Mode and Contrast Mode.

First we describe the Basic Methodology of Backtrace in Default Mode as follows :

3.2.1 Basic Methodology

Every neural network consists of multiple layers. Each layer has a variation of the following basic operation:

$$y = \Phi(Wx + b)$$

where,

- Φ = activation function
- W = weight matrix of the layer
- b = bias
- x = input
- y = output

This can be further organized as:

$$y = \Phi(X_p + X_n + b)$$

where,

- $X_p = \sum W_i x_i \quad \forall W_i x_i > 0$
- $X_n = \sum W_i x_i \quad \forall W_i x_i < 0$

Activation functions can be categorized into monotonic and non-monotonic functions.

- Non-Monotonic functions: The relevance is propagated as is.
- Monotonic functions: The relevance is switched off for positive or negative components based on saturation.

3.2.2 Relevance Propagation

The aforementioned modes represent the basic operations at each source layer for propagating relevance to the destination layer. The procedure for relevance calculation is as follows:

1. Construct a graph from the model weights and architecture with output nodes as root and input nodes as leaves.
2. Propagate relevance in a breadth-first manner, starting at the root.
3. The propagation completes when all leaves (input nodes) have been assigned relevance.

Note: Any loss of relevance during propagation is due to network bias.

The relevance of a single sample represents local importance. For global importance, the relevance of each feature can be aggregated after normalization at the sample level.

3.3 Algorithm

The algorithm has two modes of operation:

- Default Mode
- Contrastive Mode

3.3.1 Default Mode

In this mode, a single relevance is associated with each unit. The relevance is propagated by proportionately distributing it between positive and negative components. If the relevance associated with y is r_y and with x is r_x, then for the jth unit in y, we compute:

$$T_j = X_{pj} + |X_{nj}| + |b_j| \tag{1}$$

$$R_{pj} = \frac{X_{pj}}{T_j} r_{yj}, \quad R_{nj} = \frac{X_{nj}}{T_j} r_{yj}, \quad R_{bj} = \frac{b_j}{T_j} r_{yj} \tag{2}$$

R_{pj} and R_{nj} are distributed among x in the following manner:

$$r_{xij} = \begin{cases} \frac{W_{ij}x_{ij}}{X_{pj}} R_{pj} & \text{if } W_{ij}x_{ij} > 0 \\ 0 & \text{if } W_{ij}x_{ij} > 0 \text{ and } \Phi \text{ is saturated on negative end} \\ \frac{-W_{ij}x_{ij}}{X_{nj}} R_{nj} & \text{if } W_{ij}x_{ij} < 0 \\ 0 & \text{if } W_{ij}x_{ij} < 0 \text{ and } \Phi \text{ is saturated on positive end} \\ 0 & \text{if } W_{ij}x_{ij} = 0 \end{cases} \tag{3}$$

The total relevance at layer x is:

$$r_x = \sum_i r_{xi} \tag{4}$$

3.3.2 Contrastive Mode:

In this mode, each unit is assigned dual relevance, distributed between positive and negative components. This approach facilitates separate analyses of supporting and detracting influences. Unlike single-mode propagation, which combines relevance into aggregated scores, dual-mode propagation provides clarity by isolating favorable and adverse contributions. This separation enhances interpretability, enabling deeper insights into features that negatively impact predictions a capability essential for identifying counterfactuals or assessing model biases in high-stakes scenarios.

If the relevance associated with y are r_{yp}, r_{yn} and with x are r_{xp}, r_{xn}, then for the jth unit in y, we compute:

$$T_j = X_{pj} + X_{nj} + b_j \tag{5}$$

We then calculate Determine R_{pj}, R_{nj}, and Relevance Polarity as described in Algorithm 1.

Algorithm 1 Determine R_{pj}, R_{nj}, and relevance polarity in Contrastive Mode

if $T_j > 0$ **then**
  if $r_{ypj} > r_{ynj}$ **then**
    $R_{pj} \leftarrow r_{ypj}$
    $R_{nj} \leftarrow r_{ynj}$
    relevance_polarity $\leftarrow 1$
  else
    $R_{pj} \leftarrow r_{ynj}$
    $R_{nj} \leftarrow r_{ypj}$
    relevance_polarity $\leftarrow -1$
  end if
else
  if $r_{ypj} > r_{ynj}$ **then**
    $R_{pj} \leftarrow r_{ynj}$
    $R_{nj} \leftarrow r_{ypj}$
    relevance_polarity $\leftarrow -1$
  else
    $R_{pj} \leftarrow r_{ypj}$
    $R_{nj} \leftarrow r_{ynj}$
    relevance_polarity $\leftarrow 1$
  end if
end if

Afterwards, R_{pj} and R_{nj} are distributed among x as described in Algorithm 2.

Algorithm 2 Computation of $r_{xp,ij}$ and $r_{xn,ij}$ based on relevance polarity

if relevance_polarity > 0 **then**
  $r_{xp,ij} \leftarrow \frac{W_{ij}x_{ij}}{X_{pj}} R_{pj} \quad \forall W_{ij}x_{ij} > 0$
  $r_{xn,ij} \leftarrow \frac{-W_{ij}x_{ij}}{X_{nj}} R_{nj} \quad \forall W_{ij}x_{ij} < 0$
else
  $r_{xp,ij} \leftarrow \frac{-W_{ij}x_{ij}}{X_{nj}} R_{nj} \quad \forall W_{ij}x_{ij} < 0$
  $r_{xn,ij} \leftarrow \frac{W_{ij}x_{ij}}{X_{pj}} R_{pj} \quad \forall W_{ij}x_{ij} > 0$
end if

The total positive and negative relevance at layer x are:

$$r_{xp} = \sum_i r_{xp,i}, \quad r_{xn} = \sum_i r_{xn,i} \tag{6}$$

3.4 Relevance for Attention Layers:

Currently, the majority of AI models across various applications are primarily based on the attention mechanism [36]. Accordingly, we have extended our Backtrace algorithm to provide support for this attention model [37].

Attention mechanism allows the model to focus on specific parts of the input sequence, dynamically weighting the importance of different elements when making predictions. The attention function employs the equation:

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

For Multi-Head Attention, the implementation is as follows:

$$MultiHead(Q, K, V) = \textbf{Concat}\left(head_1, head_2, ..., head_n\right)W^O \tag{8}$$

where each head is computed as

$$head_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{9}$$

such that

- Q, K, V: Query, Key, Value Matrices
- W_i^Q, W_i^K, W_i^V: Weight matrices for the i-th head
- W^O: Weight matrix for combining all the heads after concatenation
- Concat: Concatenation of the outputs from all attention heads

3.4.1 Relevance Propagation for Attention Layers

Suppose the input to the attention layer is x and the output is y. The relevance associated with y is r_y. To compute the relevance using the Backtrace, we use the steps as indicated in Algorithm 3 below:

Algorithm 3 Relevance Propagation for Attention Layers

1: **Input:** x (input to attention layer)
2: **Output:** r_y (relevance associated with y)
3: We calculate the relevance r_O of **Concat** $(head_1, head_2, \ldots, head_n)$.*Where r_O represents the relevance from the linear projection layer of the Attention module.*
4: To compute the relevance of QK^T and V, use the following formulas:

$$r_{QK} = (r_O \cdot x_V) \cdot x_{QK} \tag{10}$$

$$r_V = (x_{QK} \cdot r_O) \cdot x_V \tag{11}$$

Here, x_{QK} and x_V are the outputs of QK^T and V, respectively.
5: Now that we have r_{QK}, compute the relevance of r_Q and r_K as:

$$r_Q = (r_{QK} \cdot x_Q) \cdot x_K \tag{12}$$

$$r_K = (x_K \cdot r_{QK}) \cdot x_Q \tag{13}$$

Here, x_Q and x_K are the outputs of Q and K, respectively.
6: To compute r_{Attn}, sum up r_Q, r_K, and r_V:

$$r_{Attn} = r_Q + r_K + r_V \tag{14}$$

4 Benchmarking

In this section, we present a comparative study to benchmark our proposed Backtrace algorithm against various existing explainability methods. The goal of this evaluation is to assess the effectiveness, robustness, and interpretability of Backtrace in providing meaningful insights into model predictions across different data modalities, including tabular,

image, and text data. By systematically comparing our approach with established methods, we aim to highlight the advantages and potential limitations of Backtrace in the context of explainable artificial intelligence (XAI).

4.1 Setup

The experimental setup consists of three distinct data modalities: tabular, image, and text. Each modality is associated with specific tasks, datasets, and model architectures tailored to effectively evaluate the explainability methods.

4.1.1 Tabular Modality

For the tabular data modality, we focus on a binary classification task utilizing the Lending Club dataset. This dataset is representative of financial applications, containing features that capture various attributes of borrower profiles. We employ a four-layer Multi-Layer Perceptron (MLP) neural network, which is well-suited for learning from structured data and provides a foundation for assessing the performance of explainability techniques.

4.1.2 Image Modality

In the image data modality, we conduct a multi-class classification task using the CIFAR-10 dataset. This benchmark dataset consists of images across 10 different classes, making it ideal for evaluating image classification algorithms. For this experiment, we utilize a fine-tuned ResNet-34 model, known for its deep residual learning capabilities, which enhances the model's ability to learn intricate patterns and features within the images.

4.1.3 Text Modality

The text data modality involves a binary classification task using the SST-2 dataset, which is focused on sentiment analysis. The dataset consists of movie reviews labeled as positive or negative, allowing for a nuanced evaluation of sentiment classification models. We employ a pre-trained BERT model, which leverages transformer-based architectures to capture contextual relationships in text. This approach facilitates the generation of high-quality explanations for the model's predictions, enabling a thorough assessment of explainability methods in the realm of natural language processing.

4.2 Metrics

To assess the effectiveness of explanation methods across various modalities, we utilize different metrics tailored to specific use cases. Further details are provided below:

4.2.1 Tabular Modality

- **Maximal Perturbation Robustness Test (MPRT):** This metric assesses the extent of perturbation that can be applied to an input before there are significant changes in the model's explanation of its decision. It evaluates the stability and robustness of the model's explanations rather than solely its predictions.
- **Complexity Metric:** This metric quantifies the level of detail in a model's explanation by analyzing the distribution of feature contributions.

4.2.2 Image Modality

- **Faithfulness Correlation:** Faithfulness Correlation [38] metric evaluates the degree to which an explanation aligns with the model's behavior by calculating the correlation between feature importance and changes in model output resulting from perturbations of key features.
- **Max Sensitivity:** Max-Sensitivity [39], is a robustness metric for explainability methods that evaluates how sensitive explanations are to small perturbations in the input. Using a Monte Carlo sampling-based approximation, it measures the maximum change in the explanation when slight random modifications are applied to the input. Formally, it computes the maximum distance (e.g., using L_1, L_2, or L_∞ norms) between the original explanation and those derived from perturbed inputs.
- **Pixel Flipping:** Pixel Flipping method involves perturbing significant pixels and measuring the degradation in the model's prediction, thereby testing the robustness of the generated explanation.

4.2.3 Text Modality

To evaluate the textual modality, we use the Token Perturbation for Explanation Quality (ToPEQ) metric, which assesses the robustness of model explanations by analyzing the impact of token perturbations. We employ the Least Relevant First AUC (LeRF AUC) and Most Relevant First AUC (MoRF AUC) to measure sensitivity to the least and most important tokens, respectively. Additionally, we calculate Delta AUC, the difference between LeRF AUC and MoRF AUC, to further indicate the model's ability to distinguish between important and unimportant features.

- **LeRF AUC (Least Relevant First AUC):** This metric evaluates how gradually perturbing the least important features (tokens) affects the model's confidence. The AUC measures the model's response as the least relevant features are replaced with a baseline (e.g., [UNK]), indicating the degree to which the model relies on these features.
- **MoRF AUC (Most Relevant First AUC):** This metric measures how quickly the model's performance deteriorates when the most important features are perturbed first. The AUC represents the decrease in the model's confidence as the most relevant tokens are removed, revealing the impact of these key features on the prediction.
- **Delta AUC:** This metric represents the difference between LeRF AUC and MoRF AUC. It reflects the model's sensitivity to the removal of important features (MoRF) compared to less important ones (LeRF). A larger delta suggests that the explanation method effectively distinguishes between important and unimportant features.

4.3 Experiments

4.3.1 Tabular Modality

We evaluated 1,024 samples from the test set of the Lending Club dataset, using a fine-tuned MLP checkpoint that attained an accuracy of 0.89 and a weighted average F1 score of 0.87. We assessed Backtrace against widely used metrics for tabular data, specifically LIME and SHAP [4], and employed MPRT for comparison, along with Complexity to examine the simplicity of the model explanations as illustrated in Appendix A.1.1.

Table 1: Explaination Performance metrics for explanation methods - LIME,SHAP and Backtrace, including Mean values and feature contributions across different layers (fc1 to fc4). Lower values in MPRT and Complexity indicate better performance.

Method	**MPRT (↓)**					**Complexity**
	Mean	**fc1**	**fc2**	**fc3**	**fc4**	**(↓)**
LIME	0.933	0.934	0.933	0.933	0.933	2.57
SHAP	0.684	0.718	0.65	0.699	0.669	**1.234**
Backtrace	**0.562**	**0.579**	**0.561**	**0.557**	**0.552**	2.201

The proposed method, **Backtrace**, as demonstrated in Table 1, achieves superior performance compared to both LIME and SHAP, evidenced by lower Maximal Perturbation Robustness Test (MPRT) values across various layers. This highlights its improved interpretability and robustness in explainability. However, Backtrace exhibits higher computational complexity, reflecting the fine-grained, higher entropy of its explanations. This trade-off suggests the need to balance the quality of interpretability with the simplicity of model explanations.

4.3.2 Image Modality

We conducted an evaluation on 500 samples from the CIFAR-10 test set using a supervised, fine-tuned ResNet-34 model, which achieved a test accuracy of 75.85%. We compared Backtrace against several methods as illustrated in Appendix A.1.2, including Grad-CAM, vanilla gradient, smooth gradient, and integrated gradient. The comparison utilized metrics such as Faithfulness Correlation, Max Sensitivity, and Pixel Flipping.

Table 2: Performance metrics of various explanation methods for a subset of CIFAR10 test set samples. Higher values (↑) of Faithfulness Correlation indicate better performance, while lower values (↓) of Max Sensitivity and Pixel Flipping suggest improved robustness. (*) - Indicates the presence of infinite values in some batches, for which a non-infinite mean was used to calculate the final value.

Explanation Method	Faithfulness Correlation (↑)	Max Sensitivity (↓)	Pixel Flipping (↓)
GradCAM	0.010	1070(*)	0.249
Vanilla Gradient	0.011	154(*)	0.253
Smooth Grad	0.018	158(*)	0.252
Integrated Gradient	0.009	169(*)	0.253
Backtrace	**0.199**	**0.617**	**0.199**

The Evaluations conducted as shown in Table 2 reveals that Backtrace significantly outperforms traditional methods like Grad-CAM, Vanilla Gradient, Smooth Gradient, and Integrated Gradient across all key metrics. Backtrace achieves a superior Faithfulness Correlation score of (0.199), indicating a stronger alignment between its explanations and the model's behavior. Additionally, it demonstrates robust performance with much lower Max Sensitivity (0.617) and Pixel Flipping (0.199) scores, highlighting its stability against input perturbations and better robustness in preserving the model's predictive integrity under pixel modifications. Overall, Backtrace establishes itself as a more reliable and robust explainability technique for image modality tasks.

4.3.3 Text Modality

For the text modality, evaluation was performed on the evaluation set of the SST-2 dataset using a fine-tuned BERT model[1], achieving an F1 score of 0.926. Since explainable AI (XAI) for BERT and other transformer-based models is relatively new, we employed metrics based on token perturbation for explanation quality, specifically LeRF, MoRF, and Delta AUC, as introduced in [37]. We used methods like IG, SmoothGrad, AttnRoll, GradCAM and Input Grad as illustrated in Appendix A.1.5.

Table 3: Token Perturbation for Explanation Quality metrics for various explanation methods. Lower MoRF AUC values indicate better performance, while higher LeRF AUC and Delta AUC values suggest greater robustness and better differentiation between relevant and irrelevant features.

Method	MoRF AUC (↓)	LeRF AUC (↑)	Delta AUC (↑)
IG	-6.723	46.756	53.479
Smooth Grad	14.568	38.264	23.696
Attn Roll	16.123	37.937	21.814
Backtrace	15.431	30.69	15.259
GradCAM	19.955	21.714	1.759
Random	25.068	25.684	0.616
Input Grad	27.784	19.34	-8.444

As shown in Table 3, Integrated Gradients (IG) delivered the strongest performance, achieving the lowest MoRF AUC and the highest LeRF and Delta AUC, underscoring the robustness of its explanations and precise feature attribution. Smooth Grad and Attn Roll also demonstrated commendable performance. Backtrace exhibited balanced results across LeRF and Delta AUC metrics, with a MoRF AUC of 15.431, showcasing its ability to provide meaningful explanations while allowing scope for further enhancement. These findings highlight the effectiveness of IG in explainability for transformer-based models while recognizing Backtrace's potential as a promising and competitive approach.

4.4 Observations

The quality of explanations is influenced by both the input data and the model weights. The impact of model performance is significant; low model performance tends to result in unstable explanations characterized by high entropy, while good model performance is associated with stable explanations that are more sparse. Additionally, the inference time for a sample is proportional to both the size of the model and the computational infrastructure used.

[1]Model checkpoint used `https://huggingface.co/textattack/bert-base-uncased-SST-2`

5 Discussion

Additional illustrations of Backtrace for various use cases are provided in Appendix A.

5.1 Advantages of Backtrace

5.1.1 Network Analysis

- Existing solutions involve distribution graphs and heatmaps for any network node based on node activation.
- These are accurate for that specific node but don't represent the impact of that node on the final prediction.
- Existing solutions are also unable to differentiate between the impact of input sources versus the internal network biases.

5.1.2 Feature Importance

- With each input source being assigned a fraction of the overall weightage, we can now quantify the dependence of the final prediction on each input source.
- We can also evaluate the dependence within the input source as the weight assignment happens on a per unit basis.
- Integrated Gradients and Shapley values are other methods available for calculating feature importance from Deep Learning Models. Both come with caveats and give approximate values:
  - Integrated Gradients depends on a baseline sample which needs to be constructed for the dataset and altered as the dataset shifts. This is extremely difficult for high-dimensional datasets.
  - Shapley Values are calculated on a sample set selected from the complete dataset. This makes those values highly dependent on the selection of data.

5.1.3 Uncertainty

- Instead of just relying on the final prediction score for decision-making, the validity of the decision can now be determined based on the weight distribution of any particular node with respect to the prior distribution of correct and incorrect predictions.

5.2 Applicability

The Backtrace framework is applicable in the following use-cases:

5.2.1 Interpreting the model outcomes using the local and global importance of each feature

The local importance is directly inferred from the relevance associated with input data layers. For inferring global importance, the local importance of each sample is normalized with respect to the model outcome of that sample. The normalized local importance from all samples is then averaged to provide global importance. The averaging can be further graded based on the various outcomes and binning of the model outcome.

5.2.2 Network analysis based on the relevance attributed to each layer in the network

The two modes together provide a lot of information for each layer, such as:

- Bias to input ratio
- Activation Saturation
- Positive and negative relevance (unit-wise and complete layer)

Using this information, layers can be modified to increase or decrease variability and reduce network bias. Major changes to the network architecture via complete shutdown of nodes or pathways are also possible based on the total contribution of that component.

5.2.3 Fairness and bias analysis using the feature-wise importance

This is in continuation of the global importance of features. Based on the global importance of sensitive features (e.g. gender, age, etc.) and their alignment with the data, it can be inferred whether the model or data have undue bias towards any feature value.

5.2.4 Process Compliance based on the ranking of features on local and global levels

Using the local and global importance of features and ranking them accordingly, it can be determined whether the model is considering the features in the same manner as in the business process it is emulating. This also helps in evaluating the solution's alignment with various business and regulatory requirements.

5.2.5 Validating the model outcome

Every model is analyzed based on certain performance metrics which are calculated over a compiled validation dataset. This doesn't represent the live deployment scenario.

During deployment, validation of outcomes is extremely important for complete autonomous systems. The layer-wise relevance can be used for accomplishing this. The relevance for each layer is mapped in the vector space of the same dimension as the layer outcome, yet it is linearly related to the model outcome.

Since the information changes as it passes through the network, the relevance from lower layers, even input layers, can be used to get different outcomes. These outcomes can be used to validate the model outcome. The layers are generally multi-dimensional, for which either proximity-based methods or white-box regression algorithms can be used to derive outcomes.

6 Conclusion

In this paper, we introduced the **DLBacktrace**, a new method that significantly improves model interpretability for deep learning. DLBacktrace traces relevance from output back to input, giving clear and consistent insights into which features are important and how information flows through the model. Unlike existing methods, which often rely on changing inputs or using other algorithms, DLBacktrace is stable and reliable, making it especially useful in fields that need high transparency, like finance, healthcare, and regulatory compliance. Our benchmarking results demonstrate that DLBacktrace performs better in terms of robustness and accuracy across various model types, proving it can provide practical insights. Overall, DLBacktrace contributes to the growing field of explainable AI by enhancing model transparency and trustworthiness, promoting responsible AI deployment in critical applications.

7 Future Works

Future research on DLBacktrace will aim to broaden its use and improve how it scores relevance. Key areas for development include adapting DLBacktrace for complex and evolving model architectures, like advanced transformers and multimodal models to ensure it remains effective across different AI applications. Additionally, we aim to reduce the inference time making DLBacktrace more suitable for real-time applications in production environments such as autonomous systems and dynamic decision-making scenarios.

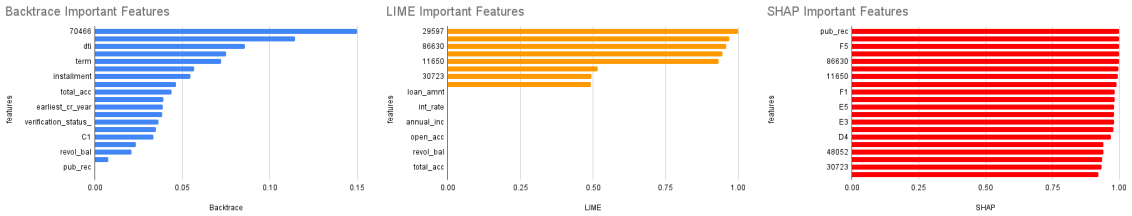
Future development will also explore the use of DLBacktrace for specific model improvements, including diagnosis and targeted editing. For example, DLBacktrace could assist in model pruning by identifying less critical components, tehreby optimizing model performance and efficiency. In addition, DLBacktrace's potential for targeted model improvements will be explored. It can assist in model pruning, especially for Mixtures of Experts (MoEs), by identifying underutilized components or redundant experts to optimize performance and efficiency. It can also help in facilitate model merging, providing insights for seamless integration of multiple models, and layer swapping, enabling selective replacement of layers to enhance adaptability or performance. We also plan to apply DLBacktrace to out-of-distribution (OOD) detection, where it can help distinguish instances that fall outside the model's training data, enhancing the robustness and reliability of AI systems.

Furthermore, extending DLBacktrace's support for model-agnostic explainability will allow it to be seamlessly applied across various architectures, making it a versatile tool in explainable AI. These improvements will make DLBacktrace more useful and establish it as an important tool for understanding and improving models across a wide range of AI applications and tasks.

A Illustrations

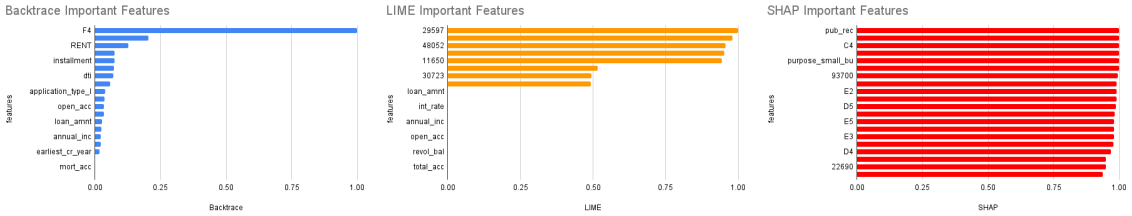
A.1 Illustrations for Various Tasks

A.1.1 Tabular Modality : Binary Classification



(a) Backtrace (b) LIME (c) SHAP

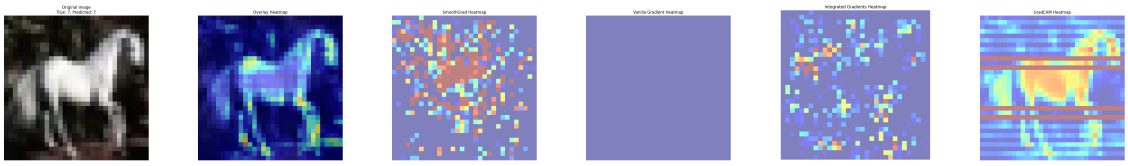
Figure 2: Illustration of Explanations of a Correctly Classified Sample from the Lending Club Dataset where Loan was Fully Paid and was predicted by MLP as Fully Paid.



(a) Backtrace (b) LIME (c) SHAP

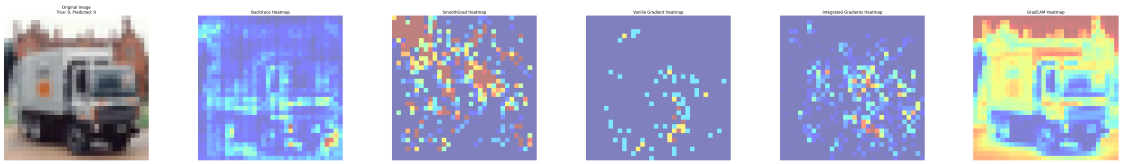
Figure 3: Illustration of Explanations of a Incorrectly Classified Sample from the Lending Club Dataset where Loan was Fully Paid and was predicted by MLP as Charged Off.

A.1.2 Image Modality : Multi-Class Classification



(a) Original Image (b) Backtrace (c) Smooth Grad (d) Vanilla Gradient (e) IG (f) GradCAM

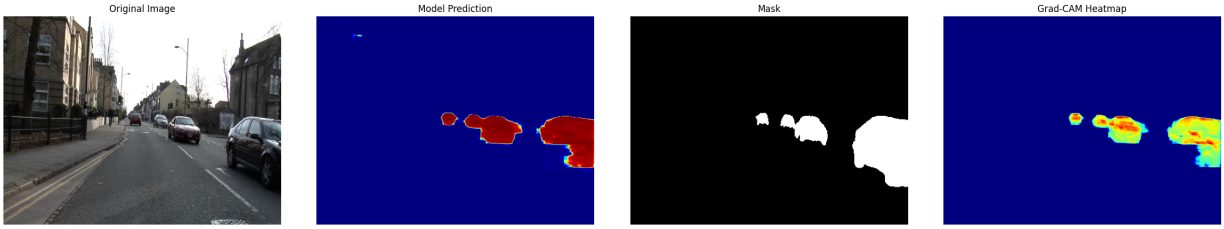
Figure 4: Visualizing ResNet's decisions on a Horse image of CIFAR10 Dataset using various explanation methods.



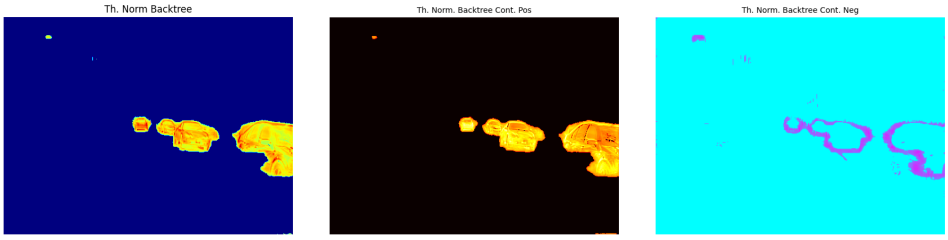
(a) Original Image (b) Backtrace (c) Smooth Grad (d) Vanilla Gradient (e) IG (f) GradCAM

Figure 5: Visualizing ResNet's decisions on a Truck image of CIFAR10 Dataset using various explanation methods.

A.1.3 Image Modality : Object Segmentation

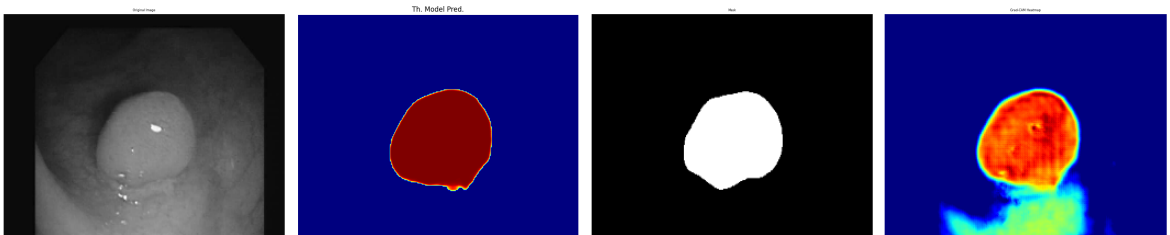


(a) Original Image (b) Model Prediction (c) Label (d) GradCAM

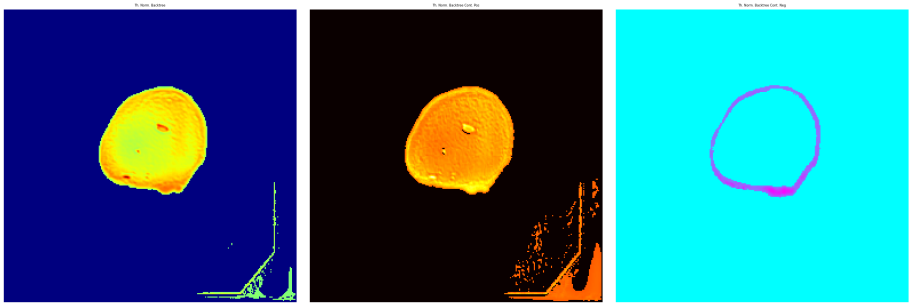


(e) Backtrace Default (f) Backtrace Contrastive (Pos) (g) Backtrace Contrastive (Neg)

Figure 6: Analysis of a U-Net segmentation model's decision-making on a CamVid Dataset Sample. The figure shows the original Image, Model Prediction, and Label, alongside Explanations of GradCAM and Backtrace visualizations in Default and Contrastive modes.



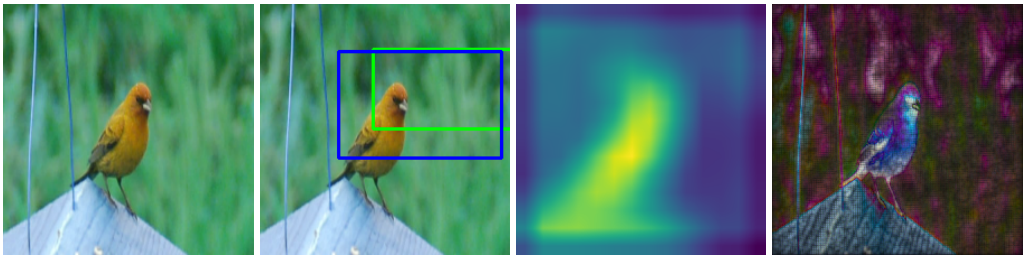
(a) Original Image (b) Prediction (c) Label (d) GradCAM



(e) Backtrace Default (f) Backtrace Contrastive (Pos) (g) Backtrace Contrastive (Neg)

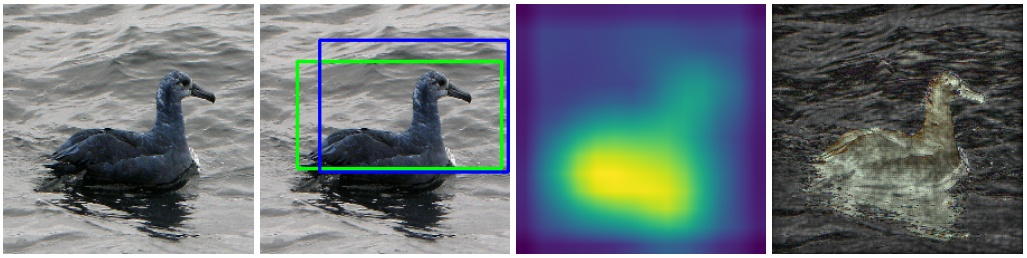
Figure 7: Analysis of a Tumour Segmentation Model's decision-making on a ClinicdB Dataset Sample. The figure shows the original Image, Model Prediction, and Label, alongside Explanations of GradCAM and Backtrace visualizations in Default and Contrastive modes.

A.1.4 Image Modality : Object Detection



(a) Original Image (b) Prediction & Label (c) GradCAM (d) Backtrace

Figure 8: Explanations of the model's decision-making process on a Bird image from the CUB-200 dataset, using Grad-CAM and Backtrace to highlight the key regions influencing the prediction.



(a) Original Image (b) Prediction & Label (c) GradCAM (d) Backtrace

Figure 9: Explanations of the model's decision-making process on a duck image from the CUB-200 dataset, using Grad-CAM and Backtrace to highlight the key regions influencing the prediction.

A.1.5 Text Modality : BERT Sentiment Classification

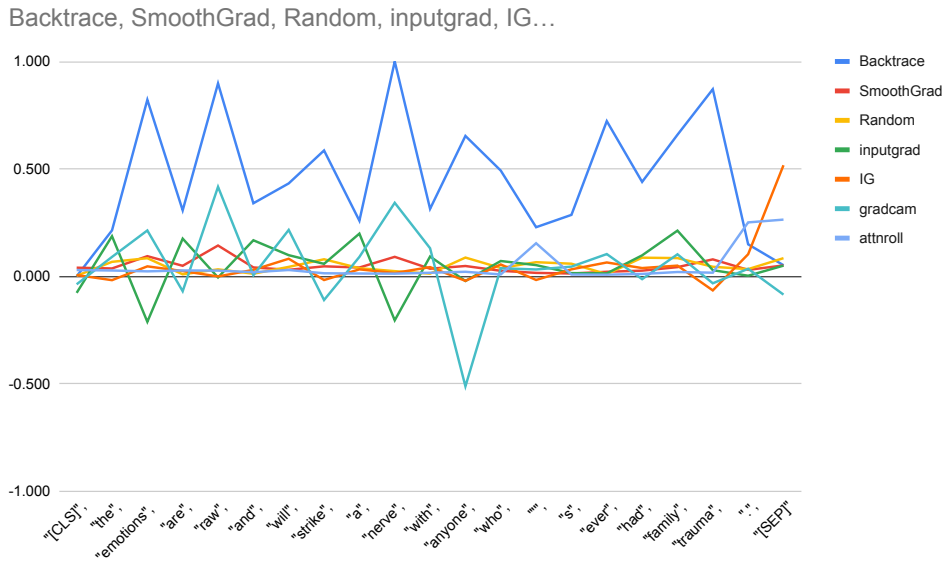
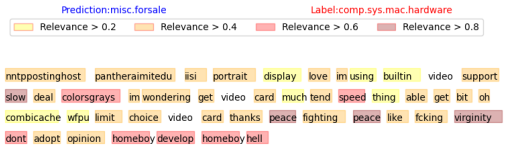
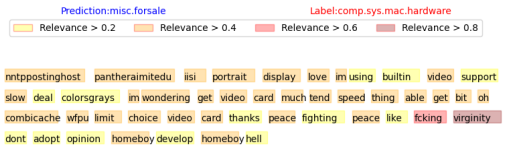


Figure 10: Explanations by different methods for model decision making for Sentiment Analysis for a sample from SST Dataset. **Input Text:** The emotions are raw and will strike a nerve with anyone who ever had family trauma. **Prediction: 1** and **Label: 1**

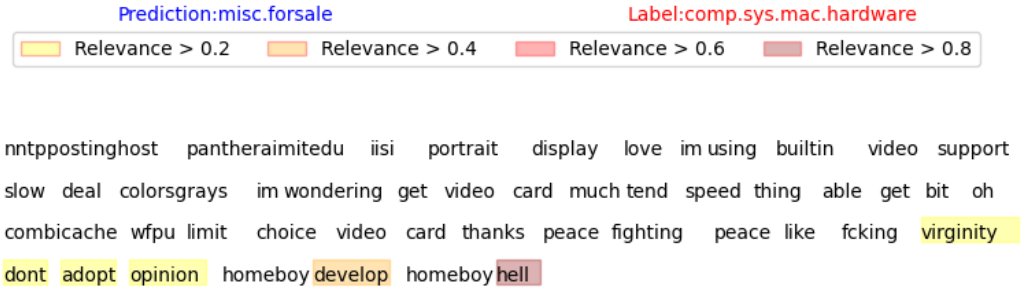
A.1.6 Text Modality : Multi-Class Classification



(a) Original Image

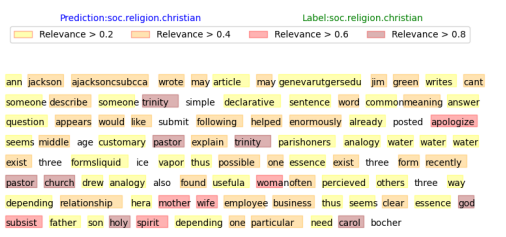


(b) Prediction & Label



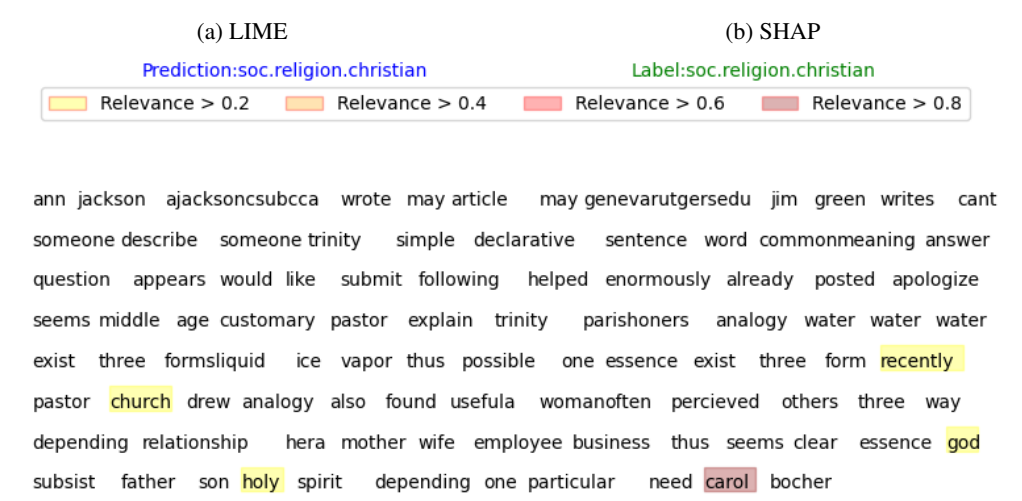
(c) Backtrace

Figure 11: Explanations of the model's decision-making process for multi-class topic detection for incorrect classification on a Model using Pre-Trained Glove Word Embedding and 1D CNN, using LIME, SHAP and Backtrace to highlight the key regions influencing the prediction.



(a) LIME

(b) SHAP



(c) Backtrace

Figure 12: Explanations of the model's decision-making process for multi-class topic detection for correct classification on a Model using Pre-Trained Glove Word Embedding and 1D CNN, using LIME, SHAP and Backtrace to highlight the key regions influencing the prediction.

A.1.7 Text Modality: Llama3.2

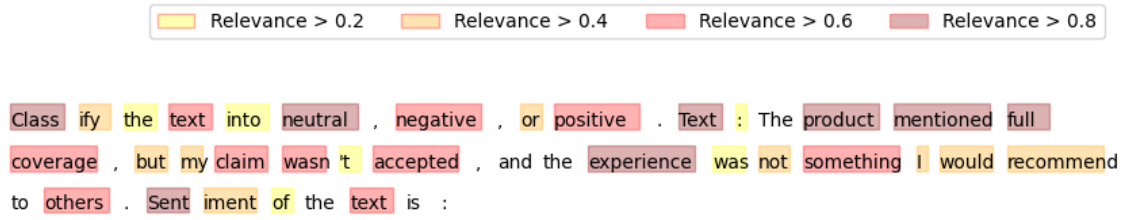


Figure 13:
Input Prompt: Classify the text into neutral, negative, or positive. Text: The product mentioned full coverage, but my claim wasn't accepted, and the experience was not something I would recommend to others. Sentiment of the text is:
Ouput : Negative. Reason: The text expresses dissatisfaction with the product and the experience, stating that the claim was not accepted and the experience was not something the author would recommend. This indicates a negative sentiment. Note: The text does not contain any explicit negative

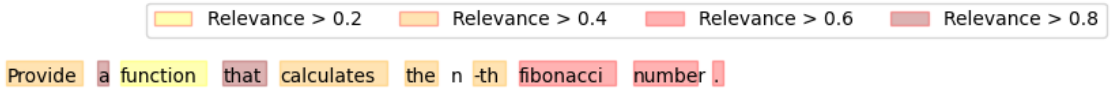


Figure 14: **Input Prompt:** Provide a function that calculates the n-th fibonacci number.
Output: Fibonacci sequence is a series of numbers in which each number is the sum of the two preceding ones, usually starting with 0 and 1.Fibonacci Function. Here's a Python function that calculates the n-th Fibonacci number using memoization to improve performance.
def fibonacci(n, memo=):
Calculate the n-th Fibonacci number.
Args: n (int): The position of the Fibonacci number to calculate. memo (dict): A dictionary to store previously calculated Fibonacci numbers.
Returns: int: The n-th Fibonacci number.
if n <= 0:
return 0
elif n == 1:
return 1
elif n not in memo:
memo[n] = fibonacci(n-1, memo) + fibonacci(n-2, memo)
return memo[n]
Example usage: print(fibonacci(10))
Output: 55

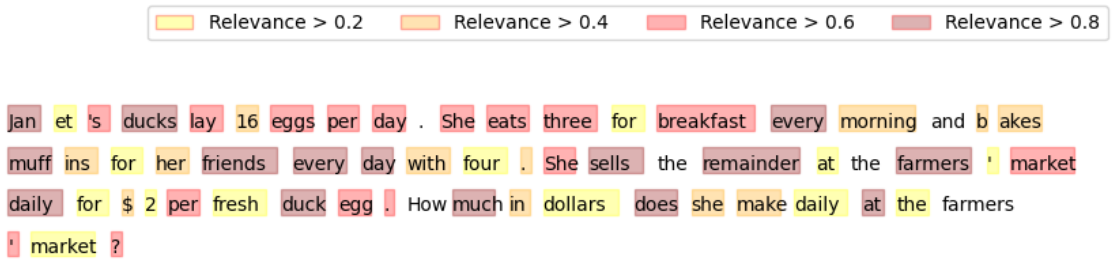


Figure 15:
Input Prompt: Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make daily at the farmers' market?
Output: Step 1: Calculate the number of eggs laid per day Janet's ducks lay 16 eggs per day. Step 2: Calculate the number of eggs eaten per day. Janet eats 3 eggs for breakfast every morning.

References

[1] OpenAI. OpenAI ChatGPT. `https://openai.com/index/chatgpt/`. [Accessed 12-11-2024].

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023.

[3] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[4] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Neural Information Processing Systems*, 2017.

[5] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, 2017.

[6] Jesse Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy, July 2019. Association for Computational Linguistics.

[7] Andreas Madsen, Himabindu Lakkaraju, Siva Reddy, and Sarath Chandar. Interpretability needs a new paradigm. *arXiv preprint arXiv:2405.05386*, 2024.

[8] Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. Rethinking interpretability in the era of large language models. *ArXiv*, abs/2402.01761, 2024.

[9] Sean Tull, Robin Lorenz, Stephen Clark, Ilyas Khan, and Bob Coecke. Towards compositional interpretability for xai. *ArXiv*, abs/2406.17583, 2024.

[10] Jonathan Dinu, Jeffrey P. Bigham, J. Zico Kolter Unaffiliated, and Carnegie Mellon University. Challenging common interpretability assumptions in feature attribution explanations. *ArXiv*, abs/2012.02748, 2020.

[11] Harmanpreet Kaur, Eytan Adar, Eric Gilbert, and Cliff Lampe. Sensible ai: Re-imagining interpretability and explainability using sensemaking theory. *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.

[12] Dario Amodei, Christopher Olah, Jacob Steinhardt, Paul Francis Christiano, John Schulman, and Dandelion Mané. Concrete problems in ai safety. *ArXiv*, abs/1606.06565, 2016.

[13] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[14] Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *ArXiv*, abs/1811.07871, 2018.

[15] Patrick Maximilian Weber, Kim Valerie Carl, and Oliver Hinz. Applications of explainable artificial intelligence in finance—a systematic review of finance, information systems, and computer science literature. *Management Review Quarterly*, 74:867–907, 2023.

[16] Nitay Calderon and Roi Reichart. On behalf of the stakeholders: Trends in nlp model interpretability in the era of llms. *ArXiv*, abs/2407.19200, 2024.

[17] Satyapriya Krishna, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. The disagreement problem in explainable machine learning: A practitioner's perspective. *Trans. Mach. Learn. Res.*, 2024, 2022.

[18] Leon Sixt, Maximilian Granz, and Tim Landgraf. When explanations lie: Why many modified bp attributions fail. In *International Conference on Machine Learning*, 2019.

[19] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus Müller. Layer-wise relevance propagation: An overview. In *Explainable AI*, 2019.

[20] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336 – 359, 2016.

[21] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[22] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *ArXiv*, abs/1706.03825, 2017.

[23] Junyi Wu, Bin Duan, Weitai Kang, Hao Tang, and Yan Yan. Token transformation matters: Towards faithful post-hoc explanation for vision transformer. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10926–10935, 2024.

[24] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Annual Meeting of the Association for Computational Linguistics*, 2020.

[25] Kenza Amara, Rita Sevastjanova, and Mennatallah El-Assady. Challenges and opportunities in text generation explainability. In *xAI*, 2024.

[26] Xuemin Yu, Fahim Dalvi, Nadir Durrani, and Hassan Sajjad. Latent concept-based explanation of nlp models. *ArXiv*, abs/2404.12545, 2024.

[27] Jérémie Bogaert and François-Xavier Standaert. A question on the explainability of large language models and the word-level univariate first-order plausibility assumption. *ArXiv*, abs/2403.10275, 2024.

[28] Aroua Hedhili Sbaï and Islem Bouallagui. Hybrid approach to explain bert model: Sentiment analysis case. In *International Conference on Agents and Artificial Intelligence*, 2024.

[29] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. https://distill.pub/2020/circuits/zoom-in.

[30] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[31] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *ArXiv*, abs/2301.05217, 2023.

[32] Anna Hedström, Leander Weber, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M.-C. Höhne. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations. *ArXiv*, abs/2202.06861, 2022.

[33] Samuel Sithakoul, Sara Meftah, and Clément Feutry. Beexai: Benchmark to evaluate explainable ai. In *xAI*, 2024.

[34] Phuong Quynh Le, Meike Nauta, Van Bach Nguyen, Shreyasi Pathak, Jörg Schlötterer, and Christin Seifert. Benchmarking explainable ai - a survey on available toolkits and open challenges. In *International Joint Conference on Artificial Intelligence*, 2023.

[35] Gilles Audemard, Jean-Marie Lagniez, Pierre Marquis, and Nicolas Szczepanski. PyXAI: An XAI Library for Tree-Based Models. In *The 33rd International Joint Conference on Artificial Intelligence*, pages 8601–8605, Jeju Island (South Korea), South Korea, August 2024.

[36] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017.

[37] Reduan Achtibat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. Attnlrp: Attention-aware layer-wise relevance propagation for transformers. *ArXiv*, abs/2402.05602, 2024.

[38] Umang Bhatt, Adrian Weller, and José M. F. Moura. Evaluating and aggregating feature-based model explanations. In *International Joint Conference on Artificial Intelligence*, 2020.

[39] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. On the (in)fidelity and sensitivity for explanations. *arXiv: Learning*, 2019.