



RealExp: Decoupling correlation bias in Shapley values for faithful model interpretations

Wen-Dong Jiang^a, Chih-Yung Chang^{a,*}, Show-Jane Yen^b, Shih-Jung Wu^a,
Diptendu Sinha Roy^c

^a Department of Computer Science and Information Engineering, Tamkang University, Taoyuan City 333, Taiwan

^b Department of Computer Science and Information Engineering, Ming Chuan University, New Taipei 25137, Taiwan

^c The Department of Computer Science and Engineering, National Institute of Technology, Shillong 793003, India

ARTICLE INFO

Keywords:

Deep learning
Explained artificial intelligence (XAI)
Machine learning
Interpretability

ABSTRACT

Deep learning has achieved significant success in handling unstructured data but remains limited by its "black box" nature, especially in sensitive applications. Existing interpretable machine learning methods partially address this issue but often overlook feature correlations and provide inadequate assessments of model decision paths. To tackle these challenges, this paper introduces Real Explainer (RealExp), a novel interpretability method that decouples the Shapley Value into individual feature importance and feature correlation importance. By integrating feature similarity computations, RealExp enhances interpretability by precisely quantifying both individual contributions and interactions, leading to more reliable explanations. Furthermore, a new interpretability evaluation criterion is proposed, focusing on decision path analysis beyond accuracy-based assessments. Experiments on image classification and sentiment analysis demonstrate RealExp's superiority in interpretability. Case studies highlight its practical benefits: RealExp aids in selecting pre-trained models for pneumonia detection, emphasizes pre-trained video segmentation for anomaly detection, and optimizes text models, achieving performance to RoBERTA finetuning model without pre-trained embeddings.

1. Introduction

Thanks to the rapid advancement of computer hardware, deep learning has made significant progress in the application of unstructured data, such as images (Cao & Chen, 2025) and text (Li et al., 2024). Specifically, the success of representation learning (Wang & Lian, 2025; Zhang et al., 2025) has gradually replaced the earlier approaches of transforming unstructured data into structured formats. The key to the success of representation learning lies in leveraging a large number of parameters for backpropagation, enabling the model to adapt to data with non-normal distributions. Although models based on backpropagation neural networks (Banerjee et al., 2023; Yang et al., 2019) have achieved significant technical advancements, their application in many sensitive domains, such as medicine (Zhang et al., 2025) and industrial inspection (Rathee et al., 2021), still faces considerable challenges due to the difficulty in understanding the basis of their decision-making.

Explainable Artificial Intelligence (XAI) aims to reveal the inner mechanisms of neural network decisions, thereby making these models more reliable for applications in sensitive domains. In recent years, several studies (Guan et al., 2024; Li et al., 2025; Jing et al.,

* Corresponding author.

E-mail address: cychang@mail.tku.edu.tw (C.-Y. Chang).

2023; Liu et al., 2024) have focused on injecting explainability into deep learning models and using various visualization techniques to explain the decisions of these "black box" models. While these models have achieved a certain level of interpretability, two pressing issues remain (Huang & Marques, 2023, 2024): first, whether the correlations between different attributes are correctly evaluated, and second, whether the model's decision-making pathway truly aligns with human reasoning, even when the model's understanding appears consistent with user expectations.

The first issue is that feature correlation was not sufficiently considered. Fig. 1 provides an example: as shown in Fig. 1, the left side depicts a binary tree with four nodes, $\{x_1\}$, $\{x_2\}$, $\{x_3\}$ and $\{x_4\}$, each representing a feature. The values on the edges, such as " $\in \{0\}$ ", " $\in \{1\}$ ", and " $\in \{0,2\}$ ", indicate the branching directions determined by the condition of each feature. The values at the leaf nodes (e.g. "6", "3", "5", "8") are the predicted outcomes of the model along that path.

By observing the decision tree, the importance of each feature can be inferred: features that influence more splits and are closer to the root node have a greater impact on the overall decision. In this tree, feature x_1 is at the root node, directly deciding which child node to proceed to, thus indicating that x_1 is the most important feature in this tree. Overall, the feature importance ranking can be represented as: $x_1 > x_4 > x_2 > x_3$.

However, when using explainable machine learning methods such as Shapley Value (Lundberg & Lee, 2017) (higher Shapley Values indicate higher importance) to compute feature importance, an incorrect result was obtained, with the rank being $x_4 > x_3 > x_1 > x_2$. There are two main reasons for this discrepancy:

First, Shapley Value does not sufficiently account for feature correlation during calculation. It primarily considers the interaction between features without considering the traditional individual importance of each feature. Secondly, when there is redundancy or strong collinearity between features, SHAP distributes the contribution among them, leading to lower Shapley Values for some features, even if they occupy important positions in the decision tree. The specific proof of this issue will be described in detail in Section 3.1.

The second limitation lies in the oversimplified evaluation framework of current interpretability methods. While these techniques typically validate whether model decisions align with human judgments on a superficial outcome level (e.g., identical object localization in classification tasks), they fundamentally ignore the critical alignment between artificial reasoning pathways and authentic cognitive processes. Fig. 2 illustrates this critical discrepancy with a four-panel visualization: starting from the original image (Panel 1) to its superpixel decomposition (Panel 2), followed by the model's activation trace (Panel 3) versus forensic expert's semantic deduction (Panel 4). Though the model accurately identifies violence-associated regions matching expert annotations, its causal chain reveals counterintuitive reasoning patterns – prioritizing apparel features (black pants → white shirt transitions) before recognizing combative gestures, whereas human experts instinctively perceive violent intent through dynamic action primitives rather than static visual elements. This mechanistic inversion exposes the fundamental divergence between artificial decision trajectories and biological cognition modes.

The goal of explainable methods is to provide domain experts with a reasonable basis for deploying models in high-risk scenarios. Therefore, even if the areas identified by the model match those identified by the expert, or even match perfectly at the pixel level, it may still be insufficient to establish trust for high-risk tasks. Clearly, such an explanation is inadequate.

To address the aforementioned issues, this paper proposes a method for interpretability computation called Real Explainer (RealExp). Specifically, this study revisits the calculation of Shapley Values, decoupling the original Shapley Value's average contribution into independent contributions and interaction contributions. Additionally, an interpretability evaluation approach combining expert annotations and the tau coefficient is proposed to further assess deep learning models.

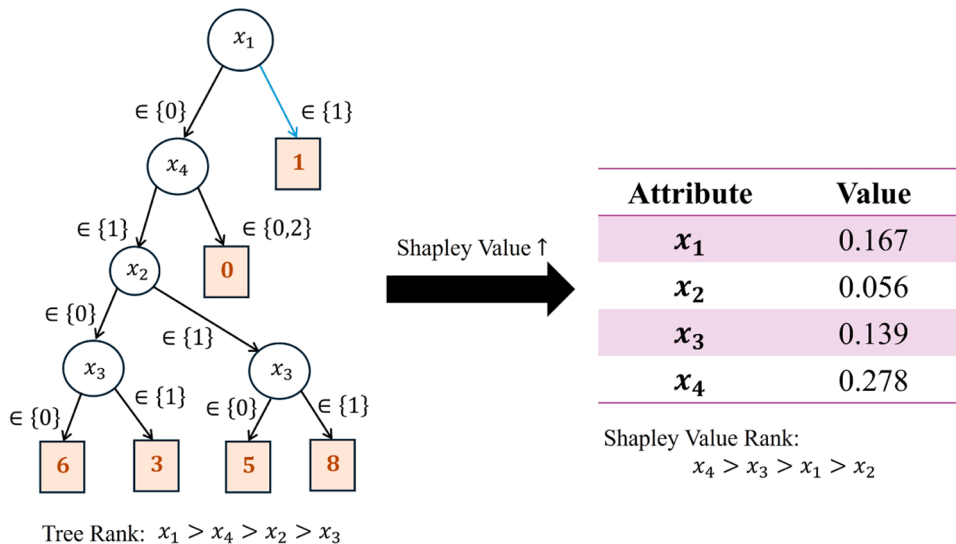


Fig. 1. Feature correlation issues in explainable machine learning.

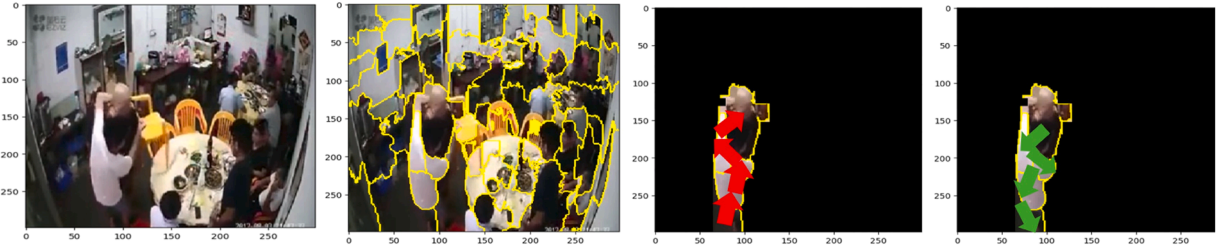


Fig. 2. Comparison of model decision path and expert decision path for violent scene identification.

In practice, the proposed RealExp is designed as an interpretable approach for surrogate models, making it capable of explaining any deep learning model. For example, in Image classification task, given an image that requires interpretation and a model, the RealExp masks certain regions of the image at a fixed ratio to generate a series of perturbed images. These perturbed images are then fed into the model to calculate changes in the model's predictions. Subsequently, RealExp fits the output of these new samples using an ensemble tree model, thereby efficiently approximating the marginal contribution of each feature. To ensure the accuracy of the fitted model results, RealExp utilizes an exponential kernel function to fine-tune the weight given to different feature combinations, where the weight is determined by the similarity between the perturbed samples and the original sample. The higher the similarity, the greater the weight assigned. The feature importance scores obtained from the fitted tree model by RealExp represent the final feature attributions, which reflect each feature's contribution to the model's prediction. The primary contributions of this paper can be summarized in three key aspects:

1. **Proposing a New Explainability Computation Method:** This paper introduces the RealExp method to address the lack of trust in existing explainability methods in high-risk scenarios. RealExp revisits the computation of Shapley Values by decoupling the average contribution into independent contributions and interaction contributions, thereby refining the understanding of feature contributions.
2. **Designing an Explainability Method for Surrogate Models:** RealExp serves as an interpretability tool for surrogate models and can explain any deep learning model. Specifically, RealExp generates perturbed images and uses an ensemble tree model to fit the output, calculating the marginal contribution of each feature to more efficiently and accurately assess the model's dependence on different features.
3. **Evaluation Method Combining Expert Annotations and Tau Coefficient:** To further evaluate the interpretability of deep learning models, this paper proposes an evaluation method that combines expert annotations with the tau coefficient. This evaluation method aims to enhance the reliability of explanations, ensuring that the model interpretations align better with domain experts' needs, especially to increase expert trust in the model in high-risk scenarios.

The remainder of the paper is organized as follows. [Section 2](#) discusses and compares previous relevant studies. [Section 3](#) details the proposed RealExp. [Section 4](#) describes the assumptions and problem formulation in detail. [Section 5](#) provides experiments and performance evaluation. The conclusions are discussed in [Section 6](#).

2. Related work

This section will review research works related to Interpretable machine learning, divided into three parts: the first part focuses on perturbation-based explainable methods; the second part covers gradient-based visualization methods for interpretability; and the third part discusses embedding-based interpretability methods.

2.1. Perturbation-based explainable methods

These methods explain models by randomly modifying parts of a single test data entry and re-inputting the modified data into the model to obtain new predictions. By comparing the changes in prediction values, the importance of features in the test data could be estimated. The core idea of these methods was to generate perturbed data and observe the sensitivity of the model's output to input changes, thereby inferring the importance of input features.

For example, (Zeiler & Fergus, 2014) slid a gray box window across images and observed changes in the model's predictions to identify the regions the model considered most important. This method intuitively revealed the areas of input data that the model focused on by occluding different regions. (Ribeiro et al., 2016) randomly masked specific regions of images or text to generate new predictions and used the slopes of linear regression models constructed from these predictions to explain the decision-making process of the model. This method was applicable not only to visual data but also to text and other data types. (Petsiuk et al., 2018) further extended this approach by generating inputs with random masks to estimate feature importance, emphasizing the broad applicability of perturbation-based methods in feature evaluation.

Additionally, (Ancona et al., 2019) proposed an approximate explanation method based on Shapley Values to assess feature importance in deep learning models. This method adopted the concept of cooperative game theory to provide a mathematically

rigorous explanation framework. (Lundberg & Lee, 2017) suggested quantifying differences between random samples and original samples by calculating the probability derivatives of Shapley Values to explore the model's decision logic. (Zafer & Khan, 2021) used clustering methods to generate perturbed samples, enhancing the interpretability of models. (Liu et al., 2024) combined interpretable tree structures with SHAP value analysis to provide more structured explanations from a tree-based model perspective. (David et al., 2024) developed the AcME method, which follows a three-stage structured process: (1) performing quantile discretization on individual features, transforming continuous variables into K equal-frequency quantile intervals; (2) implementing quantile-altered replacement operations to systematically intervene with feature values at each quantile point; and (3) constructing a feature importance function by comparing the model's predictions with the baseline output's significant deviations. (Li et al., 2023) proposed $\mathcal{E}$ LIME, which combined an improved Elastic-Net estimator to refine local explanation results by balancing global explanation distances and sparsity/feature selection in the explanations. (Tan et al., 2024) introduced GLIME with a local and unbiased sampling distribution to generate explanations with higher local fidelity.

Although these mask perturbation-based methods are applicable to any black-box model and demonstrate good flexibility in feature importance analysis, two main issues remain:

1. Limitations of the Proxy Model:

- Relying on proxy models (e.g., linear regression) for explanations introduces the inherent constraints of those models, which in turn may affect the accuracy and utility of the explanations.
- Linear Assumption Constraint: Linear models like linear regression assume a linear relationship between inputs and outputs, whereas real-world data and models often exhibit complex nonlinear behavior. This oversimplification can fail to capture the true decision-making process, thereby diminishing the reliability of the explanations.
- Feature Collinearity: When features are highly correlated, linear models struggle to isolate each feature's independent contribution, leading to potential bias in feature importance assessments and undermining the explanations' fidelity.

2. Additional Constraints on Method Applicability:

- In high-dimensional settings, the act of masking or occluding features may introduce extra noise, complicating the interpretation of results and reducing clarity.

2.2. Gradient-based visualization methods

The core purpose of these methods is to determine which input features have a significant impact on the model's output through one or more backpropagation processes. By analyzing the gradient information of the model, these methods attempt to reveal which parts of the input data are most important to the model's prediction.

For example, (Zhang et al., 2018) used the backpropagation process to trace the activation input pixels of specific nodes or classes within the network, in order to explain the model's decision-making process. This method intuitively demonstrates the correlation between input features and the network structure; (Selvaraju et al., 2017) introduced the Grad-CAM method, which generates visual heatmaps by leveraging gradient information from convolutional layers, highlighting regions crucial to the model's prediction, thus allowing users to intuitively understand the key parts the model focuses on; (Binder et al., 2016) proposed a technique that proportionally backpropagates output errors to each input feature, quantifying the impact of each feature on the prediction and representing it as feature importance; (Shrikumar et al., 2017) assigned "importance scores" to each feature by comparing the contribution of each input feature with that of a reference input, thereby further improving the accuracy of interpretations; (Smilkov et al., 2017) smoothed the gradient maps by adding random noise to the input image and calculating the average gradient, making the generated visualizations more stable and clearer; (Zahavy et al., 2016) highlighted important pixels in the input data by generating Jacobian Saliency Maps, revealing which regions have the greatest influence on the model's value or behavior predictions; (Bakchy et al., 2024) further developed gradient-based interpretability methods by proposing a lightweight CNN-based model, integrating the use of Grad-CAM to enhance usability through fast computation and efficient interpretation; (Niaz et al., 2024) proposed the Increment-CAM method, which combines Grad-CAM computation, Score-CAM computation with a modified approach, heatmap merging, and regularization to achieve model interpretability; (Li et al., 2024) proposed the Contrast-Ranking Class Activation Mapping (CR-CAM), for use with CNNs and GCNs to generate class activation maps. To address the similarity between different categories, the ranking block adopts a comparative approach to measure the distances between feature mappings in manifold space, thereby reducing the weights of the surrounding regions.

Although these methods provide intuitive and effective visual tools to explain deep learning models and reveal the internal mechanisms of model decisions, they still face three main challenges:

1. Difficulty Explaining Models with Extensive Parameters

- As the size of deep learning models grows exponentially, gradient information may become dispersed or blurred, making it hard to pinpoint the contribution of any single input feature.
- For instance, in deep convolutional neural networks, deeper layers often encounter vanishing or exploding gradients, thus weakening the precision of backpropagation-based explanations.
- Parameter redundancy and model overfitting can further hinder the interpretive process, complicating the generation of feature importance maps that hold real-world significance.

2. Difficulty Explaining Models with Extensive Parameters

- These methods are largely tailored to convolutional or graph neural networks, which perform effectively on image data. However, their direct application to other data types—such as time-series, text, or tabular data—can be problematic.
- In natural language processing tasks, for example, inputs typically take the form of discrete tokens (words or characters), making it challenging to employ gradient-based visualizations that clearly highlight the model’s attention on specific input features.
- Similarly, for structured or tabular datasets, gradient-based methods often lack the adaptability to provide meaningful explanations.

3. Difficulty Explaining Models with Extensive Parameters

- Gradient-based techniques generally yield local explanations for individual data points, helping to clarify a model’s reasoning for a specific input.
- However, these methods struggle to furnish a holistic overview of the entire dataset. For example, they cannot readily expose how the model performs on different data subsets or summarize the overarching patterns that the model learns.
- This shortcoming is especially impactful in domains like medical diagnosis or financial decision-making, where a complete understanding of the model’s global behavior is crucial.

Table 1 summarizes the comparison between previous research findings and the proposed RealExp, analyzed from four perspectives: Generality, Attribute Correlation, and Evaluation of Explanation Paths. In the table, “✓” indicates that the feature is present, “×” indicates that it is not, and “✗” indicates that the feature is present but has certain limitations.

3. The proposed RealExp

This section will introduce the specific details of the proposed RealExp, which are divided into three parts: 1. The background of the Shapley Value and a description of the existing problems. 2. The proposed RealExp and the corresponding proofs. 3. The steps for implementing RealExp in practice. The specific details are as follows:

3.1. Shapley value

The Shapley Value is a fair method used in game theory to distribute cooperative gains. It determines each participant’s contribution to the overall profit by calculating their marginal contributions in different combinations. The Shapley Value has desirable properties such as symmetry, efficiency, and gain independence, which ensure fairness in the allocation process. The Shapley Value is given by Exp. (1):

$$\begin{aligned}\Delta_s(i, \mathcal{S}) &= v(\mathcal{S} \cup \{i\}) - v(\mathcal{S}) \\ \varsigma(|\mathcal{S}|) &= \frac{|\mathcal{S}|!(|\mathcal{S}| - |\mathcal{S}| - 1)!}{|\mathcal{S}|!} \\ S_s(i) &= \sum_{\mathcal{S} \subseteq \mathcal{F} \setminus \{i\}} \varsigma(|\mathcal{S}|) \times \Delta_s(i, \mathcal{S}),\end{aligned}\quad (1)$$

the core idea of the equation is to measure the marginal contribution of feature i in different feature subsets $\mathcal{S}$ for the model output v . Herein, $\Delta_s(i, \mathcal{S})$ represents the marginal contribution of feature i to set $\mathcal{S}$. $v(\mathcal{S})$ denotes the profit generated by set $\mathcal{S}$ in model v , while $v(\mathcal{S} \cup \{i\})$ represents the profit generated after adding feature i to set $\mathcal{S}$. $\varsigma(|\mathcal{S}|)$ is a factor representing the weights of different subset combinations in the calculation. Essentially, the Shapley Value is the average of marginal contributions, which can also be represented

Table 1

The comparison between related works.

Related Works	Methods	Generality	Attribute Correlation	Paths
Zeiler & Fergus, 2014	Perturbation	✓	×	×
Ribeiro et al., 2016	Perturbation	✓	×	×
Petsiuk et al., 2018	Perturbation	✓	✗	×
Ancona et al., 2019	Perturbation	✓	✗	×
Lundberg & Lee, 2017	Perturbation	✓	✗	×
Zafer & Khan, 2021	Perturbation	×	×	×
Liu et al., 2024	Perturbation	✓	×	×
David et al., 2024	Perturbation	✓	×	×
Li et al., 2023	Perturbation	✓	×	×
Tan et al., 2024	Perturbation	✓	×	×
Zhang et al., 2018	Gradient	×	×	✗
Selvaraju et al., 2017	Gradient	×	×	✗
Binder et al., 2016	Gradient	×	×	✗
Bakchy et al., 2024	Gradient	×	×	✗
Niaz et al., 2024	Gradient	×	×	✗
Li et al., 2024	Gradient	×	✗	✗
RealEXP (Ours)	Perturbation	✓	✓	✓

by Exp. (2):

$$S_{\nu}(i) = \frac{1}{|F|!} \sum_{\pi \in \Pi} \Delta_{\nu}(i, \mathcal{S}_{\pi}), \quad (2)$$

where Π is the set of all possible feature orderings (i.e., all possible feature addition sequences), and its size is $|F|!$. $\mathcal{S}_{\pi}$ represents the set of features preceding feature i in the ordering π . Below is the specific proof:

Theorem 1: *Shapley Value averages the importance of all attributes.*

Proof 1: Let F denote the feature set. For each permutation $\pi \in \Pi$, define: $\mathcal{S}_{\pi} = \{j \in F \setminus \{i\} | \pi^{-1}(j) < \pi^{-1}(i)\}$. That is, $\mathcal{S}_{\pi}$ contains all features that appear before i in the permutation π . First, compute the number of permutations where the feature subset S serves as the predecessor set of i . For a feature subset $\mathcal{S} \in F \setminus \{i\}$, calculate the number of permutations where $\mathcal{S}_{\pi} = \mathcal{S}$. There are $|S|!$ ways to permute the subset $\mathcal{S}$ while the remaining features $F \setminus (\mathcal{S} \cup \{i\})$, can be permuted in $(|F| - |S| - 1)!$ Ways. Feature i is fixed at position $|S| + 1$. Thus, the total number of permutations where $\mathcal{S}_{\pi} = \mathcal{S}$ is given by $\text{Num}(S) = |S|! \times (|F| - |S| - 1)!$. Next, compute the proportion of all permutations in which the feature subset S occurs. Let $|F|!$ denote the total number of permutations. Then, the probability of S predecessor set of i is: $P(S) = \frac{\text{Num}(S)}{|F|!} = \frac{(|F| - |S| - 1)!}{|F|!} = \zeta(|\delta|)$. According to the definition in Exp. (1), since $P(S) = \zeta(|\mathcal{S}|)$, it follows that: $\sum_{S \subseteq (F \setminus \{i\})} P(S) \times \Delta_{\nu}(i, \mathcal{S})$, note that each $\Delta_{\nu}(i, \mathcal{S})$ appears $\text{Num}(S)$ times across all permutations. Therefore:

$$S_{\nu}(i) = \frac{1}{|F|!} \sum_{S \subseteq (F \setminus \{i\})} \text{Num}(S) \times \Delta_{\nu}(i, \mathcal{S}) = \frac{1}{|F|!} \sum_{\pi \in \Pi} \Delta_{\nu}(i, \mathcal{S}_{\pi}).$$

However, this averaging method has limitations. When features are correlated, averaging Shapley Values may underestimate the contribution of important features. This occurs because, in feature permutations, if a highly correlated feature j appears before i , the marginal contribution $\Delta_{\nu}(i, \mathcal{S})$ of i may become small or even zero, as the correlated feature j has already explained part or all the information. More specifically, two issues arise: First is Underestimation of important features. Due to feature correlation, the marginal contribution of important features is diluted during averaging. Second is incorrect feature ordering. Less significant features may receive higher Shapley Values because they lack correlated features, and their marginal contribution is not diluted during averaging. This leads to the misinterpretation illustrated in Fig. 1 of Section 1, causing inaccuracies in explanations.

Theorem 2: *Shapley Value Attenuation for Highly Correlated Features*

Proof 2: Let $F = \{x_1, x_2, \dots, x_n\}$ denote the feature set. Assume feature x_i is highly correlated with feature x_j , with correlation coefficient or similarity $s_{ij} \in [0, 1]$. In Shapley Value calculation, the marginal contribution $\Delta_{\nu}(i, \mathcal{S}_{\pi})$ depends on the feature set $\mathcal{S}_{\pi}$ preceding x_i in permutation π .

Case1: When the correlated feature $x_j \notin \mathcal{S}_{\pi}$ the marginal contribution of x_i may be large because x_j has not yet been introduced, and the information of x_i remains unexplained.

Case2: When $x_j \in \mathcal{S}_{\pi}$, due to the high correlation $s_{ij} \approx 1$, most of the explainable variance associated with x_i has already been captured by x_j . Consequently, the incremental gain $\Delta_{\nu}(i, \mathcal{S}_{\pi})$ from adding x_i may diminish significantly or even approach zero.

According to Exp. (2), since permutations are random, x_j appears before or after x_i with equal probability. Thus: when x_j appears before x_i , the number of such permutations is approximately $\frac{|F|!}{2}$, and $\Delta_{\nu}(i, \mathcal{S}_{\pi})$ is small in these cases. When x_j appears after x_i , the number of such permutations is approximately $\frac{|F|!}{2}$, and $\Delta_{\nu}(i, \mathcal{S}_{\pi})$ is large in these cases.

The Shapley Value can be divided into two parts: $S_{\nu}(i) = \frac{1}{2}E[\Delta_{\nu}(i, \mathcal{S}_{\pi}) | x_j \notin \mathcal{S}_{\pi}] + \frac{1}{2}E[\Delta_{\nu}(i, \mathcal{S}_{\pi}) | x_j \in \mathcal{S}_{\pi}]$. Since $\Delta_{\nu}(i, \mathcal{S}_{\pi})$ is smaller when $x_j \in \mathcal{S}_{\pi}$, the overall Shapley Value $S_{\nu}(i)$ is reduced. When x_j is fully correlated with x_i ($s_{ij} = 1$), $\Delta_{\nu}(i, \mathcal{S}_{\pi}) \approx 0$ when $x_j \in \mathcal{S}_{\pi}$. Assume $\Delta_{\nu}(i, \mathcal{S}_{\pi}) = \delta$ (a large positive value) when $x_j \notin \mathcal{S}_{\pi}$, and $\Delta_{\nu}(i, \mathcal{S}_{\pi}) = \epsilon$ (a value close to zero) when $x_j \in \mathcal{S}_{\pi}$. The Shapley Value becomes: $S_{\nu}(i) = \frac{1}{2}\delta + \frac{1}{2}\epsilon \approx \frac{\delta}{2}$. This is half of the Shapley Value in the absence of correlated features (δ).

3.2. RealExp

To address the aforementioned issues, this paper proposes a new calculation method called **RealExp**. This method decouples the Shapley Value into two parts and adjusts the calculation of marginal contributions based on feature similarity. Specifically, the Shapley Value ϕ_i of each feature is decomposed into two components: $\phi_i = \phi_i^{\text{independent}} + \phi_i^{\text{Margin}}$. The independent contribution $\phi_i^{\text{independent}}$ is represented as: $\phi_i^{\text{independent}} = v(\{i\}) - v(\{\})$, where $v(\{i\})$ represents the model output when only feature x_i is used, and $v(\{\})$ represents the model output when no features are used (i.e., an empty set of features). This term reflects the standalone contribution of feature x_i . The marginal contribution ϕ_i^{Margin} is expressed as: $\phi_i^{\text{Margin}} = \sum_{j \neq i} w_{ij} [v(\{i, j\}) - v(\{j\}) - \phi_i^{\text{independent}}]$, where w_{ij} reflects the similarity between features x_i and x_j defined as: $w_{ij} = \frac{1 - s_{ij}}{\sum_{k \neq i} (1 - s_{ik})}$. Herein, s_{ij} represents the similarity between features x_i and x_j , where $0 \leq s_{ij} \leq 1$. When $s_{ij} = 1$, indicates complete correlation between the features. When $s_{ij} = 0$ indicates no correlation. Thus, when $s_{ij} \approx 1$, $w_{ij} \approx 0$, reducing the impact of correlated features on the marginal contribution. The overall definition of RealExp is given by Exp. (3):

$$\phi_i = \frac{1}{|F|!} \sum_{\pi \in \Pi} \Delta_{\nu}(i, \mathcal{S}_{\pi}) \cdot Y(\mathcal{S}_{\pi}, i), \quad (3)$$

Where $Y(\mathcal{S}_\pi, i)$ is an adjustment factor defined as: $Y(\mathcal{S}_\pi, i) = \prod_{j \in \mathcal{S}_\pi} (1 - s_{ij})$. When x_i has high similarity with a feature x_j in $\mathcal{S}_\pi$, $Y(\mathcal{S}_\pi, i)$ decreases, thereby reducing the weight of x_i 's marginal contribution to the Shapley Value for that permutation. This adjustment also ensures that contributions from similar features are not double counted. By this design, the proposed RealExp

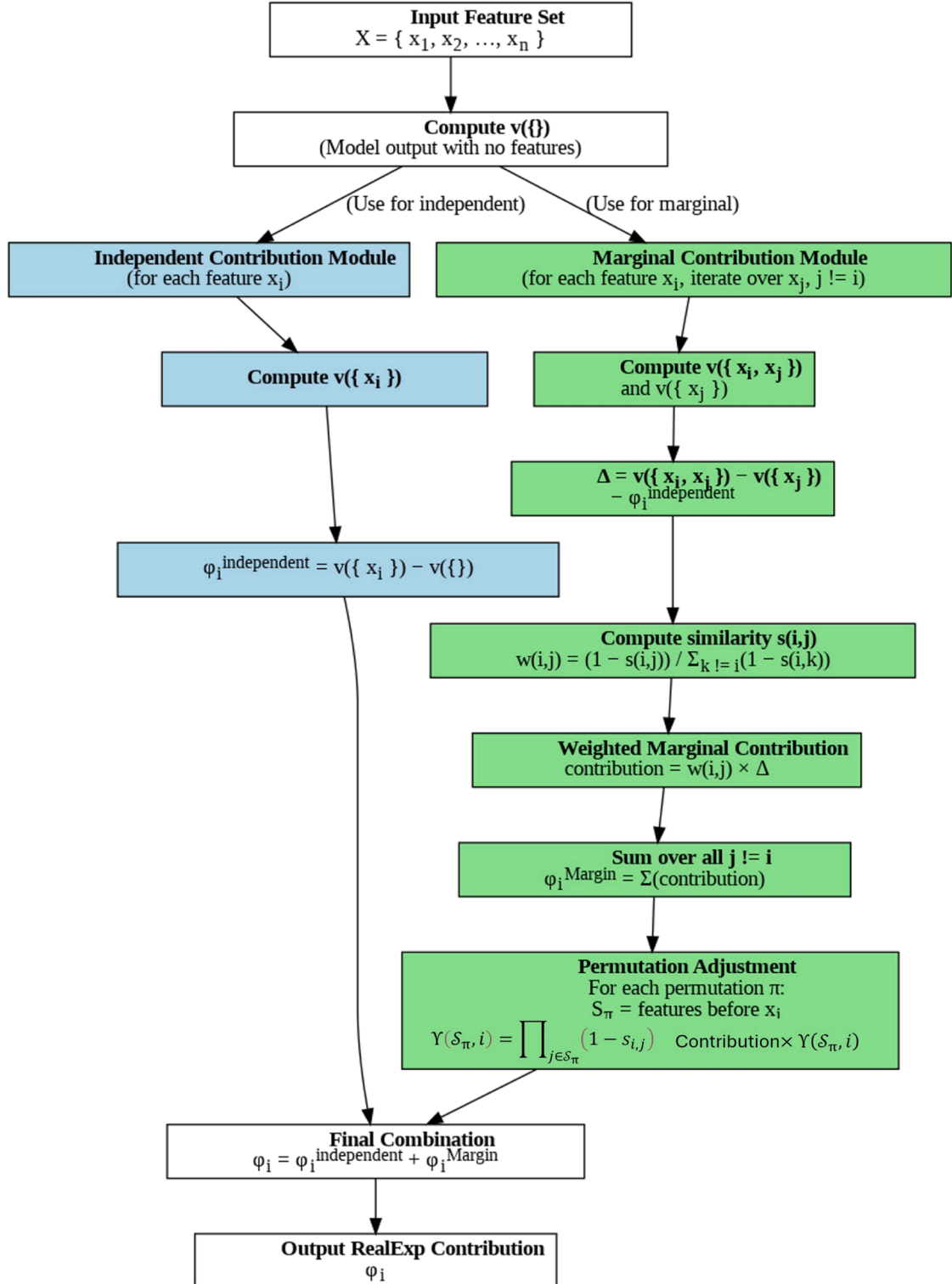


Fig. 3. RealExp workflow.

comprehensively accounts for feature correlations and avoids the averaging issues inherent in Shapley Values.

Theorem 3. *The Necessity of Similarity-Weighted Adjustment for Shapley Values.*

Proof 3: First, for each permutation $\pi \in \Pi$, the marginal contribution of feature x_i is adjusted to $\Delta_\pi(i, \mathcal{S}_\pi) \cdot Y(\mathcal{S}_\pi, i)$, when x_i has high similarity with a feature x_j in $\mathcal{S}_\pi$ ($s_{ij} \approx 1$): $Y(\mathcal{S}_\pi, i) = \prod_{j \in \mathcal{S}_\pi} (1 - s_{ij}) \approx 0$, hence, the marginal contribution of x_i for that permutation is significantly reduced or ignored. Next, according to Exp. (3): when $x_j \in \mathcal{S}_\pi$ and $s_{ij} \approx 1$, $Y(\mathcal{S}_\pi, i) \approx 0$, and the corresponding marginal contribution is ignored, avoiding dilution caused by correlated features. When $x_j \notin \mathcal{S}_\pi$ and s_{ij} is small, $Y(\mathcal{S}_\pi, i)$ is relatively large, ensuring that the marginal contribution of x_i is fully considered.

Fig. 3 presents the workflow of RealExp. The method begins with the input feature set $X = \{x_1, x_2, \dots, x_n\}$ and computes the baseline model output $v(\{\})$, which represents the model prediction when no features are used. This baseline output serves as a reference point for subsequent calculations.

Next, the RealExp method computes the independent contribution of each feature x_i , which quantifies its effect on the model output without considering interactions with other features. Specifically, this contribution is defined as $\phi_i^{\text{independent}} = v(\{i\}) - v(\{\})$. This step reflects the standalone contribution of a single feature in the absence of any interactions.

After computing independent contribution, the method proceeds to compute the marginal contribution, which evaluates the impact of feature x_i when combined with other features. For each feature pair (x_i, x_j) , the combined model output is calculated as $\Delta = v(\{i, j\}) - v(\{j\}) - \phi_j^{\text{independent}}$. This incremental value represents the additional contribution of x_i given that x_j is already present. However, since features may be correlated, directly summing all marginal contributions could lead to an overestimation of some feature contributions. To address this, the RealExp method introduces similarity-weighted importance $w(x_i, x_j)$, defined as $w_{ij} = \frac{1 - s_{ij}}{\sum_{k \neq i} (1 - s_{ik})}$. The weighted marginal contribution is computed as $\text{contribution} = w(x_i, x_j) \times \Delta$ and summed over all $j \neq i$: $\phi_i^{\text{Margin}} = \sum (\text{contribution})$. This process ensures that similar features do not disproportionately influence the final result, thereby mitigating the averaging issues in traditional Shapley Value computation. Furthermore, the RealExp method incorporates permutation adjustment to ensure fairness in feature contribution computation. For each permutation π an adjustment factor is defined as $Y(\mathcal{S}_\pi, i) = \prod_{j \in \mathcal{S}_\pi} (1 - s_{ij})$. This adjustment factor helps reduce overcounting contributions from highly similar features, making the final computation more robust. Finally, the RealExp method combines independent and marginal contributions to obtain the final feature contribution $\phi_i = \phi_i^{\text{independent}} + \phi_i^{\text{Margin}}$.

3.3. In practice

In practice, this paper applies RealExp to perturbation-based surrogate models to achieve interpretability, with the advantage of being able to explain any model. Unlike previous works (Liu et al., 2024; Ribeiro et al., 2016; Tan et al., 2024), which utilize random

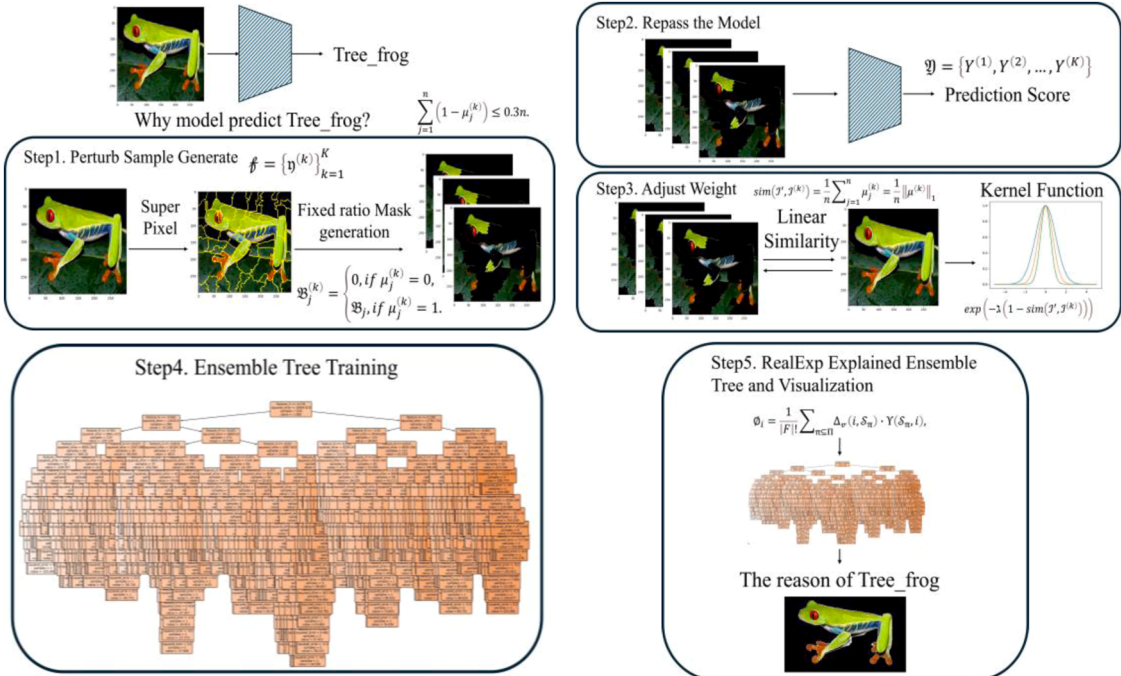


Fig. 4. RealExp's specific workflow in practice.

sampling and Monte Carlo sampling, this study adopts a fixed ratio to generate perturbation samples, effectively reducing instability. Furthermore, in constructing the surrogate model, this paper designs an ensemble tree as the surrogate model instead of traditional linear regression or its variants, as tree models inherently capture nonlinear relationships between features, avoiding potential collinearity issues associated with traditional linear models. Fig. 4 aims to take an image classification task as an example, RealExp explains why the black-box model predicts the image as a Tree Frog through five steps. First, perturbation samples are generated by segmenting the image into superpixels and applying a fixed-ratio mask. Next, these perturbed images are re-fed into the model to obtain prediction scores. Then, the similarity between the perturbation samples and the original image is computed using a linear similarity function and a kernel function. Based on these weighted perturbation samples, an ensemble tree model is trained. Finally, RealExp generates explanations and visualizations to highlight the key regions that contribute to the classification decision. The detailed process is as follows:

3.3.1. Perturb sample generation

This step is used to generate the input data for building the proxy model later. Let $\mathcal{I}$ denote the image to be interpreted and let $\mathbb{W}$ represent the watershed image segmentation algorithm. Upon segmentation, the result obtains the image $\mathcal{I}'$ such that $\mathcal{I}' = \mathbb{W}(\mathcal{I})$. The segmented image $\mathcal{I}'$ can be represented as a set of segmented blocks, denoted by $\{\mathcal{B}_j\}_{j=1}^n$, where each block $\mathcal{B}_j$ for $j \in [1, n]$ signifies the index of the block.

Let $\mathcal{M} = \{\eta^{(k)}\}_{k=1}^K$ denote a set of masks to be applied to $\mathcal{I}'$. Each mask $\eta^{(k)}$, where $k \in [1, K]$, contains a set of Boolean variables $\{\mu_j^{(k)}\}_{j=1}^n$. Each $\mu_j^{(k)}$ corresponds to block $\mathcal{B}_j$ and indicates whether it is kept or masked (0 for masked, 1 for kept). For each specific mask application, which can be define: $\mathcal{B}_j^{(k)} = \begin{cases} 0, & \text{if } \mu_j^{(k)} = 0, \\ \mathcal{B}_j, & \text{if } \mu_j^{(k)} = 1. \end{cases}$

Each set $\{\mu_j^{(k)}\}$ ensures, according to a heuristic rule, that at most 30% of the blocks are covered. This is formalized by the inequality: $\sum_{j=1}^n (1 - \mu_j^{(k)}) \leq 0.3n$. Applying each mask $\eta^{(k)}$ to $\mathcal{I}'$ yields a series of new images with selectively retained blocks: $\mathcal{I}^{(k)} = \eta^{(k)}(\mathcal{I}') = \cup_{j=1}^n \mathcal{B}_j^{(k)}$. Collectively, these form the final set of perturbed images: $\mathcal{I}'' = \{\mathcal{I}^{(1)}, \mathcal{I}^{(2)}, \dots, \mathcal{I}^{(K)}\}$, where each image $\mathcal{I}^{(k)}$ represents the outcome of a specific block retention pattern determined by the mask $\eta^{(k)}$.

Theorem 4. Fixed perturbation ratio has stronger stability.

Proof 4: Let the original image, after segmentation, be divided into n blocks, denotes as $\{B_1, B_2, \dots, B_n\}$. Let $\text{Var}[f(\mathcal{I}^{(k)})]$ denote the variance of the estimated value. For fixed proportion generation, the number of masked blocks $\eta^{(k)}$ is fixed and denoted as $m = \alpha n$, where $\alpha \leq 0.3$. Thus, the probability of masking each block is the same and satisfies: $P(\mu_j^{(k)} = 0) = \frac{m}{n} = \alpha$, $P(\mu_j^{(k)} = 1) = 1 - \alpha$. To simplify the calculation, assume that the function $f(\mathcal{I}^{(k)})$ is linear, given by: $f(\mathcal{I}^{(k)}) = f_0 + \sum_{j=1}^n c_j \mu_j^{(k)}$, where f_0 is a constant and c_j is the contribution of block B_j . For fixed proportion generation, the variance is given by: $\text{Var}_{\text{fixed}} = \alpha(1 - \alpha) \left(\left(1 + \frac{1}{n-1}\right) \sum_{i=1}^n c_i^2 - \frac{1}{n-1} \left(\sum_{i=1}^n c_i\right)^2 \right)$.

For Monte Carlo sampling, the masking probability for each block is given by $P(\mu_j^{(k)} = 0) = q$, with a variance of the masking probability $\sigma_q^2 > 0$, where $q = \alpha$. The variance is expressed as: $\text{Var}_{\text{Monte Carlo}} = (q(1-q) + \sigma_q^2) \sum_{i=1}^n c_i^2$. In random sampling, where each block is independently masked with probability α the variance is $\text{Var}_{\text{Random}} = \alpha(1 - \alpha) \sum_{i=1}^n c_i^2$. Comparison of Variances, it can be observed that: $\text{Var}_{\text{fixed}} < \text{Var}_{\text{Random}} < \text{Var}_{\text{Monte Carlo}}$. This inequality demonstrates that fixed proportion generation has the lowest variance, providing greater stability compared to both random sampling and Monte Carlo sampling.

3.3.2. Repass the model

This step is used to generate the label for building the proxy model later. Let ϕ denote the model to be explained. The generated perturbed images are represented as $\mathcal{I}'' = \{\mathcal{I}_n^{(1)}, \mathcal{I}_n^{(2)}, \dots, \mathcal{I}_n^{(K)}\}$, where each $\mathcal{I}^{(k)}$ is passed through the model ϕ for prediction. Let $\mathcal{Y} = \{Y^{(1)}, Y^{(2)}, \dots, Y^{(K)}\}$ denote the ϕ for all perturbed images.

3.3.3. Adjust weight

This step is used to generate and adjust weight for input data in proxy model. The $\mathcal{I}'$ can be regarded as a vector of one, and the perturbed image $\mathcal{I}^{(k)}$ is represented by a combination of zeros and ones (as determined by the mask $\mu^{(k)}$, the similarity measure calculates the proportion of retained blocks. The similarity can be expressed as $\text{sim}(\mathcal{I}', \mathcal{I}^{(k)}) = \frac{1}{n} \sum_{j=1}^n \mu_j^{(k)} = \frac{1}{n} \|\mu^{(k)}\|_1$, where $\mu_j^{(k)}$ represents the j -th element in the mask $\mu^{(k)}$, and n is the total number of blocks. This calculated similarity varies linearly with the number of retained blocks, which does not align with conventional nonlinear similarity measures.

To introduce nonlinearity, the similarity measure can be adjusted using an exponential function, as shown in Exp. (4):

$$\text{adj}_k = \exp(-\lambda(1 - \text{sim}(\mathcal{I}', \mathcal{I}^{(k)}))), \quad (4)$$

Where $\lambda > 0$ is a hyperparameter representing the intensity of the adjustment. The term $1 - \text{sim}(\mathcal{J}', \mathcal{J}^{(k)})$ corresponds to the normalized Hamming distance between $\mathcal{J}'$ and $\mathcal{J}^{(k)}$. This adjustment ensures that when the proportion of masked blocks is small, the adjusted similarity adj_k remains high, and when the proportion is large, adj_k becomes lower. The adjusted similarity adj_k is applied multiplicatively to each block of the perturbed image $\mathcal{J}^{(k)}$. The updated block $B_j^{(k)'} is calculated as $B_j^{(k)'} = \text{adj}_k \cdot B_j^{(k)}$, $\forall j \in \{1, 2, \dots, n\}$, where $B_j^{(k)}$ represents the block in the adjusted perturbed image, and $B_j^{(k)}$ represents the corresponding block in the original perturbed image before adjustment.$

3.3.4. Ensemble tree training

This step is used to train proxy models. Till now, building upon the established perturbed sample set $\mathcal{J}'' = \{\mathcal{J}_n^{(1)}, \mathcal{J}_n^{(2)}, \dots, \mathcal{J}_n^{(K)}\}$, where each $\mathcal{J}_n^{(k)}$ represents the k -th perturbed sample, we associate a corresponding target value set $\mathcal{Y} = \{Y^{(1)}, Y^{(2)}, \dots, Y^{(K)}\}$. Each perturbed sample $\mathcal{J}_n^{(k)}$ comprises n processed blocks $\{B_{1,k}', B_{2,k}', \dots, B_{n,k}'\}$, which serve as the feature set $\mathbf{x}^{(k)} = \{B_{j,k}'\}_{j=1}^n$.

An ensemble of regression tree models $\{\mathcal{T}_s\}_{s=1}^S$ is defined, where $s = 1, 2, \dots, S$ and S denotes the total number of trees in the ensemble.

To construct the ensemble, bootstrap sampling is performed on each feature set $\mathbf{x}^{(k)}$ to generate new training datasets $\mathcal{D}_m$. This process involves randomly drawing N samples with replacement from the original N samples to form each training set $\mathcal{D}_m$, which includes both input features and corresponding target outputs.

For each dataset $\mathcal{D}_m$, a regression tree $\mathcal{T}_s$ is recursively built. At each decision node within the tree, the objective is to find the optimal split by selecting the best split block $B_{j,k}$ and the optimal split point L_{best} that minimize the weighted total variance after the split. This optimization problem is expressed as Exp. (5):

$$(B_{j,k}, s_{\text{best}}) = \underset{B_{j,k}}{\operatorname{argmin}} [w_{\text{left}} \times \text{Var}(\mathcal{D}_{\text{left}}) + w_{\text{right}} \times \text{Var}(\mathcal{D}_{\text{right}})], \quad (5)$$

Where, $B_{j,k}$ represents the j -th processed block in the k -th perturbed sample. s denote a potential split point. $w_{\text{left}} = \frac{|\mathcal{D}_{\text{left}}|}{|\mathcal{D}|}$ and $w_{\text{right}} = \frac{|\mathcal{D}_{\text{right}}|}{|\mathcal{D}|}$ are the weights corresponding to the proportions of data in the left and right child nodes after the split. $\text{Var}(\cdot)$ denotes the variance of the target values within a dataset. $\mathcal{D}_{\text{left}}$ and $\mathcal{D}_{\text{right}}$ are the subsets of $\mathcal{D}$ resulting from the split at s .

For each leaf node in the tree $\mathcal{T}_s$, the prediction value $\hat{y}_{\text{leaf}}$ is calculated as the meaning of all target values y_i associated with the samples in that node $\hat{y}_{\text{leaf}} = \frac{1}{|\mathcal{D}_{\text{leaf}}|} \sum_{i \in \mathcal{D}_{\text{leaf}}} y_i$.

3.3.5. RealExp explained ensemble tree and visualization

This step aims to calculate the importance of the ensemble tree model using the RealExp method and output the results through visualization techniques. Specifically, each block $\mathcal{B}_i$ in the segmented set $\{\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_n\}$ is treated as a feature x_i , and the RealExp method is applied for calculation, as described in Exp. (3). After the computation, the importance scores are represented as $\{\phi_1, \phi_2, \dots, \phi_n\}$. These scores are then sorted in descending order, resulting in $\phi_{i1} \geq \phi_{i2} \geq \dots \geq \phi_{in}$, where ϕ_{ik} denotes the Shapley Value of the k -th block in the sorted sequence. Let $M: \mathbb{R} \rightarrow \mathcal{C}$ denote a mapping function which maps numerical importance scores to visual attributes, thereby intuitively presenting the importance of each block.

4. Experimental Setup and evaluation

This section introduces the Experimental Setup and Evaluation of this paper. The evaluation incorporates Kendall's τ coefficient to further analyze the feature importance paths, combining the calculation of feature importance paths and accuracy to provide deeper insights into the model's decision-making process. This design aims to help users understand whether the model's reasoning aligns with the fundamental understanding of experts and to assess the model's consistency and accuracy in feature selection and classification decisions.

For image classification tasks, let $\mathcal{M}$ denote the model to be explained, and the input image be $\mathcal{I} \in \mathbb{R}^{H \times W \times C}$, where H, W and C represent the height, width, and channel number of the image, respectively. Let $\mathcal{Q}$ denote the image segmentation algorithm, which divides $\mathcal{I}$ into n superpixel regions, resulting in the segmented set: $\mathcal{S} = \cup_{i=1}^n S_i$. Where S_i represent the i -th region, satisfying $S_i \cap S_j = \emptyset$ when $i \neq j$.

Let $\mathcal{U} = \{S_{u1}, S_{u2}, \dots, S_{um}\}$ denote the set of superpixels identified by experts as significantly contributing to the classification result, where $\mathcal{U} \subseteq \{S_1, S_2, \dots, S_n\}$, and $m \leq n$. Through an explainable algorithm, the model $\mathcal{M}$ assigns an importance score $f(S_i)$ to each superpixel S_i . Let $\mathcal{K} = \{S_{k1}, S_{k2}, \dots, S_{km}\}$ represent the superpixels sorted in descending order of importance, where $\mathcal{K} \subseteq \{S_1, S_2, \dots, S_n\}$.

To quantify the consistency between the model's ranking and the expert's ranking, two evaluation metrics are introduced:

1. Matching Count Calculation ($\mathcal{M}$ -score): Measures how well the model identifies the important features highlighted by experts.
2. Ranking Consistency Calculation (M -score): Evaluates the alignment of ranking order between the model and expert annotations.

Step 1: Matching Count Calculation ($\mathcal{M}$ -score): To determine the accuracy of the model in identifying expert-labeled features,

we calculate in Exp. (6):

$$|H| = \mathcal{H} \cap \mathcal{K}, \quad (6)$$

where $|H|$ represents the number of correctly identified superpixels. The $\mathcal{H}$ -score is then computed in Exp. (7):

$$\mathcal{H} - \text{score} = \frac{|H|}{m}, \quad (7)$$

where m is the total number of expert-labeled superpixels.

Step 2: Matching Count Calculation (M -score):

To evaluate whether the ranking order of important features is consistent between expert annotations and the model's output, we define concordant and discordant pairs among elements in H :

Concordant Pair (C): A pair (i, j) satisfies $i < j$ in expert ranking, and their order is preserved in the model ranking, i.e., $\text{rank}_K(i) < \text{rank}_K(j)$.

Discordant Pair (D): A pair (i, j) satisfies $i < j$ in expert ranking, but their order is reversed in the model ranking, i.e., $\text{rank}_K(i) > \text{rank}_K(j)$.

The total number of Label pairs is $\binom{|H|}{2} = \frac{|H|(|H|-1)}{2}$, the M -score is then calculated using Kendall's τ coefficient in Exp. (8):

$$M - \text{score} = \frac{2(C - D)}{|H|(|H| - 1)}, \quad (8)$$

Where C represents the number of concordant pairs, and D represents the number of discordant pairs.

Fig. 5 shows the Example of Experimental Setup and Evaluation in Image Classification. as shown in the figure. The process follows these steps:

Experts annotate image regions contributing to the classification decision. The expert identified important superpixels: {31, 37, 48, 57, 53, 47, 38, 33, 27, 20, 17, 18, 26, 36, 49, 10, 30, 44}. The explainability model predicts a set of important super pixels {37, 44, 38, 27, 33, 17, 10, 18, 30, 36, 43, 50, 54, 58, 31, 26, 20, 19}

Step1: Compute $\mathcal{H}$ -score: The overlapping superpixels between expert annotations and the model's predictions are: $H = \{31, 37, 38, 33, 27, 36, 10, 30, 44\}$, the total number of matched superpixels is 13. The $\mathcal{H}$ -score is computed as: $\mathcal{H} - \text{score} = \frac{13}{18} \approx 0.722$.

Step2: Compute M -score: the Expert Rank Order is [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], the model rank order is [14, 0, 2, 4, 3, 1, 6, 5, 7, 15, 9, 6, 8, 1]. Computing the total number of pairs $\binom{13}{2} = 78$. The Concordant Pairs (C) is 45, the Discordant Pairs (D) is 33. The M -score is calculated as $M - \text{score} = \frac{2(45-33)}{13(13-1)} \approx 0.1538$.

For text sentiment analysis, given a model $\mathcal{W}$ that needs to be explained and an input text $\mathcal{T}$, the output is the set of words in the text

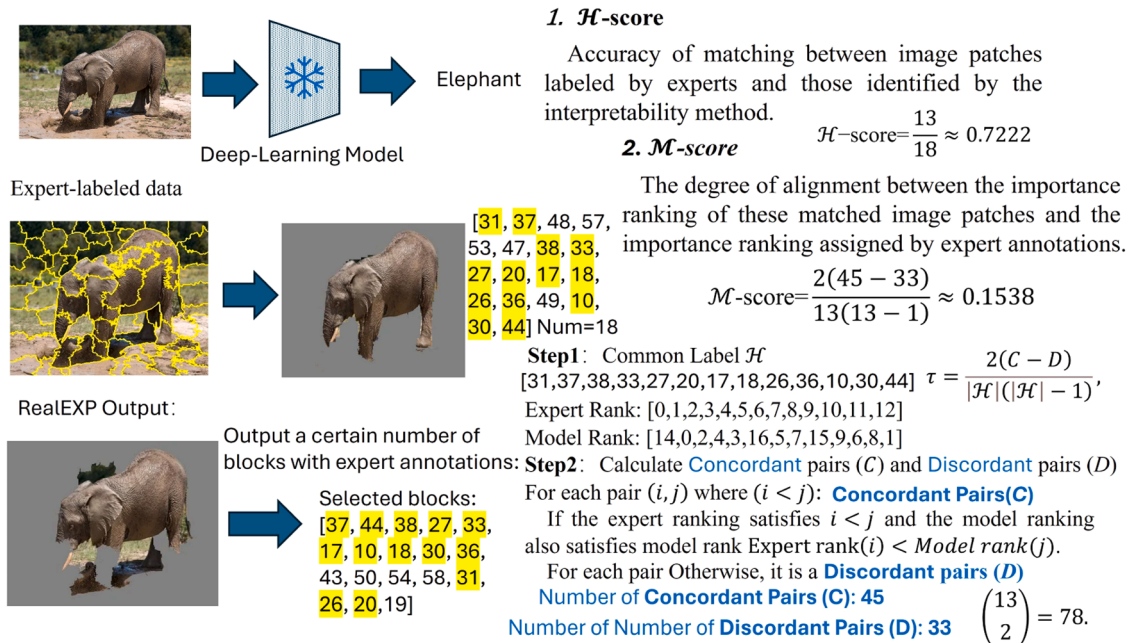


Fig. 5. An example of experimental setup and evaluation in image classification.

on which the model bases its decisions. Taking Chinese text as an example, let $\mathfrak{S}$ denote the Jieba segmentation algorithm, and let the segmented sentence be represented as $\mathfrak{T} = \{T_1, T_2, \dots, T_n\}$, where T_i denotes the i -th word or token after segmentation, and n is the total number of words in the text.

Apply masking perturbation techniques to $\mathfrak{T}$ to generate a set of perturbed texts. Let the set of perturbed samples be denoted as $\mathfrak{T}' = \{\mathfrak{T}'^{(1)}, \mathfrak{T}'^{(2)}, \dots, \mathfrak{T}'^{(K)}\}$, where K represents the number of masking perturbations applied. Each perturbed sample $\mathfrak{T}'^{(k)}$ is a subset of the original text $\mathfrak{T}$, constructed by masking certain words. Specifically, each $\mathfrak{T}'^{(k)}$ is defined as $\mathfrak{T}'^{(k)} = \{T_{j_1}^{(k)}, T_{j_2}^{(k)}, \dots, T_{j_p}^{(k)}\}$, with $0.3n < p \leq n$, where $\{j_1^{(k)}, j_2^{(k)}, \dots, j_p^{(k)}\} \subseteq \{1, 2, \dots, n\}$ indicates the indices of the unmasked words in the k -th perturbation.

For each perturbed text $\mathfrak{T}'^{(k)}$, the model $\mathcal{M}$ is used to obtain the prediction score $Y^{(k)}$. The set of prediction scores is represented as $\mathcal{Y} = \{Y^{(1)}, Y^{(2)}, \dots, Y^{(K)}\}$. The similarity between each perturbed sample $\mathfrak{T}'^{(k)}$ and the original text $\mathfrak{T}$ is calculated using a similarity function $\text{sim}(\mathfrak{T}, \mathfrak{T}'^{(k)}) = \frac{1}{n} |\mathfrak{T}'^{(k)}| = \frac{p}{n}$, where $|\mathfrak{T}'^{(k)}| = p$ is the number of unmasked words in $\mathfrak{T}'^{(k)}$. To adjust for the linearity of the similarity measure, an exponential adjustment is applied in $w^{(k)} = \exp(-\lambda(1 - \text{sim}(\mathfrak{T}, \mathfrak{T}'^{(k)})))$.

Construct the input feature set $\mathcal{X}$ for the proxy model by weighting the perturbed samples $\mathcal{X} = \{\mathfrak{T}'^{(1)} \times w^{(1)}, \mathfrak{T}'^{(2)} \times w^{(2)}, \dots, \mathfrak{T}'^{(K)} \times w^{(K)}\}$, where $\mathfrak{T}'^{(K)} \times w^{(K)}$ denotes that each word in $\mathfrak{T}'^{(k)}$ is associated with the weight $w^{(k)}$. The corresponding target labels are the prediction scores obtained from the model $\mathcal{M}$: $\mathcal{Y} = \{Y^{(1)}, Y^{(2)}, \dots, Y^{(K)}\}$. By training the interpretable model on the weighted input-output pairs $(\mathcal{X}, \mathcal{Y})$, and using Exp.(3) calculated, the importance scores for each word T_i in the original text $\mathfrak{T}$ can be derived. Let the set of importance scores be denoted as $P = \{\varnothing_1, \varnothing_2, \dots, \varnothing_n\}$, where $\varnothing_i$ represents the importance of word T_i as determined by the model. Let $\mathcal{U} = \{T_{u1}, T_{u2}, \dots, T_{um}\}$ denote the set of words selected by experts as being most indicative of the classification, where $\mathcal{U} \subseteq \{T_1, T_2, \dots, T_n\}$, and $m \leq n$.

Based on the computed importance scores P , the top m words are selected and ordered to form the set $\mathcal{H} = \{T_{k1}, T_{k2}, \dots, T_{km}\}$, where the words are sorted such that $\varphi_{k1} \geq \varphi_{k2} \geq \dots \geq \varphi_{km}$. Calculating Matching Score and Kendall's τ Coefficient by $\mathcal{H}$ -score and $\mathbf{M}$ -score. Here is an example:

Consider the input text:

$\mathfrak{T} =$ "his movie is very interesting, and the performances by the actors are impressive."

The model to be explained is a Support Vector Machine (SVM) classifier that predicts this sentence as expressing a positive sentiment. Using a standard English tokenizer $\mathfrak{S}$, the sentence is segmented into:

$\mathfrak{T} = \{T_1 = \text{"This"}, T_2 = \text{"moive"}, T_3 = \text{"is"}, T_4 = \text{"very"}, T_5 = \text{"interesting"}, T_6 = \text{"and"}, T_7 = \text{"the"}, T_8 = \text{"performance"}, T_9 = \text{"by"}, T_{10} = \text{"the"}, T_{11} = \text{"actors"}, T_{12} = \text{"are"}, T_{13} = \text{"impressive"}\}$.

Let the number of perturbed samples be $K = 3$. The perturbed texts are:

$$\mathfrak{T}'^{(1)} : \{T_1, T_2, T_3, T_4\} \cup \{T_6, T_7, T_8, T_9, T_{10}, T_{11}, T_{12}, T_{13}\},$$

$$\mathfrak{T}'^{(2)} : \{T_1, T_2, T_3, T_4, T_5\} \cup \{T_6, T_7, T_8, T_9, T_{10}, T_{11}, T_{12}\},$$

$$\mathfrak{T}'^{(3)} : \{T_1, T_2, T_3, T_4, T_5\} \cup \{T_6, T_7, T_{10}, T_{11}, T_{12}\}.$$

The prediction scores obtained from the SVM are $\mathcal{Y} = \{Y^{(1)} = 0.3, Y^{(2)} = 0.7, \dots, Y^{(3)} = 0.8\}$. The similarities between each perturbed sample and the original text are $\text{sim}(\mathfrak{T}, \mathfrak{T}'^{(1)}) = 0.9$, $\text{sim}(\mathfrak{T}, \mathfrak{T}'^{(2)}) = 0.8$ and $\text{sim}(\mathfrak{T}, \mathfrak{T}'^{(3)}) = 1$. Applying the exponential adjustment, the adjusted weights are $w^{(1)} = \exp(-\lambda(1 - 0.9))$, $w^{(2)} = \exp(-\lambda(1 - 0.8))$, $w^{(3)} = \exp(-\lambda(1 - 1)) = 1$.

The inputs to the proxy model are $\mathcal{X} = \{\mathfrak{T}'^{(1)} \times w^{(1)}, \mathfrak{T}'^{(2)} \times w^{(2)}, \mathfrak{T}'^{(3)} \times w^{(3)}\}$ with $\mathcal{Y}$. After training, RealExp calculated the proxy model's the importance scores $P = \{\varphi_1, \varphi_2, \dots, \varphi_{13}\}$ for each word T_i in $\mathfrak{T}$ are obtained. Experts consider the positive sentiment words in the sentence to be $\mathcal{U} = \{T_5, T_{13}\}$, and they rank them as: $T_5 > T_{13}$. Based on the importance scores P , the model's top two words are $\mathcal{H} = \{T_5, T_{13}\}$, with importance scores $\varphi_5 > \varphi_{13}$. The matching words are $H = 2$. The $\mathcal{H}$ -score is 1, the $\mathbf{M}$ -score is 1.

5. Experiment

This section will introduce the performance of the proposed RealExp interpretable method. All the information for the experiment is on the following the URL¹. The experiment is divided into four parts: image classification, video Classification, Pneumonia Image Classification and text sentiment analysis.

5.1. Image classification

This section introduces the explainable experiments of image classification, which are divided into three parts: dataset and evaluation metrics, parameter settings, and model evaluation.

¹ <https://github.com/1846659840/The-Research-on-a-new-LIME-based-interpretable-machine-learning-and-its-evaluation-method-imagepart>.

5.1.1. Dataset and evaluation metrics

This paper selects two datasets for study: Animal-10² and Animal-151³. The Animal-10 dataset contains 10 different types of animals, while the Animal-151 dataset includes 15 mainstream animal categories, each comprising 10 different rare species under each category. For both datasets, 1,000 images were selected, resulting in a total of 2,000 images for annotation and model testing.

To obtain expert annotations, 2,000 images were manually labeled, as shown in Fig. 6. Specifically, the front-end program Gardio was used to build a labeling platform, and a public link was distributed to volunteers. On the left side of the interface, an image from the dataset is displayed, while the system updates the segmentation results in the middle in real time. Experts then use their mouse to select the areas they believe contain dogs, ranking these areas from most important to least important. To ensure rigor, the annotations from three experts were cross-compared, and the most common choice was adopted as the final label.

In terms of model selection, this study chose five commonly used pre-trained models: MobileNet (Andrew et al., 2017), ResNet (He et al., 2016), DenseNet (Huang et al., 2017), and Vision Transformer (ViT) (Dosovitskiy et al., 2021).

For evaluation metrics, the study focuses on four aspects: the stability of explanations, the interpretability of the model, consistency with expert understanding, and the interpretability of the decision-making path.

To evaluate the stability of explanations, in terms of how well the explained regions match the original salient parts of the image and are consistent with previous works (Li et al., 2023; David et al., 2024; Liu et al., 2024; Tan et al., 2024), this study uses the Jaccard coefficient as one metric. This paper repeats the experiment five times under slight variations. For the k -th run ($k = 1, 2, \dots, 5$), the method produces a set of identified features, such as salient image regions or important feature indices in tabular data, denoted by A_k .

For two distinct runs k and l , where $k \neq l$, the overlap between the result sets A_k and A_l is quantified using the Jaccard coefficient in Exp. (9):

$$J(A_k, A_l) = \frac{|A_k \cap A_l|}{|A_k \cup A_l|}, \quad (9)$$

Where $|A_k \cap A_l|$ represents the number of shared elements in the two sets, while $|A_k \cup A_l|$ indicates the total number of unique elements across both sets. A larger Jaccard value signifies greater consistency between the explanations generated in the two runs.

Finally, a single overall stability measure across all 5 runs is obtained by calculating the average Jaccard coefficient for every pair of runs $\binom{5}{2} = 10$ pairs. If the explanation method consistently highlights the same or similar features in all runs, the average Jaccard value will be higher, reflecting a higher degree of explanation stability.

To ensure the interpretability of the model, and consistent with previous works (Li et al., 2023; Liu et al., 2024; Tan et al., 2024), this study uses the R^2 coefficient as the evaluation metric. The specific formula is $R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$, where y_i denotes the actual value output for the i -th sample (e.g., the ground-truth prediction from the original model), $\hat{y}_i$ denotes the predicted value (e.g. the reconstructed or estimated output produced by the explanation method), and $\bar{y}$ denotes the mean of the actual model outputs y_i across all samples.

A higher R^2 value indicates that the explanation method's predicted outputs $\hat{y}_i$ align more closely with the model's actual outputs y_i , signifying greater interpretability in capturing how the model truly behaves.

For consistency with expert understanding, using $\mathcal{H}$ -score is used to calculate the accuracy of matching between image patches labeled by experts and those identified by the interpretability method, as detailed in Section 4. The M -score measures the degree of alignment between the importance ranking of these matched image patches and the importance ranking assigned by expert annotations, as detailed in Section 4.

In practice, these scores are computed by traversing the entire dataset and performing a weighted averaging to evaluate overall model performance. While H-score and Kendall's τ are designed to align with expert decisions on specific tasks or samples, their computation inherently reflects global model behavior rather than being confined to individual instances. Additionally, this ensures that feature importance consistency is preserved across the dataset, providing a robust and comprehensive evaluation of the model's interpretability.

5.1.2. Parameter settings

This section introduces the hyperparameter settings. First, for the perturbation samples $\mathcal{S}'' = \{\mathcal{S}^{(1)}, \mathcal{S}^{(2)}, \dots, \mathcal{S}^{(K)}\}$, the number K is fixed at 500, and the hyperparameter λ in Exp. (4) is set to 0.25. This is consistent with previous works (Li et al., 2023; Liu et al., 2024; Tan et al., 2024). Second, for the pre-trained model weights, this study uniformly uses weights trained on ImageNet. For the construction of the tree model, the number of $\{\mathcal{S}_s\}_{s=1}^S$ is also fixed at 500.

5.1.3. Model evaluation

This section introduces the Model Evaluation in Animal-10 and Animal-151. Tables 2 and 3 present a comparison of the proposed RealEXP method with other state-of-the-art approaches (including Zeiler & Fergus, 2014, LIME (Ribeiro et al., 2016), Kernel-SHAP (Lundberg & Lee, 2017), GLIME (Li et al., 2023), HDTWM (Liu et al., 2024), AcME (David et al., 2024), and G-LIME (Tan et al.,

² <https://www.kaggle.com/datasets/alessiocorrado99/animals10>

³ <https://www.kaggle.com/datasets/sharansmenon/animals141>

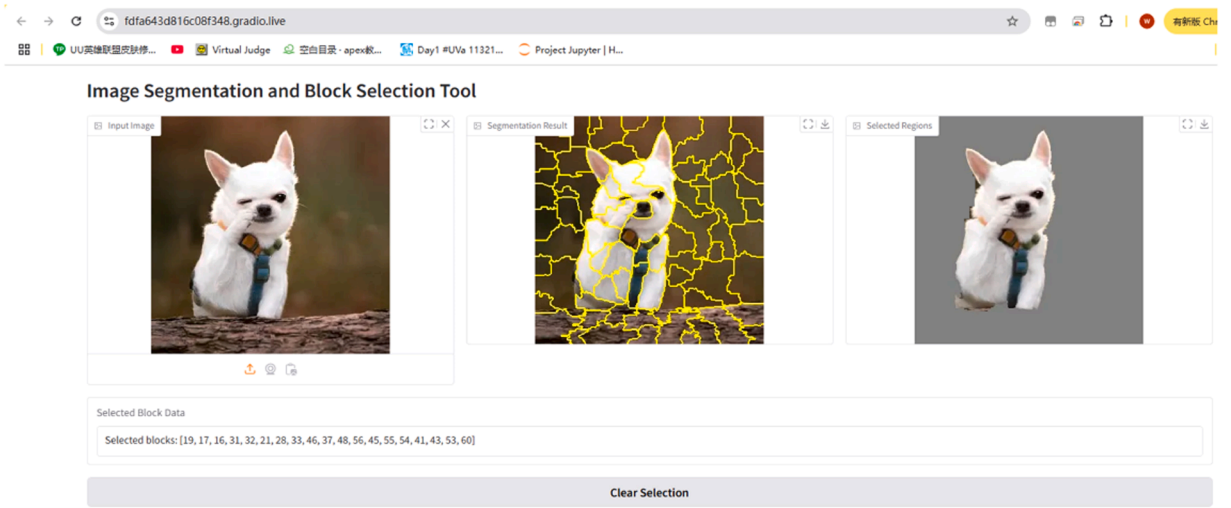


Fig. 6. Acquisition of expert-labeled data for image classification.

2024)) on the Animal-10 and Animal-151 benchmark datasets. Overall, the proposed RealEXP demonstrates superior stability, enhanced interpretability of models, and better alignment with expert judgments across both datasets. In the original paper, AcME (David et al., 2024) primarily analyzes structured data. To adapt this method to image classification tasks, the baseline is set to an image where all pixels are zero (completely black), and the original image is divided into multiple segments (considered as features). At each step, only one or several segments are restored to their original pixel values (while all other areas remain black), allowing observation of changes in the model's output for that restored region. By aggregating and ranking the influence of each segment on the prediction results, the most important regions and attributes can be identified.

Specifically, in terms of stability (measured by the stability index J), RealEXP stands out due to its novel perturbation sampling strategy. Unlike traditional methods that rely on random perturbations or Monte Carlo-based sampling, RealEXP employs a fixed-proportion perturbation approach, which significantly reduces the variance of the perturbed samples. This ensures consistent explanation results. Experimental results on stability also validate the theoretical conclusion presented in Proof 4, namely, $Var_{fixed} < Var_{Random} < Var_{Monte Carlo}$.

For model interpretability (measured by R^2), the proposed RealEXP again demonstrates significant improvements. This is primarily attributed to its innovative decomposition of Shapley Value into independent contributions and interaction contributions. Compared

Table 2

Compare the state of arts on Animal-10 Dataset.

Animal-10	Zeiler & Fergus, 2014 (Baseline)				Ribeiro et al., 2016 (LIME)			
	J	R^2	$\mathcal{H}$ -score	M -score	J	R^2	$\mathcal{H}$ -score	M -score
MobileNet	0.46	0.53	0.54	0.50	0.73	0.76	0.68	0.65
ResNet	0.46	0.51	0.45	0.50	0.73	0.77	0.71	0.60
DenseNet	0.46	0.52	0.50	0.50	0.73	0.76	0.76	0.58
ViT	0.46	0.50	0.50	0.5	0.73	0.83	0.81	0.52
Lundberg & Lee, 2017 (Kernel-SHAP)					Li et al., 2023 (GLIME)			
	J	R^2	$\mathcal{H}$ -score	M -score	J	R^2	$\mathcal{H}$ -score	M -score
MobileNet	0.71	0.77	0.69	0.66	0.74	0.77	0.69	0.68
ResNet	0.71	0.78	0.72	0.61	0.74	0.78	0.72	0.63
DenseNet	0.71	0.77	0.76	0.59	0.74	0.77	0.76	0.61
ViT	0.71	0.84	0.82	0.53	0.74	0.84	0.82	0.55
Liu et al., 2024 (HIDTWM)					David et al., 2024 (AcME)			
	J	R^2	$\mathcal{H}$ -score	M -score	J	R^2	$\mathcal{H}$ -score	M -score
MobileNet	0.75	0.78	0.70	0.67	0.78	0.82	0.75	0.69
ResNet	0.75	0.79	0.73	0.62	0.78	0.83	0.78	0.64
DenseNet	0.75	0.78	0.78	0.60	0.78	0.82	0.83	0.62
ViT	0.75	0.85	0.83	0.54	0.78	0.90	0.88	0.56
Tan et al., 2024 (GLIME)					RealEXP			
	J	R^2	$\mathcal{H}$ -score	M -score	J	R^2	$\mathcal{H}$ -score	M -score
MobileNet	0.77	0.80	0.73	0.68	0.83	0.85	0.78	0.71
ResNet	0.77	0.81	0.76	0.63	0.83	0.86	0.81	0.66
DenseNet	0.77	0.80	0.81	0.61	0.83	0.85	0.86	0.64
ViT	0.77	0.88	0.86	0.55	0.83	0.94	0.92	0.58

Table 3

Compare the state of arts on animal-151 dataset.

Animal-151	Zeiler & Fergus, 2014 (Baseline)				Ribeiro et al., 2016 (LIME)			
	J	R^2	$\mathcal{H}$ -score	M -score	J	R^2	$\mathcal{H}$ -score	M -score
MobileNet	0.46	0.43	0.42	0.42	0.73	0.64	0.53	0.55
ResNet	0.46	0.44	0.30	0.41	0.73	0.68	0.53	0.53
DenseNet	0.46	0.41	0.36	0.45	0.73	0.62	0.71	0.36
ViT	0.46	0.44	0.30	0.45	0.73	0.67	0.61	0.25
Lundberg & Lee, 2017 (Kernel-SHAP)					Li et al., 2023 (GLIME)			
	J	R^2	$\mathcal{H}$ -score	M -score	J	R^2	$\mathcal{H}$ -score	M -score
MobileNet	0.71	0.63	0.51	0.54	0.74	0.67	0.51	0.54
ResNet	0.71	0.69	0.52	0.54	0.74	0.66	0.52	0.55
DenseNet	0.71	0.62	0.66	0.37	0.74	0.62	0.71	0.39
ViT	0.71	0.68	0.63	0.42	0.74	0.68	0.63	0.44
Liu et al., 2024 (HIDTWM)					David et al., 2024 (AcME)			
	J	R^2	$\mathcal{H}$ -score	M -score	J	R^2	$\mathcal{H}$ -score	M -score
MobileNet	0.75	0.66	0.55	0.57	0.78	0.62	0.65	0.54
ResNet	0.75	0.71	0.55	0.53	0.78	0.69	0.60	0.55
DenseNet	0.75	0.58	0.73	0.38	0.78	0.57	0.73	0.57
ViT	0.75	0.69	0.64	0.43	0.78	0.71	0.66	0.45
Tan et al., 2024 (GLIME)					RealEXP			
	J	R^2	$\mathcal{H}$ -score	M -score	J	R^2	$\mathcal{H}$ -score	M -score
MobileNet	0.77	0.61	0.59	0.43	0.83	0.73	0.56	0.51
ResNet	0.77	0.65	0.54	0.50	0.83	0.66	0.65	0.51
DenseNet	0.77	0.69	0.54	0.46	0.83	0.70	0.67	0.52
ViT	0.77	0.65	0.66	0.41	0.83	0.84	0.82	0.48

to methods like Kernel-SHAP and AcME, which rely on the traditional Shapley Value for interpretability, RealEXP places greater emphasis on feature importance, leading to more accurate and insightful explanations of deep learning models.

Finally, RealEXP exhibits enhanced performance in aligning with expert evaluations, as reflected by higher $\mathcal{H}$ -score and M -score. This improvement is due to its thoughtful selection of proxy models and feature importance evaluation. Unlike GLIME, which directly outputs slopes using Gaussian ridge regression, RealEXP incorporates an ensemble tree model to capture nonlinear relationships between features. After training the tree model, RealEXP further assesses the feature importance within the ensemble, achieving more robust interpretability and better alignment with the underlying model behavior.

In the evaluation of interpretability across four pre-trained models, an intriguing phenomenon was observed: RealExp performs significantly better with ViT in $\mathcal{H}$ -score, while convolutional networks achieve higher M -scores. This suggests that ViT excels in overall interpretability performance, while CNNs align more closely with human expert reasoning in decision paths. This difference is rooted in the design philosophies of these models. ViT's multi-head attention mechanism captures global semantic information, allowing RealExp to leverage these broad attention distributions for more comprehensive interpretability. In contrast, CNNs focus on local feature extraction, which aligns better with human intuition but may result in fragmented interpretability outputs when analyzed using RealExp.

Furthermore, the choice of model should depend on task requirements. For instance, in video classification tasks requiring frame-level feature extraction, ViT's ability to model global relationships enhances temporal consistency, making it more suitable for such applications. RealExp's decomposition-based interpretability framework further strengthens this effect by ensuring that ViT's global attention maps translate into meaningful, structured explanations. Conversely, in fine-grained image classification tasks, CNNs' ability to capture local details makes them more effective, although RealExp's feature correlation analysis may be more fragmented due to the model's inherent locality.

Fig. 7 visually presents the results of applying different explainability methods on the ViT pre-trained model. It is evident that the proposed RealEXP method demonstrates higher stability and model interpretability compared to other explainability approaches. This is primarily due to two key factors: First, RealEXP places greater emphasis on feature importance, leading to more accurate and insightful explanations of deep learning models. Second, the RealEXP method prioritizes dynamic weight allocation and regional sensitivity optimization when constructing explanations. Furthermore, in multiple perturbation experiments, the RealEXP method consistently identifies regions closely related to the target classification, whereas other methods, such as LIME, Kernel-SHAP, and GLIME, may exhibit feature region drift or inconsistency under certain conditions. This indicates that RealEXP not only provides more reliable model explanations but also better meets the demand for model transparency in practical applications.

Table 4 presents the ablation study results on the Animal-151 dataset based on the ViT model, evaluating the impact of different method combinations on model interpretability. Among single methods, Fixed Perturb showed outstanding performance in the Jaccard coefficient J (0.73), while Ensemble Tree and RealEXP contributed moderately to $\mathcal{H}$ -score (0.64) and M -score (0.39), respectively, but their overall effectiveness was relatively limited. For dual-method combinations, the pairing of Fixed Perturb and RealEXP demonstrated the best results, significantly improving R^2 (0.72) and M -score (0.42). However, when all three methods (Fixed Perturb, Ensemble Tree, and RealEXP) were used together, all metrics reached their highest values: J at 0.83, R^2 at 0.84, $\mathcal{H}$ -score at 0.82, and M -score at 0.48. This indicates that the combination of all three methods maximizes interpretability, stability, and overall

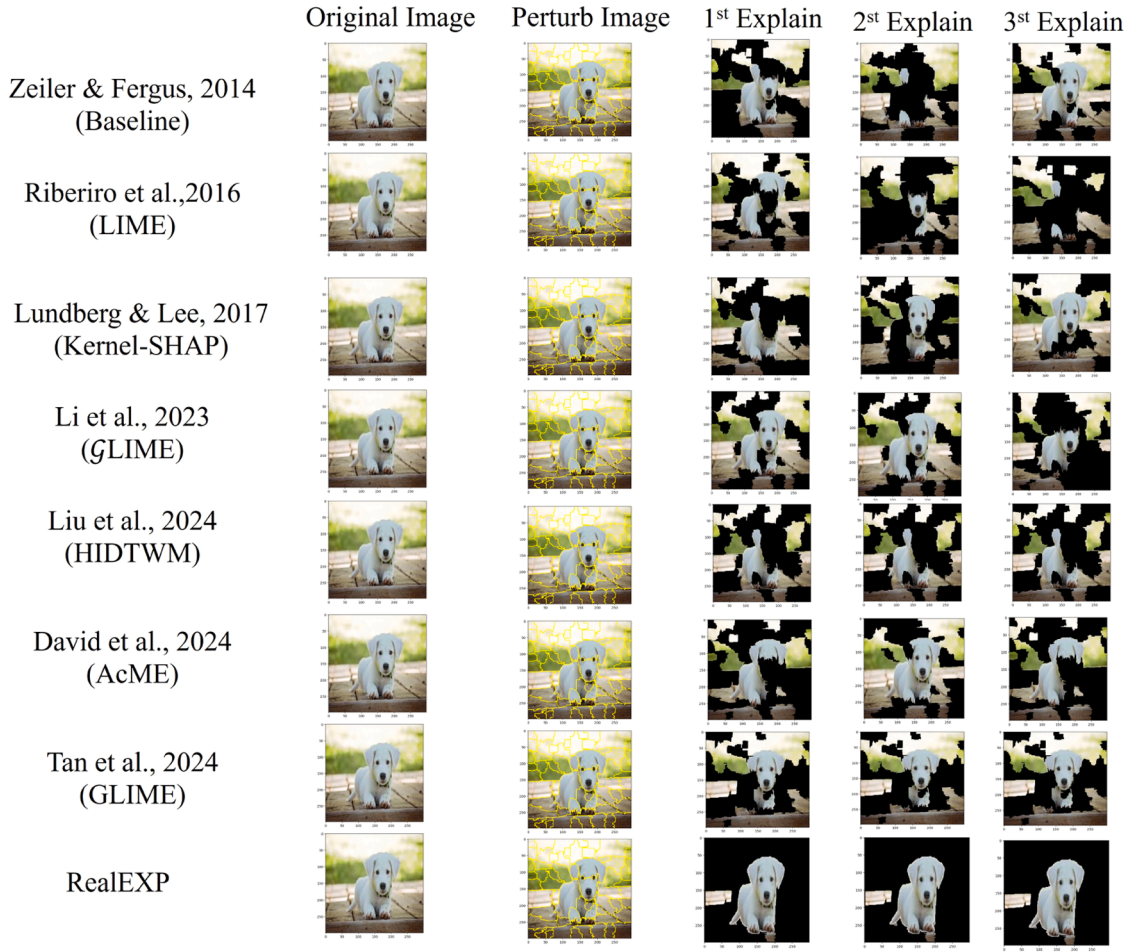


Fig. 7. Example of explainable visualization for ViT.

Table 4

Ablation study in animal-151 dataset in ViT.

Fixed Perturb	Ensemble Tree	RealEXP	J	R^2	$\mathcal{H}$ -score	M -score
✓			0.73	0.63	0.30	0.33
	✓		0.46	0.67	0.64	0.32
		✓	0.42	0.68	0.64	0.39
✓	✓		0.73	0.68	0.61	0.27
✓		✓	0.71	0.72	0.68	0.42
	✓	✓	0.48	0.54	0.67	0.33
✓	✓	✓	0.83	0.84	0.82	0.48

performance, highlighting the importance of their synergistic effects.

Fig. 9 aims to discuss the performance of RealEXP under different hyperparameter λ settings. It can be observed that variations in λ significantly impact the model's performance, with the best performance achieved when $\lambda = 0.25$. This is because, at $\lambda = 0.25$, a good balance is struck between regularization strength and the model's learning capacity, effectively preventing overfitting while ensuring sufficient learning of data features.

Fig. 7 aims to illustrate the model's performance under different ratio settings. The best performance is achieved when the Fixed Ratio equals 0.3. This is because, at this ratio, the distribution of data samples aligns most closely with the target task, allowing the model to learn key features more effectively while avoiding issues such as information loss or noise amplification caused by excessively high or low ratios. The above is the experimental part of images, followed by the experiments of text Sentiment Analysis.

Fig. 8 presents a comparison of the runtime of different interpretability methods. The experiment evaluates the computation time required by various explainable machine learning methods to process 2,000 images across two datasets, under 500 perturbations, and generate visual explanations. To ensure consistency across all methods, CUDA acceleration was not utilized, and all experiments were

conducted using only CPU processing. Specifically, a Core i7-14700K CPU was used to provide a fair runtime comparison.

As shown in Fig. 10, the RealExp method runs significantly faster (105 hours) compared to Kernel-SHAP (135 hours) and other interpretable methods. This is because traditional Shapley Value-based approaches require an exhaustive enumeration of all feature subsets, resulting in a computational complexity of $O(n \cdot 2^n)$, which grows exponentially with the number of features. In contrast, RealExp reduces the computational complexity to $O(n^2)$ by leveraging $O(n)$ single-feature evaluations, $O(n^2)$ multi-feature contribution calculations, and similarity assessments, significantly improving efficiency. Theorem 1 in the Appendix provides detailed mathematical proof, demonstrating that RealExp can accurately estimate feature contributions with lower computational costs, making it more practical for high-dimensional data scenarios.

5.2. Video classification

This section introduces the application of interpretable machine learning in video classification. The experiment explores how to construct an interpretable anomaly event detection system in the context of a smart city. The dataset used in the experiment is XD-Violence, the largest publicly available violence detection dataset, which consists of 4,754 videos with a total duration of 217 hours. It covers six types of violent incidents: abuse, car accidents, explosions, fights, riots, and shootings. The dataset is divided into a training set of 3,954 videos and a test set of 800 videos, with the latter containing 500 violent videos and 300 non-violent videos.

To validate the feasibility of the proposed interpretable method, the experiment selects 200 samples from the training set and 50 samples from the test set as experimental data, with the classification categories being coarse-grained "violent" and "non-violent."

The initial design of the neural network architecture is shown in Fig. 11. The designed violence detection system consists of the following process: First, a frame-level feature extractor is used to extract features from each frame of the video. In the experiment, MobileNet and ViT are used as frame-level feature extractors. These two pre-trained models were chosen to explore the performance of two approaches discussed in Section 5.1: one based on CNNs and the other using ViT, which divides images into patches and processes them through a Transformer encoder. After feature extraction, a Transformer Block module is applied for temporal analysis of the video sequence, followed by a classifier to complete the classification task.

Furthermore, during the training process, the system's Transformer Block adjusts the spatiotemporal relationships directly from the frozen pre-trained embeddings without significantly altering the spatial learning of the original images.

In Table 5, the experiment observed an interesting phenomenon. First, the model's overall performance was suboptimal, with a poor AP score. This suggests that the system frequently misclassified violent videos as non-violent. Second, as discussed in Section 5.1, the ViT pre-trained model was expected to perform better in learning video sequences than MobileNet. However, in practice, this advantage was not realized. Therefore, this study employed RealEXP to analyze the explainability of both systems. Fig. 12 presents two intriguing cases that aim to demonstrate model behavior in anomalous video detection.

Overall, Transformer Blocks ensure that predictions across consecutive frames remain stable, indicating that the model effectively learns temporal sequences rather than relying solely on static features from individual frames. In other words, the self-attention mechanism in Transformers enables the model to capture key patterns over longer time spans and establish long-term dependencies between different frames.

This is also verified by the spatial explainability of ViT, where the learned features tend to be aggregated together rather than being sparsely distributed. Since ViT employs Patch Embedding along with a global self-attention mechanism, its feature representations are more likely to form large, continuous regions, rather than relying primarily on local edges or color variations as CNNs do. This

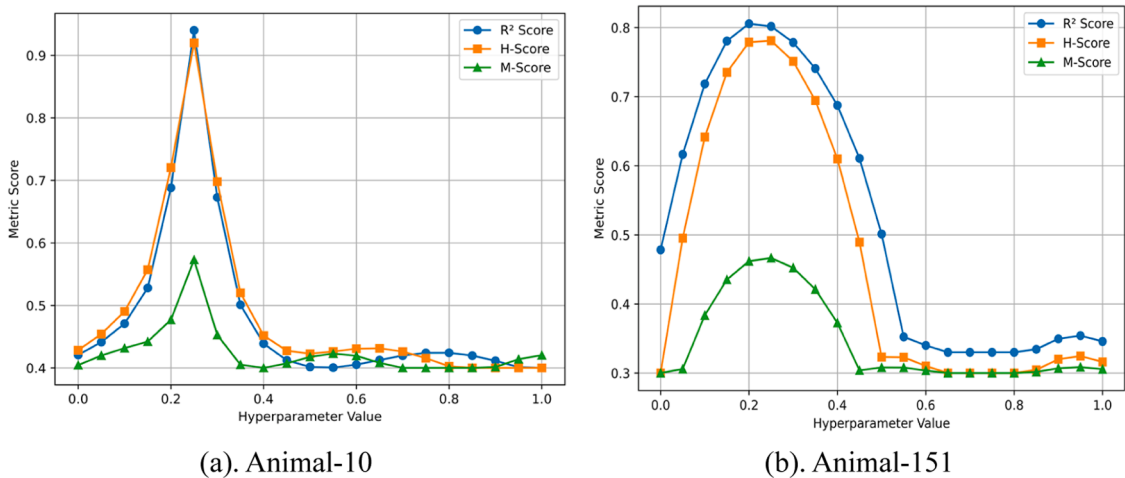


Fig. 8. Hyperparameter λ setting in Exp. (4).

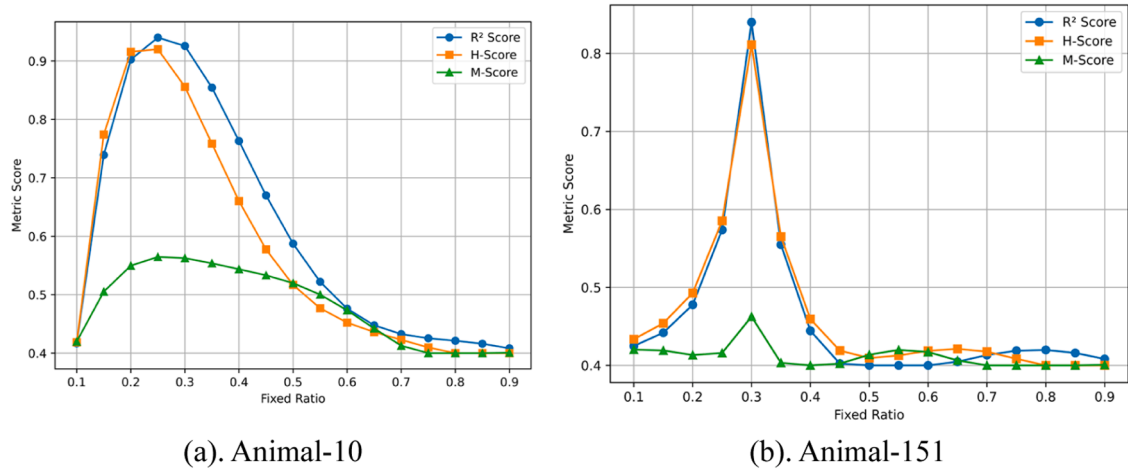


Fig. 9. Different fixed ratio.

aggregation effect further explains the stability of Transformers in video data, as they not only learn temporal information but also establish a more holistic understanding of objects in the spatial domain.

In contrast, CNNs (such as MobileNet), which rely on local convolutions, struggle to integrate information across frames, leading to limited temporal modeling capabilities and less consistent predictions over time. CNNs primarily depend on a local receptive field with a sliding window, making their feature extraction highly dependent on the spatial distribution of pixels. As a result, they inherently lack the ability to capture long-term dependencies. For instance, MobileNet's 1×1 convolutions further reduce cross-region feature interactions, preventing the model from effectively learning dependencies across consecutive frames. Consequently, CNNs are more susceptible to interference from background colors, lighting variations, or local features, leading to unstable predictions in temporal sequences and, ultimately, significant performance deviations in anomalous video detection tasks.

In Subfigure (a) – The Car Crash Scene, it can be observed that MobileNet primarily relies on color features for its predictions. This results in an interesting phenomenon: although an actual accident occurs in the foreground involving a white car, MobileNet instead makes strong predictions on the black vehicle in the background. This suggests that MobileNet might have been influenced by the high contrast in the black impact area, which coincidentally matched certain training data features, leading to correct guesses in a few key frames. However, this decision-making process does not truly capture temporal information but is instead based on local color variations. In contrast, ViT focuses primarily on the white car in the foreground and consistently attends to the accident area, demonstrating a more stable and globally aware decision-making process.

In Subfigure (b) – The Ship Under Attack Scene, an intriguing phenomenon arises: neither ViT nor MobileNet pays significant attention to the smoke and flames from the explosion, instead focusing more on changes in the water surface and the ship's structure. There are two reasons to explain it:

1. The smoke and fire share similar colors with the background, and their shapes change significantly in real-world videos, making it difficult for the model to learn stable features for distinguishing them.
2. ViT, with its ability to capture global features, is able to more consistently detect the complete ship's outline, whereas MobileNet's predictions are more fragmented, particularly showing a strong dependency on color variations in the water.

These findings highlight that environmental factors significantly affect model performance, particularly when variations in color, lighting, or background conditions occur. Furthermore, this exposes the limitations of pre-trained models—when applied to unseen, complex environments, they may fail to generalize effectively and instead rely on statistical patterns from the training data, such as color and contrast, rather than true semantic information. This phenomenon aligns with the “Wolf vs. Husky” theory proposed by Ribeiro et al. (2016), which suggests that models may not actually learn key features for classification but instead depend on non-essential environmental factors (such as the presence of snow in the background).

To address this issue, we propose an improvement: integrating a pre-trained image segmentation model into the system to remove background noise, allowing the model to focus more effectively on the target regions. This approach can help reduce the model's dependency on color and background variations, ultimately improving its generalization ability and stability.

The modified model architecture is shown in Fig. 13. Before inputting frame-level feature extraction, the system first undergoes global background removal using SAM2 (Nikhila et al., 2025). The introduction of this step aims to enhance the model's interpretability, making it more focused on foreground objects while reducing unstable decisions caused by background noise.

Table 6 presents the significant performance improvement of the retrained model on the XD-Violence dataset after incorporating SAM2 for background removal. In terms of Average Precision (AP), both MobileNet and ViT exhibited substantial enhancements, specifically: MobileNet's AP increased from 63.27 to 84.33, an improvement of +21.06. ViT's AP increased from 64.86 to 89.86, an improvement of +25.00

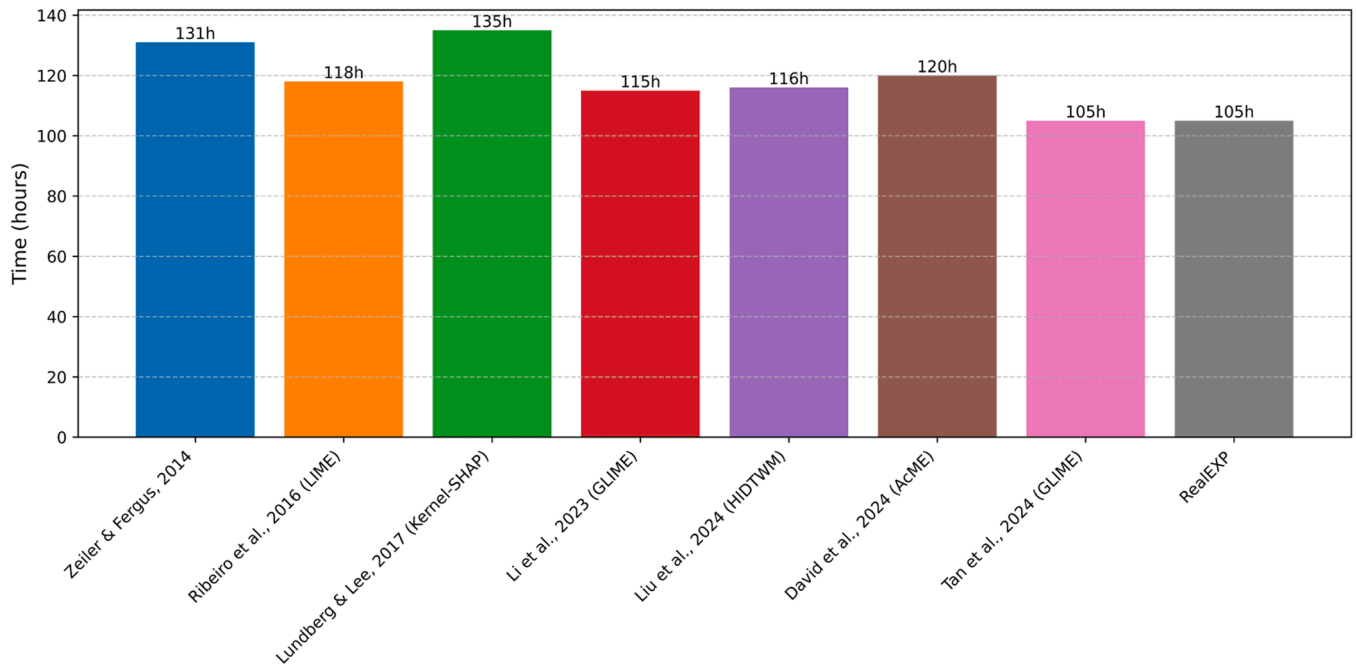


Fig. 10. Experimental comparison of runtime.

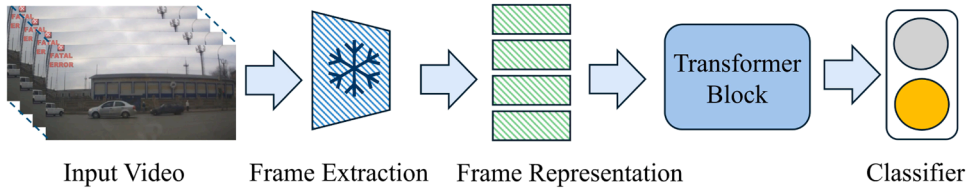


Fig. 11. Initial architecture.

Table 5

The initial model performance on XD-violence.

Feature extraction	AP value $\uparrow$
MobileNet	63.27
ViT	61.26

This demonstrates that removing background information significantly enhances the model's classification capability, particularly in video anomaly detection tasks. By eliminating irrelevant background details, the model can focus more effectively on key actions and objects within the video, thereby improving its explainability. Specific examples illustrating this impact are presented in Fig. 12, further validating how background removal influences the model's attention areas.

Fig. 14 presents the explainability analysis of the adjusted model after incorporating SAM2 for background removal. Compared to the original model, the adjusted model exhibits significant changes in attention distribution and target region focus, particularly highlighting the differences between CNN and Transformer-based architectures.

In subfigure (a), several key observations can be made:

1. After processing with SAM2, most of the background is removed, allowing the model to focus primarily on the foreground vehicles. Compared to the original model, this significantly reduces misclassifications caused by background noise, enabling the model to more reliably identify the vehicle involved in the accident.
2. ViT primarily focuses on the entire vehicle region and maintains a stable attention distribution. This aligns with the nature of the Transformer structure, where the global self-attention mechanism allows the model to learn spatial information more holistically, leading to attention that remains concentrated on complete object regions rather than being dispersed across different smaller areas.
3. MobileNet still relies heavily on color features for classification. In certain frames, MobileNet exhibits a strong preference for high-contrast regions, sometimes incorrectly focusing on the darker background vehicle rather than the actual accident vehicle. This suggests that CNNs still tend to depend on local color and edge features rather than learning the overall shape of objects.

A particularly interesting phenomenon can be observed in subfigure (b):

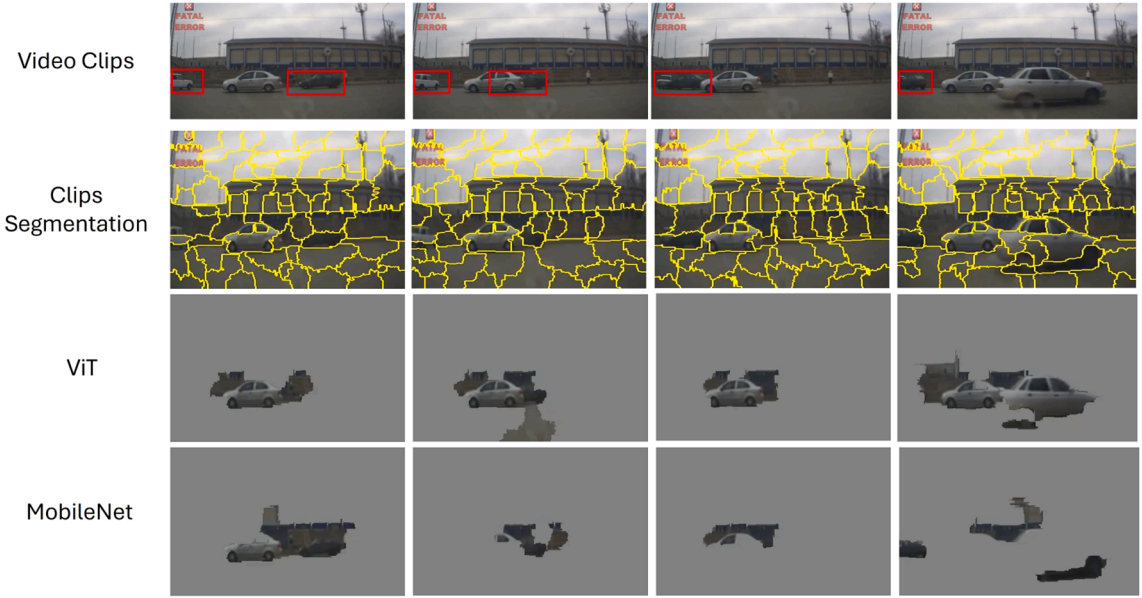
1. ViT primarily focuses on the entire fire and smoke region, whereas MobileNet tends to focus on the most color-dominant areas. Due to ViT's global self-attention mechanism, its learned features are aggregated together, treating fire and smoke as a single coherent object. As a result, its segmentation outputs display a more complete detection of fire and smoke, with attention consistently distributed across the entire affected area over multiple frames. This indicates that ViT does not simply rely on color but instead learns the temporal pattern of fire and smoke, resulting in more globally consistent attention across frames.
2. MobileNet still heavily depends on color information, particularly the brightness and contrast of the fire. This can be seen in the explainability results of MobileNet, where attention is mainly concentrated on the brightest parts of the fire, rather than the entire fire region. This suggests that CNNs still rely on local color features for classification, rather than learning the overall structure of flames and smoke like ViT does. For instance, in some frames, MobileNet fails to detect certain darker areas of the fire, whereas ViT successfully captures the entire shape of the flames and smoke.

In summary, the incorporation of SAM2 not only enhances the model's classification performance but also improves interpretability, making ViT's spatial attention distribution more stable, while MobileNet's decisions remain constrained by its reliance on color features. These results further validate Transformer's advantage in anomaly video detection, as it can more reliably learn global information in complex scenes, whereas CNNs are more prone to being influenced by local features.

5.3. Pneumonia image classification

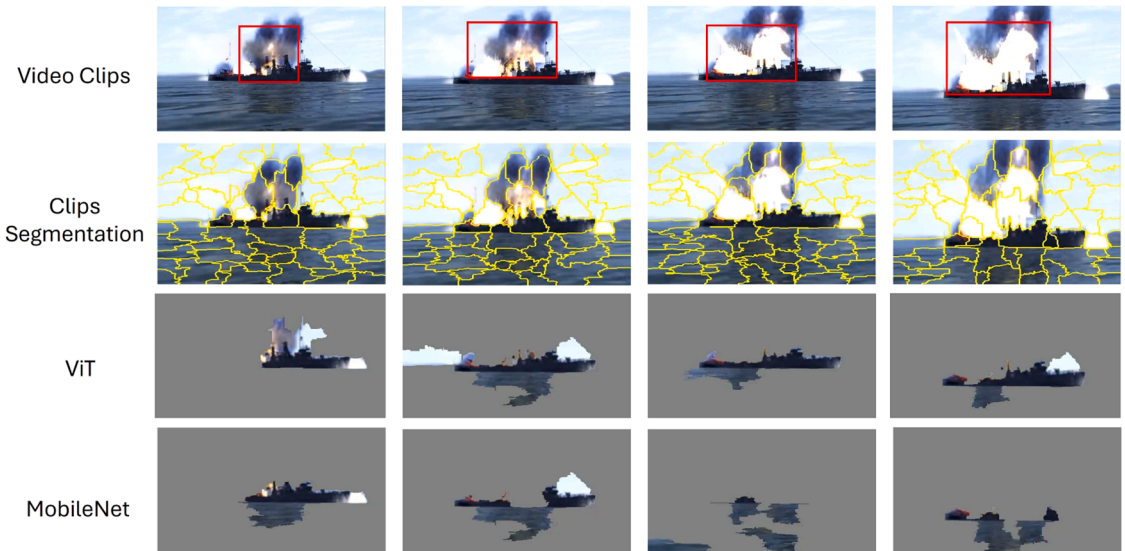
This section presents the feasibility analysis of using RealExp, along with H-score and M-score, to evaluate the selection of medical

v=TOEVc-pOEwM_#00-08-40_00-10-40_label_B6-0-0.mp4



(a) The explanation results for initial model in test data in v=TOEVc-pOEwM_#00-08-40_00-10-40_label_B6-0-0.mp4.

v=hAHXCMRvY_I_#1_label_G-0-0.mp4



(b) The explanation results for initial model in test data in v=hAHXCMRvY_I_#1_label_G-0-0.mp4.

Fig. 12. Case visualization analysis of initial model.

diagnostic models in practical applications. The dataset used in the experiment is the RSNA Pneumonia Detection Challenge from Kaggle (Anouk et al., 2018). Fig. 13 illustrates some examples from the training dataset along with their labels. As observed in the X-ray images, cases labeled 1 indicate pneumonia, characterized by extensive white opacity in the lung region. In contrast, cases labeled 0 indicate no pneumonia, where most of the lung field appears darker (black), with only structures like ribs, the heart, and blood

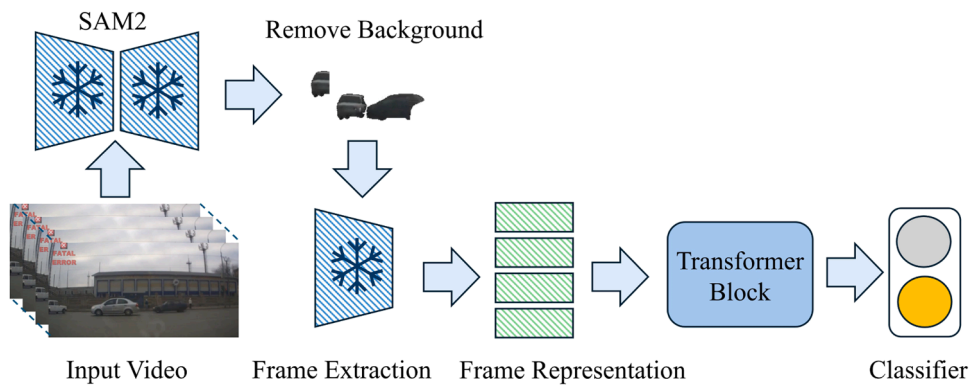


Fig. 13. Adjust architecture.

Table 6

The adjust model performance on XD-violence.

Feature extraction	AP value↑
MobileNet	84.33 (+21.06)
ViT	89.86 (+25.00)

vessels appearing in white.

The models used in this experiment are ResNet-18 and ViT. For performance evaluation, Accuracy, Precision, Recall, and F1-score are selected as key classification metrics. In the explainability analysis, 30 X-ray images from the training set were randomly selected and annotated by experts, following the RealExp framework. The details of the process are shown in Fig. 4.

Table 7 compares the performance of ResNet-18 and ViT on the RSNA Pneumonia Detection Challenge training dataset. Overall, ViT outperforms ResNet-18 in classification performance, achieving a Precision of 0.82, Recall of 0.86, F1-score of 0.84, and Accuracy of 0.86, all of which surpass those of ResNet-18 (0.74, 0.81, 0.77, and 0.79, respectively). This indicates that ViT demonstrates superior pneumonia detection capability. Regarding $\mathcal{H}$ -score (accuracy of selected regions), ViT scores 0.88, significantly higher than ResNet-18's 0.74, showing that ViT is more precise in identifying pathological regions, enhancing the reliability of medical interpretation. However, in M -score (consistency in the ranking of selected regions), ResNet-18 (0.33) outperforms ViT (0.23), suggesting that ResNet-18's selection order aligns more closely with expert evaluations, whereas ViT performs worse in this aspect.

Fig. 16 illustrates the explainability visualization results of ViT and ResNet in the task of pneumonia detection in medical imaging. There are significant differences between the two models in terms of selected key regions and the distribution of chosen blocks. The regions selected by ViT exhibit greater global coherence, typically covering larger and more continuous lung areas, which may be attributed to its global receptive field, allowing it to integrate features across the entire image for decision-making. In contrast, the blocks selected by ResNet are more scattered and localized, likely due to its CNN-based architecture, which relies more on local features for decision-making. In the visualization results, RealExp performs significantly better with ViT compared to ResNet because ViT's attention regions are broader, allowing it to capture the global distribution of pathological areas. This aligns well with RealExp's explainability approach, which benefits from models that provide coherent, high-level feature representations.

In contrast, ResNet's selected blocks are more fragmented, aligning more closely with expert-annotated lesion priorities but potentially missing critical lesion information due to its localized focus. This fragmented attention reduces the stability of RealExp's interpretability scores when applied to ResNet, which may explain its lower performance compared to ViT.

From a medical imaging perspective, ViT's global attention mechanism makes it more effective for identifying overall lesion patterns, whereas ResNet is better suited for micro-level lesion analysis. This suggests that integrating RealExp with ViT provides a more holistic interpretability framework, making it particularly well-suited for image screening tasks in medical AI applications.

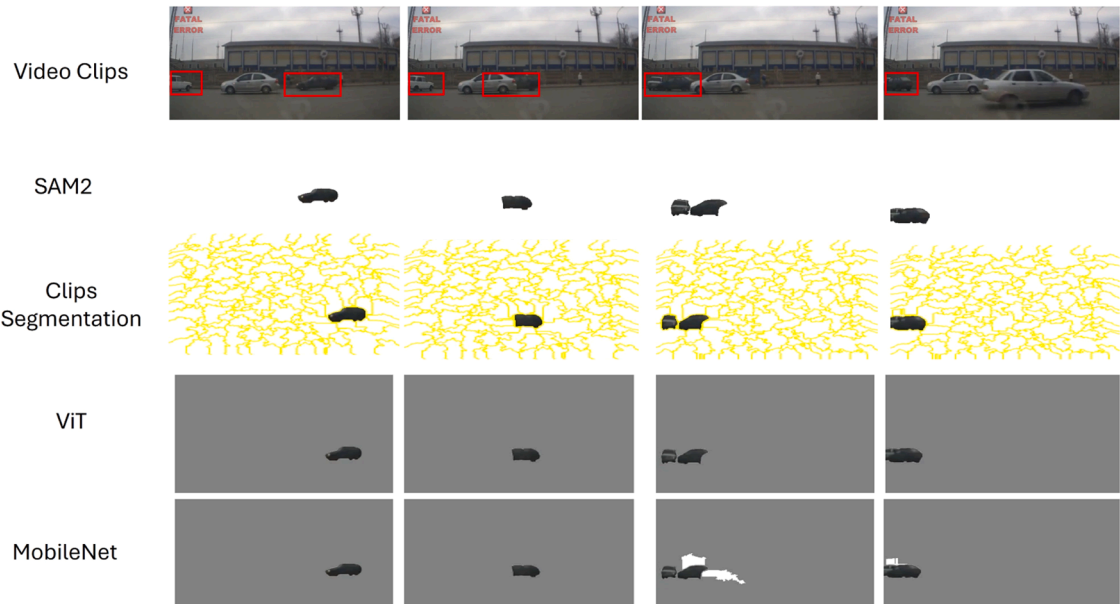
In conclusion, combining ViT's global recognition capability with ResNet's localized precision could enhance the explainability and clinical reliability of AI-based medical diagnosis systems, offering stronger support for pneumonia detection tasks.

5.4. Text sentiment analysis

This section presents explainability experiments for text sentiment analysis, focusing on addressing real-world challenges such as detecting bullying in educational environments (e.g., identifying offensive language) or filtering sensitive content in private online communities. These scenarios often necessitate the implementation of edge computing and localized models, while also requiring rapid system response times. Although Transformer-based large language models (LLMs) have demonstrated remarkable success in text processing, their extensive parameter sizes and heavy reliance on vast amounts of training data pose significant difficulties for their efficient deployment in these specific use cases.

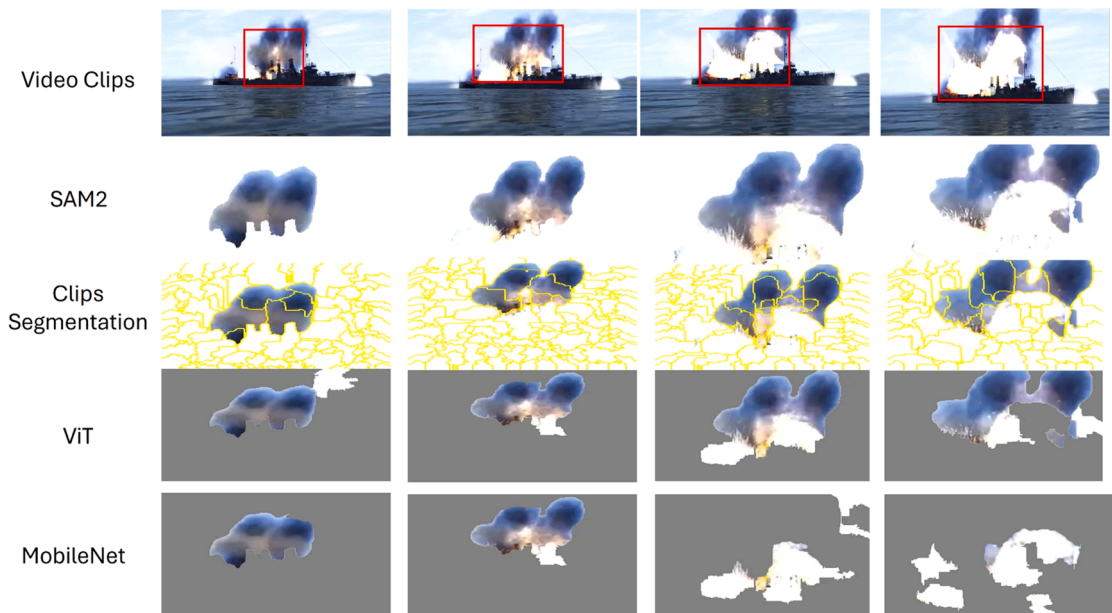
Therefore, advancing interpretable machine learning techniques to enhance the performance of traditional Bag-of-Words (BoW)

v=TOEVc-pOEwM__#00-08-40_00-10-40_label_B6-0-0.mp4



- a. The explanation results for adjust model in test data in v=TOEVc-pOEwM__#00-08-40_00-10-40_label_B6-0-0.mp4.

v=hAHXCMRvY_I__#1_label_G-0-0.mp4



- b. The explanation results for adjust model in test data in v=TOEVc-pOEwM__#00-08-40_00-10-40_label_B6-0-0.mp4.

Fig. 14. Case visualization analysis of adjust model.

methods while maintaining their lightweight nature would significantly contribute to the effective deployment of these models in such environments. In particular, if conventional BoW approaches can achieve results comparable to LLMs in identifying sensitive words and specific terms, their practical applicability and success rate would be further improved.

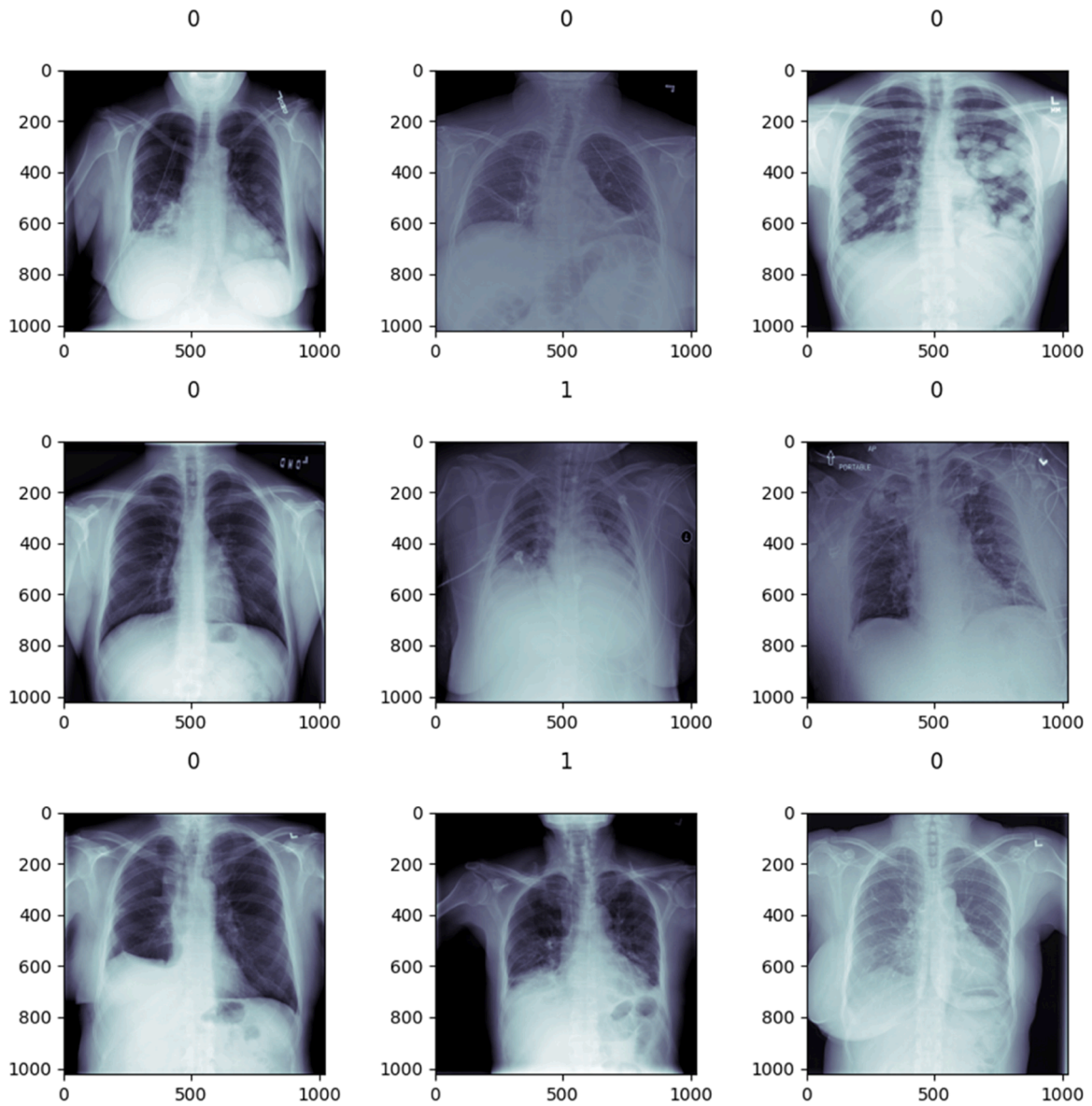


Fig. 15. Some examples in RSNA pneumonia detection challenge train dataset.

Table 7

The model performance on RSNA pneumonia detection challenge train dataset.

Model	Precision	Recall	F1	ACC	$\mathcal{H}$ -score	M -score
ResNet-18	0.74	0.81	0.77	0.79	0.74	0.33
ViT	0.82	0.86	0.84	0.86	0.88	0.23

Fig. 17 illustrates the workflow of the text analysis experiment. First, stop words are removed from the dataset, followed by stemming or lemmatization of words in the text. Next, feature extraction is performed using the TF-IDF method, and a machine learning model is trained. After the training is completed, the EnEXP method is applied to refine the machine learning model. Additionally, the experiment fine-tunes the RoBERTa model using the entire dataset and evaluates its performance through comparison.

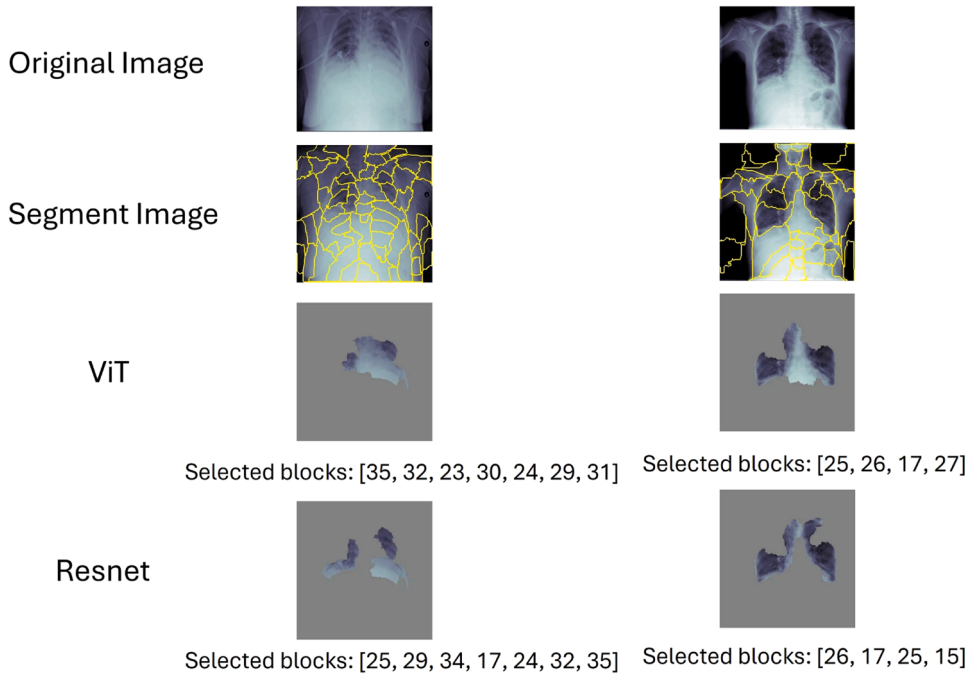


Fig. 16. Some visualization in RSNA pneumonia detection challenge test dataset.

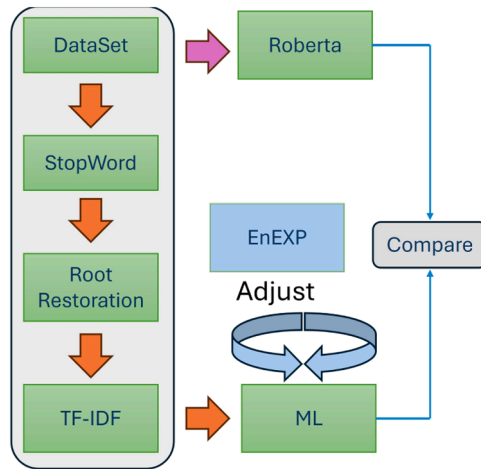


Fig. 17. Workflow for text sentiment analysis.

The data set used in the experiment is Hate Speech and Offensive Language Dataset⁴, from which the first 1,000 hate speech reviews, 1000 Offensive Language and 1,000 Neither review were selected, totaling 3000 sentences. The data was divided into training, validation, and test sets in a ratio of 7:2:1. Part-of-speech tagging was first applied to identify adjectives within the sentences, followed by the selection of words that are likely to determine the negative sentiment of the sentence.

Table 8 aims to show the performance of different machine learning models on the Hate Speech and Offensive Language Dataset. As shown in Table 8, the Random Forest and Logistic Regression achieved the best performance, followed by Decision Tree. Table 9 aims to contrast the performance of the proposed RealEXP method with other related algorithms, such as LIME (Riberiro et al., 2016), Kernel-SHAP (Lundberg & Lee, 2017), and G-LIME (Tan et al., 2024), in terms of $\mathcal{H}$ -Score. The results demonstrate that RealEXP significantly outperforms other interpretability algorithms on $\mathcal{H}$ -Score. This is attributed to RealEXP's integration of the strengths of both global and local interpretability, coupled with the introduction of a multi-tier feature weight analysis mechanism, enabling it to

⁴ <https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset>

Table 8

Different machine learning methods on the hate speech and offensive language dataset.

Random forest	Precision	Recall	F1-Score	Accuracy
0 (hate speech)	0.84	0.90	0.87	0.87
1 (Offensive Language)	0.89	0.91	0.90	
2 (Neither)	0.92	0.88	0.90	
Logistic Regression	Precision	Recall	F1-Score	Accuracy
0 (hate speech)	0.85	0.90	0.87	0.87
1 (Offensive Language)	0.89	0.84	0.86	
2 (Neither)	0.91	0.89	0.90	
Decision Tree	Precision	Recall	F1-Score	Accuracy
0 (hate speech)	0.81	0.82	0.81	0.81
1 (Offensive Language)	0.82	0.81	0.82	
2 (Neither)	0.83	0.85	0.84	

Table 9Different interpretable machine learning algorithm performance $\mathcal{H}$ -score.

$\mathcal{H}$ -score	LIME	Kernel-SHAP	GLIME	RealEXP
Random Forest	0.3758	0.4032	0.4583	0.4856
Logistic Regression	0.7621	0.8002	0.8618	0.8892
Decision Tree	0.3236	0.4189	0.4379	0.4379

more accurately identify key features in text tasks while achieving superior interpretative quality compared to traditional methods.

Additionally, an intriguing phenomenon was observed during the research: why do decision trees, typically regarded as highly interpretable, and random forests, which possess some level of interpretability, perform poorly in interpretability scores for natural language processing tasks? The primary reason lies in the fact that when perturbed data is processed through tree models, some perturbed samples yield prediction results identical to those of the original samples. This occurs because, after tokenization in text, the number of words increases significantly, and many words are filtered out during tree model construction due to their low attribute importance. Moreover, test data may include words do not present in the training data, causing the tree model to rely on the distribution of parent node data rather than effectively interpreting input data based on specific feature weights.

To tackle this problem, the study introduces a dual-phase approach, termed the RL framework, which integrates the capabilities of random forests and logistic regression. In the first phase, involving the random forest, data points where perturbed samples exhibit identical prediction probabilities to their original counterparts are preserved. Once the entire dataset has been processed, these matching data points are employed to refine the logistic regression model.

During the assessment of the pre-trained random forest model, a comprehensive analysis of 2100 data points was conducted, identifying 315 cases where perturbed samples aligned perfectly with the original samples in terms of prediction probabilities. Leveraging these insights, an updated random forest model, built on 1685 data points, was developed, and its output served as the foundation for enhancing the logistic regression model. This approach successfully mitigated the initial challenge.

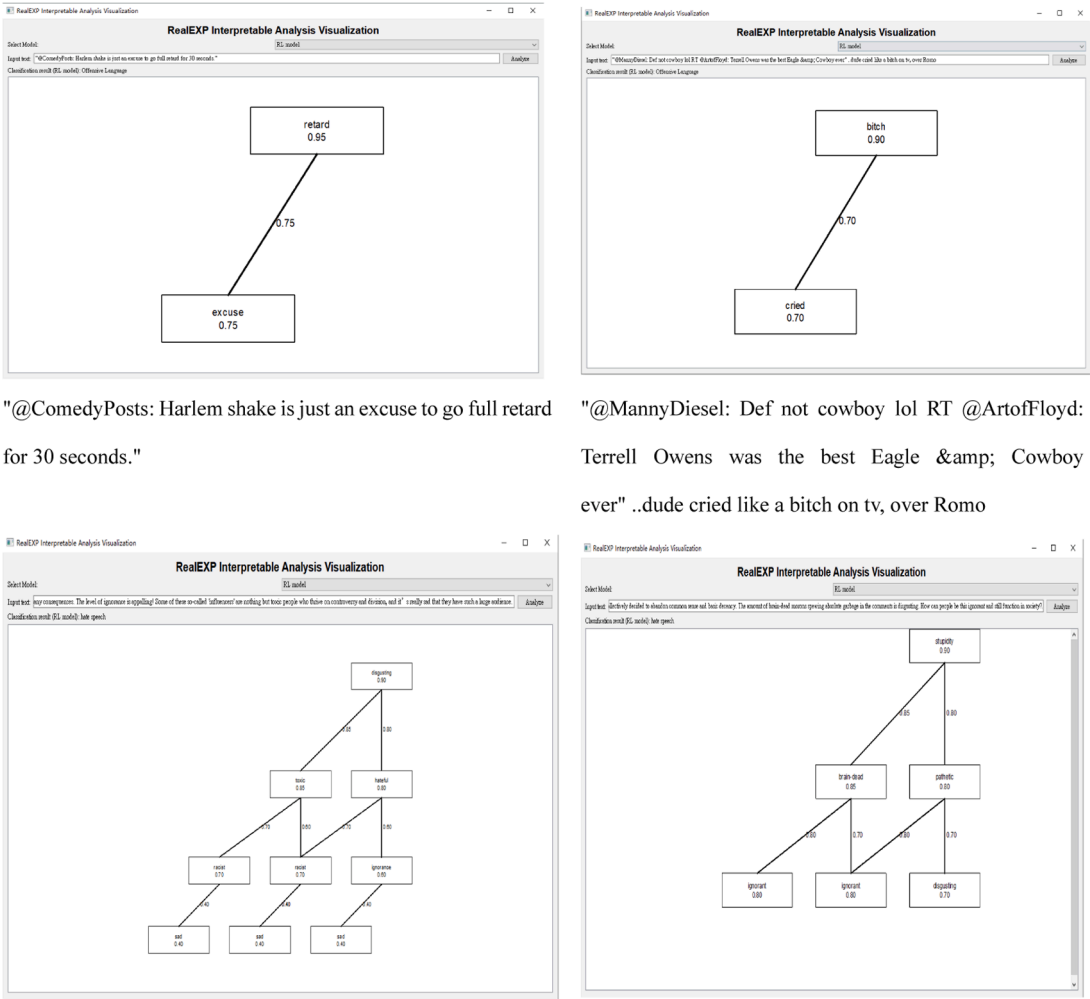
The RL Framework demonstrated significant improvements in both model performance and interpretability. Specifically, after applying RealEXP, the interpretability $\mathcal{H}$ -score of the original random forest model increased from 0.4856 to 0.9764, while the F1-Score of the combined model reached 0.95 in the performance evaluation. As shown in Table 10, the comparison between the RL model and the fine-tuned RoBERTa model revealed that the redesigned combined machine learning model outperformed the pre-trained RoBERTa model. This suggests that by incorporating interpretability mechanisms, it is possible to optimize the model's structure and feature selection without relying on large-scale pre-training, thereby enhancing both performance and generalization capabilities. Furthermore, this result demonstrates that in specific tasks, traditional machine learning methods, when combined with interpretability analysis, can effectively narrow the gap with large-scale pre-trained models and, in some cases, even achieve superior performance.

Fig. 18 illustrates the application of the RealEXP interpretability analysis tool on a hate speech and offensive language dataset. This tool enables users to understand the decision-making logic of the model when classifying such texts and identifies the key words and

Table 10

Compared with RoBERTa finetuning model.

RoBERTa	Precision	Recall	F1-Score	Accuracy
0 (hate speech)	0.92	0.93	0.92	0.96
1 (Offensive Language)	0.91	0.92	0.91	
2 (Neither)	0.93	0.94	0.93	
RL framework	Precision	Recall	F1-Score	Accuracy
0 (hate speech)	0.95	0.96	0.95	0.98
1 (Offensive Language)	0.94	0.95	0.94	
2 (Neither)	0.96	0.97	0.96	



"@ComedyPosts: I can't believe the kind of nonsense people are spreading on social media these days. It's honestly disgusting how some individuals think they can just spew hateful, racist, and completely false information without any consequences. The level of ignorance is appalling! Some of these so-called 'influencers' are nothing but toxic people who thrive on controversy and division, and it's really sad that they have such a large audience."

"I have never seen such a pathetic excuse for a community. The level of stupidity here is beyond comprehension. It's like everyone collectively decided to abandon common sense and basic decency. The number of brain-dead morons spewing absolute garbage in the comments is disgusting. How can people be this ignorant and still function in society?"

Fig. 18. Text visualization of RealExp in hate-speech-and-offensive-language-dataset.

phrases that have the most significant impact on classification outcomes. The example texts are sourced from real social media comments, covering varying degrees of offensive language. These include statements such as: "Harlem shake is just an excuse to go full retard for 30 seconds," which contains discriminatory language; "Dude cried like a bitch on TV, over Romo," a derogatory remark targeting a public figure; "I can't believe the kind of nonsense people are spreading on social media these days...", a critical comment on misinformation; and "I have never seen such a pathetic excuse for a community. The level of stupidity here is beyond comprehension...", an extremely negative evaluation of a community. These sample texts range from mildly offensive remarks to more extreme forms of hate speech, covering different types of harmful content. The visualization results from RealEXP demonstrate how the model determines whether a comment constitutes hate speech or offensive language, highlighting the specific words that play a crucial

role in the decision-making process. This analytical approach not only enhances model transparency but also helps researchers identify high-risk language patterns, ultimately allowing them to refine classification models for improved accuracy and fairness.

5.5. Limitation

This paper discusses the proposed RealEXP in the fields of traditional image classification, anomaly behavior detection classification, medical diagnostics, and text classification. In other classification domains, such as sentiment analysis, speech recognition, remote sensing image classification, and financial risk assessment, although our experimental design has significantly reduced the impact of variations in experts' perceptions of key features or regions through the majority voting mechanism, thereby improving the stability of results, the data annotation process in these domains may still be influenced by subjective factors, making it difficult to ensure annotation consistency, and leading to potential discrepancies in expert standards.

Additionally, our current experiment has not yet explored how to handle noisy data or inconsistent annotations, which may affect the reliability of the model in certain applications. For example, in medical diagnosis or financial risk assessment, annotation errors may lead to serious consequences, making it necessary to implement additional mechanisms for identifying and correcting annotation errors, such as confidence-based data filtering, automated annotation correction techniques, or semi-supervised learning methods.

Furthermore, reliance on highly qualified and suitable domain experts remains a bottleneck in practical applications. These issues may be limited by expert annotation accuracy, data quality, and consistency of annotation standards, particularly in high-precision fields such as legal text classification, genomic data analysis, and autonomous driving perception systems.

In future work, we plan to further investigate how to enhance RealEXP's adaptability in scenarios where annotation standards are inconsistent, or data contain high levels of noise. This will include exploring more robust annotation correction mechanisms, introducing automated data cleaning methods, and integrating active learning or weakly supervised learning techniques to reduce dependence on expert annotations, thereby improving the applicability and generalization of the model across different domains.

Theorem 5. Let $n = |F|$ be the number of features. The traditional Shapley Value, when computed using the exhaustive enumeration method, requires a time complexity of $O(n \cdot 2^n)$. In contrast, the RealExp method requires the following:

1. $O(n)$ single-feature evaluations to obtain $\phi_i^{\text{independent}}$.
2. $O(n^2)$ multi-feature model evaluations to obtain ϕ_i^{Margin} through the computation of $[v(\{i, j\}) - v(\{j\})]$.
3. $O(n^2)$ similarity calculations for $s_{(i, j)}$.

Therefore, in the most straightforward scenario, RealExp can approximate the contribution values of all features in $O(n^2)$ time complexity (depending on the time required for a single model evaluation), which represents a significant improvement over the

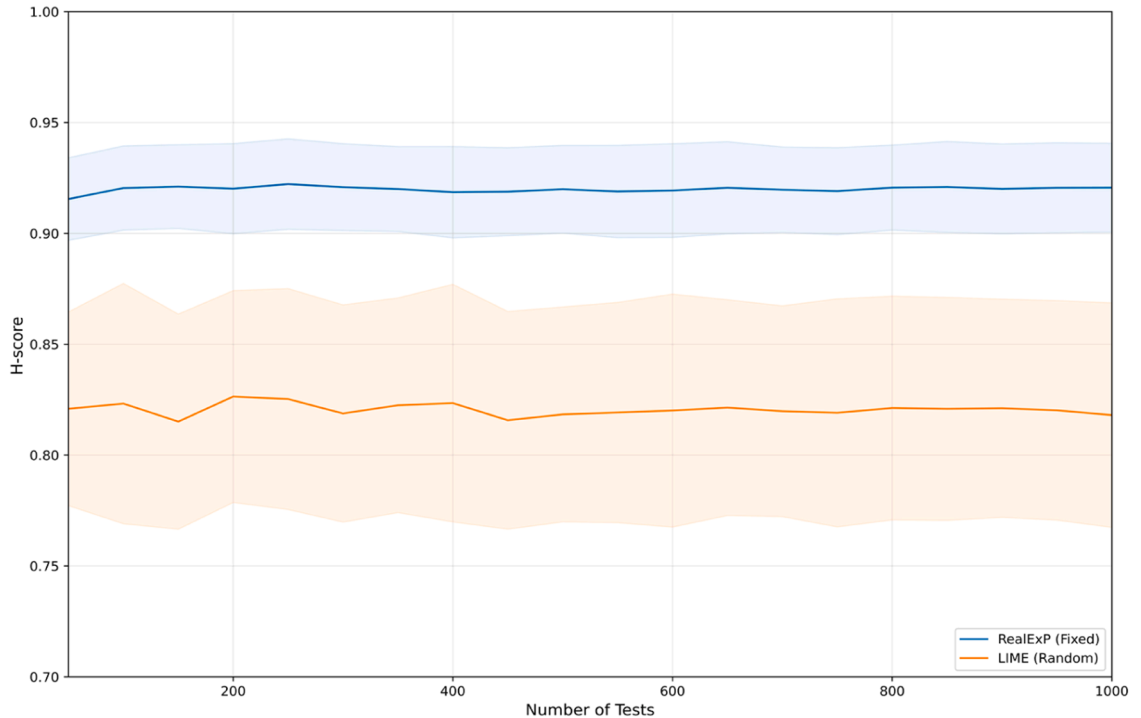


Fig. 19. Comparison of standard deviations for different sample sizes.

$O(n \cdot 2^n)$ complexity of the traditional Shapley Value.

Proof5: According to the definition in Section 3.2, $\varphi_i^{\text{independent}} = v(\{i\}) - v(\{\})$. To obtain the independent term for all i , it is sufficient to evaluate $v(\{\})$ once and $v(\{i\})$ for each feature i resulting in a total of $n + 1$ evaluations. For $\varphi_i^{\text{Margin}}$, the expression $v(\{i, j\}) - v(\{j\})$, is required for all $i \neq j$. This involves evaluating $n \cdot (n - 1)$ pairs, or $n^2 - n$ evaluations, which is of order $O(n^2)$. The similar calculation $s_{(i, j)}$ is typically obtained from external priori knowledge and is also $O(n^2)$.

Fig. 19 illustrates the performance of RealEXP and LIME in terms of $\mathcal{H}$ -score (interpretability metric) across different numbers of test iterations. The X-axis represents the number of tests, ranging from 50 to 1000, while the Y-axis indicates the $\mathcal{H}$ -score, with values approximately between 0.70 and 1.00. The results show that RealEXP (blue curve) has an average $\mathcal{H}$ -score of 0.92 with minimal variation, demonstrating high stability. In contrast, LIME (orange curve) has an average $\mathcal{H}$ -score of 0.82 but exhibits greater variability, with a wider standard deviation range, indicating a higher sensitivity to random factors and lower stability. As the number of tests increases, the mean values of both methods tend to stabilize; however, LIME still exhibits significant fluctuations. Overall, RealEXP outperforms LIME in interpretability metrics and demonstrates greater stability, whereas LIME is more affected by randomness, potentially leading to inconsistent results under different test conditions.

CAM, GradCAM, GradCAM++, and DeepSHAP utilize gradient backpropagation and visualize interpretability through heatmaps. To establish a unified evaluation standard in image classification tasks, the experiment standardizes the input of these gradient-based interpretability methods by applying a superpixel segmentation-based masking approach. Since heatmaps are presented as discrete distributions, evaluation can only be conducted using the $\mathcal{H}$ -score. In this experiment, the performance of these methods is compared with the proposed RealEXP on the ResNet model. The corresponding results are shown on Fig. 18.

Fig. 20 compares the performance of different interpretability methods (CAM, GradCAM, GradCAM++, DeepSHAP, and RealEXP) based on the $\mathcal{H}$ -score metric to evaluate their interpretability. The results show that RealEXP achieves the highest $\mathcal{H}$ -score (0.65), demonstrating its advantage in interpretability analysis. Notably, RealEXP employs a fixed perturbation ratio rather than a gradient-based interpretability technique, which contributes to its superior performance compared to other gradient-based methods, as well as its higher stability and interpretability.

Among gradient-based interpretability methods, DeepSHAP (0.63) ranks second in terms of $\mathcal{H}$ -score, indicating its strong ability to explain model decisions. GradCAM++ (0.61) and GradCAM (0.57) show improvements in H-score, reflecting advancements in gradient-based techniques, but they still fall short of RealEXP. In contrast, CAM (0.48) has the lowest $\mathcal{H}$ -score, suggesting its limited interpretability.

These results indicate that, compared to gradient-based interpretability techniques, the fixed perturbation ratio approach used by RealEXP provides more stable interpretability results and achieves the best performance in terms of the $\mathcal{H}$ -score metric, highlighting its potential in the field of XAI.

6. Conclusion

This paper introduces a novel interpretability computation method—Real Explainer (RealExp)—which decouples the Shapley Value into individual feature importance and feature correlation importance, enabling more precise interpretability analysis through feature similarity calculations. By considering both individual feature contributions and interactions among features, RealExp significantly enhances the accuracy and reliability of model explanations.

Furthermore, this paper proposes a new interpretability evaluation criterion that provides deeper insights into the decision paths of deep learning models, going beyond traditional accuracy-based evaluation metrics. In case studies on image classification, RealExp assists in selecting suitable pre-trained models, revealing the practical limitations of existing architectures (e.g., Vision Transformer) and validating its effectiveness in real-world pneumonia detection scenarios, where it also offers targeted improvement strategies.

In the analysis of weakly supervised anomaly detection in video sequences, experimental results demonstrate that by applying RealExp, pre-trained video segmentation techniques can be leveraged to enhance model interpretability. In text sentiment analysis,

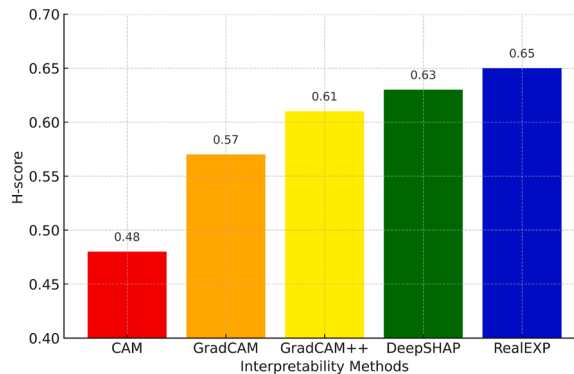


Fig. 20. Compare with gradient-based method on animal-151 dataset.

RealExp facilitates the design of a two-stage machine learning model that does not rely on pre-trained models, and this model outperforms the fine-tuned RoBERTa model.

Future research will focus on expanding the applications of interpretable machine learning across various domains of data processing and management, fostering continuous advancements and innovations in this field.

CRedit authorship contribution statement

Wen-Dong Jiang: Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Formal analysis, Data curation, Conceptualization. **Chih-Yung Chang:** Supervision, Resources, Project administration, Methodology. **Show-Jane Yen:** Writing – review & editing, Writing – original draft, Visualization, Validation. **Shih-Jung Wu:** Software, Resources. **Diptendu Sinha Roy:** Supervision, Data curation.

Data availability

Data will be made available on request.

References

- Ancona, M., Oztireli, C., & Gross, M. (2019). Explaining deep neural networks with a polynomial time algorithm for Shapley values approximation. In *Proceedings of the international conference on machine learning* (pp. 272–281).
- Anouk Stein, M. D., Wu, C., Carr, C., Shih, G., Dulkowski, J., kalpathy, L. C., Prevedello, L., Kohli, M., Mark McDonald, M. D., Phil Culliton, P., Safwan Halabi, M. D., & Xia, T. (2018). *RSNA pneumonia detection challenge*. Kaggle. <https://kaggle.com/competitions/rsna-pneumonia-detection-challenge>.
- Bakchy, S. C., Peyal, H. I., & Hoshen, M. R. (2024). Colon cancer detection using a lightweight-CNN with Grad-CAM++ visualization. In *Proceedings of the 3rd international conference on advanced electrical and electronic engineering (ICAEEE)* (pp. 1–6). IEEE.
- Banerjee, S., Mukherjee, S., Bandyopadhyay, S., & Pakray, P. (2023). An extract-then-abstract based method to generate disaster-news headlines using a DNN extractor followed by a transformer abstractor. *Information Processing & Management*, 60(3), Article 103291. <https://doi.org/10.1016/j.ipm.2023.103291>
- Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R., & Samek, W. (2016). Layer-wise relevance propagation for neural networks with local renormalization layers. In *Proceedings of the international conference on artificial neural networks* (pp. 63–71). https://doi.org/10.1007/978-3-319-44781-0_8
- Cao, Q., & Cheng, X. (2025). A hybrid feature fusion deep learning framework for multi-source medical image analysis. *Information Processing & Management*, 62(1), Article 103934. <https://doi.org/10.1016/j.ipm.2024.103934>
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the international conference on learning representations (ICLR)*.
- Guan, B., Zhu, X., & Yuan, S. (2024). A T5-based interpretable reading comprehension model with more accurate evidence training. *Information Processing & Management*, 61(2), Article 103584. <https://doi.org/10.1016/j.ipm.2023.103584>
- Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).
- Huang, X., & Marques-Silva, J. (2023). The inadequacy of Shapley values for explainability. *CoRR*. <https://doi.org/10.48550/arXiv.2302.08160>. abs/2302.08160.
- Huang, X., & Marques-Silva, J. (2024). On the failings of Shapley values for explainability. *International Journal of Approximate Reasoning*, Article 109112. <https://doi.org/10.1016/j.ijar.2023.109112>
- Jing, E., Liu, Y., Chai, Y., Sun, J., Samtani, S., Jiang, Y., & Qian, Y. (2023). A deep interpretable representation learning method for speech emotion recognition. *Information Processing & Management*, 60(6), Article 103501. <https://doi.org/10.1016/j.ipm.2023.103501>
- Li, X., Guo, S., Wu, J., & Zheng, C. (2025). An interpretable polytomous cognitive diagnosis framework for predicting examinee performance. *Information Processing & Management*, 62(1), Article 103913. <https://doi.org/10.1016/j.ipm.2024.103913>
- Li, X., Xiong, H., Li, X., Zhang, X., Liu, J., Jiang, H., Chen, Z., & Dou, D. (2023). G-LIME: Statistical learning for local interpretations of deep neural networks using global priors. *Artificial Intelligence*, 314, Article 103823. <https://doi.org/10.1016/j.artint.2022.103823>
- Li, Y., Liang, H., Zheng, H., & Yu, R. (2024). CR-CAM: Generating explanations for deep neural networks by contrasting and ranking features. *Pattern Recognition*, 149, Article 110251. <https://doi.org/10.1016/j.patcog.2024.110251>
- Li, Z., Jian, Y., Xue, Z., Zheng, Y., Zhang, M., Zhang, Y., Hou, X., & Wang, X. (2024). Text-enhanced knowledge graph representation learning with local structure. *Information Processing & Management*, 61(5), Article 103797. <https://doi.org/10.1016/j.ipm.2024.103797>
- Liu, Y., Wang, T., & Chu, F. (2024). Hybrid machine condition monitoring based on interpretable dual tree methods using Wasserstein metrics. *Expert Systems with Applications*, 235, Article 121104. <https://doi.org/10.1016/j.eswa.2023.121104>
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Neural information processing systems* (pp. 4765–4774).
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the neural information processing systems* (pp. 4765–4774).
- Niaz, A., Soomro, S., Zia, H., & Choi, K. N. (2024). Increment-CAM: Incrementally-weighted class activation maps for better visual explanations. *IEEE Access*, 12, 88829–88840. <https://doi.org/10.1109/ACCESS.2024.3413859>
- Nikhila, R., Valentin, G., Yuan-Ting, H., Ronghang, H., Chaitanya, R., Tengyu, M., Haitham, K., Roman, R., Chloe, R., Laura, G., Eric, M., Junting, P., Kalyan Vasudev, A., Nicolas, C., Chao-Yuan, W., Ross, G., Piotr, D., & Christoph, F. (2025). SAM 2: Segment anything in images and videos. In *Proceedings of the international conference on learning representations (ICLR)*.
- Petsiuk, V., Das, A., & Saenko, K. (2018). RISE: Randomized input sampling for explanation of black-box models. In *Proceedings of the British machine vision conference*. Article no.
- Rathee, G., Ahmad, F., Sandhu, R., Kerrache, C. A., & Azad, M. A. (2021). On the design and implementation of a secure blockchain-based hybrid framework for industrial Internet-of-Things. *Information Processing & Management*, 58(3), Article 102526. <https://doi.org/10.1016/j.ipm.2021.102526>
- Ribeiro, M., Singh, S., & Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144). <https://doi.org/10.1145/2939672.2939778>
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE international conference on computer vision* (pp. 618–626). <https://doi.org/10.1109/ICCV.2017.74>
- Shrikumar, A., Greenside, P., Shcherbina, A., & Kundaje, A. (2017). Learning important features through propagating activation differences. In *Proceedings of the international conference on machine learning* (pp. 3145–3153).
- Smilkov, D., Thorat, N., Kim, B., Viegas, F., & Wattenberg, M. (2017). SmoothGrad: Removing noise by adding noise. *arXiv*, 1706.03825.
- Tan, Z., Tian, Y., & Li, J. (2024). GLIME: General, stable and local LIME explanation. *Advances in Neural Information Processing Systems*, 36.
- Wang, L., & Lian, T. (2025). Category-guided multi-interest collaborative metric learning with representation uniformity constraints. *Information Processing & Management*, 62(1), Article 103937. <https://doi.org/10.1016/j.ipm.2024.103937>
- Yang, C., Zhang, H., Jiang, B., & Li, K. (2019). Aspect-based sentiment analysis with alternating co attention networks. *Information Processing & Management*, 56(3), 463–478. <https://doi.org/10.1016/j.ipm.2018.12.004>

- Zafar, M. R., & Khan, N. (2021). Deterministic local interpretable model-agnostic explanations for stable explainability. *Machine Learning & Knowledge Extraction*, 3, 525–541. <https://doi.org/10.3390/make3030027>
- Zahavy, T., Ben-Zrihem, N., & Mannor, S. (2016). Graying the black box: Understanding DQNs. In *Proceedings of the international conference on machine learning* (pp. 1899–1908).
- Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Proceedings of the European conference on computer vision* (pp. 818–833). https://doi.org/10.1007/978-3-319-10590-1_53
- Zhang, J., Bargal, S. A., Lin, Z., Brandt, J., Shen, X., & Sclaroff, S. (2018). Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10), 1084–1102. <https://doi.org/10.1007/s11263-018-1078-2>
- Zhang, P., Niu, Z., Ma, R., & Zhang, F. (2025). Multi-view graph contrastive representation learning for bundle recommendation. *Information Processing & Management*, 62(1), Article 103956. <https://doi.org/10.1016/j.ipm.2024.103956>
- Zhang, Y., Sun, F., Zhang, X., & Ma, X. (2025). Are multiple information sources better? The effect of multiple physicians in online medical teams on patient satisfaction. *Information Processing & Management*, 62(1), Article 103889. <https://doi.org/10.1016/j.ipm.2024.103889>