Causal Explanations for Image Classifiers

HANA CHOCKLER*, King's College London, UK
DAVID A. KELLY, King's College London, UK
DANIEL KROENING†, Amazon.com, Inc., USA
YOUCHENG SUN, Mohamed bin Zayed University of Artificial Intelligence, UAE and University of Manchester, UK

Existing algorithms for explaining the output of image classifiers use different definitions of explanations and a variety of techniques to find them. However, none of the existing tools use a principled approach based on formal definitions of cause and explanation.

In this paper we present a novel black-box approach to computing explanations grounded in the theory of actual causality. We prove relevant theoretical results and present an algorithm for computing approximate explanations based on these definitions. We prove termination of our algorithm and discuss its complexity and the amount of approximation compared to the precise definition.

We implemented the framework in a tool, rex, and we present experimental results and a comparison with state-of-the-art tools. We demonstrate that rex is the most efficient black-box tool and produces the smallest explanations, in addition to outperforming other black-box tools on standard quality measures.

JAIR Associate Editor: J. Christopher Beck

JAIR Reference Format:
Hana Chockler, David A. Kelly, Daniel Kroening, and Youcheng Sun. 2026. Causal Explanations for Image Classifiers. *Journal of Artificial Intelligence Research* 86, Article 9 (June 2026), 30 pages. DOI: 10.1613/jair.1.21939

1 Introduction

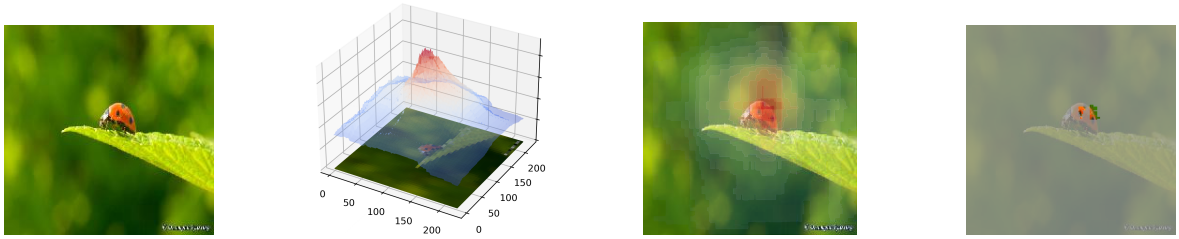
Neural networks (NNs) are now a primary building block of many computer vision systems. NNs are complex non-linear functions with algorithmically generated coefficients. In contrast to traditionally engineered image processing pipelines, it is difficult to retrace how the pixel data are interpreted by the layers of the network. Moreover, in many application areas, in particular healthcare, the networks are proprietary, making the analysis of the internal layers impossible. The "black box" nature of neural networks creates demand for eXplainable AI (XAI) techniques that show why a particular input yields the observed output without understanding the model's parameters and their influence on that output.

An explanation of an output of an automated procedure is essential in many areas, including verification, planning, diagnosis and the like. An informative explanation may alter a user's confidence in the result. Explanations are also useful for determining whether there is a fault in the automated procedure: if the explanation does not

*Corresponding Author.
†The work reported in this paper was done prior to joining Amazon.

Authors' Contact Information: Hana Chockler, ORCID: 0000-0003-1219-0713, hana.chockler@kcl.ac.uk, King's College London, London, UK; David A. Kelly, ORCID: 0000-0002-5368-6769, david.a.kelly@kcl.ac.uk, King's College London, London, UK; Daniel Kroening, ORCID: 0000-0002-6681-5283, dkr@amazon.com, Amazon.com, Inc., , USA; Youcheng Sun, ORCID: 0000-0002-1893-6259, youcheng.sun@mbzuai.ac.ae, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE and University of Manchester, Manchester, UK.

DOI: 10.1613/jair.1.21939



(a) Ladybug (b) Responsibility (c) Heat map (d) Explanation

Fig. 1. A ladybug (a), its responsibility map (b), the heat map (c), which is a projection of the responsibility map on a plane overlaid on the original image, and a causal explanation (d). The minimal causal explanation computed by our tool REX is less than 1% of the image.

make sense, it may indicate that the procedure is faulty. As argued by Bhusal et al. [2025], it is important to recognize that a "good" explanation should reveal what information a model is using to make its decision, even if that information is not intuitive to a human.

There have been a number of definitions of explanations over the years in various domains of computer science [Chajewska and Halpern 1997; Gärdenfors 1988; Pearl 1988], philosophy [Hempel 1965] and statistics [Salmon 1989]. Black-box explanations for the results of image classifiers are typically based on or are given in the form of a *ranking* of the pixels, which is a numerical measure of importance: the higher the score, the more important the pixel is for the NN's classification outcome. Often these rankings are presented in a form of a *heat map*, with higher scores corresponding to hotter areas.

This paper addresses the following research questions:

RQ1 What is a formal, rigorous definition of explanation, suitable for analyzing image classifiers?
RQ2 Can we compute explanations based on the definition above without opening the black box? What is the complexity of such computation?
RQ3 Is there an efficiently computable approximation for explanations?
RQ4 What are suitable quality measures for explanations?
RQ5 What is the quality of explanations computed by our algorithms compared to other XAI methods?
RQ6 Is there a trade-off between quality and compute cost of the explanations?
RQ7 Can black-box methods achieve the same quality of explanations as white-box methods?

Our algorithms are based on the formal definition of explanation in the theory of actual causality by Halpern [2019] and its adaptation to image classifiers by Chockler and Halpern [2024]. Essentially, an explanation in the context of image classifiers is a minimal subset of pixels with their original values that is sufficient for obtaining the same classification as the original image, given that the rest of the image has its values replaced with data from a predefined dataset. We view a black-box image classifier (a neural network) as a *causal model* (Section 3) of depth 2 (that is, one layer of input nodes, and one internal layer), with the classification itself computed in the single internal node, thus capturing the opacity of the network. The algorithm (Section 5) then calculates an approximate explanation, where the approximation is in the size of the explanation (Figure 1). That is, an explanation computed by our algorithm is sufficient for the classification, but it might not be minimal, though it is very close to minimal on real images: in our experiments, explanations average between $3\% - 5\%$ of the image, which is smaller than any of the other black-box tools. Size is a useful metric here as it indicates the degree of extraneous information contained in the explanation. We formally prove termination and complexity of our algorithm and discuss its approximation ratio.

We compare the experimental results of our implementation of the algorithm in a tool, ReX, with those of other XAI tools, both white-box and black-box, on a number of benchmark datasets: ImageNet, VOC2012, ECSSD, and a set of partially occluded images. Our results show that ReX is on par with the most efficient black-box tools in terms of execution time, while producing the smallest explanations. We also analyze the explanations using the standard measures of insertion and deletion curves [Petsiuk et al. 2018] and the intersection with the background or the occlusion. The results show that ReX outperforms the other tools on insertion curves and on the overlap with irrelevant parts of the image. We discuss the logic behind deletion curves and demonstrate that low deletion is a measure of the quality of the model for particular types of images, rather than for the quality of an explanation (Section 6.3). We also argue that explanations computed by ReX are so small that any further reduction in their size is not useful [D. Kelly et al. 2025].

The rest of the paper is organized as follows. Section 2 presents an overview of the related work on black-box explainable AI. Section 3 gives the necessary background on actual causality. Section 4 provides the theoretical foundations of our approach to explainability. Section 5 is a detailed overview of our algorithm, with the evaluation results discussed in Section 6. The tool ReX is open source and available at https://github.com/ReX-XAI/ReX. All models and benchmark sets are publicly available and referenced in Section 6.

2 Related Work

The taxonomy of XAI methods and styles is large [Schwalbe and Finzel 2024]. Our method and its implementation in a tool, ReX, produces local, post hoc, black-box causal explanations of image classifications. By local, we mean that the explanation applies only to one image, not to all images with the same classification. This is in contrast to a global explanation, which explains an entire class. Post hoc means that the method is applied to a model which is not inherently explainable and which has not been designed with explainability in mind. By black-box, we mean that there is no access at all to model architecture, gradients, or internals. An XAI method with free access to any layer of the model is *white-box*. An XAI tool with access to discretionary information, such as the gradient, we term *grey-box*. A proprietary system need not expose any information to the user other than the top-1 class. As our primary interest is in black-box, post hoc approaches, we only give a brief overview of other algorithms here.

The existing approaches to post hoc explainability can largely be grouped into two categories: propagation and perturbation. Propagation-based explanation methods are usually more computationally efficient. They back-propagate a model's decision to the input layer to determine the weight of each input feature for the decision. Grad-CAM [Selvaraju et al. 2017] only needs one backward pass and propagates the class-specific gradient into the final convolutional layer of a neural network to coarsely highlight important regions of an input image. Guided Back-Propagation (GBP) [Springenberg et al. 2015] computes the single and average partial derivatives of the output to attribute the prediction of a neural network. LRP [Bach et al. 2015] uses layer-wise back-propagation applied to all layers of the model. The model's output is decomposed into different degrees of relevance by applying different rules for each layer.

In contrast to propagation-based explanation methods, perturbation-based explanation approaches explore the input space directly in search of an explanation. The exploration/search often requires a large number of inference passes, which may incur significant computational cost, especially when compared to propagation methods. There are many ways to create these perturbations, most of which are based on random search or heuristics. The chief advantage of perturbation-based methods is that they are model independent and do not, in general, require any access to model internals.

Integrated Gradients (IG) [Sundararajan et al. 2017] relies only on the model's classification and gradient. As a model does not need to expose the gradient, we consider IG a *grey-box* method. It progressively introduces

pixels from an image onto an underlying baseline value and computes an explanation based on the effect of these introductions on the output gradient.

SHAP (SHapley Additive exPlanations) [Lundberg and Lee 2017b] is a family of different techniques, all of which seek to estimate Shapley values [Winter 2002]. Shapley values measure the contribution, either positive or negative, of a feature towards an outcome. GradientSHAP is a gradient-based method that approximates Shapley values via gradient integration, in a similar fashion to IG. There are numerous other methods for approximating Shapley values, among which KernelSHAP [Lundberg and Lee 2017b] and DeepLiftSHAP [Lundberg and Lee 2017b] are probably the most used. DeepLiftSHAP uses the DeepLift [Shrikumar et al. 2017] method whereas KernelSHAP uses the LIME framework.

Given a particular input, LIME [Ribeiro et al. 2016] samples its neighborhood and builds an *inherently explainable* model to approximate the system's local behavior. An inherently explainable model for LIME is usually a variant on some linear regression model. Owing to the high computational cost of this approach, LIME relies on a separate segmentation algorithm to break an image into *superpixels*, that is, clusters of connected pixels that share similar characteristics, such as color, intensity, or texture. The quality of this initial segmentation is vital for good performance, as LIME does not refine it further [Knab et al. 2024]. A LIME explanation is therefore necessarily tied to the provided segmentation. When this segmentation does not align with relevant features, LIME's performance suffers [Blake et al. 2025]. For instance, if the segmentation is too large or too crude, LIME's explanations contain irrelevant or unnecessary pixels.

In RISE [Petsiuk et al. 2018], the importance of a pixel is computed as the expectation over all local perturbations conditioned on the event that the pixel is observed. Rather than relying on an initial segmentation, RISE creates random occlusions of the input image. More recently, spectrum-based fault localization (SBFL) has been applied to explaining image classifiers. The technique has been implemented in the tool Deepcover [Sun et al. 2020]. While not an independent XAI method, NoiseTunnel [Adebayo et al. 2018] adds gaussian noise to the image before applying another XAI method, usually IG. The noise serves to smooth the resultant saliency map.

None of the methods mentioned above agree on what constitutes an explanation. LIME produces a locally explainable model, but the user is usually presented with a heat map of the local model's weights. RISE also produces a heat map, but its output is not directly comparable with LIME, due to the lack of a common definition. GradientSHAP output is also usually visualized as a heatmap, but one with both positive and negative values.

REX, the method and tool presented in this paper, is a perturbation-based approach and addresses the limitations of existing black-box methods in two aspects. REX does not rely on an initial segmentation as does LIME, nor does it simply rely on unguided random occlusions, as does RISE. The feature masking in REX uses causal reasoning to guide the creation of occlusions. REX output is also different from other methods: REX shows the user which pixels are sufficient for a classification. REX tests this sufficiency against the model itself, unlike LIME for example, which uses a locally trained model. Owing to its guided iterative refinement, REX is computationally efficient, while still producing explanations that are state of the art.

3 Background on Actual Causality

In this section, we briefly review the definitions of causality and causal models introduced by Halpern and Pearl [2005a] and relevant definitions of causes and explanations in image classification by Chockler and Halpern [2024]. The reader is referred to Halpern [2019] for further reading.

We assume that the world is described in terms of variables and their values. Some variables may have a causal influence on others. This influence is modeled by a set of *structural equations*. It is conceptually useful to split the variables into two sets: the *exogenous* variables, whose values are determined by factors outside the model, and the *endogenous* variables, whose values are ultimately determined by the exogenous variables. The structural equations describe how these values are determined.

Formally, a *causal model* M is a pair $(\mathcal{S}, \mathcal{F})$, where $\mathcal{S}$ is a *signature*, which explicitly lists the endogenous and exogenous variables and characterizes their possible values, and $\mathcal{F}$ defines a set of *(modifiable) structural equations*, relating the values of the variables. A signature $\mathcal{S}$ is a tuple $(\mathcal{U}, \mathcal{V}, \mathcal{R})$, where $\mathcal{U}$ is a set of exogenous variables, $\mathcal{V}$ is a set of endogenous variables, and $\mathcal{R}$ associates with every variable $Y \in \mathcal{U} \cup \mathcal{V}$ a nonempty set $\mathcal{R}(Y)$ of possible values for Y (i.e., the set of values over which Y *ranges*). For simplicity, we assume here that $\mathcal{V}$ is finite, as is $\mathcal{R}(Y)$ for every endogenous variable $Y \in \mathcal{V}$. The set $\mathcal{F}$ associates with each endogenous variable $X \in \mathcal{V}$ a function denoted F_X (i.e., $F_X = \mathcal{F}(X)$) that expresses the dependency of X on other variables. If all variables in M are Boolean, the model is called a *binary model*.

The structural equations define what happens in the presence of external interventions. Setting the value of some variable X to x in a causal model $M = (\mathcal{S}, \mathcal{F})$ results in a new causal model, denoted $M_{X \leftarrow x}$, which is identical to M, except that the equation for X in $\mathcal{F}$ is replaced by $X = x$.

Some salient features of causal models can be captured by causal graphs, where the nodes represent variables, and the arrows depict the direction of causality. A causal model M is *recursive* if its causal graph is acyclic. If M is a recursive causal model, then given a *context*, that is, a setting $\vec{u}$ for the exogenous variables in $\mathcal{U}$, the values of all the other variables are determined. In this paper, following the convention, we restrict the discussion to recursive models.

We call a pair $(M, \vec{u})$ consisting of a causal model M and a context $\vec{u}$ a *(causal) setting*. A causal formula ψ is true or false in a setting. We write $(M, \vec{u}) \models \psi$ if the causal formula ψ is true in the setting $(M, \vec{u})$. The $\models$ relation is defined inductively. $(M, \vec{u}) \models X = x$ if the variable X has value x in the unique solution to the equations in M in context $\vec{u}$. Finally, $(M, \vec{u}) \models [\vec{Y} \leftarrow \vec{y}]\varphi$ if $(M_{\vec{Y}=\vec{y}}, \vec{u}) \models \varphi$, where $M_{\vec{Y} \leftarrow \vec{y}}$ is the causal model that is identical to M, except that the variables in $\vec{Y}$ are set to $Y = y$ for each $Y \in \vec{Y}$ and its corresponding value $y \in \vec{y}$. The setting of $\vec{Y}$ to $\vec{y}$ effectively cuts the incoming causal transitions into $\vec{Y}$, replacing the variables in $\vec{Y}$ with the values $\vec{y}$. These values are then propagated to the variables that causally depend on $\vec{Y}$. This change of the values of some of the variables is called an *intervention*.

A standard use of causal models is to define *actual causation*: that is, what it means for some particular event that occurred to cause another particular event. There have been a number of definitions of actual causation given for acyclic models (e.g. [Beckers 2021; Glymour and Wimberly 2007; Hall 2007; Halpern and Pearl 2005a; Halpern 2019; Hitchcock 2007, 2001; Weslake 2015; Woodward 2003]). In this paper, we are focusing on *explainability*, which is traditionally defined using some form of actual causation. As we argue in the next section, in our setting, the causal models have a simple structure, and hence all the relevant definitions are simplified.

4 Theoretical Foundations of Our Approach

We approach the first two research questions theoretically.

RQ1 What is a formal rigorous definition of explanation, suitable for image classification?

Given an image classifier (e.g., a neural network) $\mathcal{N}$ and an input image x, we define a binary causal model $M_{\mathcal{N},x}$ as follows. The set $\mathcal{V} = \vec{V} \cup \{O\}$ of endogenous variables consists of a set $\vec{V}$ corresponding to the set of pixels $P(x)$ of x and the single output variable O. Essentially, $\vec{V}$ is a *mask*, indicating which pixels of x are visible and which are occluded, and the output variable O indicates whether the classification of a partially masked image stays the same as of the original image. To keep in line with the formal definitions of structural causal models, the values of $\vec{V}$ are determined by the set of exogenous variables $\mathcal{U}$.

Assigning 1 to a variable $v_i \in \vec{V}$ means that the pixel p_i, corresponding to v_i, has its original value (taken from x). Assigning 0 to this variable means that p_i is masked – replaced with some predefined masking value. The masking operation of $\mathcal{V}$ on $P(x)$ is denoted by $\mathcal{V} \odot P(x)$ and is the Hadamard product of these sets viewed as matrices of the same size (corresponding to the input size and shape of $\mathcal{N}$).

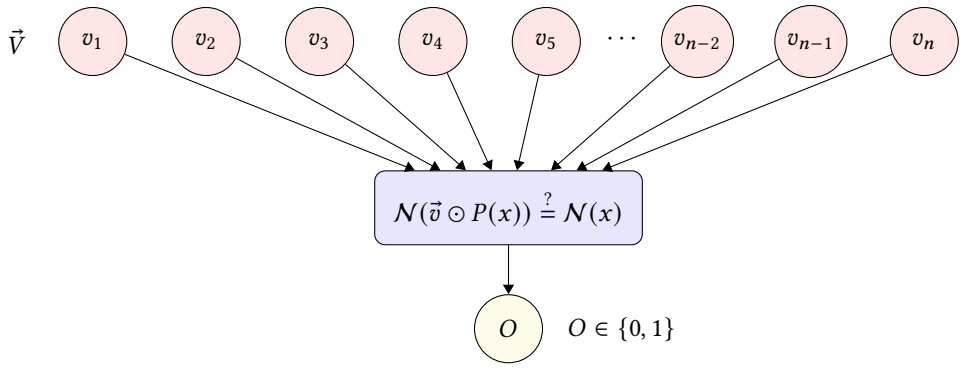


Fig. 2. A depth-2 binary causal model $M_{\mathcal{N},x}$ for an image x and a classifier $\mathcal{N}$. $\vec{v}$ is the vector of values of $\vec{V}$. The output $O \in \{0, 1\}$ indicates whether the classification of the Hadamard product of x and $\vec{V}$ is the same as the original classification.

The context $\vec{u}$ that assigns all variables in $\vec{V}$ the value 1 (i.e., none of the pixels are masked) corresponds to the fully unmasked image x. The value of O is 1 iff the output of $\mathcal{N}$ on $\mathcal{V} \odot P(x)$, the partially masked image defined by applying $\vec{V}$ to x, is $\mathcal{N}(x)$ and is 0 otherwise. Clearly, $O = 1$ if the image is fully unmasked, that is, $(M_{\mathcal{N},x}, \vec{u}) \models (O = 1)$.

We depict the reasoning process of $M_{\mathcal{N},x}$ in Figure 2, omitting the set of exogenous variables. The causal model has depth 2. In what follows, we omit the subscript $\mathcal{N}, x$ from the causal model notation if it is clear from the context. This construction assumes causal independence between the variables in $\vec{V}$. This is common to many approaches to causal and counterfactual explainable AI [Beckers 2022; Chockler and Halpern 2024; Mothilal et al. 2021; Poyiadzi et al. 2020; Sharma et al. 2020; Ustun et al. 2019], and is the *de facto* approach in all black-box explainable AI tools. We argue, however, that this is an accurate depiction in the case of images, and not just a convenient approximation. Image classification models perform classifications over data, not over what the image represents. This data encodes correlations which are due to the underlying data production method being causal, *i.e.* the real world. Obscuring or permuting pixels does not lead to a cascading change of other pixels values, as one would expect if pixels were causally connected, instead these permutations disrupt correlations in the data only.

Consider the "Photobombing" dataset, described in Section 6.1. This dataset is comprised of partially obscured images, obtained by overlaying random color patches over Imagenet images. An input from this dataset is a perfectly valid image. Indeed, obscuring a part of the input image either by introducing an artificial object or by positioning a real object in front of the primary subject of the classification does not lead to any change in unobscured pixels. Thus, pixel independence holds on the general data representation of images.

We refer the reader to Chockler and Halpern [2024] for a more in-depth discussion of causal independence between pixel values in image classification. We note that the causal independence assumption may not hold for concept bottleneck models and other models which claim to learn causes and not just correlations [Koh et al. 2020]. For these types of models, assuming independence may result in approximation which might lead to inaccurate results.

As we discuss below, there is a strong *correlation* between pixels in a given image, which is not surprising given that images usually capture objects in the real world. This does not contradict our claim of the lack of causal connection between the pixels in the image, due to the argument above.

Given a neural network $\mathcal{N}$ and an input image x, let $\vec{u}_1$ be the context that assigns 1 to all variables in $\vec{V}$, and let $\vec{u}_0$ be the context that assigns 0 to all these variables. We introduce the following definition of explanation.

Definition 4.1 (Single-Context Explanation)**.** A subset $\vec{V}_{exp}$ of $\vec{V}$ is a *single-context explanation* of a classification $\mathcal{N}(x)$ of an input image x by a classifier $\mathcal{N}$ if the following conditions hold:

EXIM1. $(M, \vec{u}_0) \models [\vec{V}_{exp} = 1](O = 1)$.

EXIM2. $\vec{V}_{exp}$ is minimal; there is no strict subset $\vec{V}'_{exp}$ of $\vec{V}_{exp}$ that satisfies EXIM1.

When there is no confusion, we call *single-context explanation* simply an *explanation*. As there is a one-to-one correspondence between the variables in $\vec{V}$ and the pixels of x, we also call the subset of pixels P_{exp} of x that corresponds to $\vec{V}_{exp}$ a *single-context explanation* of $\mathcal{N}(x)$.

In other words, an explanation is a minimal subset of pixels of a given input image x that is sufficient for the model $\mathcal{N}$ to classify the image, with all other pixels masked. Note that we do not assume that the classification of a fully masked input by $\mathcal{N}$ is different from $\mathcal{N}(x)$; if they are equal, the (single) explanation is an empty set.

Definition 4.1 matches other definitions of explanations in the causality landscape. So as not to interrupt the flow for the reader, we define and prove the relevant equivalences in appendix A. The reader can safely ignore this appendix, as it is not directly related to the methods and algorithms presented in this paper.

We now return to the task at hand, which is *computing* explanations.

RQ2 Can we compute explanations based on the definition above without opening the black box? What is the complexity of such computation?

Unfortunately, the precise computation of explanations is intractable—see appendix A for the proof. It is of little surprise, as all the existing definitions of explanations are intractable, except in some very simple special cases. There is, therefore, a justification for an approximation algorithm for explanations. Section 5, described in the next section, computes approximate explanations. It uses an approximate ranking of pixels according to their importance for the classification of the input image as a heuristic to guide explanation discovery. We formalize the notion of importance as follows.

Definition 4.2 (Sufficient responsibility)**.** The *degree of sufficient responsibility* of an explanation $\vec{V}_{exp}$ for the classification of x by $\mathcal{N}$ is defined as $1/|\vec{V}_{exp}|$. We also extend this value to all pixels in $\vec{V}_{exp}$. That is, v_i (and its matching pixel p_i) have the degree of sufficient responsibility $1/|\vec{V}_{exp}|$ for the classification of x by $\mathcal{N}$, where $\vec{V}_{exp}$ is the smallest explanation for the classification of x that contains v_i. If there is no explanation that contains v_i, then its degree of sufficient responsibility is defined as 0.

Similarly to the definition of explanation, Definition 4.2 has a match in the existing causality landscape, as we show in appendix A, where we also prove its intractability. The following observation describes a brute-force approach to computing sufficient responsibility, which is clearly exponential in the number of pixels of the image.

Observation 4.3. *Given an image x and its classification o, we can calculate the degree of sufficient responsibility of each pixel p_i of x by directly applying Definition 4.1, that is, by checking the conditions EXIM1 and EXIM2 for all subsets of pixels of x.*

In the next section, we describe an efficient approximation algorithm for computing explanations that is based on an efficiently computable approximate degree of responsibility.

5 The REX Algorithm

In this section we introduce our algorithm for computing approximate explanations. The high-level view of the algorithm is presented in Figure 3. The algorithm consists of two independent parts. The first part ranks pixels according to their approximate degree of sufficient responsibility (Definition 4.2); the second part is a *greedy algorithm* that extracts (approximate) explanations from this ranking.

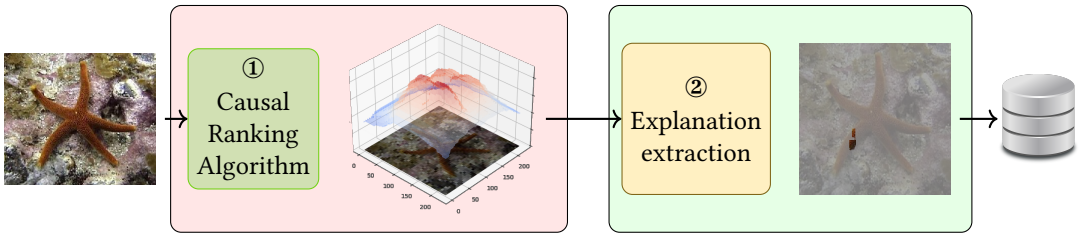


Fig. 3. High level overview of Algorithm 3. The causal ranking algorithm produces an approximate responsibility map (①). The pixels in the image are then ordered by their approximate responsibility, and the explanation extraction algorithm uses this ranking to produce an approximately minimal sufficient explanation (②), which captures the information required for the DNN to give the classification ('starfish' in this example).

The algorithm for computing an approximate degree of responsibility is based on the notion of a *superpixel* P_i. We slightly abuse the definition of a superpixel as used in LIME, where it refers to pixels which are clustered via some segmentation algorithm. In our context, however, it is simply some subset of pixels of a given image which have not been masked. REX does not rely on any externally provided initial segmentation. The algorithm instead partitions an input image into a small number of rectilinear superpixels, computes their degree of sufficient responsibility for the output, and then *iteratively refines* the superpixels with responsibility exceeding some predefined threshold.

The computation of the degree of sufficient responsibility of each superpixel for the output is precise and is described in Section 5.1. The approximation comes from the iterative refinement step in the algorithm, described in Section 5.2, and is due to the fact that the final ranking of pixels depends on the selected partition. To ameliorate the effect of a particular partition, the algorithm is repeated a number of times with partitions selected independently at random, and the results are averaged across all the partitions. Section 5.3 describes the averaging process and the greedy construction of an approximate sufficient explanation from the ranked list of pixels. Section 5.4 gives a step-by-step example to illustrate the working of the algorithm.

The scalability of the approach relies on the following observation.

Observation 5.1. *The pixels with the highest responsibility for the model's decision are located in superpixels with the highest responsibility.*

Intuitively, the observation holds when pixels with high responsibility do not appear in the superpixels surrounded by other pixels with very low responsibility for the input image classification outcome. Although this can happen in principle, especially in the case of adversarial images, we do not encounter this in practice. While the model has access to data only, this data has strong *correlations* due to the underlying data production mechanism.

5.1 Computing the Degree of Responsibility of a Superpixel

Given a set of pixels $\mathcal{P}$, we use $\mathbb{P}_i$ to denote a *partition* of $\mathcal{P}$, that is, a set $\{P_{i,j} : \bigcup P_{i,j} = \mathcal{P}$ and $\forall j \neq k, P_{i,j} \cap P_{i,k} = \emptyset\}$. The number of elements in $\mathbb{P}_i$ is a parameter, denoted by s; in our implementation, REX, we set $s = 4$ as this provides convenient partitioning. We refer to $P_{i,j}$ as *superpixels*. It is insufficient to consider each superpixel in isolation: there is no reason why any one superpixel should be the cause of a classification. Therefore we create combinations of superpixels from the power set of $\mathbb{P}_i$, denoted $2^{\mathbb{P}_i}$.

For an NN $\mathcal{N}$, an input x, and a partition $\mathbb{P}_i$, we can generalize Definition 4.2 to the set of *superpixels* defined by $\mathbb{P}_i$. We denote by $r_i(P_{i,j}, x, \mathcal{N}(x))$ the *sufficient responsibility* of a superpixel $P_{i,j}$ for $\mathcal{N}$'s classification of x, given $\mathbb{P}_i$.

Algorithm 1 *responsibility*$(x, \mathbb{P}_i)$

INPUT: an image x, a partition $\mathbb{P}_i$
OUTPUT: a responsibility map $R : \mathbb{P}_i \to \mathbb{Q}$

1: **for** each $P_{i,j} \in \mathbb{P}_i$ **do**
2: $\quad k \leftarrow \min_{x_m}\{\mathit{diff}(x_m, x) \mid x_m \in \tilde{X}_i^j\}$
3: $\quad r_{i,j} \leftarrow \frac{1}{k+1}$
4: **end for**
5: **return** $r_{i,0}, \ldots, r_{i,|P_i|-1}$

For a partition $\mathbb{P}_i$, we denote by X_i the set of *mutant images* obtained from x by masking subsets of $2^{\mathbb{P}_i}$, and by $\tilde{X}_i$ the subset of X_i that is classified as the original image x. Formally,

$$\tilde{X}_i = \{x_m : \mathcal{N}(x_m) = \mathcal{N}(x)\}.$$

We compute $r_i(P_{i,j}, x, \mathcal{N}(x))$, which is an approximation of the degree of sufficient responsibility (Definition 4.2) of each superpixel $P_{i,j}$ for the classification of x, in Algorithm 1. For a superpixel $P_{i,j}$, we define the set

$$\tilde{X}_i^j = \{x_m : P_{i,j} \text{ is not masked in } x_m\} \cap \tilde{X}_i.$$

For a mutant image x_m, we define $\mathit{diff}_i(x_m, x)$ as the number of superpixels in the partition $\mathbb{P}_i$ that are masked in x_m (that is, the difference between x and x_m with respect to $\mathbb{P}_i$). For an image y, we denote by $y(P_{i,j})$ an image that is obtained by masking the superpixel $P_{i,j}$ in y. The responsibility of a superpixel $P_{i,j}$ is calculated by Algorithm 1 as a minimum difference between a mutant image and the original image over all mutant images x_m that do not mask $P_{i,j}$, are classified the same as the original image x, and masking $P_{i,j}$ in x_m changes the classification.

5.2 Iterative Refinement of Responsibility

Algorithm 1 calculates the responsibility of each superpixel, subject to a given partition. It then proceeds with only the high-responsibility superpixels. It returns a responsibility map, R, which maps superpixels to the rational numbers $\mathbb{Q}$, indicating their degree of responsibility.

Note that in general, it is possible that all superpixels in a given partition have the same responsibility. Consider, for example, a situation where the explanation is right in the middle of the image, and our partition divides the image into four quadrants. Each quadrant would be equally important for the classification, hence we would not gain any insight into why the image was classified in that particular way. In this case, the algorithm starts again from another partition. It can also be the case that there might be multiple disjoint combinations of high responsibility superpixels in $\mathbb{P}_i$. Refining them all can be computationally expensive. Rex allows the user to choose from various pruning strategies to reduce the amount of work performed.

Different combinations of superpixels may have equal explanatory power. In the event that one superpixel is all that is required, therefore having responsibility 1, the further iterative subdivision of that superpixel is easy. However, if responsibility is split over multiple superpixels, we need a more refined approach. Take, for example, a situation where the left hand side of the image contains the explanation, as can be seen in Figure 10b. Here responsibility is split over two super pixels. Let us say that the passing combination of superpixels $P_c = \{0, 2\}$, assuming that we have numbered $P_{i,j}$ consecutively, and that $s = 4$. Each superpixel $P_{i,j}$ has responsibility 0.5. We cannot simply mask superpixel 2 while refining superpixel 0 as both together are required for the classification. We handle this problem by holding one superpixel to its original value while refining the other and then reversing the procedure. The superpixel held to its original value, but not being refined, does not get any additional responsibility.

Algorithm 2 *iterative_responsibility_refinement*$(x, \mathbb{P}_i)$

INPUT: an image x and a partition $\mathbb{P}_i$
OUTPUT: a responsibility map $R : \mathbb{P}_i \longrightarrow \mathbb{Q}$

1: $R \leftarrow$ *responsibility*$(x, \mathbb{P}_i)$
2: **if** R meets termination condition **then**
3: **return** R
4: **end if**
5: $R' \leftarrow \emptyset$
6: **for** each $P_c \in (2^{\mathbb{P}_i} - \emptyset)$ s.t $R(P_c) \neq 0$ **do**
7: $R' \leftarrow R' \cup$ *iterative_responsibility_refinement*(x, P_c)
8: **end for**
9: **return** R'

One partition is rarely sufficient for a high-quality (*i.e.* small) explanation, therefore we compute Algorithm 1 many times and compose the results, as shown in Algorithm 2. This calculates the responsibility for each superpixel (Line 1). If the termination condition is met (Lines 2–3), the responsibility map Q is updated accordingly. Otherwise, for each superpixel in $\mathbb{P}_i$, we refine it and call the algorithm recursively. We use $\cup$ to include these newly computed values in the returned map. The algorithm terminates when: 1) the superpixels in $\mathbb{P}_i$ are sufficiently refined (containing only very few pixels), or 2) when all superpixels in $\mathbb{P}_i$ have the same responsibility (this condition is for efficiency). In particular, if no further subdivision of a superpixel results in the desired classification, responsibility for all superpixels is 0 and the algorithm terminates.

5.3 Explanation Extraction

So far, we assume one particular partition $\mathbb{P}_i$, which Algorithm 2 recursively refines and calculates the corresponding responsibilities of superpixels in each step by calling Algorithm 1. We note that the choice of the initial partition over the image can affect the values calculated by Algorithm 2, as this partition determines the set of possible mutants in Algorithm 1. We ameliorate the influence of the choice of any particular partition by iterating the algorithm over a set of initial partitions. Twenty iterations of the algorithm will therefore yield 20 starting partitions, chosen at random. rex allows the user to choose from a number of different types of random to build these partitions, with uniform being the default.

In Algorithm 3, we consider N initial partitions and compute an average of the degrees of responsibility induced by each of these partitions. In the algorithm, $\mathbb{P}^x$ stands for a specific partition chosen randomly from the set of partitions, and r_p denotes the degree of responsibility of a pixel p w.r.t. $\mathbb{P}^x$.

Algorithm 3 has two parts: ranking all pixels (Lines 1–9) and constructing the explanation (Lines 10–17). The algorithm ranks the pixels of the image according to their responsibility for the model's output. Each time a partition is randomly selected (Line 3), the iterative responsibility refinement (Algorithm 2) is called to refine it into a set of fine-grained superpixels and calculate their responsibilities (Line 4). A superpixel's responsibility is evenly distributed to all its pixels, and the pixel-level responsibility is updated accordingly for each sampled partition (Lines 5–7). After N iterations, all pixels are ranked according to their responsibility r_p.

The remainder of Algorithm 3 follows the method for explaining the result of an image classifier by Sun et al. [2020]. That is, we construct a subset of pixels $\mathcal{E}$ to explain $\mathcal{N}$'s output on this particular input x *greedily*. We add pixels to $\mathcal{E}$ as long as $\mathcal{N}$'s output on $\mathcal{E}$ does not match $\mathcal{N}(x)$. This process terminates when $\mathcal{N}$'s output is the same as on the whole image x. The set $\mathcal{E}$ is returned as an explanation. We note that this extraction procedure is

Algorithm 3 *explanation*(x)

INPUT: an input image x, a parameter $N \in \mathbb{N}$
OUTPUT: an explanation $\mathcal{E}$

$r_p \leftarrow 0$ for all pixels p
for c in 1 to N **do**
  $\mathbb{P}^x \leftarrow$ sample a partition
  $R \leftarrow$ *iterative_responsibility_refinement*$(x, \mathbb{P}^x)$
  for each $P_{i,j} \in$ domain of R **do**
    $\forall p \in P_{i,j} : r_p \leftarrow r_p + \frac{R(P_{i,j})}{|P_{i,j}|}$
  end for
end for
pixel_ranking $\leftarrow$ pixels from high r_p to low
$\mathcal{E} \leftarrow \emptyset$
for each pixel $p_i \in$ *pixel_ranking* **do**
  $\mathcal{E} \leftarrow \mathcal{E} \cup \{p_i\}$
  $x^{exp} \leftarrow$ mask pixels of x that are **not** in $\mathcal{E}$
  if $\mathcal{N}(x^{exp}) = \mathcal{N}(x)$ **then**
    return $\mathcal{E}$
  end if
end for

not normally followed by other familiar XAI tools, as their several different definitions of explanation do not extend to finding minimal sets of pixels which induce the desired classification.

While we approximate the computation of an explanation in order to ensure efficiency of our approach, the algorithm is built on solid theoretical foundations, which distinguishes it from other random or heuristic-based approaches. In practice, while our algorithm uses an iterative average of a greedy approximation, it yields highly accurate results (Section 6). Furthermore, our approach is simple and general, and uses the neural network as a black-box.

5.4 Illustrative Example

To illustrate how REX works, consider Figure 4, which is classified as 'bus' by a ResNet50, even though there is an occlusion in the middle. Initially, REX picks an arbitrary partition of the image. This results in 15 combinations of masking superpixels, though in practice, we only need to examine 14 combinations as the 15^{th} is the entire image, which we have already tested. We use Algorithm 1 to calculate the responsibility of each superpixel. Those superpixels or combinations of superpixels which contribute towards the correct classification are further refined. This iterative refinement is a recursive application of the same initial random partitioning into four superpixels, as in Algorithm 2.

The responsibility maps shown in Figure 5 demonstrate the importance of Algorithm 3. The initial partitioning of the image can greatly affect the quality of the responsibility map. Figure 5a shows the responsibility map after one iteration of the algorithm. While it still indicates well the areas of interest in the image it is also rather crude. Figure 5b contains more refined information about the distribution of responsibility over the pixels in the image. We now have two distinct peaks of responsibility on either side of the central occlusion (a person). Further iterations bring about further refinement. By iteration 100 (Figure 5e), it is clear that the 'dominant' explanation is around the rear tire of the bus. The other explanation however, while depressed in relation to the tire, has not

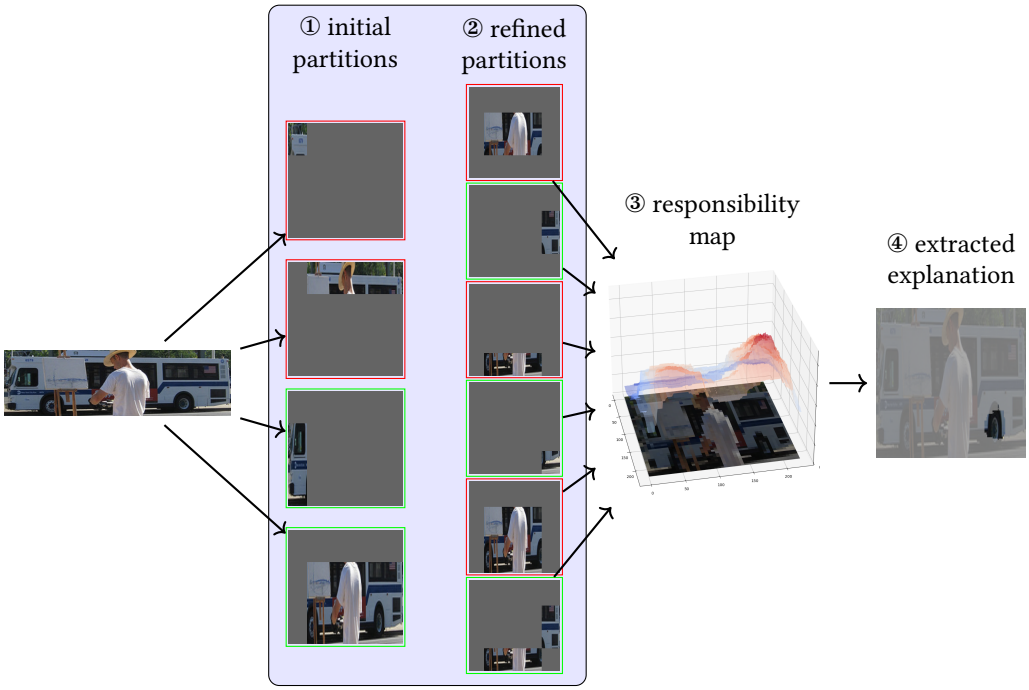
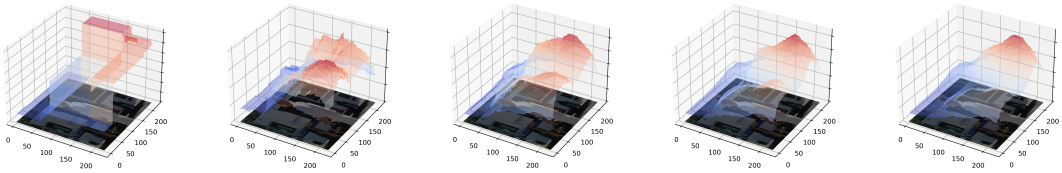


Fig. 4. The REX algorithm in action: REX creates an initial random partition of an image into 4 sections (①). All combinations of these sections are queried by the model, with further refinement applied to those sections or combinations of sections that meet the requirements (②). Some combinations, highlighted in green, are classified as bus, others, in red, are not. This results, after several iterations, in a detailed responsibility map (③), from which a minimal passing explanations can be extracted (④).



(a) $N = 1$ (b) $N = 10$ (c) $N = 30$ (d) $N = 60$ (e) $N = 100$

Fig. 5. Improvement of REX's pixel ranking on 'bus' (Figure 4) as the number of iterations N increases (Algorithm 3)

vanished and can still be found through searching the responsibility map [Chockler, D. A. Kelly, et al. 2025]. Moreover, a disjoint explanation exists for this image, as given in Figure 6a.

5.5 Termination and Complexity

It is clear that the algorithm necessarily terminates as soon as the parts can no longer be divided into superpixels. Moreover, the number of calls it performs to the model is linear in the size of the image, as proved in the following lemma.

(a) Non-contiguous explanation

(b) Contiguous explanation

Fig. 6. Two different explanations extracted from the 30 iteration ranking in Figure 5. Figure 6a contrasts interestingly with Figure 6b: it seems that the front tire by itself is not sufficient (a very small rectangular section of the rear tire is included in the explanation), but the rear tire alone is sufficienct.

Lemma 5.2. *The number of calls of Algorithm 3 to the model is $O(2^s nN)$, where s is the size of the partition in each step (in our setting $s = 4$), n is the number of pixels in the original image x, and N is the number of initial partitions.*

Proof. The computation of responsibilities of superpixels in one partition is $O(2^s)$, as the algorithm examines the effect of mutating each subset of the superpixels in the current partition. Note that s is a constant independent of the size of the image. The number of steps is determined by the termination condition on the size of a single superpixel, which in the worst case is the same as a single pixel. In our setting, the algorithm terminates when a single superpixel is contains fewer than 10 pixels[1]. However, in general, the algorithm can continue down to the level of a single pixel, thus resulting in n pixels in the last step. The algorithm performs N iterations, and every iteration uses a different initial partition. The parameter N is independent of the size of the image. □

Recall the research question **RQ3**:

RQ3 Is there an efficiently computable approximation to explanations?

The complexity analysis of Algorithm 3 in Lemma 5.2 provides an affirmative answer to this research question.

6 Evaluation

We conduct a large scale investigation into rex, comparing it with a host of popular XAI tools. Our answers to the research questions will largely remain empirical. Just as there is no universally agreed definition of explanation, there is also no single best way to evaluate the quality of an explanation.

RQ4 What are suitable quality measures for explanations?

A causal explanation is a minimal set of pixels from an image which are sufficient to obtain the same top-1 classification as the complete set of all pixels of the image. This immediately suggests size of explanation as a robust quality measure. As none of the tools (apart from rex) compute minimal, sufficient explanations, we use our mechanism of extraction (Section 5.3) on the output of all tools tested.

We also use a number of complementary measures as proxies for the quality of the explanations. Insertion and deletion curves, introduced in Petsiuk et al. [2018], are widely used to assess the quality of saliency maps. For insertion curves, the model confidence on a target class is measured as pixels are inserted over a baseline value. The order of insertion is derived from the map. If a map has accurately identified the most important pixels for a class, then the class confidence should rise quickly. This results in a large AUC (area under curve). Deletion curves are calculated in the opposite way, replacing the pixels with a baseline value. Again, if the map has identified the most important pixels accurately, then the model confidence should drop quickly, resulting in a low AUC.

[1] rex also terminates when a user-defined work budget is exceeded.

While a REX explanation is not a map, we calculate insertion and deletion over the responsibility map. The AUC for both insertion and deletion are obviously tied to the original confidence of the model. To allow for a fair comparison between images with different initial confidences, we follow Calderón-Peña et al. [2024] and normalize all curves by the initial confidence of the model on the image under test. Note that this can result in the the insertion curve AUC being greater than 1. This phenomenon occurs when the confidence on the entire image is lower than the confidence of intermediate insertion stages. This makes intuitive sense: if the pixel ranking is accurate, then those pixels that either do not contribute towards the classification, or reduce confidence, are left to be added last. Without these 'negative' pixels, confidence is potentially much higher.

Comparison with human segmentation. For those datasets that contain segmentation information, we also measure the intersection of the minimal explanation with that segmentation. This segmentation is human-provided and we argue corresponds most closely with what a human considers acceptable. On VOC2012, for example, we would like a minimal explanation to reside mostly inside the human-provided segmentation mask and therefore contain only a small number of pixels outside the mask. For datasets that feature occluded objects, a good explanation would have as few pixels from inside the occlusion region as possible, as the occlusion should not make a substantial contribution towards the model's classification.

To answer **RQ4**, we argue that the size of explanations, as well as insertion curves and the intersection with the human segmentation are measures that match our intuition. However, we argue that deletion curves are not, as they remain relatively high in presence of multiple detected explanations (Section 6.3).

6.1 Experimental Setup

We answer research questions **RQ5–RQ7** experimentally. We have implemented the proposed explanation approach in the publicly available tool REX. For all other XAI tools, apart from RISE, we used the Captum PyTorch library [Kokhlikyan et al. 2020][2]. For RISE, we used the authors' implementation[3], which we lightly altered to use PyTorch rather than the original TensorFlow framework.

In the evaluation, we compare REX with a wide range of explanation tools, specifically GRAD-CAM [Selvaraju et al. 2017], KERNELSHAP [Lundberg and Lee 2017b], GRADIENTSHAP [Lundberg and Lee 2017a], RISE [Petsiuk et al. 2018], LIME [Ribeiro et al. 2016], IG [Sundararajan et al. 2017] and NOISETUNNEL [Smilkov et al. 2017]. For the ResNet50 model only, we also include LRP [Bach et al. 2015]. Unfortunately, the Captum version of LRP could not run on the other tested models.

While our main interest is in black-box explainability, we have included methods which require access to the model gradient, such as GRADIENTSHAP and IG, and internal model layers, *i.e.* GRAD-CAM. This is largely due to the relative paucity of purely black-box XAI methods for image data.

We use 4 data sets: ImageNet-1k-mini validation [Russakovsky et al. 2015], ECSSD [Shi et al. 2016], Pascal VOC2012 [Everingham et al. n.d.], and a "Photobombing" dataset we created. Imagenet1k-mini comes with labels for ground truth. VOC2012 has labels and segmentation data. ECSSD comprises complex images which come with a human-provided segmentation. We created the "Photobombing" dataset by inserting black occlusions into ImageNet images, meaning we have the original label and also know the pixel coordinates of the occlusions. These occlusions are placed so as not to change the model classification. We use the TORCHVISION[4] implementations of ResNet50, ViT and ConvNext-Large with default weights on all data sets.

All experiments were conducted using a server running Ubuntu 20.04 with an Nvidia A40 GPU. All tools were used with default settings. For REX in particular, this means that it performs 20 iterations of the algorithm (N in Algorithm 3), with a minimum superpixel size of 10 pixels. The pruning strategy for the work queue is "area"

[2] https://www.captum.ai
[3] https://github.com/eclique/RISE
[4] https://pytorch.org/vision/stable/index.html

(the passing mutants are ordered by size, smallest to largest) and only one item is kept in the work queue at a time, so we only refine the smallest passing mutant. Partitions are created uniformly at random. Other strategies and partitioning distributions are available in the configuration.

rex uses four superpixels per partition. This is practical for images as we can split the image with just one call to a random number generator. The number of mutants produced is also relatively small (15), which can usually be fit into a single batch for model inference. Having more initial superpixels does not lead to greater expressivity: we iteratively refine passing combinations of superpixels, essentially recreating the more detailed initial partition which would be produced by a greater number of superpixels. The termination conditions for the partition refinement in Algorithm 2 are: 1) the area of a superpixel is less than 10 pixels of (resized) input image, or 2) the four superpixels share the same responsibility.

6.2 Results

RQ5 What is the precision of explanations computed by our algorithm compared to other XAI methods?
RQ6 Is there a trade-off between precision and compute cost of the explanations?
RQ7 Can black-box methods achieve the same quality of explanations as white and grey-box methods?

Tables 1 to 3 show the experimental results of evaluation of rex against 8 other XAI tools over 4 datasets, with three different models. The columns in the tables are: mean area (area), σ of area (standard deviation), normalized insertion curve AUC (ins), normalized deletion curve AUC (del), proportion of explanation inside the relevant segment (IN), and the fraction of the explanation that is outside the relevant segment (OUT), shown only where appropriate. Bold indicates best result in category for all columns. Note that while we have indicated lowest AUC for deletion as 'best', lower here is not necessarily better, as we argue in Section 6.3.

Resnet. Table 1 shows the results with the ResNet50 model. rex consistently produces the most smallest explanations. This is reinforced by the observations that rex also produces the highest insertion AUC of any of the tools. rex does particularly well on the 'Photobombing' dataset, where its explanations are almost entirely (0.97) disjoint from the inserted occlusion (Figure 13). Figure 7 shows that rex has the most concentrated explanations and consistent results, with very few outliers. The grey-box tools perform worse than the other methods. Note that rex even outperforms Grad-CAM, a white-box method.

ViT. Table 2 shows the results with the 'vit_b_32' model from torchvision. Explanations across all tools are much larger than for the ResNet50, which seems to be a feature of this model. rex, however, still manages to produce explanations with the lowest number of redundant pixels. In general, rex, lime and Grad-CAM are the best performing tools. Even in cases where lime slightly outperforms rex (e.g., on ECSSD IN), with 0.73 of its explanation inside the segmentation against 0.72, the effect is very small. Figure 8 shows that rex again has the most concentrated explanations and consistent results, with very few outliers. lime is the closest comparable tool, though it has many more outliers and a higher median value. The significant difference in explanation size between the ResNet model and ViT model is, perhaps, of independent interest.

ConvNext. Table 3 shows the results with the 'convnext_large' model from torchvision. rex, lime, Grad-CAM and rise are the best performing tools. It is interesting to note that 3 of these tools fall into our black-box category. Figure 9 shows that rex has the best performance again, this time slightly ahead of Grad-CAM, as Grad-CAM has many more outliers than rex and a slightly larger inter-quartile range. Explanation size is comparable with that of the ResNet model, revealing the ViT model to be somewhat of an outlier.

Table 1. Results for ResNet50 model.

(a) Results on Voc

tool	area (↓)	std	ins (↑)	del	IN (↑)	OUT (↑)
Grad-CAM	0.0832	0.1202	1.0488	0.2756	0.3202	–
noiseTunnel	0.2867	0.2627	0.6953	0.0851	0.3241	–
lrp	0.3936	0.2828	0.5267	0.2953	0.2093	–
ig	0.4904	0.2768	0.4899	0.1228	0.2215	–
GradientShap	0.3894	0.3002	0.5935	**0.0651**	0.2872	–
KernelShap	0.7741	0.2084	0.188	0.1941	0.1636	–
lime	0.1124	0.1616	0.963	0.2329	0.3039	–
rex	**0.0427**	**0.0505**	**1.2218**	0.2148	**0.6432**	–
rise	0.1271	0.1547	0.9358	0.2264	0.2861	–

(b) Results on Imagenet

tool	area (↓)	std	ins (↑)	del	IN (↑)	OUT (↑)
Grad-CAM	0.0486	0.0839	1.0059	0.4073	–	–
noiseTunnel	0.1847	0.2131	0.7612	0.1274	–	–
lrp	0.2749	0.2418	0.5972	0.3567	–	–
ig	0.4063	0.264	0.534	0.1767	–	–
GradientShap	0.2861	0.2696	0.6474	**0.0983**	–	–
KernelShap	0.6686	0.2472	0.2549	0.2649	–	–
lime	0.0649	0.0969	0.9871	0.3313	–	–
rex	**0.0333**	**0.0351**	**1.1101**	0.3403	–	–
rise	0.1026	0.1254	0.8898	0.3728	–	–

(c) Results on "Photobombing"

tool	area (↓)	std	ins (↑)	del	IN (↑)	OUT (↑)
Grad-CAM	0.0455	0.0642	0.9843	0.3625	–	0.8505
noiseTunnel	0.1565	0.1832	0.7604	0.1348	–	0.8732
lrp	0.2429	0.205	0.6049	0.3687	–	0.9058
ig	0.4184	0.2311	0.5178	0.1853	–	0.8806
GradientShap	0.296	0.2373	0.637	**0.114**	–	0.8533
KernelShap	0.6601	0.2246	0.2527	0.2612	–	0.9188
lime	0.0507	0.0523	0.9937	0.3046	–	0.8921
rex	**0.0297**	**0.026**	**1.0481**	0.2947	–	**0.9739**
rise	0.0787	0.0942	0.8906	0.2977	–	0.8798

(d) Results on ECSSD

tool	area (↓)	std	ins (↑)	del	IN (↑)	OUT (↑)
Grad-CAM	0.0836	0.1288	1.0226	0.2915	**0.6654**	–
noiseTunnel	0.2695	0.276	0.7194	0.119	0.5955	–
lrp	0.3545	0.292	0.5721	0.305	0.3418	–
ig	0.4437	0.3016	0.5552	0.1377	0.4187	–
GradientShap	0.3525	0.3182	0.6444	**0.0977**	0.5379	–
KernelShap	0.7257	0.2403	0.2348	0.2369	0.244	–
lime	0.1013	0.1579	1.0024	0.2404	0.6538	–
rex	**0.0423**	**0.0522**	**1.2176**	0.2214	0.6205	–
rise	0.1165	0.1599	0.95	0.2752	0.5451	–

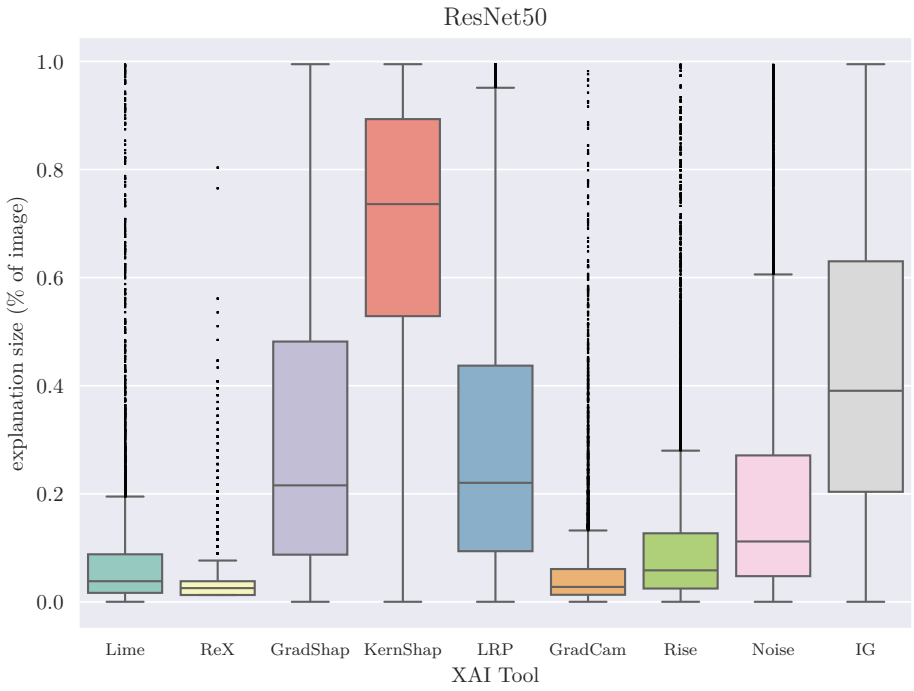


Fig. 7. Box plot of all tools over all datasets with a ResNet50 model. rex has the lowest median value and also the smallest number of outliers.

Table 2. Results for ViT model.

(a) Results on Voc

tool	area (↓)	std	ins (↑)	del	IN (↑)	OUT (↑)
Grad-CAM	0.7148	0.181	0.2466	0.1889	0.1727	–
noiseTunnel	0.4988	0.3019	0.4565	0.0633	0.2315	–
ig	0.5652	0.2632	0.4274	0.0943	0.1946	–
GradientShap	0.5021	0.2992	0.4761	**0.0551**	0.2267	–
KernelShap	0.669	0.2603	0.2579	0.2579	0.1686	–
lime	0.277	0.2218	**0.7191**	0.112	0.26	–
rex	**0.2206**	**0.1711**	0.6985	0.1377	**0.5496**	–
rise	0.4049	0.2491	0.5986	0.097	0.2365	–

(b) Results on Imagenet

tool	area (↓)	std	ins (↑)	del	IN (↑)	OUT (↑)
Grad-CAM	0.6277	0.203	0.2943	0.2538	–	–
noiseTunnel	0.3903	0.2711	0.5341	**0.098**	–	–
ig	0.4706	0.2505	0.4833	0.1481	–	–
GradientShap	0.3871	0.2691	0.5586	0.0916	–	–
KernelShap	0.5483	0.2651	0.3541	0.3519	–	–
lime	0.2246	0.1937	**0.7264**	0.1668	–	–
rex	**0.1689**	**0.133**	0.687	0.2015	–	–
rise	0.3356	0.2333	0.5997	0.1664	–	–

(c) Results on "Photobombing"

tool	area (↓)	std	ins (↑)	del	IN (↑)	OUT (↑)
Grad-CAM	0.6025	0.1907	0.303	0.2715	–	0.9196
noiseTunnel	0.3647	0.2447	0.5439	0.1012	–	0.8835
ig	0.4598	0.2251	0.4806	0.1486	–	0.8889
GradientShap	0.3726	0.2465	0.5607	**0.0967**	–	0.8802
KernelShap	0.5168	0.233	0.37	0.3665	–	0.9186
lime	0.1748	0.1458	**0.7365**	0.1778	–	0.8891
rex	**0.1402**	**0.1046**	0.6883	0.2073	–	**0.9551**
rise	0.2803	0.2056	0.6119	0.1741	–	0.8915

(d) Results on ECSSD

tool	area (↓)	std	ins (↑)	del	IN (↑)	OUT (↑)
Grad-CAM	0.6741	0.2181	0.2731	0.2344	0.2494	–
noiseTunnel	0.4298	0.3001	0.5371	0.0731	0.4296	–
ig	0.4727	0.2723	0.5184	0.1082	0.3409	–
GradientShap	0.4179	0.2929	0.5772	**0.0685**	0.4134	–
KernelShap	0.6192	0.2729	0.321	0.3188	0.2466	–
lime	0.2698	0.2319	**0.7392**	0.123	**0.5548**	–
rex	**0.2008**	**0.1653**	0.7286	0.1475	0.5334	–
rise	0.3687	0.2585	0.6597	0.1178	0.4468	–

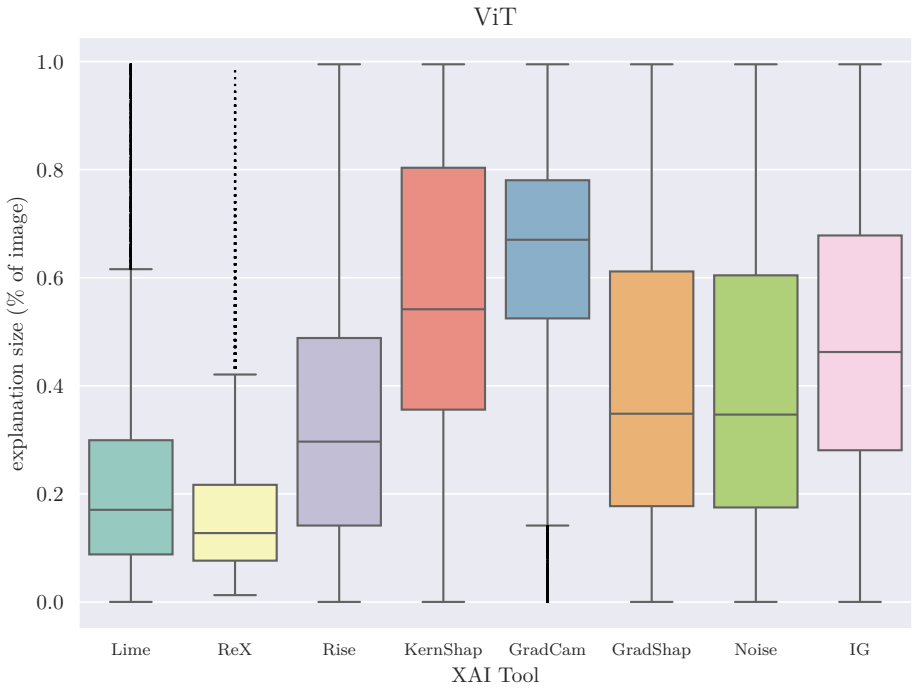


Fig. 8. Box plot of all tools over all datasets with a ViT model. rex has the lowest median value and also the smallest number of outliers. ViT explanations are noticeably larger than those for ResNet50 and ConvNext.

Table 3. Results for convnext model.

(a) Results on Voc

tool	area (↓)	std	ins (↑)	del	IN (↑)	OUT (↑)
Grad-CAM	0.0556	0.0948	0.8635	0.2497	0.3345	–
noiseTunnel	0.1972	0.2443	0.689	0.117	0.3614	–
ig	0.4152	0.2565	0.465	0.2387	0.2025	–
GradientShap	0.2268	0.2609	0.6411	**0.0998**	0.3327	–
KernelShap	0.5739	0.3001	0.2926	0.2924	0.1678	–
lime	0.0689	0.0972	0.8096	0.2908	0.3042	–
ReX	**0.0361**	**0.038**	**0.876**	0.2057	**0.6295**	–
rise	0.0759	0.1129	0.8136	0.3005	0.2863	–

(b) Results on Imagenet

tool	area (↓)	std	ins (↑)	del	IN (↑)	OUT (↑)
Grad-CAM	0.0337	0.0553	0.8613	0.4413	–	–
noiseTunnel	0.1116	0.1608	0.7682	**0.1886**	–	–
ig	0.3414	0.2168	0.4993	0.3471	–	–
GradientShap	0.1394	0.183	0.706	0.1678	–	–
KernelShap	0.4357	0.2855	0.4042	0.4061	–	–
lime	0.0493	0.0707	0.8228	0.4747	–	–
ReX	**0.0287**	**0.0236**	**0.8788**	0.3487	–	–
rise	0.0615	0.084	0.8058	0.5136	–	–

(c) Results on "Photobombing"

tool	area (↓)	std	ins (↑)	del	IN (↑)	OUT (↑)
Grad-CAM	0.0426	0.0527	0.8584	0.3688	–	0.8695
noiseTunnel	0.0768	0.0925	0.7916	**0.2077**	–	0.8755
ig	0.3144	0.1731	0.5454	0.3847	–	0.8783
GradientShap	0.1335	0.1351	0.7095	0.2186	–	0.85
KernelShap	0.3829	0.2405	0.4683	0.468	–	0.9183
lime	0.0513	0.0551	**0.8643**	0.4161	–	0.8947
ReX	**0.0296**	**0.0236**	0.8475	0.3173	–	**0.9748**
rise	0.063	0.0723	0.8129	0.4769	–	0.9015

(d) Results on ECSSD

tool	area (↓)	std	ins (↑)	del	IN (↑)	OUT (↑)
Grad-CAM	0.0574	0.1023	0.8865	0.2694	**0.6846**	–
noiseTunnel	0.2164	0.2734	0.6424	0.1067	0.5913	–
ig	0.421	0.2925	0.4202	0.2102	0.363	–
GradientShap	0.2434	0.284	0.5911	**0.0987**	0.5671	–
KernelShap	0.5616	0.3152	0.2898	0.2977	0.2468	–
lime	0.0849	0.1377	0.7157	0.3183	0.5841	–
rex	**0.0364**	**0.0405**	**0.9012**	0.2077	0.5722	–
rise	0.0823	0.1387	0.7996	0.3073	0.5045	–

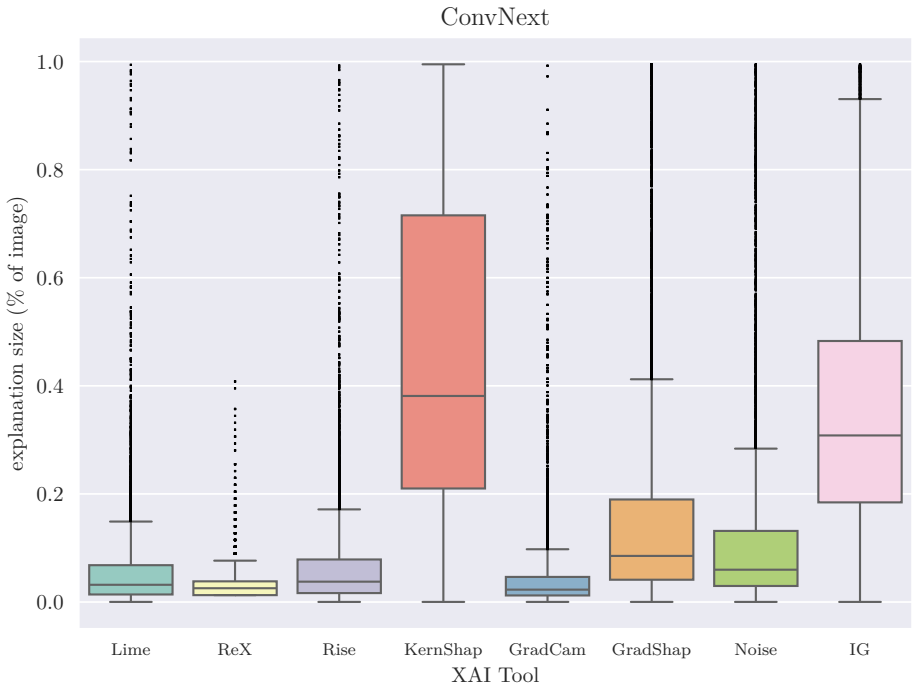
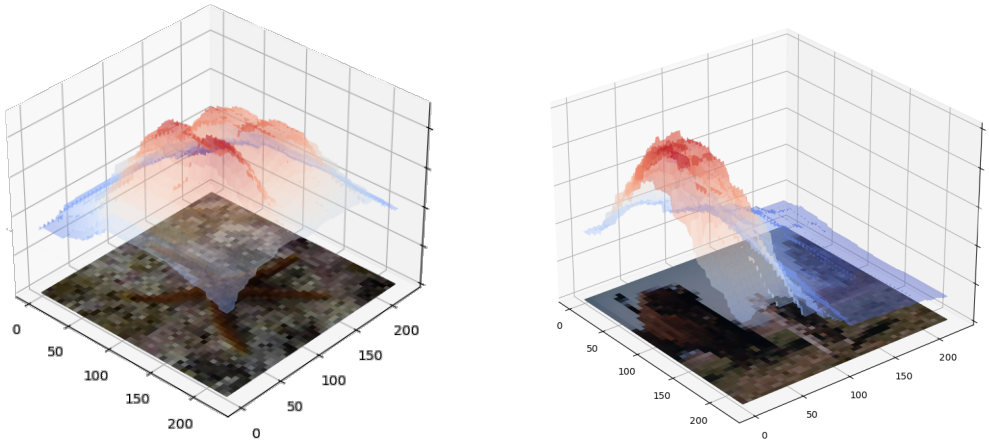


Fig. 9. Box plot of all tools over all datasets with a ConvNext model. rex has the lowest median value and also the smallest number of outliers. It even outperforms Grad-CAM, a purely white-box tool.



(a) Starfish (b) Bison

Fig. 10. An image with a high deletion curves does not indicate a low quality explanation. The starfish (Figure 10a) has a normalized deletion of 0.32 because their are multiple points of high responsibility in the image. These points indicate the presence of multiple explanations. The bison (Figure 10b) has a low deletion of 0.1 because there is only one point of interest in the image. Deletion is a difficult measure to interpret.

Runtime. We measured the running time of all tools. As all tools were evaluated on the same hardware with the same models and datasets, the results should be consistent. The white and grey-box methods are clearly the fastest, as expected. Indeed, Grad-CAM is essentially instantaneous, as the overhead over one call to the model is insignificant. No black-box tool can compete with this, by the nature of black-box tools. All the gradient-based methods took $\approx$ 3 seconds to produce initial maps. Note that this is the time taken to generate initial saliency, as minimal pixel subsets still need to be extracted. Both rex and lime took $\approx$ 4s to produce maps, with rise being much slower ($\approx$ 10s). Taking into account the extra time for the explanation extraction (which only rex does by default), the efficiency picture changes slightly. rex's total compute time increases only slightly (to $\approx$ 6s) due the the uniformly small size of its explanations. Other tools suffer more, even Grad-CAM's average compute time increases to a total of $\approx$ 4s. This is due to the relatively noisy output of Grad-CAM, requiring more work from the extraction algorithm as a result. rex is an efficient black-box tool: for **RQ6** therefore, our results show that there is little trade-off between quality and compute cost.

6.3 Discussion

Utility and interpretation of deletion curves. Petsiuk et al. [2018] stated that a high insertion and a low deletion is preferable. This assertion relies on the assumption that there is only one explanation for the image's classification, or that the XAI tool finds only one explanation. For deletion AUC to remain low, saliency values outside of the single 'explanation' must be strictly uninformative: the saliency map cannot provide information about any other substructures present in the image. It must destroy that signal. Essentially, pixel deletion after the first discovered explanatory pixel set must be close to random, breaking up information from other potential explanations. Given that a large body of work is dedicated to 'cleaning up' saliency maps, especially from Grad-CAM-style tools [Adebayo et al. 2018; Chattopadhay et al. 2018; Smilkov et al. 2017], it is not surprising that this genuine signal is lost.

(a) Explanation 1 (b) Explanation 2 (c) Explanation 3 (d) Explanation 4

Fig. 11. The responsibility map REX provides is sufficiently detailed to extract multiple valid explanations for most images. These starfish explanations were extracted from Figure 10a.

REX does not work like this: the responsibility map provides detailed information over the entire pixel space. Because the map is not itself the explanation, there is no motivation to remove or delete 'inconvenient' hot-spots. Moreover, because REX is not using activation, which can be noisy, every hot-spot in the REX map is indeed associated to some degree with the desired classification. REX utilizes the richness of its responsibility map to discover multiple, different explanations of an image [Chockler, D. A. Kelly, et al. 2025]. This is unlike the other tools in our experimental comparison, all of which return one explanation by design.

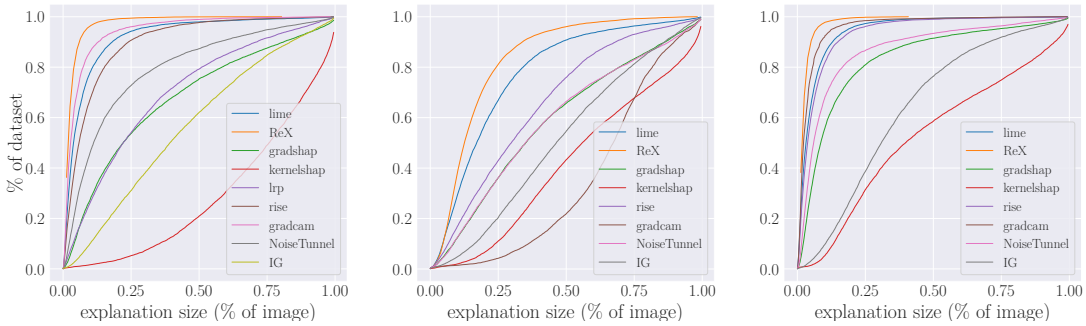
As an example, if we examine Figure 10a, we have an image with a normalized insertion curve of 0.97 but a normalized deletion curve of 0.32. There would appear to be tension between the two numbers, as a high insertion suggests a high quality explanation whereas a high deletion suggests a low quality explanation. The mystery is resolved, however, by noting that the image contains multiple sufficient explanations. Indeed, there are four distinct peaks of responsibility in Figure 10a, each of which corresponds to an independent explanation for starfish (Figure 11). It takes a long time to delete all of the pixels from the well-defined peaks in the responsibility map.

In Tables 1 to 3 we follow the usual procedure of indicating the lowest deletion curve value, but stress that we do not consider the deletion curve as a robust measure for explanation quality.

Differences between methods. Broadly speaking, the tools can be split into three different sets. The gradient-based methods perform least well on all metrics, including overlap with a human-provided segmentation mask. The explanations they provide are diffuse and large. NOISETUNNEL uses IG as its primary saliency method, and in return for a relatively small computational overhead, NOISETUNNEL greatly improves the explanation quality over the IG baseline.

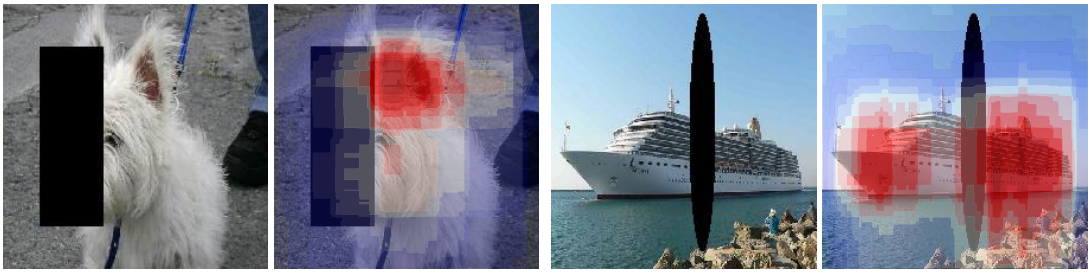
Figure 12 shows cumulative plots of explanation sizes over the different tested models. Again, REX dominates the other curves. LIME and GRAD-CAM are, in general, the next best performing tools. The poor performance (almost linear) of KERNELSHAP is interesting, suggesting that image explainability is not a natural fit for this method.

LIME, when used for image classification, works best when provided with a segmentation mask of the image. For common datasets such as the ones we investigate, many high quality algorithms exist. REX does not require such existing segmentation masks. Blake et al. [2025] conducted an investigation on brain MRI explainability using a different model and different data set and a similar, though not overlapping, set of explainability tools. On this dataset, LIME's performance is hampered by medically meaningless segmentation. They found that REX did not suffer from this problem, and performed well compared to the other tools on this different model and data. The reason for that is that REX does not rely on segmentation at all. Instead, as described in Section 5, it performs an iterative refinement by randomly partitioning the input and averages the results over a number of different random partitions.



(a) ResNet50 (b) ViT (c) ConvNext

Fig. 12. Cumulative plots of explanations sizes with 3 different models over all datasets. REX consistently produces smaller explanations, meaning it identifies precisely those pixels required for the classification, with fewer unnecessary pixels.



(a) 'West Highland white terrier' (b) 'Ocean liner'

Fig. 13. Photo Bombing images and responsibility maps from REX

The performance of REX is notable across all datasets for being very consistent in terms of the evaluation metrics. The other tools examined are not as consistent. Note that the values for IN and OUT in Tables 1 to 3 are percentages of the explanation contained (or outside) the segmentation. As REX's explanations are already smaller than the others, this means that, in absolute terms, REX also has fewer extraneous pixels. This holds because other tools (such as LIME) may have similar IN values, but a larger explanation size.

We evaluated the results produced by REX against those produced by other explainability tools on a number of standard measures proposed in the literature, such as the size of explanations, insertion and deletion curves. For the datasets containing the ground truth or some part of it, such as the segmentation information or the partial occlusion, we also measured the percentage of the explanation that is clearly outside of an area where it should be contained, such as the occlusion in partially occluded areas (Figure 13). For all other measures, REX shows superior performance, which answers **RQ5**.

Finally, for **RQ7**, we did not exhaustively compare the results against all possible explainability methods. Rather, we chose the state-of-the-art primary attribution methods from the Captum library.

7 Conclusions and Future Work

In this paper we described an approach to explainability of image classifiers that is rooted in actual causality and computes explanations based on the formal definitions of sufficient explanations. As the exact computation is intractable, we described a modular algorithm for computing approximate explanations using a *causal ranking* of parts of the input image, viewing the classifier as black-box. Our experiments demonstrate that an implementation of our algorithm in the tool Rex produces results that are superior to those of state-of-the-art tools according to standard measures.

In the future, we will apply Rex to other modalities that lend themselves to occlusion-based reasoning, such as tabular data, time series, and spectroscopy. We will also explore different domains of application, where precise explanations are instrumental, in particular mission-critical and safety-critical domains, such as in healthcare AI and autonomous vehicles. The healthcare domain will require some adaptations of our algorithms, since the quality of medical images is different from the quality of general images (e.g., there are fewer colors in an X-ray than in a general image). In the automotive domain, we will adapt Rex to analyze object detectors and also devise new algorithms that work in (near) real-time, to address the needs of explainability for autonomous vehicles.

Acknowledgments

A part of this research, describing explanations for partially occluded images, appeared in the Proceedings of the ICCV conference in 2021, titled "Explanations for Occluded Images".

Hana Chockler and David A. Kelly are partially supported by the UKRI AI program and the Engineering and Physical Sciences Research Council for CHAI – Causality in Healthcare AI Hub [grant number EP/Y028856/1].

The authors are very grateful to Sander Beckers for his careful reading of an earlier draft and constructive suggestions regarding a simplified causal framework.

The authors are extremely grateful to Joe Halpern, who sadly passed away in February 2026. We discussed the core ideas with Joe, and his comments shaped the way we think about causality in the context of image classification. His intellectual curiosity and his vision continue to inspire us.

References

J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. 2018. "Sanity checks for saliency maps." In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (NIPS'18). Curran Associates Inc., Montréal, Canada, 9525–9536.

S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. 2015. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation." *PLoS ONE*, 10, 7.

S. Beckers. 2021. "Causal sufficiency and actual causation." *Journal of Philosophical Logic*, 50, 1341–1374.

S. Beckers. 2022. "Causal Explanations and XAI." In: *1st Conference on Causal Learning and Reasoning, CLeaR 2022, Sequoia Conference Center, Eureka, CA, USA, 11-13 April, 2022* (Proceedings of Machine Learning Research). Vol. 177. PMLR, 90–109.

D. Bhusal, M. Clifford, S. Rampazzi, and N. Rastogi. 2025. "FACE: Faithful Automatic Concept Extraction." In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

N. Blake, D. Kelly, S. Peña, A. Chanchal, and H. Chockler. 2025. "MRxaI: Black-Box Explainability for Image Classifiers in a Medical Setting." *Ceur Workshop Proceedings*, 4059.

S. Calderón-Peña, H. Chockler, and D. A. Kelly. 2024. "Real-Time Incremental Explanations for Object Detectors." *arXiv preprint arXiv:2408.11963*.

U. Chajewska and J. Y. Halpern. 1997. "Defining Explanation in Probabilistic Systems." In: *Uncertainty in Artificial Intelligence (UAI)*. Morgan Kaufmann, 62–71.

A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. 2018. "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks." In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 839–847.

H. Chockler and J. Y. Halpern. 2024. "Explaining Image Classifiers." In: *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR*.

H. Chockler and J. Y. Halpern. 2004. "Responsibility and Blame: A Structural-Model Approach." *J. Artif. Intell. Res.*, 22, 93–115.

H. Chockler, D. A. Kelly, and D. Kroening. 2025. "Multiple Different Explanations for Image Classifiers." In: *ECAI European Conference on Artificial Intelligence*.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. N.d. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. ().

P. Gärdenfors. 1988. *Knowledge in Flux*. MIT Press.

C. Glymour and F. Wimberly. 2007. "Actual causes and thought experiments." In: *Causation and Explanation*. Ed. by J. Campbell, M. O'Rourke, and H. Silverstein. MIT Press, Cambridge, MA, 43–67.

N. Hall. 2007. "Structural equations and causation." *Philosophical Studies*, 132, 109–136.

J. Y. Halpern and J. Pearl. 2005a. "Causes and explanations: a structural-model approach. Part I: causes." *British Journal for Philosophy of Science*, 56, 4, 843–887.

J. Y. Halpern and J. Pearl. 2005b. "Causes and explanations: a structural-model approach. Part II: explanations." *British Journal for Philosophy of Science*, 56, 4, 889–911.

J. Y. Halpern. 2015. "A Modification of the Halpern–Pearl Definition of Causality." In: *Proceedings of IJCAI*. AAAI Press, 3022–3033.

J. Y. Halpern. 2019. *Actual Causality*. The MIT Press.

C. G. Hempel. 1965. *Aspects of Scientific Explanation*. Free Press.

C. Hitchcock. 2007. "Prevention, preemption, and the principle of sufficient reason." *Philosophical Review*, 116, 495–532.

C. Hitchcock. 2001. "The intransitivity of causation revealed in equations and graphs." *Journal of Philosophy*, XCVIII, 6, 273–299.

D. Kelly, A. Chanchal, and N. Blake. 2025. "I Am Big, You Are Little; I Am Right, You Are Wrong." In: *International Conference on Computer Vision*.

P. Knab, S. Marton, and C. Bartelt. 2024. "Beyond Pixels: Enhancing LIME with Hierarchical Features and Segmentation Foundation Models." In: https://api.semanticscholar.org/CorpusID:268363283.

P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. 13–18 Jul 2020. "Concept Bottleneck Models." In: *Proceedings of the 37th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Ed. by H. D. III and A. Singh. Vol. 119. PMLR, (13–18 Jul 2020), 5338–5348. https://proceedings.mlr.press/v119/koh20a.html.

N. Kokhlikyan et al.. 2020. "Captum: A unified and generic model interpretability library for PyTorch." *arXiv*. eprint: 2009.07896 (cs.LG).

S. M. Lundberg and S.-I. Lee. 2017a. "A Unified Approach to Interpreting Model Predictions." In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

S. M. Lundberg and S.-I. Lee. 2017b. "A Unified Approach to Interpreting Model Predictions." In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30, 4765–4774.

R. K. Mothilal, D. Mahajan, C. Tan, and A. Sharma. 2021. "Towards Unifying Feature Attribution and Counterfactual Explanations: Different Means to the Same End." In: *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*. ACM, 652–663.

C. H. Papadimitriou and M. Yannakakis. 1984. "The Complexity of Facets (and Some Facets of Complexity)." *J. Comput. Syst. Sci.*, 28, 2, 244–259.

C. Papadimitriou. 1984. "The Complexity of Unique Solutions." *Journal of ACM*, 31, 492–500.

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann.

V. Petsiuk, A. Das, and K. Saenko. 2018. "RISE: Randomized Input Sampling for Explanation of Black-box Models." In: *British Machine Vision Conference (BMVC)*. BMVA Press.

R. Poyiadzi, K. Sokol, R. Santos-Rodríguez, T. D. Bie, and P. A. Flach. 2020. "FACE: Feasible and Actionable Counterfactual Explanations." In: *AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020*. ACM, 344–350.

M. T. Ribeiro, S. Singh, and C. Guestrin. 2016. ""Why Should I Trust You?" Explaining the Predictions of Any Classifier." In: *Knowledge Discovery and Data Mining (KDD)*. ACM, 1135–1144.

M. T. Ribeiro, S. Singh, and C. Guestrin. 2018. "Anchors: High-Precision Model-Agnostic Explanations." In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*. AAAI Press, 1527–1535.

O. Russakovsky et al.. 2015. "ImageNet Large Scale Visual Recognition Challenge." *International Journal of Computer Vision (IJCV)*, 115, 3, 211–252. DOI: 10.1007/s11263-015-0816-y.

W. C. Salmon. 1989. *Four Decades of Scientific Explanation*. University of Minnesota Press.

G. Schwalbe and B. Finzel. 2024. "A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts." *Data Mining and Knowledge Discovery*, 38, 5, 3043–3101.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. 2017. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization." In: *International Conference on Computer Vision (ICCV)*. IEEE, 618–626.

S. Sharma, J. Henderson, and J. Ghosh. 2020. "CERTIFAI: A Common Framework to Provide Explanations and Analyse the Fairness and Robustness of Black-box Models." In: *AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020*. ACM, 166–172.

J. Shi, Q. Yan, L. Xu, and J. Jia. 2016. "Hierarchical Image Saliency Detection on Extended CSSD." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 4, 717–729. DOI: 10.1109/TPAMI.2015.2465960.

A. Shrikumar, P. Greenside, and A. Kundaje. 2017. "Learning important features through propagating activation differences." In: *ICML*. Vol. 70. JMLR.org, 3145–3153.
D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. 2017. "SmoothGrad: removing noise by adding noise." *arXiv*. https://arxiv.org/abs/1706.03825 eprint: 1706.03825 (cs.LG).
J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. 2015. "Striving for Simplicity: The All Convolutional Net." In: *ICLR (Workshop Track)*. http://arxiv.org/abs/1412.6806.
Y. Sun, H. Chockler, X. Huang, and D. Kroening. 2020. "Explaining Image Classifiers using Statistical Fault Localization." In: *ECCV, Part XXVIII* (LNCS). Vol. 12373. Springer, 391–406.
M. Sundararajan, A. Taly, and Q. Yan. 2017. "Axiomatic Attribution for Deep Networks." In: *International Conference on Machine Learning*. PMLR, 3319–3328.
B. Ustun, A. Spangher, and Y. Liu. 2019. "Actionable Recourse in Linear Classification." In: *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*. ACM, 10–19.
B. Weslake. 2015. "A partial theory of actual causation." *British Journal for the Philosophy of Science*, To appear.
E. Winter. 2002. "The shapley value." *Handbook of game theory with economic applications*, 3, 2025–2054.
J. Woodward. 2003. *Making Things Happen: A Theory of Causal Explanation*. Oxford University Press, Oxford, U.K.

A Our Definitions in the Actual Causality Landscape

For the ease of presentation, we first restate our definition of explanation.

Definition 4.1 (Single-Context Explanation)**.** A subset $\vec{V}_{exp}$ of $\vec{V}$ is a *single-context explanation* of a classification $\mathcal{N}(x)$ of an input image x by a classifier $\mathcal{N}$ if the following conditions hold:

EXIM1. $(M, \vec{u}_0) \models [\vec{V}_{exp} = 1](O = 1)$.

EXIM2. $\vec{V}_{exp}$ is minimal; there is no strict subset $\vec{V}'_{exp}$ of $\vec{V}_{exp}$ that satisfies EXIM1.

When there is no confusion, we call *single-context explanation* simply an *explanation*. As there is a one-to-one correspondence between the variables in $\vec{V}$ and the pixels of x, we also call the subset of pixels P_{exp} of x that corresponds to $\vec{V}_{exp}$ a *single-context explanation* of $\mathcal{N}(x)$.

The following is the latest definition of actual cause due to Halpern [Halpern 2015] (see [Halpern and Pearl 2005a] for the original definition).

Definition A.1. [Actual cause [Halpern 2015]] $\vec{X} = \vec{x}$ is an *actual cause* of φ in $(M, \vec{u})$ if the following three conditions hold:

AC1. $(M, \vec{u}) \models (\vec{X} = \vec{x})$ and $(M, \vec{u}) \models \varphi$.

AC2. There is a a setting $\vec{x}'$ of the variables in $\vec{X}$, a (possibly empty) set $\vec{W}$ of variables in $\mathcal{V} - \vec{X}'$, and a setting $\vec{w}$ of the variables in $\vec{W}$ such that $(M, \vec{u}) \models \vec{W} = \vec{w}$ and $(M, \vec{u}) \models [\vec{X} \leftarrow \vec{x}', \vec{W} \leftarrow \vec{w}]\neg\varphi$, and moreover

AC3. $\vec{X}$ is minimal; there is no strict subset $\vec{X}'$ of $\vec{X}$ such that $\vec{X}' = \vec{x}''$ can replace $\vec{X} = \vec{x}'$ in AC2, where $\vec{x}''$ is the restriction of $\vec{x}'$ to the variables in $\vec{X}'$.

The following lemma states their equivalence in our setting.

Lemma A.2. *Given a image classifier $\mathcal{N}$, an input x, and a causal model M derived from them as described in Section 4, a subset $\vec{V}_{exp}$ of $\vec{V}$ is a single-context explanation of $\mathcal{N}(x)$ according to Definition 4.1 iff $\vec{V}_{exp} = 0$ (that is, all variables in $\vec{V}_{exp}$ have the value 0) is an actual cause of $O = 0$ in $(M, \vec{u}_0)$ according to Definition A.1, under the assumption that a fully masked image has a different classification than x.*

Proof. First we observe that AC1 simply states that in the current context $\vec{u}_0$, the value of all variables in $\vec{V}$ is 0, and the classification of the fully masked image is different from $\mathcal{N}(x)$. Assume that EXIM1 holds. Then, assigning $\vec{V}_{exp}$ to 1 results in restoring the original classification $\mathcal{N}(x)$, and hence $O = 1$, hence AC2 holds

with $\vec{W} = \emptyset$. For the other direction, assume that AC2 holds. Since the variables in $\vec{V}$ are causally independent, changing $\vec{V}_{exp}$ to 1 does not change any other input variable, and hence we can take $\vec{W} = \emptyset$. Then, AC2 implies EXIM1. Finally, AC3 is simply the minimality condition, and is equivalent to EXIM2. □

We note that the theory of actual causality includes the notion of explanations as well [Halpern and Pearl 2005b], but these are defined over a *set* of contexts, rather than a single context, so are not suitable for our purposes. We can, in general, imagine a setting in which we are searching for a subset of pixels of the input image that is sufficient to result in the original classification, regardless of the values of the rest of the pixels—this would be similar to the setting explored in Anchors [Ribeiro et al. 2018]. However, such a setting leads to significantly less intuitive results for explaining image classification, as also evident from the results of applying Anchors to images.

Let us now place our definition of sufficient responsibility into the actual causality landscape. For the ease of presentation, we first restate the definition.

Definition 4.2 (Sufficient responsibility)**.** The *degree of sufficient responsibility* of an explanation $\vec{V}_{exp}$ for the classification of x by $\mathcal{N}$ is defined as $1/|\vec{V}_{exp}|$. We also extend this value to all pixels in $\vec{V}_{exp}$. That is, v_i (and its matching pixel p_i) have the degree of sufficient responsibility $1/|\vec{V}_{exp}|$ for the classification of x by $\mathcal{N}$, where $\vec{V}_{exp}$ is the smallest explanation for the classification of x that contains v_i. If there is no explanation that contains v_i, then its degree of sufficient responsibility is defined as 0.

In the literature, the degree of responsibility is defined by Chockler and Halpern [Chockler and Halpern 2004] as the quantification of actual causality, hence its precise definition is tied to the definition of the actual cause. The latest version is as follows.

Definition A.3. [Degree of responsibility [Halpern 2019]] For $\vec{X} = \vec{x}$ being an *actual cause* of φ in $(M, \vec{u})$, the degree of responsibility of $\vec{X} = \vec{x}$ for the value of φ in $(M, \vec{u})$ is defined as

$$dr(\vec{X} = \vec{x}, \varphi, M, \vec{u}) = \frac{1}{|\vec{X}| + |\vec{W}|},$$

for the smallest $|\vec{X}| + |\vec{W}|$ that satisfy Definition A.1.

The following is immediate from the definitions, given that in our case $\vec{W} = \emptyset$.

Proposition A.4. *Given a image classifier $\mathcal{N}$, an input x, a causal model M derived from them as described in Section 4, and a single-context explanation $\vec{V}_{exp} \subseteq \vec{V}$ of $\mathcal{N}(x)$ according to Definition 4.1, the degree of sufficient responsibility of $\vec{V}_{exp}$ for φ in M is the degree of responsibility of $O = 0$ in $(M, \vec{u}_0)$ according to Definition A.3, under the assumption that a fully masked image has a different classification than x.*

Indeed, the degree of sufficient responsibility of $\vec{V}_{exp}$ is $\frac{1}{\vec{V}_{exp}}$, which is exactly the degree of responsibility according to Definition A.3 for $(M, \vec{u}_0)$, given that in our model M there is no dependency between the variables.

B The Complexity of Our Definitions

In order to analyze the complexity of our definition of single-context explanation, we first need to introduce a complexity class. C. H. Papadimitriou and Yannakakis [1984] introduced the complexity class DP, which consists of all languages L such that there exists a language L_1 in NP and a language L_2 in co-NP such that $L = L_1 \cap L_2$. They showed that a number of problems of interest were DP complete.

For the complexity discussion below, we use the classifier $\mathcal{N}$ as an oracle, not taking its complexity into account. Instead, we assume that the value of O is computed in polynomial time using the values of variables in $\vec{V}$. Since all variables in our causal models are binary, the computation is simply a Boolean formula.

Halpern proved that the decision problem of actual causality as per Definition A.1 is DP-complete, and the result holds for binary models as well [Halpern 2015]. The proof, however, relies on causal models having causal dependencies between their variables. As we show below, our definition, while still intractable, is in a lower complexity class. But first we observe that for singletons, our definition is computable in polynomial time.

Proposition B.1. *If $\vec{V}_{exp}$ is a single variable V, the complexity of deciding whether it satisfies Definition 4.1 is polynomial in the size of the input.*

Proof. The proof is straightforward by observing that EXIM1 is checkable in polynomial time, and EXIM2 holds trivially, as singletons have no subsets. □

We note that the decision problem of actual cause is NP-complete for singletons [Halpern 2015], indicating that our setting significantly simplifies the problem. The reduction in complexity also holds for the general case, as proved in the following theorem.

Theorem B.2. *The decision problem of explanations in Definition 4.1 is co-NP-complete.*

Proof. For the membership in co-NP, we prove that the complementary problem, of showing that a given $\vec{V}_{exp}$ is not an explanation, is in NP. Given a set $\vec{V}_{exp}$, checking whether it satisfies EXIM1 is polynomial. For refuting EXIM2, it suffices to find one witness subset $\vec{V}' \subset \vec{V}$ that satisfies EXIM1, hence the complementary problem is indeed in NP.

For the co-NP-hardness, we show a reduction from the classic co-NP-complete problem $UNSAT = \{$ unsatisfiable propositional Boolean formulae $\}$. Given a propositional Boolean formula φ over the set of variables $X_1, \ldots, X_n$, we construct a depth-2 binary causal model M_φ as follows. The set of endogenous variables $\mathcal{V}$ is the set $X_1, \ldots, X_n, O$, where the values of $X_1, \ldots, X_n$ are determined directly by the exogenous variables, and

$$O = (\varphi \vee \bigwedge_{i=1}^{n} X_i) \wedge (\bigwedge_{i=1}^{n} X_i \rightarrow \neg\varphi).$$

Essentially, the value of O is an exclusive OR of the value of φ and the conjunction of all variables of φ. Then, $\varphi \in UNSAT$ iff the set $\vec{X} = \{X_1, \ldots X_n\}$ is a single-context explanation for $O = 0$ in $(M, \vec{u}_0)$.

First, we claim that O has the value 1 only in the assignment that sets all variables in $\vec{X}$ to 1 iff φ is unsatisfiable. Indeed, if φ is unsatisfiable, then its value is 0 under all assignments, and the value of the conjunction is 1 iff all variables in $\vec{X}$ are set to 1. For the other direction, if the value of O is 1 under the assignment that sets all variables in $\vec{X}$ to 1, then φ is falsified under this assignment, and since this is the only assignment that results in $O = 1$, φ is falsified under all other assignments to $\vec{X}$ as well, hence φ is unsatisfiable.

We are now ready to prove the reduction. Assume first that $\varphi \in UNSAT$. Then, by the previous discussion, $(M, \vec{u}_0) \models [\vec{X} \leftarrow 1](O = 1)$, satisfying EXIM1, and $\vec{X}$ is the only such subset of variables, satisfying EXIM2. Therefore, $\vec{X}$ is a single-context explanation to O in M_φ, as required.

For the other direction, assume that $\vec{X}$ is a single-context explanation to O in M_φ. Then, $(M, \vec{u}_0) \models [\vec{X} \leftarrow 1](O = 1)$, and no subset of $\vec{X}$ satisfies this condition, hence by the previous discussion, $\varphi \in UNSAT$, completing the proof. □

There is, therefore, little hope to find an efficient algorithm for computing exact explanations for image classification, and the answer to **RQ2** wrt precise explanations is 'no', as computing explanations is intractable.

While this is not directly relevant to this paper, the following observation is a direct corollary from Lemma A.2 and Theorem B.2.

Observation B.3. *In depth-2 binary causal models, the decision problem of actual causality is co-NP-complete.*

We now turn to examining the complexity of the degree of sufficient responsibility. First, we need to introduce a new complexity hierarchy, specifically, the hierarchy of functional problems. This is because the output of the degree of the responsibility computation is a value, and not an accept/reject decision. For a complexity class A, the class $\mathrm{FP}^{\mathrm{A}[\log n]}$ consists of all functions that can be computed by a polynomial-time Turing machine with an oracle for a problem in A, which on input x asks a total of $O(\log |x|)$ queries (cf. [C. Papadimitriou 1984]). A function $f(x)$ is $\mathrm{FP}^{\mathrm{A}[\log n]}$ hard iff for every function $g(x)$ in $\mathrm{FP}^{\mathrm{A}[\log n]}$ there exist polynomially computable functions $R, S : \Sigma^* \rightarrow \Sigma^*$ (where Σ is the common alphabet) such that $g(x) = S(f(R(x)))$. A function $f(x)$ is complete in $\mathrm{FP}^{\mathrm{A}[\log n]}$ iff it is in $\mathrm{FP}^{\mathrm{A}[\log n]}$ and is $\mathrm{FP}^{\mathrm{A}[\log n]}$-hard.

We prove the following result.

Theorem B.4. *The problem of computing the degree of sufficient responsibility is $FP^{\mathrm{NP}[\log n]}$-complete.*

Proof. The proof is very similar to the proof of complexity of the degree of responsibility by Chockler and Halpern [2004], so we present only the key points here.

For the membership in $\mathrm{FP}^{\mathrm{NP}[\log n]}$, we observe that the polynomial time algorithm for computing the degree of responsibility can perform a binary search on the size of $\vec{V}_{exp}$, in each step querying an oracle for the existence of a single-context explanation $\vec{V}_{exp}$ of size k. By Theorem B.2, the decision problem of the existence of explanation is co-NP-complete, and hence can be decided by an NP-oracle.

For hardness in $\mathrm{FP}^{\mathrm{NP}[\log n]}$, the reduction is from the $\mathrm{FP}^{\mathrm{NP}[\log n]}$-complete function problem $MINSAT$, defined in [Chockler and Halpern 2004] as $MINSAT(\varphi) = k$, where k is the minimal number of 1's in a satisfying assignment for φ. The reduction is the same as in [Chockler and Halpern 2004]. □

Theorem B.4 shows that computing the degree of sufficient responsibility in our setting is intractable, hence the need for approximation algorithms.

Received 13 February 2026; accepted 25 March 2026