Fast Hierarchical Games for Image Explanations

Jacopo Teneggi [1] **Alexandre Luster** [2] **Jeremias Sulam** [1]

Abstract

As modern neural networks keep breaking records and solving harder problems, their predictions also become less intelligible. The current lack of interpretability undermines the deployment of accurate machine learning tools in sensitive settings. In this work, we present a model-agnostic explanation method for image classification based on a hierarchical extension of Shapley coefficients –*Hierarchical Shap (h-Shap)*– that resolves some limitations of current approaches. Unlike other Shapley-based explanation methods, h-Shap is scalable and it does not need approximation. Under certain distributional assumptions, which are common in multiple instance learning, h-Shap retrieves the exact Shapley coefficients with an exponential improvement in computational complexity. We compare our hierarchical approach with popular Shapley-based and non-Shapley-based methods on a synthetic dataset, a medical imaging scenario, and a general computer vision problem. We show that h-Shap outperforms the state of the art in both accuracy and runtime.

1. Introduction

Explainability has become a question of increasing relevance in machine learning, where the growing complexity of deep neural networks often renders them *opaque* to us, the humans interacting with them. This issue is commonly referred to as the *black-box problem* and comprises theoretical, technical, and regulatory questions (Zednik, 2019; Tomsett et al., 2018). As deep neural networks take on sensitive tasks in medical, legal, and financial settings, they need to achieve both high accuracy and high transparency for a safe deployment. For example, uninterpretable predictions could mislead clinicians in their decision-making rather than support it (Amann et al., 2020). Furthermore, it is sometimes required by law (Kaminski, 2019) to provide an explanation of how data lead an automated algorithm, for example, to reject a loan application (Kaminski, 2019; Kaminski & Malgieri, 2019; Hacker et al., 2020). Finally, opaque models can conceal dataset bias, and lead to socially unfair models (Shin, 2021).

In this work, we are particularly interested in explaining models in supervised learning scenarios in order to gain insights about the concept related to a specific response. For example, assume one has a model that predicts the presence of brain tumor in MRI scans with very high accuracy. What are the most relevant features that indicate the presence of tumor, and where are they located? Can we discover new features of the disease from what the model has learned? Many important problems of this kind exist, but the necessary tools to answer these questions effectively and efficiently are still lacking.

The foundational work by Ribeiro et al. (2016) spurred exciting advances in local feature attribution methods, such as Grad-CAM (Selvaraju et al., 2017), Integrated Gradients (Sundararajan et al., 2017), and DeepLIFT (Shrikumar et al., 2017). Lundberg and Lee (2017) provide a unified framework for several different approaches under their SHAP method, which leverages Shapley coefficients –a game-theoretic measure (1953)– and feature removal strategies. Unlike other perturbation-based alternatives, these methods produce attributions that enjoy of important consistency results and theoretical properties. Since then, a plethora of different explanation methods has been developed[1] for tabular, sequential, or imaging data; both based on Shapley coefficients (Chen et al., 2018) as well as other information theoretic quantities (MacDonald et al., 2019; Heiß et al., 2020; Merrick & Taly, 2020). Although previous work explores structured and hierarchical approaches (Chen et al., 2020b; 2018; Singh et al., 2018), they remain limited for high-dimensional data.

Notwithstanding the recent advances in image attribution

[1]Department of Biomedical Engineering, Johns Hopkins University, Baltimore, USA [2]School of Life Sciences, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. Correspondence to: Jacopo Teneggi <jtenegg1@jhu.edu>, Jeremias Sulam <jsulam1@jhu.edu>.

Workshop on Interpretable ML in Healthcare at International Conference on Machine Learning (ICML). Copyright 2021 by the author(s).

[1] To our knowledge, Covert et al. (2020) compiled the most comprehensive review of currently available explanation methods based on feature removal.

methods based on Shapley coefficients, several limitations hinder their use for "large" images –a standard image contains $\approx 10^6$ pixels, and larger images are used in several important applications. The contribution of this work is threefold: first, we present a fast explanation method based on Shapley coefficients that is exponentially faster than popular SHAP methods. Second, under some distributional assumptions similar to those in multiple instance learning, we show that the coefficients provided by h-Shap are exact, and can be further approximated in a controlled manner by trading off computational cost. Third, we compare h-Shap with other popular explanation methods on three benchmarks of varied complexity and dimension, demonstrating that h-Shap outperforms the state of the art both in terms of runtime and retrieval of relevant features.

This paper is organized as follows. In Sec. 2 we briefly summarize the necessary background. We present h-Shap in Sec. 3, and the experiments with their results in Sec. 4 and 5. Finally, we discuss our limitations in Sec. 6, and we conclude in Sec. 7.

2. Background

In supervised learning scenarios, we are interested in approximating a response or label, $Y \in \mathcal{Y}$, from a given input random sample $X \in \mathcal{X}$. Herein we assume a realizable setting where the response $Y = f^*(X) \in \mathcal{Y}$, for some $f^* : \mathcal{X} \to \mathcal{Y}$, and denote the joint distribution of (X, Y) as $\mathcal{D}$. We look for a function $f : \mathcal{X} \to \mathcal{Y}'$ that approximates $f^*(X)$. Given a loss function $L : \mathcal{Y} \times \mathcal{Y}' \to \mathbb{R}$ that penalizes the dissimilarity between the predicted and real label, we look for f in a suitable functional class with minimal risk, $\mathcal{R} = \mathbb{E}_{\mathcal{D}}[L(Y, f(X))]$. However, $\mathcal{D}$ is typically unknown and instead we are provided with a training set $\{(X^{(i)}, Y^{(i)})\}_{i=1}^N$ of observed data. As a result, we search for a function that minimizes the empirical risk,

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=i}^{N} L(Y^{(i)}, f(X^{(i)})), \tag{1}$$

where $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$, with parameters θ (such as a neural network model). We focus on binary classification problems, where $\mathcal{Y} = \{0, 1\}$ and $\mathcal{Y}' \in \mathbb{R}$, though our general methodology is applicable to multi-class settings as well. We will refer to images as vectors in the n-dimensional real space, i.e. $\mathcal{X} \subseteq \mathbb{R}^n$.

Explaining predictions via Shapley coefficients. Modern machine learning models can often provide solutions that perform remarkably well. In many settings, however, one would like to know the contribution of x_i, the i^{th} entry of X, towards the output. Let us define by C a subset of the entries of X, so that $C \subseteq [n] := \{1, \ldots, n\}$, and define $X_C \in \mathbb{R}^n$ the input that coincides with X in the entries denoted by C but takes a different, *baseline*, value in its complement, $\bar{C}$. In the context of interpretability, we look for a vector $\Phi_{(X,\hat{f})} \in \mathbb{R}^n$, where the i^{th} coordinate reflects the importance of x_i towards $\hat{f}(X)$. Broadly speaking, the features C provide an explanation for $\hat{f}(X)$ if $\hat{f}(X) \approx \hat{f}(X_C)$. Different measures of importance have been proposed to study model interpretability, and thus to compute $\Phi_{(X,\hat{f})}$. In this work, we focus on the general approach presented originally by (Lundberg & Lee, 2017) that employs Shapley coefficients (1953) as the measure of contribution of every pixel toward the output, which has gained great popularity (Sundararajan & Najmi, 2020). We now briefly introduce some game theory notation to define Shapley coefficients.

Let $g = (X, f, [n])$ be an n-person cooperative game with players $[n]$ and characteristic function $f : \mathcal{X} \mapsto \mathbb{R}$ which maps the input space $\mathcal{X}$ to a score. In particular, $f(X_C)$ is the score that the players in C would earn by collaborating in the game, with $f(X_\emptyset) = 0$ by convention[2]. A *solution concept* is a rule that assigns a fair contribution to each player in the game. Notably, Shapley coefficients, denoted by $\phi_1(f), \ldots, \phi_n(f)$, are the only solution concept of $(X, f, [n])$ that simultaneously satisfy the properties of efficiency, nullity, and symmetry (Shapley, 1953). In the context of model explanations, input features are regarded as players, and these properties imply that: **i**) feature attributions sum up to the model prediction; **ii**) the attributions of irrelevant features are simply 0; and **iii**) the attributions of equally important features are equal, respectively. These equip Shapley-based methods with a useful set of properties, which are not generally satisfied by others attributions methods.

Shapley coefficients can be derived axiomatically (Shapley, 1953), and they are defined as

$$\phi_i(f) = \sum_{C \subseteq [n] \setminus \{i\}} w_C \left[f(X_{C \cup \{i\}}) - f(X_C) \right] \tag{2}$$

where $w_C = |C|!(n - |C| - 1)!/n!$. This way, $\phi_i(f)$ represents the averaged marginalized contribution of x_i over all possible subsets of $[n]$. Eq. (2) also illustrates what is arguably the most important limitation of Shapley coefficients: their computation is exponential in the dimension of the input features, and it requires 2^n unique evaluations of f. This quickly becomes intractable in image classification problems when f is a convolutional neural network and $n \approx 10^6$, or larger. As a result, all state-of-the-art image explanation methods based on Shapley coefficients rely on some approximation strategy to work

[2] Game theory (Owen, 1995) requires a characteristic function $v : \mathcal{P}(X) \to \mathbb{R}$, where $\mathcal{P}(X)$ is the power set of X. Following prior work (Lundberg & Lee, 2017), we assume $v(C) = f(X_C)$, $\forall C \subseteq X$, and therefore use f for the sake of simplicity.

around this computational limitation. For instance, GradientExplainer (Lundberg & Lee, 2017) extends Integrated Gradients (Sundararajan et al., 2017) by sampling multiple references from the background dataset to integrate on. Similarly, DeepExplainer (Lundberg & Lee, 2017; Chen et al., 2021) builds upon DeepLIFT (Shrikumar et al., 2017) by choosing a per-node attribution rule that can approximate Shapley coefficients when integrated over many background samples. Finally, PartitionExplainer employs a hierarchical clustering approach to compute Owen's coefficients (Owen, 1977; López & Saboya, 2009), which can improve runtime compared to the naive Shapley coefficients. While these approximations can sometimes work in practice, they only provide consistency results and lack results when one can only use a small amount of model evaluations (Merrick & Taly, 2020). As a result, it is hard to understand when they will and will not be effective. We will compare extensively with these approaches later in Sec. 4.

We remark that one of the most important details of any explanation method based on feature removal is the baseline, which defines the value that X_C takes in the entries not in C. There are different approaches to removing features, ranging from using the default value of 0, to using their conditional distribution (refer to (Covert et al., 2020) for further details). Computing the latter can be challenging, and recent work has explored various approximations (Aas et al., 2021; Frye et al., 2019). The effects of using different baselines have also been investigated in images (Sturmfels et al., 2020) and tabular data (Haug et al., 2021). We follow the standard approach of setting the baseline to their expected value over the training dataset (Lundberg & Lee, 2017; Janzing et al., 2020), and comment on potential extensions later.

Multiple Instance Learning. In this work, we focus on problems with particular joint distributions of samples and labels. Our guarantees will apply to settings broadly known as *multiple instance learning* (MIL) (Weidmann et al., 2003). In MIL, each *instance* x_i is assumed to have an instance-label, and the sample X is regarded as a *bag* that aggregates all instances. The bag, X, has its own label $Y \in \{0, 1\}$ determined by its constituent instances. In its simplest version, the bag is assumed to be positive if at least one of its instances is positive. As an example, an image of cells will be labeled with `infection` if at least one cell in it is `infected`. Importantly, the learner does not have access to the instance-labels, but only to the bag-label Y. Such an MIL setting appears in several important problems (Han et al., 2020; Hashimoto et al., 2020; Fu et al., 2012). In the context of our work, we assume that the prediction rule satisfies such an MIL assumption:

$$f^*(X) = 1 \iff \exists\, C \subseteq [n]:\ f^*(X_C) = 1. \quad (3)$$

In words, Eq. (3) implies that $f^*(X)$ will be 1 as soon as there is at least one subset C of $[n]$ that contains the *concept* of interest. This is simply a formalization of the setting we were describing earlier, where the concept can be a specific morphological feature in a brain scan, a sick cell in a blood smear, or something as general as a traffic light in a street image.

To recap, $\hat{f}$ is trained to detect a binary concept in a sample image, and we would like to detect which subsets of the input, X_C, are relevant for this task. While this could in principle be done via Shapley coefficients, it is computationally intractable. We now move on to present our approach, which will address this limitation.

3. Hierarchical-Shap

Our motivating observation is that if an area of an image is uninformative (i.e. it does not contain the concept), so will be its constituent sub-areas. Therefore, the exploration of relevant areas of an image can be done in a hierarchical manner. There exists extensive literature on hierarchies of games and their properties (Faigle & Peis, 2008; Algaba & van den Brink, 2019). Our contribution is to deploy these ideas for the purpose of image explanations. We now make this more precise.

Let $\mathcal{T}_0 = (S_0, \mathcal{T}_1, \dots, \mathcal{T}_\gamma)$ be a recursive γ-partition tree of X, where S_0 is the root node containing all features of X, i.e. $S_0 = [n], |S_0| = n$, and $\mathcal{T}_1, \dots, \mathcal{T}_\gamma$ are the subtrees branching off of S_0. Let $c(S_i) = \{C_1, \dots, C_\gamma\}$ denote the children of S_i, and $h_{\hat{f}} : S_i \mapsto (X, \hat{f}, c(S_i))$ be a mapping from the node S_i of $\mathcal{T}_i$ to the γ-person cooperative game $(X, \hat{f}, c(S_i))$. Succinctly, $\mathcal{G}_0 = h_{\hat{f}}(\mathcal{T}_0)$ is a hierarchy of γ-person games, and we denote by $\phi_{i,1}(\hat{f}), \dots, \phi_{i,\gamma}(\hat{f})$ the Shapley coefficients of $g_i \in \mathcal{G}_0$. In simpler words, we partition an image X into *a few disjoint components*, compute the Shapley coefficients ϕ_i of each component, and then partition further in a hierarchical manner. In particular, the number of such partitions per level (specified by γ) is very small: if X is a one dimensional vector, we set $\gamma = 2$ and $\mathcal{T}_0$ is a binary tree; when X is a $(\sqrt{n} \times \sqrt{n})$ image, $\gamma = 4$ and $\mathcal{T}_0$ is a quadtree. As a result, computing all 2^γ unique evaluations of $\hat{f}$ required for each game $(X, \hat{f}, c(S_i))$ is trivial. For images, each coefficient requires only 16 model evaluations. We have chosen to employ symmetric disjoint partitions in this work (i.e. halves for vectors, quadrants for images, etc) for simplicity only. More sophisticated (and potentially data-dependent) hierarchical partitions are possible as well. We will comment on this in the discussion.

Given such nested partitions, h-Shap relies on evaluating the resulting hierarchy of games while only visiting nodes that are relevant. More precisely, beginning at S_0, it computes the coefficients $\phi_{0,1}, \dots, \phi_{0,\gamma}$ of g_0. Under Eq. (3), if any $\phi_{0,i} = 0$, all features in the corresponding subtrees will also be irrelevant. As a result, they can be ignored altogether, and we only proceed by exploring the S_i for which $\phi_i > 0$. This process finishes when all relevant leaves have been visited. In practice, we introduce two parameters to add flexibility. We set a relevance tolerance, τ, which determines the threshold to be used to declare a partition relevant, and therefore expand on its subtrees. We further introduce a minimal feature size, s, that serves as a condition for termination. These two parameters are naturally motivated by application and easy to set. For example, it might not be that useful for a domain expert to know the exact pixel-level explanation of a given input. Rather, it would be more informative to have a coarser aggregation of the features that inform the model prediction. Later in this section, we will precisely characterize how the minimal feature size s affects the dissimilarity between h-Shap's attributions and the exact Shapley coefficients. On the other hand, model deviations and noise in the input may result in positive coefficients very close to 0. Requiring $\phi_i > \tau > 0$ provides control over the sensitivity of the method. Finally, when $\tau = 0, s = 1$, h-Shap simply explores all relevant nodes in $\mathcal{T}_0$ as described above.

Fixed τ and s, h-Shap explores $\mathcal{T}_0$ starting from S_0, and it visits all relevant nodes $S_i : \phi_i > \tau, |S_i| \geq s$. This tree exploration can be naturally done in a depth-first or breadth-first manner. Please refer to Supplementary Material A for both algorithms. The only difference between them is that the former defines τ as an absolute value (e.g. 0), whereas the latter does so relative to the pooled Shapley coefficients of all nodes at the same depth (e.g. 50^{th} percentile). Both algorithms return the set of relevant leaves $L \subseteq [n]$ with coefficients greater than τ, and the saliency map $\widehat{\Phi}$ is finally computed as

$$\widehat{\phi}_i = \begin{cases} 1/|L| & \text{if } i \in L, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

This choice will ensure that $\widehat{\Phi}$ is consistent with the exact Shapley attributions Φ under the MIL assumption, as we will formalize shortly.

To mask features out (i.e. as baseline), h-Shap uses their expected value (or *unconditional distribution* (Janzing et al., 2020)) for simplicity, as done by other works (Covert et al., 2020). As pointed out by (Covert et al., 2020; Lundberg & Lee, 2017), this is valid under the assumptions of model linearity and feature independence[3]. Yet, as we will argue later in Sec. 7, the feature independence property holds approximately in the cases we are interested in this work, whereas our MIL assumption is enough to provide specific guarantees without requiring linearity of the model. We will also show in Sec. 4 that these assumptions are sufficient for h-Shap to work well in practice. More generally, our contribution is independent of the particular method employed for sampling the baseline, and follow-up work can employ better approximations of both the observational and interventional conditional distributions in appropriate tasks (Chen et al., 2020a).

Computational analysis. The benefit of h-Shap relies on decoupling the dimensionality of the sample X (i.e. n), from the number of players in each game (i.e. γ). As we will explain in this section, this leads to an exponential computational advantage over the general expression in Eq. (2) in explaining $\hat{f}$. In the analysis that follows, we do not include the computation of the baseline value –which we assume fixed, see discussion in Sec. 7– and we refer the reader to the proofs of all the results in this section to the Supplementary Material B. Let us denote by $\hat{\mathcal{T}}_0$ the subtree of $\mathcal{T}_0$ explored by h-Shap (i.e. the one with the visited nodes only). We will also assume in this section that n is a power of γ for simplicity of the expressions. We begin by making the following remark.

Remark 3.1 (Computational cost)**.** *Given $X \in \mathbb{R}^n$, h-Shap requires at most $2^\gamma k \log_\gamma(n)$ model evaluations, where k is the number of relevant leaves in $\hat{\mathcal{T}}_0$.*

This result follows directly by noting that the cost of splitting each node is always 2^γ, and by realizing that each important leaf takes, at most, $\log_\gamma(n)$ nodes, which is exponentially better than the cost of Eq. (2). The reader should recall that the number of internal nodes of a full and complete γ-partition tree is $(n-1)/(\gamma-1)$. Then, the above result is relevant whenever $k \log_\gamma n < (n-1)/(\gamma-1)$. This implies that further benefit is obtained whenever $k = \mathcal{O}(n/\log_\gamma n)$, which is only a mild requirement in the number of relevant features.

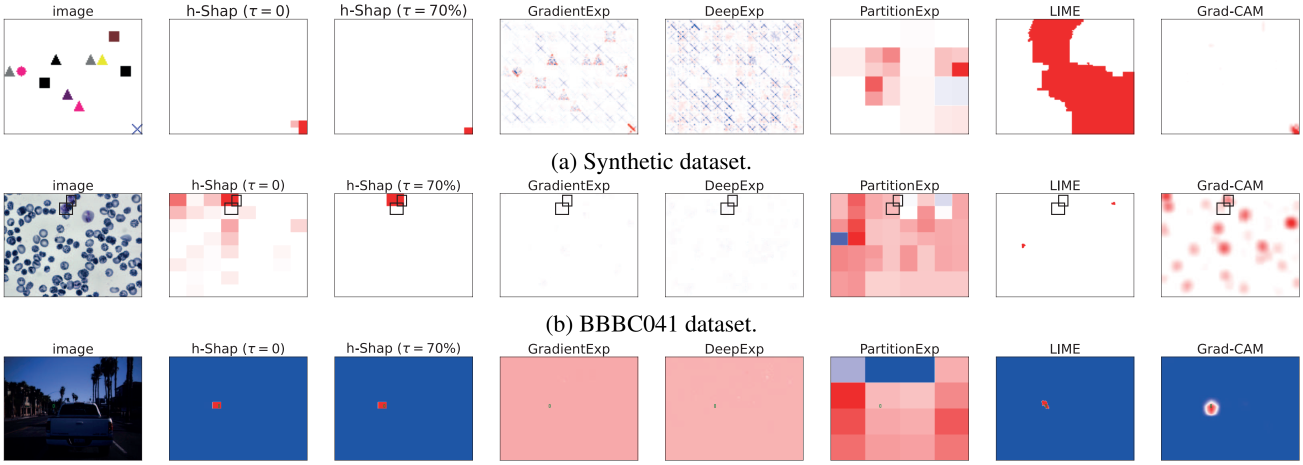
Moreover, it is of interest to know the expected computational cost, which can be significantly smaller than the upper bound above. Throughout the rest of this section, and to provide more precise results, we will let the data X be drawn from a distribution of *important* and *non-important* features. A distribution is "important" in the sense that it leads to positive responses.

Assumption A1. *The data $X \in \mathbb{R}^n$ is drawn so that each entry $x_i \sim a_i\mathcal{I} + (1-a_i)\mathcal{I}^c$, where $a_i \sim$ Bernoulli(ρ) is a binary random variable that indicates whether the feature x_i comes from an* important *distribution $\mathcal{I}$, or its* non-important *complement $\mathcal{I}^c$, so that*

$$\hat{f}(X_C) = 1 \iff \exists i \in C : x_i \sim \mathcal{I}, C \subseteq [n]. \tag{5}$$

With these elements, we present the following result.

[3] We refer to (Chen et al., 2020a; Sundararajan & Najmi, 2020; Merrick & Taly, 2020; Janzing et al., 2020) for recent discussion on the use of *observational* vs *interventional* conditional distributions in the context of removal-based explanation methods.



(a) Synthetic dataset.

(b) BBBC041 dataset.

(c) LISA dataset.

Figure 1. A few saliency maps for the three settings studied in this work, where blue pixels have negative, white pixels have negligible, and red pixels have positive Shapley coefficients. The color mapping is adapted to each saliency map and centered around 0.

Theorem 3.2 (Expected number of visited nodes)**.** *Assume X and $\hat{f}(X)$ satisfy A1, $\tau = 0$, and $s = 1$. Then, the expected number of visited nodes in $\hat{\mathcal{T}}_0$ is*

$$\mathbb{E}[|\hat{\mathcal{T}}_0|] = 1 + \gamma(1 - p(S_0))\mathbb{E}[|\hat{\mathcal{T}}_1|], \tag{6}$$

where

$$p(S_i) = \begin{cases} (1-\rho)^{\frac{|S_i|}{\gamma}} & \text{if } i = 0, \\ (1-\rho)^{\frac{|S_i|}{\gamma}} \left(\frac{1-(1-\rho)^{|S_i|\frac{\gamma-1}{\gamma}}}{1-(1-\rho)^{|S_i|}} \right) & \text{otherwise.} \end{cases}$$

See Proof B.1. Hence, $\mathbb{E}[|\hat{\mathcal{T}}_0|]$ is a monotonically increasing function of the Bernoulli probability ρ, and it tends to $(n-1)/(\gamma-1)$ as $\rho \to 1$.

Accuracy and Approximation. Recall that h-Shap provides image attributions by means of a hierarchy of cooperative games. As a result, the attributions are different, in general, from those estimated by analyzing the grand coalition directly –that is, by the general Shapley approach in Eq. (2). Yet, we now show that under A1, h-Shap can in fact provide exact Shapley coefficients while being exponentially faster.

We begin by noting that under the MIL assumption, all positive features have the same importance. This agrees with intuition that the number of times the positive concept appears in the input image does not affect its label. We denote as Φ and $\widehat{\Phi}$ the exact and hierarchical Shapley coefficients, respectively, for simplicity.

Remark 3.3. *Under A1, and denoting $k = \|\Phi\|_0$, it holds that the exact saliency map Φ satisfies*

$$\phi_i = \begin{cases} 1/k & \text{if } x_i \sim \mathcal{I} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

This remark follows simply from the nullity and symmetry properties of Shapley coefficients. As a result, the saliency map computed by h-Shap, $\widehat{\Phi}$, as in Eq. (4), coincides with Φ under the MIL assumption. We now derive a more general similarity lower bound between Φ and $\widehat{\Phi}$ that allows for minimal feature sizes $s > 1$. For simplicity, we assume that n and s are powers of γ, and $1 \leq s \leq n$. First of all, because of the MIL assumption, h-Shap will *only* keep exploring nodes that have at least one important feature in them at each level of the hierarchy. Thus, for each important feature i with $\Phi_i = 1/k$ there will be a non-zero coefficient produced by h-Shap. The following result precisely quantifies to what extent these two vectors Φ and $\widehat{\Phi}$ match.

Theorem 3.4 (Similarity lower bound)**.** *Assume $X \in \mathbb{R}^n$ and $\hat{f}(X)$ satisfy A1, and $k = \|\Phi\|_0$. Then*

$$\frac{\langle \Phi, \widehat{\Phi} \rangle}{\|\Phi\|_2 \|\widehat{\Phi}\|_2} \geq \max\{1/\sqrt{s}, \sqrt{k/n}\}. \tag{8}$$

See Proof B.2. This result shows that not only does h-Shap provide faster image attributions, but it retrieves the exact Shapley coefficients defined in Eq. (7) under the MIL assumption if $s = 1$.

4. Experiments

We now move to demonstrate the performance of h-Shap and of other state-of-the-art methods for image attributions. Our objective is mainly to compare with other Shapley-based methods, such as GradientExplainer (Lundberg & Lee, 2017), DeepExplainer (Lundberg & Lee, 2017; Chen et al., 2021), and PartitionExplainer. We also include LIME (Ribeiro et al., 2016) given its relation to Shapley coefficients, and Grad-CAM (Selvaraju et al., 2017) because of its popularity. We study three comple-

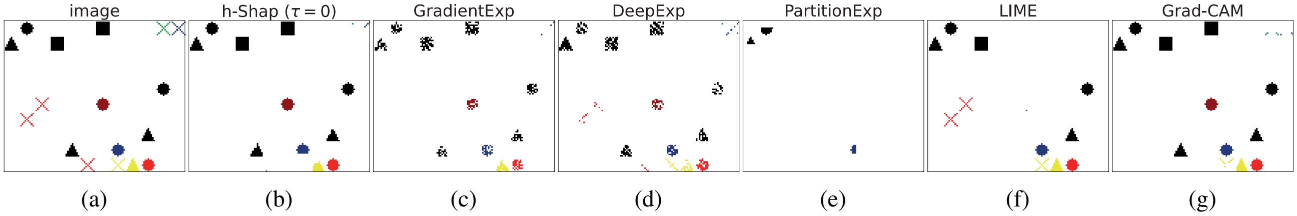


Figure 2. Ablation examples for all explanation methods removing all important pixels from the original image 2a. The model is trained to predict if a given image does contain a cross or not.

mentary binary classification problems of different complexity and input dimension: a simple synthetic benchmark, a medical imaging dataset, and a general computer vision task. We focus on scenarios where the ground truth of the image attributions (i.e. what defines the label) is well defined and available for evaluation. Our code is made available for the purpose of reproducibility[4]. When possible, each method was set to use as much GPU memory as possible, so as to minimize their runtime. DeepExplainer, GradientExplainer, and PartitionExplainer were constrained the most by memory, reflecting their limitation in analyzing large images. We use h-Shap with both an absolute threshold $\tau = 0$, and a relative threshold τ equal to the 70^{th} percentile, which we refer to as $\tau = 70\%$. Finally, we perform *full* randomization sanity checks (Adebayo et al., 2018) on the network used in the synthetic dataset for all explanation methods. We refer the reader to Supplementary Material D for these results.

Synthetic dataset. We created a controlled setting where the joint data distribution is completely known, giving us maximal flexibility for sampling. We generate images of size 100×120 pixels with a random number of non-overlapping geometric shapes of size 10×10 and of different colors, uniformly distributed across the image. Each image that contains at least one cross receives a positive label, and each image without any crosses receives a negative label. Alongside with the images, we generate the ground truth saliency maps by setting all pixels that precisely lie on a cross to 1, and every other pixel to 0. We generate 8000 positive and negative images, and we randomly sample train, validation, and test splits, with size $5000, 1000$ and 2000 images, respectively. We train a simple ConvNet architecture and achieve an accuracy greater than $99\,\%$ on the test set –implying that the model has effectively satisfied the MIL assumption for this problem. From the true positive predictions on the test set, we choose 300 example images with 1 cross and as many with 6 crosses to evaluate the saliency maps.

P. vivax (malaria) dataset. Moving on to a real and high-dimensional problem, we explore the BBBC041 dataset[5] (Ljosa et al., 2012). The dataset consists of 1328, 1200×1600 pixels blood smears with uninfected and malaria-infected cells. The dataset also comprises bounding-box annotations of both healthy and sick cells. We consider the binary problem of detecting images that contain at least one trophozoite. We apply transfer learning to a ResNet18 (He et al., 2016) network pretrained on ImageNet using cross-entropy loss. Our model achieves a test accuracy of $\approx 94\,\%$. We finally aggregate all 107 true positive predictions for evaluation.

LISA traffic light dataset. We finally look at a general computer vision dataset consisting of driving sequences[6] (Jensen et al., 2016; Philipsen et al., 2015). The complete dataset counts 43 007 frames of size 960×1280 pixels, and 113 888 annotated traffic lights. From this set, we take daytime traffic images, and train a model to predict the presence of a green light in a sample image. As before, we apply transfer learning on a pretrained ResNet18 with cross-entropy loss. After training, we achieve a test accuracy of $\approx 93\,\%$. Finally, we randomly sample 300 true positive examples to evaluate the different attribution methods on.

We refer to Supplementary Material C for a detailed description of the training procedures. Fig. 1a, 1b, and 1c show an example image for each dataset, and the respective saliency maps obtained with different explanation models (for more examples, see Fig. E.1).

5. Results

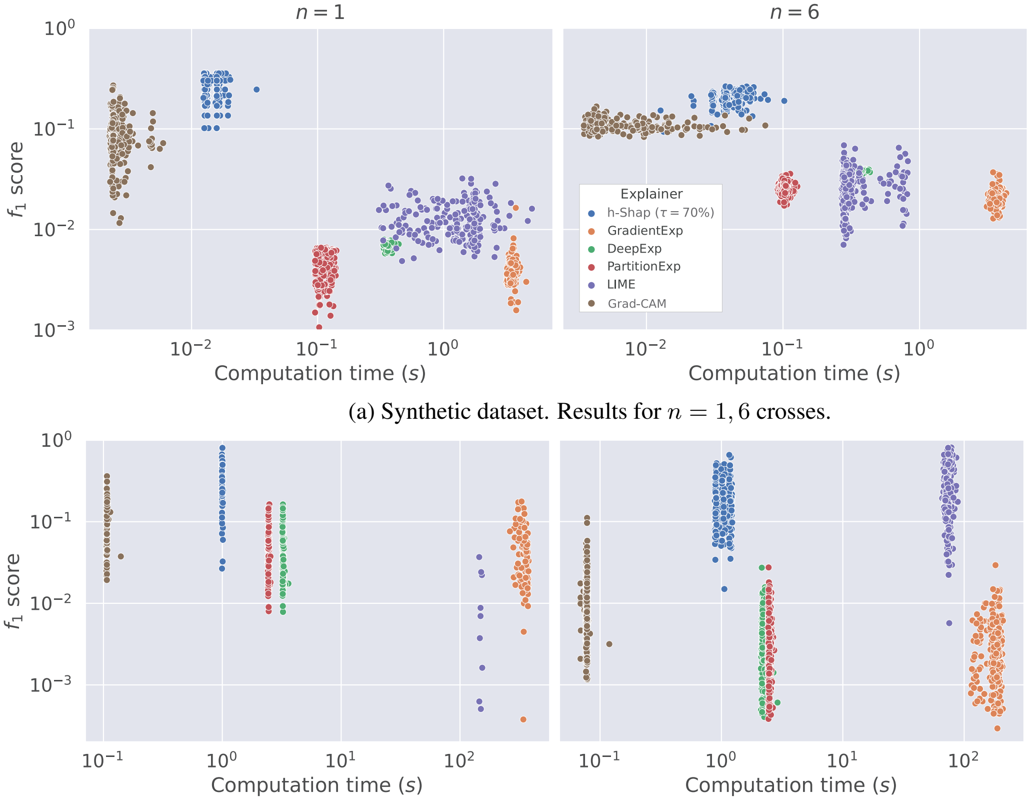
We evaluate the explanation methods by means of three performance measures: ablation tests, accuracy, and runtime.

Ablation tests. As commonly done in literature (Lundberg & Lee, 2017; Sturmfels et al., 2020; Haug et al., 2021) we remove the top k scoring features of all methods by setting them to their expected value, and plot the logit of the prediction as a function of k. For these experiments, we use $\tau = 0$ so as to find *all* the features that are relevant for the model. Fig. 2 shows full ablation results on one example image from the synthetic dataset for all explanation

[4] https://github.com/Sulam-Group/h-shap

[5] https://www.kaggle.com/kmader/malaria-bounding-boxes.

[6] https://www.kaggle.com/mbornoe/lisa-traffic-light-dataset.



(a) Synthetic dataset. Results for $n = 1, 6$ crosses.

(b) BBBC041 dataset.

(c) LISA dataset.

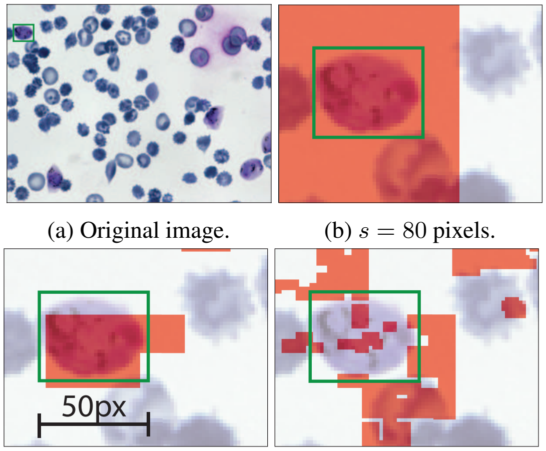
Figure 3. f_1 scores as a function of runtime for all explanation methods in all three experiments.

methods. We expect a perfect method to remove all crosses from the image –and only those. We can appreciate how h-Shap removes mostly only the crosses, while other methods also erase other shapes which should not be identified as important. Furthermore, removing more relevant features should produce a steeper drop of the prediction logit. We include the respective curves in Fig. E.2, depicting that h-Shap's logit curves either drop the fastest towards 0 or provide the lowest logit at complete ablation. Indeed, h-Shap quickly identifies the most relevant features in the image. Naturally, as tasks become harder, the accuracy of $\hat{f}$ decreases, and the model gets further away from the oracle function f^*. In these cases, $\hat{f}$ might not satisfy Eq. (3), resulting in noisier saliency maps, and subsequently, in non-monotonic curves.

Accuracy and Runtime. Since we have ground-truth explanations in all these cases (i.e. a cross, a sick cell, or a green traffic light), we use f_1 scores (Guidotti, 2021) as a measure of goodness of explanation. Fig. 3 depicts the f_1 scores as a function of runtime for every explanation method and experiment. The relevance tolerance τ allows to take into account the risk of the model $\hat{f}$ and discard noisy attributions, while also decreasing runtime. These results reflect how the computational cost and accuracy guarantees described earlier translate into application. Not only does h-Shap decrease runtime by almost two orders of magnitude compared to current Shapley-based explanation methods, but also it increases the f_1 score by more than one order of magnitude. In all experiments –both synthetic and real– h-Shap consistently provides more accurate and faster saliency maps.

6. Discussion and Limitations

h-Shap's most important limitation is its MIL assumption on the data distribution. Indeed, h-Shap is designed to identify local *findings* that produce a positive global response, accurately and efficiently. These are precisely the important features C analyzed in Sec. 3. This setting fails when the minimal feature size is much smaller than the size of the findings that define the label. As an example, Fig. 4 depicts a zoomed-in version of the map produced by h-Shap for one sample from the P. vivax dataset, for different values of s. We see that when s is somewhat smaller than the object, h-Shap still recognizes the important features in the image. Once s is too small, however, the resulting map breaks down, as our assumption does not hold anymore. Indeed, this simply implies that a small (e.g. 5×5 pixels) image patch of a cell is no longer necessary for the model to recognize the cell. In practice, these failure cases can easily be identified by deploying simple conditions searching over decreasing sizes of s (which would not increase



(a) Original image. (b) $s = 80$ pixels.

(c) $s = 20$ pixels. (d) $s = 5$ pixels.

Figure 4. Degradation of h-Shap's maps as the minimal feature size s becomes smaller than the target concept.

the computational cost).

Another limitation of h-Shap pertains the way hierarchical partitions are created. We have chosen to use quadrants for simplicity, and this is sub-optimal: important features can fall in-between quadrants, impacting performance. This limitation is minor, as it can be easily fixed by applying ideas of cycle spinning and averaging the resulting estimates. Furthermore, and more interestingly, hierarchical data-dependent partitions could also be employed. We regard this as future work.

Finally, we turn our attention to the masking strategy, i.e. how to sample the baseline. We recall that in this work we defined the variable X_C as

$$(X_C)_i = \begin{cases} X_i & \text{if } i \in C \\ R_i & \text{otherwise,} \end{cases} \tag{9}$$

where $R \in \mathbb{R}^{n-|C|}$ is a baseline value. Throughout this work, we have treated R as a fixed, deterministic quantity. However, more generally, reference inputs are random variables. Let this masked input be the random variable $X_C = [\bar{X}_C, R] \in \mathbb{R}^n$, where $\bar{X}_C \in \mathbb{R}^{|C|}$ is fixed, and R is a random variable. Here, we follow the original approach in (Lundberg & Lee, 2017). Indeed, the definition of ϕ_i in Eq. (2) can be made more precise by writing the expectation $\mathbb{E}[\hat{f}(X_{C\cup\{i\}}) - \hat{f}(X_C)]$ as

$$\mathbb{E}_R[\hat{f}([\bar{X}_{C\cup\{i\}}, R])|\bar{X}_{C\cup\{i\}}] - \mathbb{E}_R[\hat{f}([\bar{X}_C, R])|\bar{X}_C]. \tag{10}$$

Then, if the model $\hat{f}$ is linear, and the features are independent, Eq. (10) simplifies to

$$\hat{f}([\bar{X}_{C\cup\{i\}}, \mathbb{E}[R]]) - \hat{f}([\bar{X}_C, \mathbb{E}[R]]), \tag{11}$$

where $\mathbb{E}[R]$ is an unconditional expectation. $\mathbb{E}[R]$ can be easily computed over the training data, and is precisely the fixed baseline we employed in this work.

How realistic are these assumptions in our case? First, the cases that we study approximately satisfy feature independence in a local sense, i.e. when s is greater or similar to the size of the concept we are interested in detecting. This precisely holds in the synthetic dataset, where each 10×10 pixels shape is sampled independently from the others. This assumption is still approximately valid in the other two experiments, where the presence or absence of a cell does not affect the content of the image many pixels apart. On the other hand, while we have chosen general models $\hat{f}$ which are far from linear, we argue that A1 is enough to obtain a weaker sense of interpretability. Looking at

$$\hat{f}([X_C, \mathbb{E}[R]]), \tag{12}$$

and under the MIL assumption, there are only two mutually exclusive events for the subset C: (a) C contains at least one relevant feature, and (b) C does not contain any relevant features. When event (a) occurs, Eq. (12) will necessarily yield a value ≈ 1. It follows that if both $C \cup \{i\}$ and C contain important features, Eq. (11) will be ≈ 0; which agrees with intuition that all important features are equally important. As a result, because $\mathbb{E}[R]$ is fixed and A1 holds, a positive value of Eq.(11) is only attained if (i.e. implies that) i is an important feature (and it also implies that $\mathbb{E}[R]$ is not important).

Even though we have focused on binary classification tasks in this work, h-Shap could also easily be applied to multi-class settings by adapting the problem to a *1 vs all* scenario. Lastly, note that our method relies on $\hat{f}$ satisfying A1, and one should wonder when this holds. Such an assumption is true when f^* –the true classification rule $Y = f^*(X)$– satisfies A1 (which is true for a variety of problems, including the ones studied in our experiments), and $\hat{f}$ constitutes a good approximation for f^*. We show that such assumptions are reasonable in practical settings.

7. Conclusion

We presented a fast, scalable, and exact explanation method for image classification based on a hierarchical extension of Shapley coefficients. We showed that when the data distribution satisfies some multiple instance learning assumption, our method gains an exponential computational advantage while producing accurate –or approximate– results. Furthermore, we studied synthetic and real settings of varying complexity, demonstrating that h-Shap outperforms the current state-of-the-art methods in both accuracy and runtime, and suggesting that h-Shap acts as a weakly-supervised object detector. We have also presented and illustrated limitations of our approach, and addressing them is matter of future work.

References

Aas, K., Jullum, M., and Løland, A. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, pp. 103502, 2021.

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

Algaba, E. and van den Brink, R. The shapley value and games with hierarchies. *Handbook of the Shapley Value*, pp. 49, 2019.

Amann, J., Blasimme, A., Vayena, E., Frey, D., and Madai, V. I. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC Medical Informatics and Decision Making*, 20(1):1–9, 2020.

Chen, H., Janizek, J. D., Lundberg, S., and Lee, S.-I. True to the model or true to the data? *arXiv preprint arXiv:2006.16234*, 2020a.

Chen, H., Zheng, G., and Ji, Y. Generating hierarchical explanations on text classification via feature interaction detection. *arXiv preprint arXiv:2004.02015*, 2020b.

Chen, H., Lundberg, S., and Lee, S.-I. Explaining models by propagating shapley values of local components. In *Explainable AI in Healthcare and Medicine*, pp. 261–270. Springer, 2021.

Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. L-shapley and c-shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610*, 2018.

Covert, I., Lundberg, S., and Lee, S.-I. Explaining by removing: A unified framework for model explanation. *arXiv preprint arXiv:2011.14878*, 2020.

Faigle, U. and Peis, B. A hierarchical model for cooperative games. In *International Symposium on Algorithmic Game Theory*, pp. 230–241. Springer, 2008.

Frye, C., Feige, I., and Rowat, C. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *arXiv preprint arXiv:1910.06358*, 2019.

Fu, G., Nan, X., Liu, H., Patel, R. Y., Daga, P. R., Chen, Y., Wilkins, D. E., and Doerksen, R. J. Implementation of multiple-instance learning in drug activity prediction. In *BMC bioinformatics*, volume 13, pp. 1–12. BioMed Central, 2012.

Guidotti, R. Evaluating local explanation methods on ground truth. *Artificial Intelligence*, 291:103428, 2021.

Hacker, P., Krestel, R., Grundmann, S., and Naumann, F. Explainable ai under contract and tort law: legal incentives and technical challenges. *Artificial Intelligence and Law*, pp. 1–25, 2020.

Han, Z., Wei, B., Hong, Y., Li, T., Cong, J., Zhu, X., Wei, H., and Zhang, W. Accurate screening of covid-19 using attention-based deep 3d multiple instance learning. *IEEE transactions on medical imaging*, 39(8):2584–2594, 2020.

Hashimoto, N., Fukushima, D., Koga, R., Takagi, Y., Ko, K., Kohno, K., Nakaguro, M., Nakamura, S., Hontani, H., and Takeuchi, I. Multi-scale Domain-adversarial Multiple-instance CNN for Cancer Subtype Classification with Unannotated Histopathological Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2020.

Haug, J., Zürn, S., El-Jiz, P., and Kasneci, G. On baselines for local feature attributions. *arXiv preprint arXiv:2101.00905*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Heiß, C., Levie, R., Resnick, C., Kutyniok, G., and Bruna, J. In-distribution interpretability for challenging modalities. *arXiv preprint arXiv:2007.00758*, 2020.

Janzing, D., Minorics, L., and Blöbaum, P. Feature relevance quantification in explainable ai: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pp. 2907–2916. PMLR, 2020.

Jensen, M. B., Philipsen, M. P., Møgelmose, A., Moeslund, T. B., and Trivedi, M. M. Vision for looking at traffic lights: Issues, survey, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, 17(7):1800–1815, 2016.

Kaminski, M. E. The right to explanation, explained. *Berkeley Tech. LJ*, 34:189, 2019.

Kaminski, M. E. and Malgieri, G. Algorithmic impact assessments under the gdpr: producing multi-layered explanations. *U of Colorado Law Legal Studies Research Paper*, (19-28), 2019.

Ljosa, V., Sokolnicki, K. L., and Carpenter, A. E. Annotated high-throughput microscopy image sets for validation. *Nature methods*, 9(7):637–637, 2012.

López, S. and Saboya, M. On the relationship between shapley and owen values. *Central European Journal of Operations Research*, 17(4):415, 2009.

Lundberg, S. and Lee, S.-I. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

MacDonald, J., Wäldchen, S., Hauch, S., and Kutyniok, G. A rate-distortion framework for explaining neural network decisions. *arXiv preprint arXiv:1905.11092*, 2019.

Merrick, L. and Taly, A. The explanation game: Explaining machine learning models using shapley values. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pp. 17–38. Springer, 2020.

Owen, G. Values of games with a priori unions. In *Mathematical economics and game theory*, pp. 76–88. Springer, 1977.

Owen, G. *Game Theory*. Academic Press New York, 3rd edition, 1995. ISBN 0125311516.

Philipsen, M. P., Jensen, M. B., Møgelmose, A., Moeslund, T. B., and Trivedi, M. M. Traffic light detection: A learning algorithm and evaluations on challenging dataset. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 2341–2345. IEEE, 2015.

Ribeiro, M. T., Singh, S., and Guestrin, C. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Shapley, L. S. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

Shin, D. The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable ai. *International Journal of Human-Computer Studies*, 146:102551, 2021.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pp. 3145–3153. PMLR, 2017.

Singh, C., Murdoch, W. J., and Yu, B. Hierarchical interpretations for neural network predictions. *arXiv preprint arXiv:1806.05337*, 2018.

Sturmfels, P., Lundberg, S., and Lee, S.-I. Visualizing the impact of feature attribution baselines. *Distill*, 5(1):e22, 2020.

Sundararajan, M. and Najmi, A. The many shapley values for model explanation. In *International Conference on Machine Learning*, pp. 9269–9278. PMLR, 2020.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

Tomsett, R., Braines, D., Harborne, D., Preece, A., and Chakraborty, S. Interpretable to whom? a role-based model for analyzing interpretable machine learning systems. *arXiv preprint arXiv:1806.07552*, 2018.

Weidmann, N., Frank, E., and Pfahringer, B. A two-level learning method for generalized multi-instance problems. In *European Conference on Machine Learning*, pp. 468–479. Springer, 2003.

Zednik, C. Solving the black box problem: a normative framework for explainable artificial intelligence. *Philosophy & Technology*, pp. 1–24, 2019.

Supplementary Material

A. Algorithms

Here, we describe the two versions of h-Shap presented in Sec. 3. Algorithm 1 details *d*h-Shap (depth-first h-Shap), and Algorithm 2 describes *b*h-Shap (breadth-first h-Shap). We recall that both implementations return the set of relevant leaves $L \subseteq [n] \coloneqq \{1, \dots, n\}$ such that their Shapley coefficients are greater than a relevance tolerance τ. The former implementation, *d*h-Shap, uses an absolute tolerance, while the latter, *b*h-Shap, uses a relative tolerance.

Algorithm 1 Depth-first h-Shap

```
procedure $d$H-SHAP$(X, \mathcal{T}_0, \hat{f})$
inputs: image $X$, threshold $\tau \geq 0$, trained model $\hat{f}$
    $g_0 \leftarrow (X, \hat{f}, c(S_0))$
    $\phi_{0,1}, \dots, \phi_{0,\gamma} \leftarrow \texttt{shap}(g_0)$
    for all $\phi_i$ do
        if $\phi_i > \tau$ then
            if $|S_i| \leq s$ then
                return $S_i$
            else
                return $d$h-Shap$(X, \mathcal{T}_i, \hat{f})$
            end if
        end if
    end for
end procedure
$L \leftarrow d$h-Shap$(X, \mathcal{T}_0, \hat{f})$
```

Algorithm 2 Breadth-first h-Shap

```
procedure $b$H-SHAP$(X, \mathcal{T}_0, \hat{f})$
inputs: image $X$, threshold $\tau \geq 0$, trained model $\hat{f}$
    $L \leftarrow \emptyset$
    $l \leftarrow S_0$
    while $l$ is not empty do
        $\Phi_l \leftarrow \emptyset$
        for all $S_i \in l$ do
            $g_i \leftarrow (X, \hat{f}, c(S_i))$
            $\phi_{i,1}, \dots, \phi_{i,\gamma} \leftarrow \texttt{shap}(g_i)$
            $\Phi_l \leftarrow \Phi_l \cup \phi_{i,1}, \dots, \phi_{i,\gamma}$
        end for
        $\tau \leftarrow \tau(\Phi_l)$
        $l' \leftarrow \emptyset$
        for all $\phi_i \in \Phi_l$ do
            if $\phi_i \geq \tau$ then
                if $|S_i| \leq s$ then
                    $L \leftarrow L \cup S_i$
                else
                    $l' \leftarrow l' \cup S_i$
                end if
            end if
        end for
        $l \leftarrow l'$
    end while
    return $L$
end procedure
$L \leftarrow b$h-Shap$(X, \mathcal{T}_0, \hat{f})$
```

B. Proofs

We summarize here the assumptions and notation used in the following results. Let $X \in \mathbb{R}^n$ be drawn so that each entry $x_i \sim a_i\mathcal{I} + (1 - a_i)\mathcal{I}^c$, where $a_i \sim \text{Bernoulli}(\rho)$ is a binary random variable that indicates whether the feature x_i comes from an *important* distribution $\mathcal{I}$, or its *non-important* complement $\mathcal{I}^c$. Let

$$\hat{f}(X_C) = 1 \iff \exists i \in C : \; x_i \sim \mathcal{I}, \; C \subseteq [n],$$

where $n \coloneqq \{1, \dots, n\}$ and $X_C \in \mathbb{R}^n$ is equal to X in the entries of C and takes value in the baseline in its complement $\bar{C}$. We denote with $\Phi_{(X,\hat{f})} = \{\phi_1(\hat{f}), \dots, \phi_n(\hat{f})\} \in \mathbb{R}^n$ the saliency map of X where $\phi_i(\hat{f})$ is the Shapley coefficient of x_i. Let $k = \|\Phi_{(X,\hat{f})}\|_0$ be the number of reported important features by the exact Shapley coefficients. We showed earlier (see Eq. (7)) that under these assumptions, it follows:

$$\phi_i(\hat{f}) = \begin{cases} 1/k & \text{if } x_i \sim \mathcal{I} \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, let $\mathcal{T}_0 = (S_0, \mathcal{T}_1, \dots, \mathcal{T}_\gamma)$ be the recursive definition of a γ-partition tree of X such that $S_0 = [n]$; $\mathcal{T}_i, \dots, \mathcal{T}_\gamma$ are the subtrees branching off of S_0; and $c(S_i)$ are the γ children of the node S_i. Recall that h-Shap explores $\mathcal{T}_0$ from S_0 and returns all relevant leaves $L \subseteq [n]$ such that their Shapley coefficient is greater than a relevance tolerance τ. We denote with $\widehat{\mathcal{T}}_0$ the subtree composed of the nodes visited by h-Shap, and with $\widehat{\Phi}_{(X,\hat{f})}$ the saliency map computed by h-Shap, such that

$$\widehat{\phi}_i(\hat{f}) = \begin{cases} 1/|L| & \text{if } i \in L \\ 0 & \text{otherwise.} \end{cases}$$

Now, we will provide proof of the Theorems presented in Sec. 3.

B.1. Expected number of visited nodes 3.2

Here, we are interested in evaluating the expected number of nodes visited by h-Shap, to better characterize its computational advantage.

Proof. Recall that S_0 contains all features of X. That is,

$S_0 = [n]$. Since $x_1, \dots, x_n \in X$ are iid, so are groups of features. Then, it suffices to analyze each child of S_0 independently. Consider the two mutually exclusive events on the child node $c_i \in c(S_0)$: **1)** it does not contain any important features, i.e. $\nexists\, j \in c_i : \ \hat{f}(X_j) = 1$; and **2)** it contains at least one important feature, i.e. $\exists\, j \in c_i : \ \hat{f}(X_j) = 1$. Let $p_1(S_0)$ be the probability of event 1, and $1 - p_1(S_0)$ be the probability of event 2. When event 2 occurs, we add one node to $\hat{\mathcal{T}}_0$, and then we explore the subtree $\hat{\mathcal{T}}_i$ branching off of c_i. We can recursively apply this strategy to each subtree of $\hat{\mathcal{T}}_0$, which yields

$$E[|\hat{\mathcal{T}}_0|] = 1 + \gamma(1 - p_1(S_0))E[|\hat{\mathcal{T}}_1|]. \tag{13}$$

We are left with evaluating $p_1(S_0)$, which simply is

$$p_1(S_0) = (1-\rho)^{|S_0|/\gamma} \tag{14}$$

since the probability for x_k not to be important, i.e $x_k \sim \mathcal{I}^c$, is $(1-\rho)$, and all the children $c(S_0)$ have cardinality $n/\gamma = |S_0|/\gamma$ because they form a disjoint symmetric partition of S_0. When analyzing the i^{th} subtree branching off of S_0, $\hat{\mathcal{T}}_i$, one has to condition on the probability of the event that S_i contains at least one important feature. The probability $p'(S_i)$ of the event that S_i contains at least one important feature is, again, simply $1 - (1-\rho)^{|S_i|}$. Therefore

$$p_1(S_i) = \frac{(1-\rho)^{|S_i|/\gamma}(1 - (1-\rho)^{|S_i|(\gamma-1)/\gamma})}{1-(1-\rho)^{|S_i|}} \tag{15}$$

is the conditioned probability that a child of S_i does not contain any important features. □

B.2. Similarity lower bound 3.4

Here, we want to find the lower bound of the similarity between Φ and $\widehat{\Phi}$, defined as

$$\alpha = \frac{\langle \Phi, \widehat{\Phi} \rangle}{\|\Phi\|_2 \|\widehat{\Phi}\|_2}.$$

Proof. Let $k = \|\Phi\|_0$ be the number of reported important features in X as returned by the Shapley coefficients. Let $L \subseteq [n]$ be the relevant leaves returned by h-Shap. From Eq. (7) and (4) it follows that

$$\|\Phi\|_2 = \sqrt{\frac{1}{k^2}k} = \sqrt{\frac{1}{k}}, \tag{16}$$

$$\|\widehat{\Phi}\|_2 = \sqrt{\frac{1}{(\ell s)^2}\ell s} = \sqrt{\frac{1}{\ell s}}, \tag{17}$$

where $|L| = \ell s$, ℓ is the number of relevant leaves, and s is the minimal feature size. Furthermore, we know that

$$\langle \Phi, \widehat{\Phi} \rangle = k\left(\frac{1}{k}\frac{1}{\ell s}\right) = \frac{1}{\ell s}. \tag{18}$$

Therefore

$$\alpha = \frac{\langle \Phi, \widehat{\Phi} \rangle}{\|\Phi\|_2 \|\widehat{\Phi}\|_2} = \frac{\frac{1}{\ell s}}{\frac{1}{\sqrt{\ell s k}}} = \sqrt{\frac{k}{\ell s}}. \tag{19}$$

Fixed s and k, α is a monotonically decreasing function of ℓ, which means that minimizing the similarity between Φ and $\widehat{\Phi}$ is equivalent to maximizing the number of leaves returned by h-Shap. When $k \leq n/s$, $\ell \leq k$, so $\alpha \geq \sqrt{k/(ks)} = 1/\sqrt{s}$. When $k > n/s$, $|L| = n$, therefore $\alpha \geq \sqrt{k/n}$. □

C. Experimental details

C.1. Synthetic dataset

Table 1 represents the network architecture used in the synthetic dataset experiment. We optimize for 50 epochs with Adam optimizer, learning rate 0.001 and cross-entropy loss.

Layer	Filter size	Input size
Conv_1	$6 \times (3 \times 5 \times 5)$	$3 \times 100 \times 120$
ReLU_1	–	$6 \times 96 \times 116$
MaxPool_1	2×2	$6 \times 96 \times 116$
Conv_2	$16 \times (6 \times 4 \times 4)$	$6 \times 48 \times 58$
ReLU_2	–	$16 \times 45 \times 55$
MaxPool_2	5×5	$16 \times 45 \times 55$
FC_1	1584×120	1584×1
ReLU_3	–	120×1
Dropout_1	–	120×1
FC_2	120×84	120×1
ReLU_4	–	84×1
Dropout_2	–	84×1
FC_3	84×2	84×1

Table 1. Network architecture for the synthetic dataset experiment

C.2. P. vivax (malaria), LISA datasets

In both experiments, we optimize all parameters of a pretrained ResNet18 for 25 epochs with stochastic gradient descent – learning rate 0.001, momentum 0.9. We use cross-entropy loss and learning rate decay of 0.2 every 10 epochs.

D. Sanity checks

Some interpretability methods have been shown (Adebayo et al., 2018) to be unreliable in that they do not truly rely on what the model has learned, i.e. the precise parametrization of $\hat{f}$. For this reason, (Adebayo et al., 2018) advocates for some *sanity checks*. Following this observation, we perform full model randomization tests on all methods compared in this work. The intuition

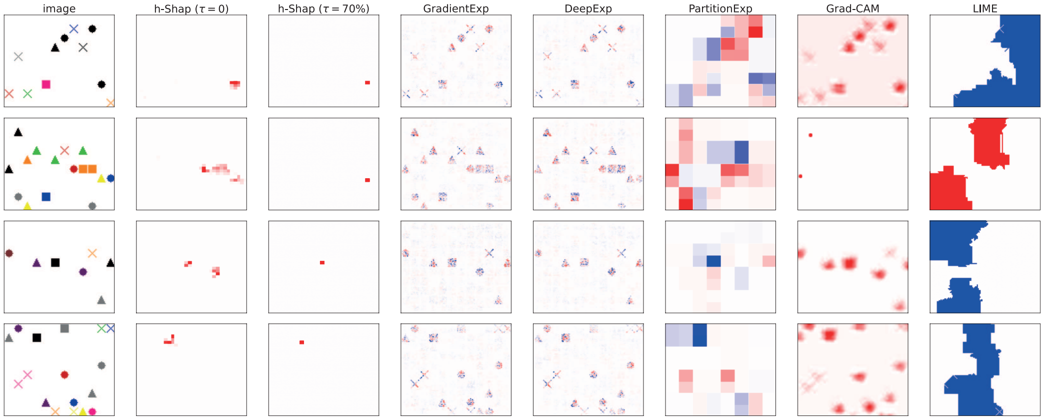
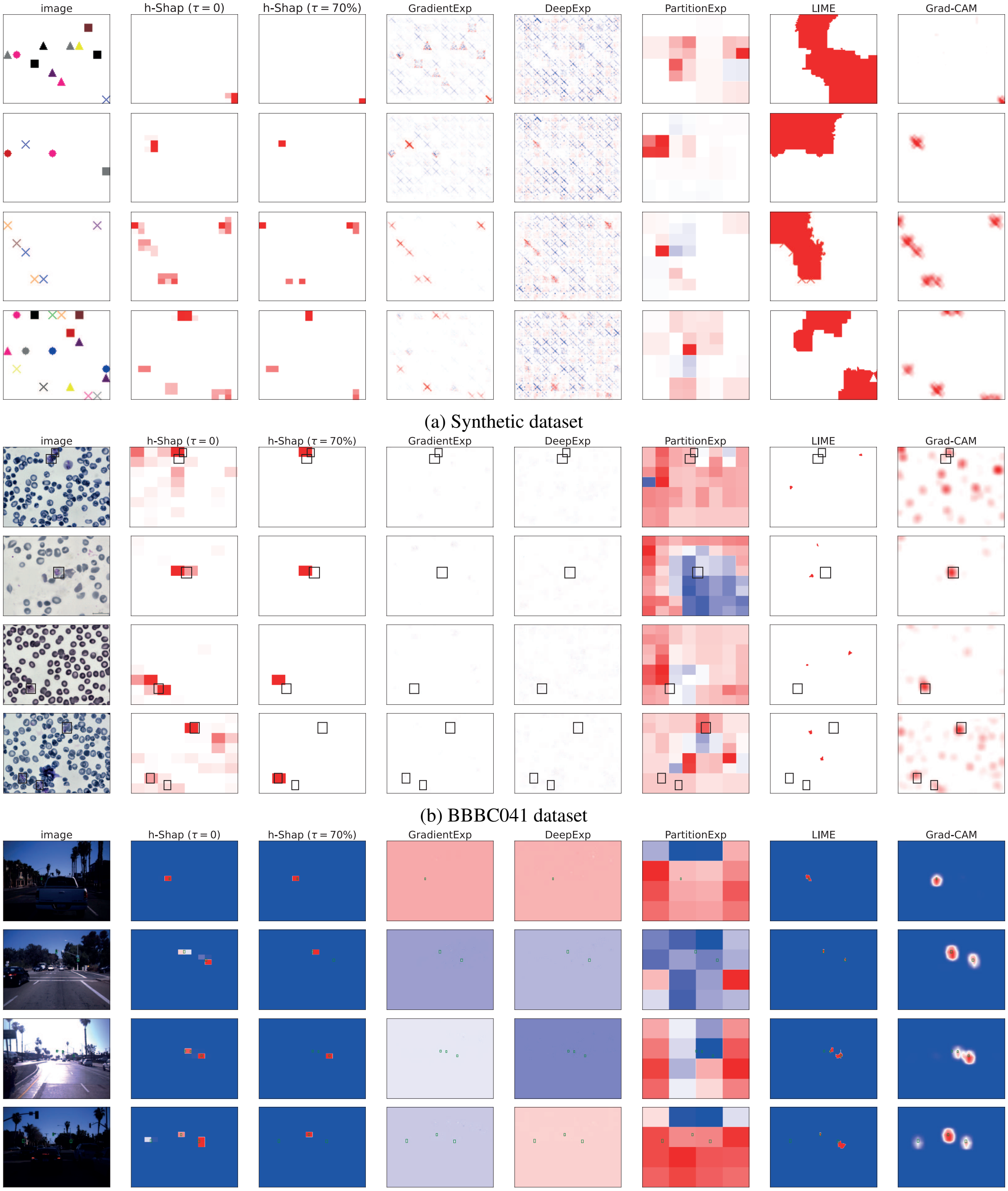


Figure D.1. Examples of full model randomization tests in the synthetic dataset.

behind model randomization tests is that if the explanation method actually depends on features learned by the model, the explanations should degrade as model weights are randomized. We perform *full* randomization tests in the sense that we randomly initialize *all* the parameters in the simple network described above in Table 1. Fig. D.1 shows that all explanation methods employed in this work pass the model randomization test, in the sense that the saliency maps degrade completely with a random model.

E. Figures

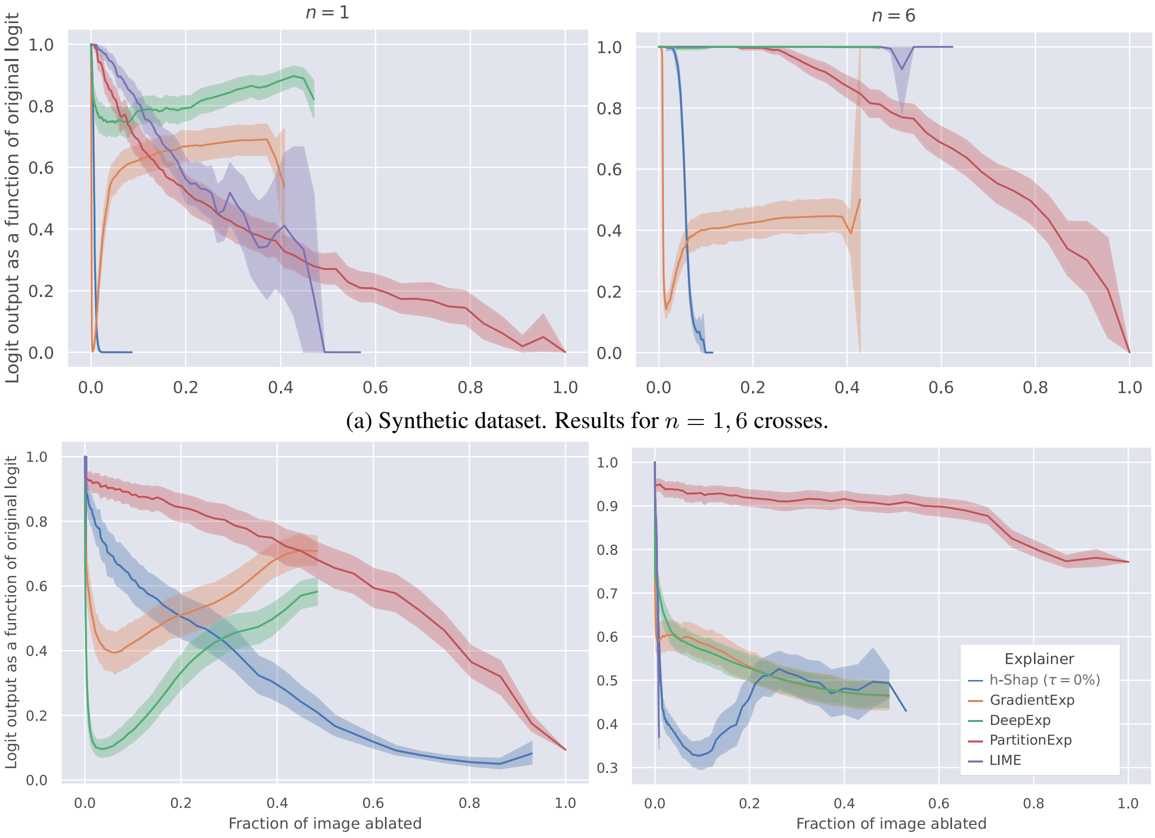


(a) Synthetic dataset

(b) BBBC041 dataset

(c) LISA dataset

Figure E.1. More examples of saliency maps.



(a) Synthetic dataset. Results for $n = 1, 6$ crosses.

(b) BBBC041 dataset

(c) LISA dataset

Figure E.2. Logit output compared to original logit output as a function of image ablation.