Journal of Artificial Intelligence Research 70 (2021) 351–387 Submitted 08/2020; published 01/2021

The Computational Complexity of Understanding Binary Classifier Decisions

Stephan Wäldchen STEPHANW@MATH.TU-BERLIN.DE
Jan Macdonald MACDONALD@MATH.TU-BERLIN.DE
Sascha Hauch SASCHA.HAUCH@ALUMNI.TU-BERLIN.DE
Institut für Mathematik,
Technische Universität Berlin,
Berlin, Germany

Gitta Kutyniok KUTYNIOK@MATH.LMU.DE
Mathematisches Institut,
Ludwig-Maximilians-Universität München,
München, Germany

Department of Physics and Technology,
University of Tromsø,
Tromsø, Norway

Abstract

For a d-ary Boolean function $\Phi\colon \{0,1\}^d \to \{0,1\}$ and an assignment to its variables $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ we consider the problem of finding those subsets of the variables that are sufficient to determine the function value with a given probability δ. This is motivated by the task of interpreting predictions of binary classifiers described as Boolean circuits, which can be seen as special cases of neural networks. We show that the problem of deciding whether such subsets of relevant variables of limited size $k \leq d$ exist is complete for the complexity class $\mathsf{NP}^{\mathsf{PP}}$ and thus, generally, unfeasible to solve. We then introduce a variant, in which it suffices to check whether a subset determines the function value with probability at least δ or at most $\delta - \gamma$ for $0 < \gamma < \delta$. This promise of a probability gap reduces the complexity to the class $\mathsf{NP}^{\mathsf{BPP}}$. Finally, we show that finding the minimal set of relevant variables cannot be reasonably approximated, i.e. with an approximation factor $d^{1-\alpha}$ for $\alpha > 0$, by a polynomial time algorithm unless $\mathsf{P} = \mathsf{NP}$. This holds even with the promise of a probability gap.

1. Introduction

Algorithmic problem solving in real-world scenarios often requires reasoning in an uncertain environment. This necessity leads to the investigation of probabilistic satisfiability problems and probabilistic computational complexity classes such as PP and $\mathsf{NP}^{\mathsf{PP}}$. One prototypical example, the E-Maj-Sat problem (Littman et al., 1998, 2001), is an extension of the classical satisfiability problem that includes an element of model counting. The class of $\mathsf{NP}^{\mathsf{PP}}$-complete problems contains many relevant artificial intelligence (AI) problems such as probabilistic conformant planning (Littman et al., 1998), calculating maximum expected utility (MEU) solutions (de Campos & Ji, 2008), and maximum a posteriori (MAP) hypotheses (Park, 2002).

We connect these probabilistic reasoning tasks to a key problem in machine learning, namely the problem of interpreting the decisions of neural network classifiers. For this, we extend the concept of prime implicants (Marquis, 1991, 2000) for Boolean functions to a probabilistic setting, which formalises existing practical attempts to interpret neural network classifiers.

1.1 Motivation

Neural networks are parameter-rich, highly nonlinear models and can be seen as continuous generalisations of Boolean circuits. This is briefly visualised in Figure 1. They have achieved impressive success in classification (Graves et al., 2013; Krizhevsky et al., 2012; Szegedy et al., 2013), regression (Sun et al., 2013; Toshev & Szegedy, 2014; Taigman et al., 2014) and reconstruction tasks (Kang et al., 2017; Xie et al., 2012).[1]

The same expressiveness that allows for hierarchical reasoning (Fukushima, 1980) and universal approximation (Hornik, 1991) makes understanding and interpreting these models more challenging compared to traditional machine learning methods such as linear regression or decision trees. Further, treating neural networks as "black box" solvers without accessible reasoning is not feasible in critical applications such as medical imaging and diagnosis (McBee et al., 2018; Shen et al., 2017).

A significant first step towards understanding network decisions is to distinguish the relevant input variables from the less relevant ones for a specific prediction, as illustrated in Figure 2. This has been pursued predominantly in the form of visual maps that assign an importance value to each input parameter (Bach et al., 2015; Erhan et al., 2009; Simonyan et al., 2014; Zeiler & Fergus, 2014).

A stringent logical concept that captures the idea of relevance for Boolean circuits are prime implicant explanations (Shih et al., 2018). These consist of subsets of the input variables that, if held fixed, guarantee that the function value remains unchanged, independent of the assignment to the rest of the variables. The problem of finding small prime implicants is $\mathsf{NP}^{\mathsf{coNP}}$-hard (Eiter & Gottlob, 1995), and practical algorithms rely on highly optimised SAT or MILP-solvers even for relatively low-dimensional cases (Ignatiev et al., 2019). In general, since the classifier function is fixed within each problem instance, such complexity results carry over to all types of classifiers that are able to efficiently represent Boolean circuits. A prominent example are ReLU-neural networks with weights and biases in $\{-1, 0, 1\}$. They can emulate Boolean circuits of comparable width and depth (Mukherjee & Basu, 2017; Parberry, 1996), as illustrated in Figure 1.

Prime implicant explanations can be seen as a type of explanation under worst-case conditions: the explaining set of variables is required to be sufficient for the function value to remain unchanged for *all* possible assignments to the other variables.

In this paper, we argue to relax this notion and allow the function value to change with a small probability over random assignments to the non-relevant variables. This has two main reasons. First, a worst case analysis might be feasible for binary classifiers of a few variables, but it is too rigid for very high dimensional cases, such as modern image classification. In many cases, it would lead to unnecessarily large sets of relevant variables that are not able

[1] For excellent introductions to neural networks we recommend the books by Goodfellow et al. (2016) and Nielsen (2018).

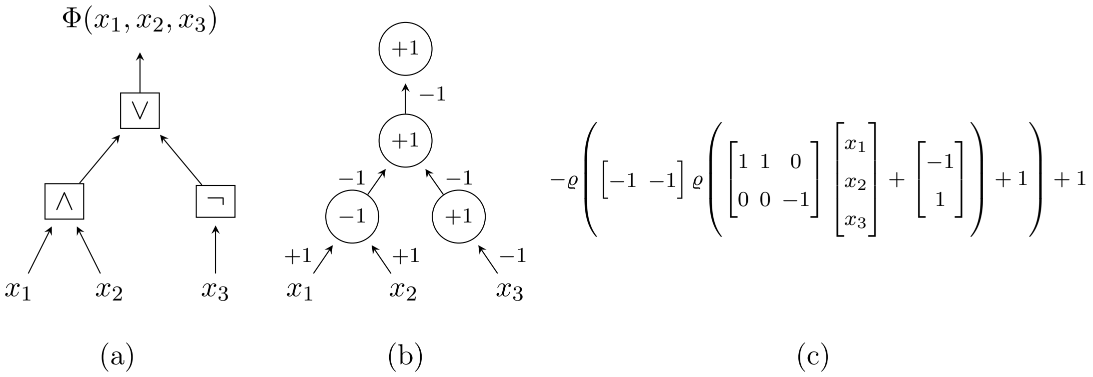
$\Phi(x_1, x_2, x_3)$
$\vee$
$\wedge$
$\neg$
x_1
x_2
x_3
(a)
$+1$
-1
$+1$
-1
-1
-1
$+1$
$+1$
$+1$
-1
x_1
x_2
x_3
(b)
$-\varrho\left(\begin{bmatrix}-1 & -1\end{bmatrix}\varrho\left(\begin{bmatrix}1 & 1 & 0\\ 0 & 0 & -1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}-1\\ 1\end{bmatrix}\right)+1\right)+1$
(c)

Figure 1: The Boolean function $\Phi(x_1, x_2, x_3) = (x_1 \wedge x_2) \vee (\neg x_3)$ viewed as a Boolean circuit (a) and a rectified linear unit (ReLU) neural network in its graphical (b) and algebraic representation (c). The neural network weights and biases are denoted at the edges and nodes respectively. The ReLU activation $\varrho(x) = \max\{x, 0\}$ is applied components-wise.

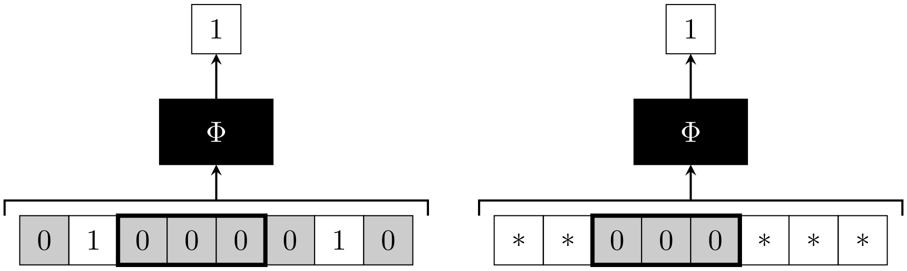

Figure 2: The Boolean function $\Phi\colon \{0,1\}^8 \to \{0,1\}$ decides if a binary input string contains a substring of three consecutive zeros. A relevant subset of input variables for an exemplary input is highlighted by a box. In this simple case the three consecutive zeros are relevant, because it is sufficient to know them to predict the decision made by Φ, independent of all other input variables. Note, that in this example the relevant set is not unique, as there are two sets of three consecutive zeros.

to pinpoint the true importance of variables. This is explained in more detail in Section 2.1. Secondly, practical heuristic algorithms for determining sets of important variables (Fong & Vedaldi, 2017; Ribeiro et al., 2018; Khosravi et al., 2019) as well as methods to numerically evaluate and compare them (Fong & Vedaldi, 2017; Samek et al., 2017a; Zeiler & Fergus, 2014), already implicitly rely on this relaxed probabilistic formulation of relevance. They estimate the expected change in the function value via random sampling of non-relevant variables. Thus, practical interpretation algorithms necessarily need to solve the problem defined in the next section and are subject to our hardness results. A rigorous analysis of this setting is long overdue and of high importance.

1.2 Notation

Throughout the paper $d \in \mathbb{N}$ denotes the arity of the Boolean function $\Phi\colon \{0,1\}^d \to \{0,1\}$, and $\mathbf{x} = (x_1, \dots, x_d) \in \{0,1\}^d$ is an arbitrary fixed assignment to its variables. We denote the d-dimensional vectors of all zeros or ones by $\mathbf{0}_d$ and $\mathbf{1}_d$ respectively. We refer to Φ as a Boolean circuit when its description is in terms of standard logical gates like AND, OR, and NOT. The description length is the number of gates. We use the usual symbols $\wedge$, $\vee$, $\neg$, and $\oplus$ for the logical conjunction, disjunction, negation, and exclusive disjunction respectively. We denote $[d] = \{1, \dots, d\}$, and for a subset $S \subseteq [d]$ we denote the restriction of $\mathbf{x}$ to the components with index in S by $\mathbf{x}_S = (x_i)_{i \in S}$. Further, we will use Boolean functions also interchangeably as logical propositions, in the sense that $\Phi(\mathbf{x})$ is shorthand for the logical proposition $\Phi(\mathbf{x}) = 1$. Whenever we discuss statements concerning probabilities of logical propositions to hold, we assume independent uniform distributions for all involved variables. Thus,

$$P_{\mathbf{y}}(\Phi(\mathbf{y})) = \frac{\left|\{\, \mathbf{y} \in \{0,1\}^d \,:\, \Phi(\mathbf{y}) = 1 \,\}\right|}{\left|\{\, \mathbf{y} \in \{0,1\}^d \,\}\right|},$$

and, conditioned to some event $A(\mathbf{y})$,

$$P_{\mathbf{y}}(\Phi(\mathbf{y}) \,|\, A(\mathbf{y})) = \frac{\left|\{\, \mathbf{y} \in \{0,1\}^d \,:\, \Phi(\mathbf{y}) = 1, A(\mathbf{y}) = 1 \,\}\right|}{\left|\{\, \mathbf{y} \in \{0,1\}^d \,:\, A(\mathbf{y}) = 1 \,\}\right|}.$$

We omit the subscript whenever it is clear from the context over which variables the probability is taken. If the probability is taken over all variables of a Boolean function, we simply write $P(\Phi)$ instead of $P_{\mathbf{y}}(\Phi(\mathbf{y}))$.

2. Problem Formulation and Complexity Results

Let us now give a formal definition of our probabilistic notion of prime implicant explanations and state the two main results of this paper.

A subset $S \subseteq [d]$ of variables is *relevant* for the function value $\Phi(\mathbf{x})$ if fixing $\mathbf{x}$ on S and randomising it on the complement S^c does not change the value of Φ with high probability. The complement then consists of the *non-relevant* variables.

Definition 2.1. Let $\Phi\colon \{0,1\}^d \to \{0,1\}$, $\mathbf{x} \in \{0,1\}^d$, and $\delta \in [0,1]$. We call $S \subseteq [d]$ a *δ-relevant* set for Φ and $\mathbf{x}$, if

$$P_{\mathbf{y}}(\Phi(\mathbf{y}) = \Phi(\mathbf{x}) \,|\, \mathbf{y}_S = \mathbf{x}_S) \geq \delta.$$

For δ close to one this means that the input $\mathbf{x}$ supported on S already determines the output $\Phi(\mathbf{x})$ with high probability. It is clear that $S = [d]$ is always one-relevant, and any subset $S \subseteq [d]$ is at least zero-relevant. Now, the question arises whether for a given δ there exists a δ-relevant set of a certain size. Similarly, one could ask to find the smallest δ-relevant set. This set would then be composed of the most important variables for the function value $\Phi(\mathbf{x})$. This introduces a trade-off since a larger δ will generally require a larger set S.

Definition 2.2. For $\delta \in (0,1]$ we define the δ-Relevant-Input problem as follows.

Given: A Boolean circuit $\Phi\colon \{0,1\}^d \to \{0,1\}$, $\mathbf{x} \in \{0,1\}^d$, and $k \in \mathbb{N}$, $1 \leq k \leq d$.

Decide: Does there exist $S \subseteq [d]$ with $|S| \leq k$ such that S is δ-relevant for Φ and $\mathbf{x}$?

Note that in our formulation δ is not a part of the problem instance, but instead each choice of δ represents a problem class, similar to k-SAT. We show that the problem is hard for any fixed δ. The minimisation formulation of the above decision problem can be defined in the obvious way.

Definition 2.3. For $\delta \in (0, 1]$ we define the Min-δ-Relevant-Input problem as follows.

Given: A Boolean circuit $\Phi \colon \{0,1\}^d \to \{0,1\}$ and $\mathbf{x} \in \{0,1\}^d$.

Task: Find the minimal $k \in \mathbb{N}$ such that there exists $S \subseteq [d]$ with $|S| \leq k$ and S is δ-relevant for Φ and $\mathbf{x}$.

The majority of the remainder of the paper will deal with analysing the computational complexity of δ-Relevant-Input, Min-δ-Relevant-Input, and related variants thereof. Our first main contribution shows that the δ-Relevant-Input problem is generally hard to solve.

Theorem 2.4. *For $\delta \in (0, 1)$ the δ-Relevant-Input problem is $\mathsf{NP}^{\mathsf{PP}}$-complete.*

Intuitively, the NP-part of the problem complexity arises from the necessity to check all subsets $S \subseteq [d]$ as possible candidates for being δ-relevant. The PP-part of the complexity arises from the fact that for any given set S checking if it is δ-relevant is by itself a hard (in fact PP-hard)[2] problem. The problem class $\mathsf{NP}^{\mathsf{PP}}$ is beyond the scope of conventional computing. In particular, Min-δ-Relevant-Input is at least as hard to solve as the corresponding decision problem, which makes it unfeasible to solve exactly. However, in applications it is rarely required to exactly find the smallest relevant set. It would be desirable to obtain good approximate solutions within feasible computational complexity.

We present two potential ways for simplifying the problem by allowing approximations: First, we relax the requirement that a solutions set has to be exactly δ-relevant. Secondly, we allow an approximation of the the minimal relevant set in terms of its size. The former would address the PP part whereas the latter would address the NP aspect.

Calculating probabilities or expectation values may be hard in theory, yet it is often easy to calculate them (approximately) in practice, e.g. by sampling. Checking whether a logical proposition is satisfied with probability more than δ by sampling only fails if the true probability can be arbitrarily close to δ both from above and below. These edge cases cause the hardness of the problem, but in our scenario we do not necessarily care about their resolution. We make this notion formal by stating a promise version of our problem where, if the true probability is smaller than δ, it will be smaller by at least γ with $0 \leq \gamma < \delta$. We refer to this as the γ-Gapped-δ-Relevant-Input problem and it is formally defined in Section 4. We will see that for positive γ this reduces the problem complexity from $\mathsf{NP}^{\mathsf{PP}}$ to $\mathsf{NP}^{\mathsf{BPP}}$. The associated optimisation problem is called Min-γ-Gapped-δ-Relevant-Input and made formal in the same section. Unfortunately, even in this simplified case, it remains NP-hard to approximate the size of the optimal set S within any reasonable approximation factor.

[2] Checking if a subset is one-relevant is in coNP instead of PP. Thus, we excluded $\delta = 1$ in Theorem 2.4.

Theorem 2.5. *Let $\delta \in (0,1)$ and $\gamma \in [0,\delta)$. Then, for any $\alpha \in (0,1)$ there is no polynomial time approximation algorithm for* Min-γ-Gapped-δ-Relevant-Input *with an approximation factor of $d^{1-\alpha}$ unless* $\mathsf{P} = \mathsf{NP}$.

The complete proofs of both main theorems as well formal definitions, detailed discussions, and analyses of the problem variants are given in Section 3 and Section 4. Already here, we can draw an important corollary which follows from Theorem 2.5 for the special case $\gamma = 0$.

Corollary 2.6. *Let $\delta \in (0,1)$. Then, for any $\alpha \in (0,1)$ there is no polynomial time approximation algorithm for* Min-δ-Relevant-Input *with an approximation factor of $d^{1-\alpha}$ unless* $\mathsf{P} = \mathsf{NP}$.

2.1 Related Works

Prime Implicant Explanations A concept closely related to δ-relevant sets are prime implicant explanations (Shih et al., 2018). An implicant explanation of $\Phi(\mathbf{x})$ is a subset S of the variables such that $\mathbf{x}_S$ is sufficient for $\Phi(\mathbf{x})$. In other words, any completion $\mathbf{y}$ satisfying $\mathbf{y}_S = \mathbf{x}_S$ yields $\Phi(\mathbf{y}) = \Phi(\mathbf{x})$. In our terminology, implicants are precisely the one-relevant sets. A prime implicant is an implicant that is minimal with respect to set inclusion and can therefore not be reduced further. The δ-Relevant-Input problem with $\delta = 1$ answers the question if there exists a prime implicant of size at most k. This is known to be hard for $\mathsf{NP}^{\mathsf{coNP}}$ (Eiter & Gottlob, 1995) in general. However, certain representations of Boolean functions such as Binary Decision Diagrams (BDD) (Akers, 1978) allow for an efficient search over the prime implicants (Coudert & Madre, 1992; Manquinho et al., 1998).

As already briefly mentioned in the introduction, the case $\delta = 1$ is often too strict, especially for high-dimensional problems as commonly found in modern machine learning. Let us illustrate this with the task of image classification as an example. In this case, often small regions of the input image can be manipulated in a way that changes a classifier prediction, e.g. through adversarial patches (Brown et al., 2017; Liu et al., 2018). Thus, prime implicants will have to cover large portions of the input image, independent of the size of the actual object in the image that led to the original classifier prediction.

Thus, we extend the complexity analysis for $\delta < 1$, which adds a model counting component. Although model counting can be done efficiently for various classes of representations of Boolean functions, e.g. BDDs (Bryant, 1986), deterministic Decomposable Negation Normal Forms (d-DNNF) (Darwiche, 2000) and Sentential Decision Diagrams (SDD) (Darwiche, 2011), this alone does not solve the inapproximability of our problem as we will prove in Section 4.2. Going further, we do not see a straightforward way to extend the prime implicant finding algorithm of Coudert and Madre (1992) for BDDs to our problem setting with $\delta < 1$. The basic observation underlying the algorithm is that a set of variables not containing x_j is an implicant for $\Phi(\mathbf{x})$ exactly if it is an implicant for both $\Phi(x_1, \ldots, x_j = 0, \ldots, x_d)$ and $\Phi(x_1, \ldots, x_j = 1, \ldots, x_d)$. This is not true for δ-relevance.

Sufficient Explanations Khosravi et al. (2019) introduced sufficient explanations for binary decision functions obtained from thresholding a continuous prediction model, e.g. a logistic regression classifier. As in our approach, the authors consider a probabilistic version of the prime implicant problem. In this case, the classification decision is required to remain unchanged in expectation instead of for all possible assignments to the non-fixed variables.

More precisely, let $f\colon \mathcal{X} \to [0,1]$ be a continuous prediction model on a domain $\mathcal{X}$ (e.g. a logistic regression model), $\theta\colon [0,1] \to \{0,1\}$ be a binarisation function (e.g. thresholding at 0.5), and $\mathcal{D}$ be a distribution on $\mathcal{X}$. A variable x_i of an input $\mathbf{x} \in \mathcal{X}$ is called a supporting variable, if

$$\begin{cases} \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}\big(f(\mathbf{y}) \,\big|\, \mathbf{y}_{\{i\}^c} = \mathbf{x}_{\{i\}^c}\big) \leq f(\mathbf{x}) & \text{if } \theta(f(\mathbf{x})) = 1, \\ \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}\big(f(\mathbf{y}) \,\big|\, \mathbf{y}_{\{i\}^c} = \mathbf{x}_{\{i\}^c}\big) > f(\mathbf{x}) & \text{if } \theta(f(\mathbf{x})) = 0. \end{cases}$$

In other words, randomising x_i conditioned on fixing all other variables does not increase the classification margin in expectation. A sufficient explanation is a cardinality minimal subset S of all supporting variables satisfying

$$\theta(\mathbb{E}_{\mathbf{y}\sim\mathcal{D}}(f(\mathbf{y}) \,|\, \mathbf{y}_S = \mathbf{x}_S)) = \theta(f(\mathbf{x})).$$

For an already binary function f and $\mathcal{D} = \mathcal{U}\big(\{0,1\}^d\big)$, this approach is essentially the same as finding small $\frac{1}{2}$-relevant sets. The only difference is that sufficient explanations only consider subsets of supporting variables, while we make no such distinction. This is however a minor difference and we conjecture that our hardness results carry over.

Anchors Anchors were introduced recently by Ribeiro et al. (2018) as local model-agnostic explanations. Given a generic function $f\colon \mathcal{X} \to \mathcal{Z}$ from a domain $\mathcal{X}$ to a codomain $\mathcal{Z}$ (for example a set of class labels) and a threshold $\delta \in [0,1]$, an anchor for an input $\mathbf{x} \in \mathcal{X}$ is some predicate $A\colon \mathcal{X} \to \{0,1\}$ satisfying

$$A(\mathbf{x}) = 1 \qquad \text{and} \qquad P_{\mathbf{y}\sim D_{\mathbf{x}}}(f(\mathbf{y}) = f(\mathbf{x}) \,|\, A(\mathbf{y})) \geq \delta,$$

where $D_{\mathbf{x}}$ is a local distribution in the neighbourhood of $\mathbf{x}$. The description of feasible predicates A is rather vague, however the predicates explicitly considered by Ribeiro et al. are of the form

$$A(\mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{y}_S = \mathbf{x}_S, \\ 0 & \text{otherwise,} \end{cases}$$

for some subset S of the variables in $\mathcal{X}$, just as in our formulation. Choosing the domain $\mathcal{X} = \{0,1\}^d$ and codomain $\mathcal{Y} = \{0,1\}$, the only difference to our δ-Relevant-Input problem is that we consider a global uniform distribution instead of local perturbations around $\mathbf{x}$. Ribeiro et al. suggested to search for an anchor with the largest possible coverage, defined as $\operatorname{cov}(A) = P_{\mathbf{y}\sim D_{\mathbf{x}}}(A(\mathbf{y}))$. For the uniform distribution this is exactly equivalent to searching for the smallest set S. We conjecture that our hardness results carry over to the problem of finding anchors for many possible perturbation distributions $D_{\mathbf{x}}$.

Shapley Values Another concept for measuring the relevance or the *contribution* of individual variables to a collective are the Shapley values (Shapley, 1953) in cooperative game theory. Here, the variables are seen as players of a coalitional game, and the Shapley values describe a method to distribute the value achieved by a coalition of players to the individual players. This distribution fulfils a set of game theoretic properties that make it "fair".

Let $\nu\colon 2^{[d]} \to \mathbb{R}$ be a function that assigns a value to each subset of variables (coalition of players). It is called the *characteristic function* of the game. Then, the Shapley value of the i-th variable (i-th player) is defined as

$$\varphi_{i,\nu} = \sum_{S \subseteq [d] \setminus \{i\}} \frac{|S|!(d - |S| - 1)!}{d!} (\nu(S \cup \{i\}) - \nu(S)),$$

which can be interpreted as the marginal contribution of the i-th variable to the value ν averaged over all possible coalitions. In general it is #P-hard to compute Shapley values (Deng & Papadimitriou, 1994). However, in some cases efficient approximation algorithms exist (Fatima et al., 2008).

In our scenario the value of a subset of variables S can be measured by the expected difference in Φ when fixing variables in S and randomising the remaining variables. Kononenko et al. (2010) proposed to use

$$\nu(S) = \frac{1}{2^{d-|S|}} \sum_{\substack{\mathbf{y} \in \{0,1\}^d \\ \mathbf{y}_S = \mathbf{x}_S}} \Phi(\mathbf{y}) - \mathbb{E}_{\mathbf{y}}(\Phi(\mathbf{y}))$$

for the analysis of classifier decisions, which uses the expectation of the completely randomised classifier score as a reference value to determine the coalition value. We observe that

$$\begin{aligned} P_{\mathbf{y}}(\Phi(\mathbf{y}) = \Phi(\mathbf{x}) \,|\, \mathbf{y}_S = \mathbf{x}_S) &= 1 - \frac{1}{2^{d-|S|}} \sum_{\substack{\mathbf{y} \in \{0,1\}^d \\ \mathbf{y}_S = \mathbf{x}_S}} |\Phi(\mathbf{y}) - \Phi(\mathbf{x})| \\ &= 1 - |\nu(S) + \mathbb{E}_{\mathbf{y}}(\Phi(\mathbf{y})) - \Phi(\mathbf{x})|, \end{aligned}$$

hence $S \subseteq [d]$ is δ-relevant for Φ and $\mathbf{x}$ exactly if $|\nu(S) + \mathbb{E}_{\mathbf{y}}(\Phi(\mathbf{y})) - \Phi(\mathbf{x})| \leq 1 - \delta$.

Despite this relation between δ-relevant sets and the characteristic function ν our problem formulation is considerably different from the Shapley value approach. The task considered in this paper is not to distribute the value of coalitions amongst the variables but to find (small) coalitions that are guaranteed to have a certain value.

3. Computational Complexity of δ-Relevant-Input

Recall the first main theorem, which shows that the δ-Relevant-Input problem is generally hard to solve for $\delta \in (0, 1)$.

Theorem 2.4. *For $\delta \in (0, 1)$ the δ-Relevant-Input problem is $\mathsf{NP}^{\mathsf{PP}}$-complete.*

The proof of Theorem 2.4 will be split into two parts. We will show that δ-Relevant-Input is $\mathsf{NP}^{\mathsf{PP}}$-hard in Section 3.1 and that it is contained in $\mathsf{NP}^{\mathsf{PP}}$ in Section 3.2.

3.1 δ-Relevant-Input is $\mathsf{NP}^{\mathsf{PP}}$-hard

We now give the first part of the proof of Theorem 2.4. This is done by constructing a polynomial-time reduction of a $\mathsf{NP}^{\mathsf{PP}}$-complete problem to δ-Relevant-Input. The canonical complete problem for $\mathsf{NP}^{\mathsf{PP}}$ is E-Maj-Sat (Littman et al., 1998).

Definition 3.1. The E-Maj-Sat problem is defined as follows.

Given: A Boolean function $\Phi\colon \{0,1\}^d \to \{0,1\}$ in conjunctive normal form (CNF) and $k \in \mathbb{N}$, $1 \leq k \leq d$.

Decide: Does there exist $\mathbf{x} \in \{0,1\}^k$ such that $P_{\mathbf{y}}\big(\Phi(\mathbf{y}) \,\big|\, \mathbf{y}_{[k]} = \mathbf{x}\big) > \frac{1}{2}$?

In other words, E-Maj-Sat asks whether there is an assignment to the first k variables of Φ such that the majority of assignments to the remaining $d-k$ variables satisfies Φ. There are three hurdles to take if we want to reduce this to δ-Relevant-Input.

1. Instead of freely assigning values to a subset of variables we are given an assignment to all variables and can only choose which to fix and which to randomise.
2. Instead of assigning values to a given set of k variables we can freely choose the set S of size at most k.
3. Instead of checking whether the majority of assignments satisfies Φ we check if the fraction of satisfying assignments is greater than or equal to some δ.

We address each of these hurdles and give a chain of polynomial-time reductions

$$\text{E-Maj-Sat} \preceq_p \text{IP1} \preceq_p \text{IP2} \preceq_p \delta\text{-Relevant-Input} \tag{1}$$

in three steps with intermediate auxiliary problems IP1 and IP2. The following observations will turn out to be useful.

Remark 3.2. Let Φ and Ψ be Boolean functions, not necessarily of different variables. Then,

$$\begin{aligned} P(\Psi) = 0 \quad &\Rightarrow \quad P(\Phi \oplus \Psi) = P(\Phi), \\ P(\Psi) = 1 \quad &\Rightarrow \quad P(\Phi \oplus \Psi) = 1 - P(\Phi), \end{aligned}$$

and if Φ and Ψ are independent, i.e. $P(\Phi \wedge \Psi) = P(\Phi)P(\Psi)$, also

$$P(\Psi) = \frac{1}{2} \quad \Rightarrow \quad P(\Phi \oplus \Psi) = \frac{1}{2}.$$

Lemma 3.3. *Let* $\mathrm{EQ}\colon \{0,1\}^k \times \{0,1\}^k \to \{0,1\}$ *and* $\Psi\colon \{0,1\}^k \times \{0,1\}^k \times \{0,1\} \to \{0,1\}$ *be defined as*

$$\mathrm{EQ}(\mathbf{u},\mathbf{v}) = \bigwedge_{i=1}^{k} \neg(u_i \oplus v_i) \qquad \text{and} \qquad \Psi(\mathbf{u},\mathbf{v},t) = \left(\bigvee_{i=1}^{k} (u_i \oplus v_i)\right) \wedge t.$$

Then, for any $\Phi\colon \{0,1\}^k \times \{0,1\}^{d-k} \to \{0,1\}$ *and* $A\colon \{0,1\}^k \times \{0,1\}^k \to \{0,1\}$ *with*

$$P(A(\mathbf{u},\mathbf{v}) \wedge \mathrm{EQ}(\mathbf{u},\mathbf{v})) > 0 \quad \text{and} \quad P(A(\mathbf{u},\mathbf{v}) \wedge \neg\,\mathrm{EQ}(\mathbf{u},\mathbf{v})) > 0,$$

we have

$$P(\Phi(\mathbf{u},\mathbf{r}) \oplus \Psi(\mathbf{u},\mathbf{v},t) \,|\, A(\mathbf{u},\mathbf{v})) > \frac{1}{2} \quad \Longleftrightarrow \quad P(\Phi(\mathbf{u},\mathbf{r}) \,|\, A(\mathbf{u},\mathbf{v}), \mathrm{EQ}(\mathbf{u},\mathbf{v})) > \frac{1}{2}.$$

The condition EQ determines whether $\mathbf{u} = \mathbf{v}$ or not. As soon as there exists an i with $u_i \neq v_i$, $\Psi(\mathbf{u}, \mathbf{v}, t)$ has the value of t, which is 1 with probability $\frac{1}{2}$. That means by modulo-adding Ψ to Φ we only have to consider the cases where $\mathbf{u} = \mathbf{v}$ to decide whether the majority of assignments to Φ evaluates to true. This is independent from any additional condition $A(\mathbf{u}, \mathbf{v})$.

Proof. We can rewrite $\Psi(\mathbf{u}, \mathbf{v}, t) = (\neg \mathrm{EQ}(\mathbf{u}, \mathbf{v})) \wedge t$ and therefore

$$P(\Psi \,|\, \mathrm{EQ}) = 0 \qquad \text{and} \qquad P(\Psi \,|\, \neg \mathrm{EQ}) = \frac{1}{2}.$$

Since $\Phi \,|\, A$ and $\Psi \,|\, A$ are conditionally independent given $\neg$ EQ (in this case Ψ depends on t only), we obtain from Remark 3.2 that

$$P(\Phi \oplus \Psi \,|\, A, \mathrm{EQ}) = P(\Phi \,|\, A, \mathrm{EQ}) \qquad \text{and} \qquad P(\Phi \oplus \Psi \,|\, A, \neg \mathrm{EQ}) = \frac{1}{2}.$$

Therefore,

$$\begin{aligned} P(\Phi \oplus \Psi \,|\, A) &= P(\Phi \oplus \Psi \,|\, A, \mathrm{EQ}) P(\mathrm{EQ}) + P(\Phi \oplus \Psi \,|\, A, \neg \mathrm{EQ}) P(\neg \mathrm{EQ}) \\ &= P(\Phi \,|\, A, \mathrm{EQ}) P(\mathrm{EQ}) + \frac{1}{2}(1 - P(\mathrm{EQ})) \\ &= \frac{1}{2} + \left(P(\Phi \,|\, A, \mathrm{EQ}) - \frac{1}{2} \right) P(\mathrm{EQ}). \end{aligned}$$

This directly implies $P(\Phi \oplus \Psi \,|\, A) > \frac{1}{2}$ if and only if $P(\Phi \,|\, A, \mathrm{EQ}) > \frac{1}{2}$. □

3.1.1 Fixing or Randomising Variables

Let us now come to the first step of the reductive chain (1). In this, we translate the possibility of freely assigning the first k variables into the choice of fixing or randomising variables from a given assignment. This choice is however still restricted to the first k variables.

Definition 3.4. We define the Intermediate Problem 1 (IP1) as follows.

Given: A Boolean circuit $\Phi \colon \{0,1\}^d \to \{0,1\}$, $\mathbf{x} \in \{0,1\}^d$ and $k \in \mathbb{N}$, $1 \leq k \leq d$.

Decide: Does there exist $S \subseteq [k]$ such that $P_{\mathbf{y}}(\Phi(\mathbf{y}) \,|\, \mathbf{y}_S = \mathbf{x}_S) > \frac{1}{2}$?

In other words, IP1 asks the questions whether there exists a subset of the first k variables of Φ such that fixing these to the values given by $\mathbf{x}$ implies that the majority of assignments to the remaining variables satisfies Φ.

Lemma 3.5. E-Maj-Sat $\preceq_p$ IP1*, in particular* IP1 *is* $\mathsf{NP}^{\mathsf{PP}}$*-hard.*

Proof. Let $\{\Phi, k\}$ be an E-Maj-Sat instance. We will construct $\{\Phi', \mathbf{x}', k'\}$ that is a *Yes*-instance for IP1 if and only if $\{\Phi, k\}$ is a *Yes*-instance for E-Maj-Sat. For convenience we split the d variables of Φ into the first k variables and the remaining $d - k$ variables and denote this $\Phi(\mathbf{x}) = \Phi(\mathbf{u}, \mathbf{r})$. The main idea is to duplicate the first k variables and choose $\mathbf{x}'$ in such a way that fixing the original variables or their duplicates corresponds to assigning zeros or ones in the E-Maj-Sat instance respectively. More precisely, we define

- $\Phi' \colon \{0,1\}^k \times \{0,1\}^k \times \{0,1\}^{d-k} \times \{0,1\} \to \{0,1\}$ as

$$\Phi'(\mathbf{u},\mathbf{v},\mathbf{r},t) = \Phi(\mathbf{u},\mathbf{r}) \oplus \left(\bigvee_{i=1}^{k} (u_i \oplus v_i) \wedge t \right),$$

- $\mathbf{x}' = (\mathbf{0}_k, \mathbf{1}_k, \mathbf{0}_{d-k}, 0) \in \{0,1\}^k \times \{0,1\}^k \times \{0,1\}^{d-k} \times \{0,1\}$,
- $k' = 2k$.

This is a polynomial time construction. With Ψ defined as in Lemma 3.3 we can rewrite $\Phi'(\mathbf{u},\mathbf{v},\mathbf{r},t) = \Phi(\mathbf{u},\mathbf{r}) \oplus \Psi(\mathbf{u},\mathbf{v},t)$.

Necessity: Assume that $\{\Phi, k\}$ is a *Yes*-instance for E-Maj-Sat. Then, there exists an assignment $\mathbf{u}^* \in \{0,1\}^k$ to the first k variables of Φ such that $P_{\mathbf{r}}(\Phi(\mathbf{u}^*,\mathbf{r})) > \frac{1}{2}$. Now, choose $S' = \{\, i \in \{1,\dots,k\} \,:\, u_i^* = 0 \,\} \cup \big\{\, i \in \{k+1,\dots,2k\} \,:\, u_{i-k}^* = 1 \,\big\} \subseteq [k'] = [2k]$. Let $A \colon \{0,1\}^k \times \{0,1\}^k \to \{0,1\}$ be given by

$$A(\mathbf{u},\mathbf{v}) = \left(\bigwedge_{i \in S' \cap \{1,\dots,k\}} \neg u_i \right) \wedge \left(\bigwedge_{i \in S' \cap \{k+1,\dots,2k\}} v_{i-k} \right)$$

and EQ as in Lemma 3.3. Note that A depends on S' and thus implicitly on $\mathbf{u}^*$. In fact, $A(\mathbf{u},\mathbf{v}) = 1$ holds if and only if both $u_i^* = 0$ implies $u_i = 0$ and $u_i^* = 1$ implies $v_i = 1$ for all $i \in [k]$. In particular, we have $A(\mathbf{u},\mathbf{u}) = 1$ if an only if $\mathbf{u} = \mathbf{u}^*$. Also $\mathrm{EQ}(\mathbf{u},\mathbf{v}) = 1$ if and only if $\mathbf{u} = \mathbf{v}$. Thus, we have

$$\begin{aligned} P_{\mathbf{r}}(\Phi(\mathbf{u}^*,\mathbf{r})) &= P(\Phi(\mathbf{u},\mathbf{r}) \,|\, \mathbf{u} = \mathbf{u}^*) \\ &= P(\Phi(\mathbf{u},\mathbf{r}) \,|\, A(\mathbf{u},\mathbf{u})) \\ &= P(\Phi(\mathbf{u},\mathbf{r}) \,|\, A(\mathbf{u},\mathbf{v}), \mathrm{EQ}(\mathbf{u},\mathbf{v})), \end{aligned}$$

and by the choice of $\mathbf{x}'$, A, and S' we get

$$\begin{aligned} P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) &= P\big(\Phi'(\mathbf{u},\mathbf{v},\mathbf{r},t) \,\big|\, A(\mathbf{u},\mathbf{v})\big) \\ &= P(\Phi(\mathbf{u},\mathbf{r}) \oplus \Psi(\mathbf{u},\mathbf{v},t) \,|\, A(\mathbf{u},\mathbf{v})). \end{aligned}$$

We use Lemma 3.3 together with $P_{\mathbf{r}}(\Phi(\mathbf{u}^*,\mathbf{r})) > \frac{1}{2}$ to conclude $P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) > \frac{1}{2}$ which shows that $\{\Phi', \mathbf{x}', k'\}$ is a *Yes*-instance for IP1.

Sufficiency: Now, conversely, assume that $\{\Phi', \mathbf{x}', k'\}$ is a *Yes*-instance for IP1. Then, there exists $S' \subseteq [k'] = [2k]$ such that $P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) > \frac{1}{2}$. Following the same grouping of variables as before, we write $\mathbf{x}' = (\mathbf{u}', \mathbf{v}', \mathbf{r}', t')$. We can translate this into a satisfying assignment $\mathbf{u}^*$ for E-Maj-Sat where $u_i^* = 0$ when $i \in S$ and $u_i^* = 1$ when $i + k \in S'$. For that, we need two statements to be true. First, not both i and $i+k$ can be in S'. And second, if neither i nor $i+k$ are in S', then there is always the possibility of adding one of them to S' and still satisfy $P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) > \frac{1}{2}$.

To prove the first statement, assume towards a contradiction that there exists an $i \in [k]$ with $i \in S'$ and $i + k \in S'$. Since $\mathbf{u}' = \mathbf{0}_k$ and $\mathbf{v}' = \mathbf{1}_k$, we have that $(\mathbf{u}, \mathbf{v})_{S'} = (\mathbf{u}', \mathbf{v}')_{S'}$ implies $u_i = 0 \neq 1 = v_i$ and hence

$$P_{\mathbf{u},\mathbf{v},t}\big(\Psi(\mathbf{u},\mathbf{v},t) \,\big|\, (\mathbf{u},\mathbf{v})_{S'} = (\mathbf{u}',\mathbf{v}')_{S'}\big) = P_{\mathbf{u},\mathbf{v},t}\big(t \,\big|\, (\mathbf{u},\mathbf{v})_{S'} = (\mathbf{u}',\mathbf{v}')_{S'}\big) = \frac{1}{2}.$$

Thus, Remark 3.2 would imply $P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) = \frac{1}{2}$, which contradicts the assumption that $\{\Phi', \mathbf{x}', k'\}$ is a *Yes*-instance for IP1.

For the second statement, assume there exists an $i \in [k]$ with neither $i \in S'$ nor $i+k \in S'$. Then $A(\mathbf{u}, \mathbf{v})$ is a condition on $\mathbf{u}$ and $\mathbf{v}$ that does not include the variables u_i and v_i. Therefore,

$$\begin{aligned}
P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v})\big) = {} & \frac{1}{4}P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v}), u_i = 0, v_i = 0\big)\\
& + \frac{1}{4}P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v}), u_i = 0, v_i = 1\big)\\
& + \frac{1}{4}P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v}), u_i = 1, v_i = 0\big)\\
& + \frac{1}{4}P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v}), u_i = 1, v_i = 1\big).
\end{aligned}$$

For the second and third summand we have $u_i \neq v_i$, thus using Remark 3.2 again, we get

$$P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v}), u_i = 0, v_i = 1\big) = \frac{1}{2} = P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v}), u_i = 1, v_i = 0\big),$$

and obtain

$$\begin{aligned}
P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v})\big) = {} & \frac{1}{4}P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v}), u_i = 0, v_i = 0\big)\\
& + \frac{1}{4}P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v}), u_i = 0, v_i = 1\big)\\
& + \frac{1}{4}P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v}), u_i = 0, v_i = 1\big)\\
& + \frac{1}{4}P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v}), u_i = 1, v_i = 1\big)\\
= {} & \frac{1}{2}P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v}), u_i = 0\big)\\
& + \frac{1}{2}P_{\mathbf{u},\mathbf{v},t}\big(\Phi'(\mathbf{u},\mathbf{v},t) \,\big|\, A(\mathbf{u},\mathbf{v}), v_i = 1\big).
\end{aligned}$$

Altogether, if $P_{\mathbf{u},\mathbf{v},t}(\Phi'(\mathbf{u},\mathbf{v},t) \,|\, A(\mathbf{u},\mathbf{v})) > \frac{1}{2}$, then at least one of the additional conditions $u_i = 0$ or $v_i = 1$ must also yield a probability greater than $\frac{1}{2}$. This implies that if $P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) > \frac{1}{2}$ then either i or $i+k$ can be added to S' while keeping the probability greater than $\frac{1}{2}$.

So, without loss of generality, we can assume that for each $i \in [k]$ exactly one of the cases $i \in S'$ or $i + k \in S'$ occurs. Then, we can define $\mathbf{u}^* \in \{0,1\}^k$ as $u_i^* = 0$ if $i \in S'$ and $u_i^* = 1$ otherwise. We observe that S' and $\mathbf{u}^*$ are exactly as in the previous step and the rest of the proof follows analogously. Again we use Lemma 3.3 and conclude from $P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) > \frac{1}{2}$ that $P_{\mathbf{r}}(\Phi(\mathbf{u}^*, \mathbf{r})) > \frac{1}{2}$. This shows that $\{\Phi, k\}$ is a *Yes*-instance for E-Maj-Sat. □

3.1.2 Allowing for all variables to be chosen

We continue with the second step of the reductive chain (1). Instead of choosing from the first k variables we are free to chose from all d variables but at most k many.

Definition 3.6. We define the Intermediate Problem 2 (IP2) as follows.

Given: A Boolean circuit $\Phi\colon \{0,1\}^d \to \{0,1\}$, $\mathbf{x} \in \{0,1\}^d$ and $k \in \mathbb{N}$, $1 \leq k \leq d$.

Decide: Does there exist $S \subseteq [d]$ with $|S| \leq k$ such that $P_{\mathbf{y}}(\Phi(\mathbf{y}) \,|\, \mathbf{y}_S = \mathbf{x}_S) > \frac{1}{2}$?

In other words, IP2 asks the question whether there exists a subset of at most k variables of Φ such that fixing these to the values given by $\mathbf{x}$ implies that the majority of the possible assignments to the remaining variables satisfies Φ.

Lemma 3.7. *We have* IP1 $\preceq_p$ IP2*. In particular,* IP2 *is* $\mathsf{NP}^{\mathsf{PP}}$*-hard.*

Proof. Let $\{\Phi, \mathbf{x}, k\}$ be an IP1 instance. We will construct $\{\Phi', \mathbf{x}', k'\}$ that is a *Yes*-instance for IP2 if and only if $\{\Phi, \mathbf{x}, k\}$ is a *Yes*-instance for IP1. For convenience, we split the d variables of Φ into the first k variables and the remaining $d-k$ variables and denote this $\Phi(\mathbf{x}) = \Phi(\mathbf{u}, \mathbf{r})$. The main idea is to extend Φ with clauses that force the set S to be chosen from the first k variables. More precisely, we define

- $\Phi'\colon \{0,1\}^k \times \{0,1\}^k \times \{0,1\}^{d-k} \times \{0,1\}^{d-k} \times \{0,1\}^{d-k} \to \{0,1\}$ with
$$\Phi'(\mathbf{u}, \mathbf{v}, \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) = \Phi(\mathbf{u}, \mathbf{r}_1 \oplus \mathbf{r}_2 \oplus \mathbf{r}_3) \wedge \left(\bigwedge_{i=1}^{k} ((u_i \oplus \neg x_i) \vee v_i) \right),$$
where $\mathbf{r}_1 \oplus \mathbf{r}_2 \oplus \mathbf{r}_3$ is understood component-wise,
- $\mathbf{x}' = \left(\mathbf{x}_{[k]}, \mathbf{1}_k, \mathbf{x}_{[k]^c}, \mathbf{x}_{[k]^c}, \mathbf{x}_{[k]^c}\right) \in \{0,1\}^k \times \{0,1\}^k \times \{0,1\}^{d-k} \times \{0,1\}^{d-k} \times \{0,1\}^{d-k}$,
- $k' = k$.

This is a polynomial time construction.

Necessity: Assume that $\{\Phi, \mathbf{x}, k\}$ is a *Yes*-instance for IP1. Then, there exists $S \subseteq [k]$ such that $P_{\mathbf{y}}(\Phi(\mathbf{y}) \,|\, \mathbf{y}_S = \mathbf{x}_S) > \frac{1}{2}$. Now, choose

$$S' = S \cup \{\, i \in \{k+1, \dots, 2k\} \,:\, i - k \notin S \,\}.$$

Then, $|S'| = |S| + (k - |S|) = k = k'$ and for each $i \in [k]$ exactly one of the cases $i \in S'$ or $i + k \in S'$ occurs. The former corresponds to fixing $u_i = x_i' = x_i$ and the latter to fixing $v_i = 1$. Therefore,

$$P_{(\mathbf{u},\mathbf{v})}\left(\bigwedge_{i=1}^{k} ((u_i \oplus \neg x_i) \vee v_i) \,\middle|\, (\mathbf{u}, \mathbf{v})_{S'} = \mathbf{x}'_{S'} \right) = 1,$$

which means, conditioned on $(\mathbf{u},\mathbf{v})_{S'} = \mathbf{x}'_{S'}$, the probability of satisfying Φ' only depends on $\Phi(\mathbf{u}, \mathbf{r}_1 \oplus \mathbf{r}_2 \oplus \mathbf{r}_3)$. Now, since the random vector $\mathbf{r}_1 \oplus \mathbf{r}_2 \oplus \mathbf{r}_3$ is independent of this condition, it has the exact same distribution as the random vector $\mathbf{r}$, and we obtain

$$\begin{aligned} P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) &= P\big(\Phi'(\mathbf{u},\mathbf{v},\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_3) \,\big|\, (\mathbf{u},\mathbf{v})_{S'} = \mathbf{x}'_{S'}\big) \\ &= P\big(\Phi'(\mathbf{u},\mathbf{v},\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_3) \,\big|\, \mathbf{u}_S = \mathbf{x}_S, \mathbf{v}_{[k]\setminus S} = \mathbf{1}_{k-|S|}\big) \\ &= P(\Phi(\mathbf{u},\mathbf{r}) \,|\, \mathbf{u}_S = \mathbf{x}_S) \\ &= P_{\mathbf{y}}(\Phi(\mathbf{y}) \,|\, \mathbf{y}_S = \mathbf{x}_S) > \frac{1}{2}. \end{aligned} \tag{2}$$

Hence, $\{\Phi', \mathbf{x}', k'\}$ is a *Yes*-instance for IP2.

Sufficiency: Now, conversely, assume that $\{\Phi', \mathbf{x}', k'\}$ is a *Yes*-instance for IP2. Then, there exists a set $S' \subseteq [2k + 3(d-k)]$ with $|S'| \leq k' = k$ and

$$P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) > \frac{1}{2}.$$

First, we show that S' can contain at most two indices that are not in $[2k]$. For any $i \in [k]$, consider the term $(u_i \oplus \neg x_i) \vee v_i$, which is true if $u_i = x_i$ or $v_i = 1$. This could be assured by $i \in S'$ or $i + k \in S'$ respectively. Otherwise, $P_{u_i,v_i}((u_i \oplus \neg x_i) \vee v_i) = \frac{3}{4}$. Let $N = |\{i \in [k] \,|\, i \notin S' \wedge i + k \notin S'\}|$, then

$$P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) \leq P\left(\bigwedge_{i=1}^{k} ((u_i \oplus \neg x_i) \vee v_i) \,\middle|\, (\mathbf{u},\mathbf{v})_{S'} = \mathbf{x}'_{S'}\right) = \left(\frac{3}{4}\right)^N,$$

and since $\left(\frac{3}{4}\right)^3 < \frac{1}{2}$ but $P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) > \frac{1}{2}$, we know that $N \leq 2$.

Therefore, at most two variables out of $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ can be fixed and thus $\mathbf{r}_1 \oplus \mathbf{r}_2 \oplus \mathbf{r}_3$ conditioned on $\mathbf{y}'_{S'} = \mathbf{x}'_{S'}$ has the same distribution as $\mathbf{r}_1 \oplus \mathbf{r}_2 \oplus \mathbf{r}_3$ without the condition. So, without loss of generality, we can even assume $S' \cap [2k]^c = \emptyset$.

Similarly, if $i \in S'$, we have $P\big((u_i \oplus \neg x_i) \vee v_i \,\big|\, (\mathbf{u},\mathbf{v})_{S'} = \mathbf{x}'_{S'}\big) = 1$ and additionally having $i + k \in S'$ could not increase the probability of satisfying Φ'. Hence, we can assume $i + k \notin S'$ in this case. Contrary, if $i \notin S'$, we have

$$P\big((u_i \oplus \neg x_i) \vee v_i \,\big|\, (\mathbf{u},\mathbf{v})_{S'} = \mathbf{x}'_{S'}\big) = \frac{1}{2} + \frac{1}{2} P\big(v_i \,\big|\, (\mathbf{u},\mathbf{v})_{S'} = \mathbf{x}'_{S'}\big),$$

which is one if $i + k \in S'$ and $\frac{3}{4}$ otherwise. So including $i + k$ in S' does not decrease the probability.

Altogether, without loss of generality, we can assume $S' \subseteq [2k]$, $|S'| = k$ and for each $i \in [k]$ exactly one of the cases $i \in S'$ or $i + k \in S'$ occurs. We now choose $S = S' \cap [k]$. Then, the rest of the proof proceeds exactly as in (2), and we conclude

$$P_{\mathbf{y}}(\Phi(\mathbf{y}) \,|\, \mathbf{y}_S = \mathbf{x}_S) = P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}'_{S'}) \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) > \frac{1}{2},$$

implying that $\{\Phi, \mathbf{x}, k\}$ is a *Yes*-instance for IP1. □

3.1.3 Changing the Probability Threshold

Now, we want to change the probability threshold from $\frac{1}{2}$ to an arbitrary number $\delta \in (0,1)$ and show that the hardness does not depend on δ. Our reduction will depend on whether $\delta > \frac{1}{2}$ or $\delta < \frac{1}{2}$ since we either have to raise or lower the probability threshold. To raise the probability threshold, we make use of the following lemma.

Lemma 3.8 (Raising the probability, $>$ to $\geq$)**.** *Given $0 \leq \delta_1 < \delta_2 < 1$, for any $d \in \mathbb{N}$ there exists a monotone function $\Pi\colon \{0,1\}^n \to \{0,1\}$ such that for all $\Phi\colon \{0,1\}^d \to \{0,1\}$ we have*

$$P_{\mathbf{y}}(\Phi(\mathbf{y})) > \delta_1 \quad \Longleftrightarrow \quad P_{(\mathbf{y},\mathbf{r})}(\Phi(\mathbf{y}) \vee \Pi(\mathbf{r})) \geq \delta_2$$

with $n \in \mathcal{O}(d^2)$. The function Π can be constructed in $\mathcal{O}(n)$ time.

The constructive proof of Lemma 3.8 can be found in Appendix A. An analogous lemma is used to lower the probability threshold.

Lemma 3.9 (Lowering the probability, $>$ to $\geq$)**.** *Given $0 < \delta_1 \leq \delta_2 \leq 1$, for any $d \in \mathbb{N}$ there exists a monotone function $\Pi\colon \{0,1\}^n \to \{0,1\}$ such that for all $\Phi\colon \{0,1\}^d \to \{0,1\}$ we have*

$$P_{\mathbf{y}}(\Phi(\mathbf{y})) > \delta_2 \quad \Longleftrightarrow \quad P_{(\mathbf{y},\mathbf{r})}(\Phi(\mathbf{y}) \wedge \Pi(\mathbf{r})) \geq \delta_1$$

with $n \in \mathcal{O}\big(d^2\big)$. The function Π can be constructed in $\mathcal{O}(n)$ time.

Lastly, we introduce an auxiliary operation that allows us to make $\Phi(\mathbf{x})$ true for the initial assignment $\mathbf{x}$, while not changing the overall probability for random assignments.

Lemma 3.10 (Neutral Operation)**.** *Given $0 < \delta < 1$, for any $d \in \mathbb{N}$ there exists a monotone function $\Gamma_{d,\delta} : \{0,1\}^r \to \{0,1\}$ and positive integer $n_{d,\delta}$ with $r + n_{d,\delta} \in \mathcal{O}\big(d^2\big)$ such that for all $\Phi\colon \{0,1\}^d \to \{0,1\}$ we have*

$$P_{\mathbf{y}}(\Phi(\mathbf{y})) \geq \delta \quad \Longleftrightarrow \quad P_{(\mathbf{y},\mathbf{r},\mathbf{t})}\left((\Phi(\mathbf{y}) \wedge \Gamma_{d,\delta}(\mathbf{r})) \vee \left(\bigwedge_{i=1}^{n} t_i\right)\right) \geq \delta$$

for all $n \geq n_{d,\delta}$. The function $\Gamma_{d,\delta}$ can be consturcted in $\mathcal{O}(r)$ time.

The constructive proof of Lemma 3.10 can be found in Appendix D. Now, we are able to prove the following lemma.

Lemma 3.11. *For $\delta \in (0,1)$ we have* IP2 $\preceq_p$ δ-Relevant-Input. *In particular, the δ-*Relevant-Input *problem is $\mathsf{NP}^{\mathsf{PP}}$-hard.*

Proof. We start with the reduction for the case of $\delta \in (\frac{1}{2}, 1)$. Let $\{\Phi, \mathbf{x}, k\}$ be an IP2 instance. We will construct $\{\Phi', \mathbf{x}', k'\}$ that is a *Yes*-instance for δ-Relevant-Input if and only if $\{\Phi, \mathbf{x}, k\}$ is a *Yes*-instance for IP2. Let $\Pi\colon \{0,1\}^\ell \to \{0,1\}$ be as in Lemma 3.8 for $\delta_1 = \frac{1}{2}$ and $\delta_2 = \delta$. Let $\Gamma = \Gamma_{d+\ell,\delta}$ and $n_{d+\ell,\delta}$ be defined according to Lemma 3.10 and set $n = n_{d+\ell,\delta} + k$. We define

- $\Phi'\colon \{0,1\}^d \times \{0,1\}^\ell \times \{0,1\}^m \times \{0,1\}^n \to \{0,1\}$,

  $(\mathbf{y}, \mathbf{r}, \mathbf{s}, \mathbf{t}) \mapsto ((\Phi(\mathbf{y}) \vee \Pi(\mathbf{r})) \wedge \Gamma(\mathbf{s})) \vee (\bigwedge_{i=1}^{n} t_i)$

- $\mathbf{x}' = (\mathbf{x}, \mathbf{0}_\ell, \mathbf{0}_m, \mathbf{1}_n) \in \{0,1\}^d \times \{0,1\}^\ell \times \{0,1\}^m \times \{0,1\}^n$,
- $k' = k$.

This is a polynomial time construction. By the choice of Φ' and $\mathbf{x}'$, we guarantee $\Phi'(\mathbf{x}') = 1$ regardless of the value of $\Phi(\mathbf{x})$ since $\bigwedge_{i=0}^{n} 1 = 1$.

Necessity: Assume that $\{\Phi, \mathbf{x}, k\}$ is a *Yes*-instance for IP2 with satisfying set S. Then set $S' = S$ and from the definition of Π and n we get

$$\begin{aligned} & P_{\mathbf{y}}(\Phi(\mathbf{y}) \,|\, \mathbf{y}_S = \mathbf{x}_S) > \frac{1}{2} \\ \iff \quad & P_{(\mathbf{u},\mathbf{r})}(\Phi(\mathbf{u}) \vee \Pi(\mathbf{r}) \,|\, \mathbf{u}_S = \mathbf{x}_S) \geq \delta \\ \iff \quad & P_{(\mathbf{u},\mathbf{r},\mathbf{s},\mathbf{t})}\left((\Phi(\mathbf{u}) \vee \Pi(\mathbf{r})) \wedge \Gamma(\mathbf{s}) \vee \left(\bigwedge_{i=1}^{n} t_i \right) \middle| \, \mathbf{u}_S = \mathbf{x}_S \right) \geq \delta \\ \iff \quad & P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') = \Phi'(\mathbf{x}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) \geq \delta. \end{aligned}$$

Hence, $\{\Phi', \mathbf{x}', k'\}$ is a *Yes*-instance for δ-Relevant-Input.

Sufficiency: Now assume that $\{\Phi', \mathbf{x}', k'\}$ is a *Yes*-instance for δ-Relevant-Input. Then there exists a subset S' with $|S'| \leq k' = k$ and $P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') = \Phi'(\mathbf{x}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) \geq \delta$. Since Π and Γ are monotone and their initial input assignments are $\mathbf{0}_\ell$ and $\mathbf{0}_m$, including any of their variables in S' does not increase the probability that Φ' evaluates to $\Phi'(\mathbf{x}') = 1$. Thus, without loss of generality, we can assume that S' does no include variables from Γ. At most $k' = k$ of the n variables in the conjunction from Lemma 3.10 can be included in S', which by the choice of n does not affect whether the overall probability threshold of δ is reached or not. Thus,

$$P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') = \Phi'(\mathbf{x}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) \geq \delta \quad \Longrightarrow \quad P_{\mathbf{y}'}\Big(\Phi'(\mathbf{y}') = \Phi'(\mathbf{x}') \,\Big|\, \mathbf{y}'_{S' \cap [d]} = \mathbf{x}'_{S' \cap [d]}\Big) \geq \delta.$$

We set $S = S' \cap [d]$. Clearly $|S| \leq |S'| = k' = k$. Then analogous to before,

$$P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') = \Phi'(\mathbf{x}') \,\big|\, \mathbf{y}'_{S} = \mathbf{x}'_{S}\big) \geq \delta \quad \iff \quad P_{\mathbf{y}}(\Phi(\mathbf{y}) \,|\, \mathbf{y}_S = \mathbf{x}_S) > \frac{1}{2},$$

implying that $\{\Phi, \mathbf{x}, k\}$ is a *Yes*-instance for IP2.

The reduction for $\delta \in (0, \frac{1}{2}]$ can be done analogously by using Lemma 3.9 instead of Lemma 3.8 and we omit the details for brevity. □

3.2 δ-Relevant-Input is contained in $\mathsf{NP}^{\mathsf{PP}}$

We now come to the second part of the proof of Theorem 2.4. We will show that δ-Relevant-Input is indeed contained in $\mathsf{NP}^{\mathsf{PP}}$, meaning that it can be solved in polynomial time by a non-deterministic Turing machine with access to a PP-oracle. The following lemmas, very similar to Lemmas 3.8 and 3.9, will be useful.

Lemma 3.12 (Raising the probability, $\geq$ to $>$). *Given $0 \leq \delta_1 \leq \delta_2 < 1$, for any $d \in \mathbb{N}$ there exists a monotone function $\Pi\colon \{0,1\}^n \to \{0,1\}$ such that for all $\Phi\colon \{0,1\}^d \to \{0,1\}$ we have*

$$P_{\mathbf{y}}(\Phi(\mathbf{y})) \geq \delta_1 \quad \iff \quad P_{(\mathbf{y},\mathbf{r})}(\Phi(\mathbf{y}) \vee \Pi(\mathbf{r})) > \delta_2$$

with $n \in \mathcal{O}\big(d^2\big)$. The function Π can be constructed in $\mathcal{O}(n)$ time.

The constructive proof of Lemma 3.12 can be found in Appendix A.

Lemma 3.13 (Lowering the probability, $\geq$ to $>$)**.** *Given $0 < \delta_1 < \delta_2 \leq 1$, for any $d \in \mathbb{N}$ there exists a monotone function $\Pi\colon \{0,1\}^n \to \{0,1\}$ such that for all $\Phi\colon \{0,1\}^d \to \{0,1\}$ we have*

$$P_{\mathbf{y}}(\Phi(\mathbf{y})) \geq \delta_2 \quad \iff \quad P_{(\mathbf{y},\mathbf{r})}(\Phi(\mathbf{y}) \wedge \Pi(\mathbf{r})) > \delta_1$$

with $n \in \mathcal{O}(d^2)$. The function Π can be constructed in $\mathcal{O}(n)$ time.

The constructive proof of Lemma 3.13 can be found in Appendix B.

Lemma 3.14. *For $\delta \in (0,1)$ the δ-Relevant-Input problem is contained in $\mathsf{NP}^{\mathsf{PP}}$.*

We will prove this for $\delta \in \left(\frac{1}{2}, 1\right)$ by lowering the probability threshold from δ to $\frac{1}{2}$. The case $\delta \in \left(0, \frac{1}{2}\right]$ can be treated analogously by raising the threshold.

Proof. Let $\{\Phi, \mathbf{x}, k\}$ be an instance of δ-Relevant-Input. It suffices to show that the decision problem whether a given set $S \subseteq [d]$ is δ-relevant for Φ and $\mathbf{x}$ is in PP. Without loss of generality we can assume $\Phi(\mathbf{x}) = 1$. Otherwise, we could consider $\neg\Phi$ instead. Now, choose $\Pi\colon \{0,1\}^n \to \{0,1\}$ as in Lemma 3.13 for $\delta_1 = \frac{1}{2}$ and $\delta_2 = \delta$. Then,

$$P_{\mathbf{y}}(\Phi(\mathbf{y}) \,|\, \mathbf{y}_S = \mathbf{x}_S) \geq \delta \quad \iff \quad P_{(\mathbf{y},\mathbf{r})}(\Phi(\mathbf{y}) \wedge \Pi(\mathbf{r}) \,|\, \mathbf{y}_S = \mathbf{x}_S) > \frac{1}{2}.$$

A probabilistic Turing machine can now draw a random assignment $(\mathbf{y}, \mathbf{r})$ conditioned on $\mathbf{y}_S = \mathbf{x}_S$ and evaluate $\Phi(\mathbf{y}) \wedge \Pi(\mathbf{r})$. Thus, the machine will answer *Yes* with probability strictly greater than $\frac{1}{2}$ if and only if S is δ-relevant. This means the subproblem of checking a set for δ-relevance is contained in PP.

A non-deterministic Turing-machine with a PP-oracle can thus guess a set $S \subseteq [d]$ with $|S| \leq k$ and, using the oracle, check whether it is δ-relevant. □

4. Variations of the Problem Formulation

We want to consider two variations of the δ-Relevant-Input problem. The first variation relaxes the requirement to check if a candidate set S is exactly δ-relevant or not by introducing a probability gap γ. In short, we then ask if a δ-relevant set of size k exists or if all sets of size k are not even $(\delta - \gamma)$-relevant.

The second variation concerns the optimisation version of the problem. Here, we introduce a set size gap and relax the requirement to find the smallest δ-relevant set. Instead, for $k < m$ we ask if a δ-relevant set of size k exists or if all relevant sets must be of size at least m.

We show that these problems remain hard to solve (even in combination, that is with both a gap in probability and set size). This can be used to show that no polynomial time approximation algorithm for Min-δ-Relevant-Input with approximation factor better than the trivial factor d can exists unless $\mathsf{P} = \mathsf{NP}$. Due to the connection between Boolean circuits and neural networks, as described in Section 1, this inapproximability result shows theoretical limitations of interpretation methods for neural network decision.

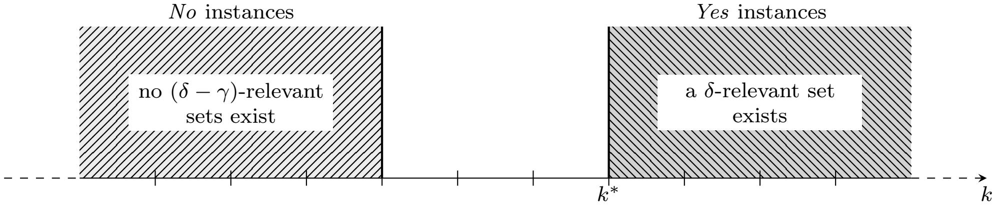
No instances
no $(\delta-\gamma)$-relevant sets exist
Yes instances
a δ-relevant set exists
k^*
k

Figure 3: Visualization of the γ-Gapped-δ-Relevant-Input problem for some fixed Φ and $\mathbf{x}$ and for various k. In the unmarked region in the centre no δ-relevant set exists but $\widetilde{\delta}$-relevant sets could exist for any $\widetilde{\delta} < \delta$, in particular also for $\widetilde{\delta} = \delta - \gamma$. In this region we do not expect an answer for the gapped problem. The solution k^* of the ungapped optimisation problem Min-δ-Relevant-Input is the left boundary of the *Yes*-instance region.

4.1 The Probability Gap

As explained in the problem formulation, see Section 2, probabilities and expectation values may be hard to calculate in theory, yet are often easy to approximate in practice via sampling. The edge cases where the true probability can be arbitrarily close to the threshold δ cause the hardness of problems in PP.

It seems impractical to defend the hardness of the δ-Relevant-Input problem with the exact evaluation of probabilities. Therefore, we introduce a variant of the problem including a probability gap. This can be seen as a promise problem with the promise that all sets S are either δ-relevant or not even $(\delta-\gamma)$-relevant. Alternatively, this can be seen as the δ-Relevant-Input problem where we want to answer *Yes* if a δ-relevant set of size k exists but only want to answer *No* if all sets of size k are not even $(\delta-\gamma)$-relevant. For cases in between we do not expect an answer at all or do not care about the exact answer. This is illustrated in Figure 3.

Definition 4.1. For $\delta \in (0,1]$ and $\gamma \in [0,\delta)$ we define the γ-Gapped-δ-Relevant-Input problem as follows.

Given: A Boolean circuit $\Phi\colon \{0,1\}^d \to \{0,1\}$, $\mathbf{x} \in \{0,1\}^d$, and $k \in \mathbb{N}$, $1 \leq k \leq d$.

Decide:

Yes: There exists $S \subseteq [d]$ with $|S| \leq k$ and S is δ-relevant for Φ and $\mathbf{x}$.

No: All $S \subseteq [d]$ with $|S| \leq k$ are not $(\delta-\gamma)$-relevant for Φ and $\mathbf{x}$.

For $\gamma = 0$ we exactly retrieve the original δ-Relevant-Input problem, but for $\gamma > 0$ this is an easier question.

Lemma 4.2. *For $\delta \in (0,1)$ and $\gamma \in (0,\delta)$ the γ-Gapped-δ-Relevant-Input problem is contained in $\mathsf{NP}^{\mathsf{BPP}}$.*

Proof. Let $\{\Phi, \mathbf{x}, k\}$ be an instance of γ-Gapped-δ-Relevant-Input. It suffices to show that the decision problem whether a given set $S \subseteq [d]$ is either δ-relevant (*Yes*) or not

$(\delta-\gamma)$-relevant (*No*) for Φ and $\mathbf{x}$ is in BPP. To see this, we describe an explicit algorithm with bounded error probability:

Draw $n = \left\lceil \frac{2\ln(3)}{\gamma^2} \right\rceil$ independent random binary vectors $\mathbf{b}^{(i)} \in \{0,1\}^{d-|S|}$ for $i \in [n]$ from the uniform distribution on $\{0,1\}^{d-|S|}$ and define $\mathbf{y}^{(i)} \in \{0,1\}^d$ as $\mathbf{y}^{(i)}_S = \mathbf{x}_S$ and $\mathbf{y}^{(i)}_{S^c} = \mathbf{b}^{(i)}$. Set

$$\xi = \frac{1}{n}\sum_{i=1}^{n} \xi_i, \quad \text{where} \quad \xi_i = \begin{cases} 1, & \text{if } \Phi(\mathbf{x}) = \Phi\big(\mathbf{y}^{(i)}\big) \\ \\ 0, & \text{if } \Phi(\mathbf{x}) \neq \Phi\big(\mathbf{y}^{(i)}\big) \end{cases} \quad \text{for} \quad i = 1, \dots, n.$$

Then, answer *No* if $\xi < \delta - \frac{\gamma}{2}$ and *Yes* if $\xi \geq \delta - \frac{\gamma}{2}$.

The random variables ξ_i are independently and identically Bernoulli distributed variables with

$$p = \mathbb{E}[\xi_i] = \mathbb{E}[\xi] = P_{\mathbf{y}}(\Phi(\mathbf{y}_S) = \Phi(\mathbf{x}) \,|\, \mathbf{y}_S = \mathbf{x}_S).$$

Therefore, S is δ-relevant if $p \geq \delta$ and not $(\delta-\gamma)$-relevant if $p < \delta - \gamma$. We use Hoeffding's inequality (Hoeffding, 1994) to bound the error probability of the algorithm. Firstly, assume $p \geq \delta$. Then, we make an error if $\xi < \delta - \frac{\gamma}{2}$, which implies $p - \xi > \frac{\gamma}{2}$. The probability for this event can be bounded by

$$P\Big(p - \xi > \frac{\gamma}{2}\Big) \leq e^{-\frac{n\gamma^2}{2}} \leq \frac{1}{3}.$$

Secondly, assume $p < \delta - \gamma$. Then, we can bound the probability that $\xi \geq \delta - \frac{\gamma}{2}$, and thus $\xi - p > \frac{\gamma}{2}$, by

$$P\Big(\xi - p > \frac{\gamma}{2}\Big) \leq e^{-\frac{n\gamma^2}{2}} \leq \frac{1}{3}.$$

Altogether the algorithm answers correctly with probability $\frac{2}{3}$, showing that the problem lies in BPP.

A non-deterministic Turing machine with BPP-oracle can thus guess a set $S \subseteq [d]$ with $|S| \leq k$ and, using the oracle, check if it is δ-relevant or not $(\delta-\gamma)$-relevant . □

Similar to the original problem formulation, we can also state an optimisation version of the gapped problem. In this case, we relax the optimality condition on the set size k by allowing also sizes in the region between *Yes*- and *No*-instances of γ-Gapped-δ-Relevant-Input (cf. Figure 3). In other words, we want to find any k that is large enough so that it is not a *No*-instance for the gapped problem but not larger than the optimal solution of the ungapped minimization problem. Strictly speaking, this results in a search problem and not an optimisation problem. However, problems of this type can be referred to as weak optimisation problems (Grötschel et al., 1988).

Definition 4.3. For $\delta \in (0,1]$ and $\gamma \in [0,\delta)$ we define the Min-γ-Gapped-δ-Relevant-Input problem as follows.

Given: A Boolean circuit $\Phi\colon \{0,1\}^d \to \{0,1\}$ and $\mathbf{x} \in \{0,1\}^d$.

Find: $k \in \mathbb{N}$, $1 \leq k \leq d$ such that

(i) There exists $S \subseteq [d]$ with $|S| = k$ and S is $(\delta-\gamma)$-relevant for Φ and $\mathbf{x}$.

(ii) All $S \subseteq [d]$ with $|S| < k$ are not δ-relevant for Φ and $\mathbf{x}$.

Note that both for γ-Gapped-δ-Relevant-Input and Min-γ-Gapped-δ-Relevant-Input a solution for γ_1 will always also be a solution for $\gamma_2 > \gamma_1$. Specifically, being able to solve the ungapped problems introduced in Section 2 provides a solution to the gapped problems for any $\gamma > 0$.

4.2 The Set Size Gap (Approximability)

Even the gapped version of the minimisation problem is hard to approximate. We prove this by introducing another intermediate problem which we show to be NP-hard but which would be in P if there exists a "good" polynomial time approximation algorithm for Min-γ-Gapped-δ-Relevant-Input. As mentioned above, strictly speaking Min-γ-Gapped-δ-Relevant-Input is not an optimisation but a search problem. In order to give a meaning to the concept of approximation factors we use the following convention.

Definition 4.4. An algorithm for Min-γ-Gapped-δ-Relevant-Input has an approximation factor $c \geq 1$ if, for any instance $\{\Phi, \mathbf{x}\}$, it produces an approximate solution k such that there exists a true solution $\widetilde{k}$ (satisfying both conditions in Definition 4.3) with $\widetilde{k} \leq k \leq c\widetilde{k}$.

An algorithm that always produces the trivial approximate solution $k = d$ achieves an approximation factor d. We will show that it is generally hard to obtain better factors. More precisely, for any $\alpha > 0$ an algorithm achieving an approximation factor $d^{1-\alpha}$ can not be in polynomial time unless $\mathsf{P} = \mathsf{NP}$.

Definition 4.5. For $\delta \in (0, 1]$ and $\gamma \in [0, \delta)$ we define the Intermediate Problem 3 (IP3) as follows.

Given: A Boolean circuit $\Phi\colon \{0,1\}^d \to \{0,1\}$, $\mathbf{x} \in \{0,1\}^d$, and $k, m \in \mathbb{N}$, $1 \leq k \leq m \leq d$.

Decide:

Yes: There exists $S \subseteq [d]$ with $|S| \leq k$ and S is δ-relevant for Φ and $\mathbf{x}$.

No: All $S \subseteq [d]$ with $|S| \leq m$ are not $(\delta - \gamma)$-relevant for Φ and $\mathbf{x}$.

The restriction to the case $k = m$ is exactly the γ-Gapped-δ-Relevant-Input problem. However, here we also allow the case $k < m$ with a gap in the set sizes. This is illustrated in Figure 4.

Lemma 4.6. *For $\delta \in (0, 1)$ and $\gamma \in [0, \delta)$ we have* SAT $\preceq_p$ IP3*, in particular, in this case* IP3 *is* NP*-hard.*

The idea for this proof is rather simple. Given a SAT-formula Φ with d variables, we replace each variable by a conjunction of sufficiently many variables, i.e.

$$u_i = \bigwedge_{j=1}^{q} u_i^{(j)},$$

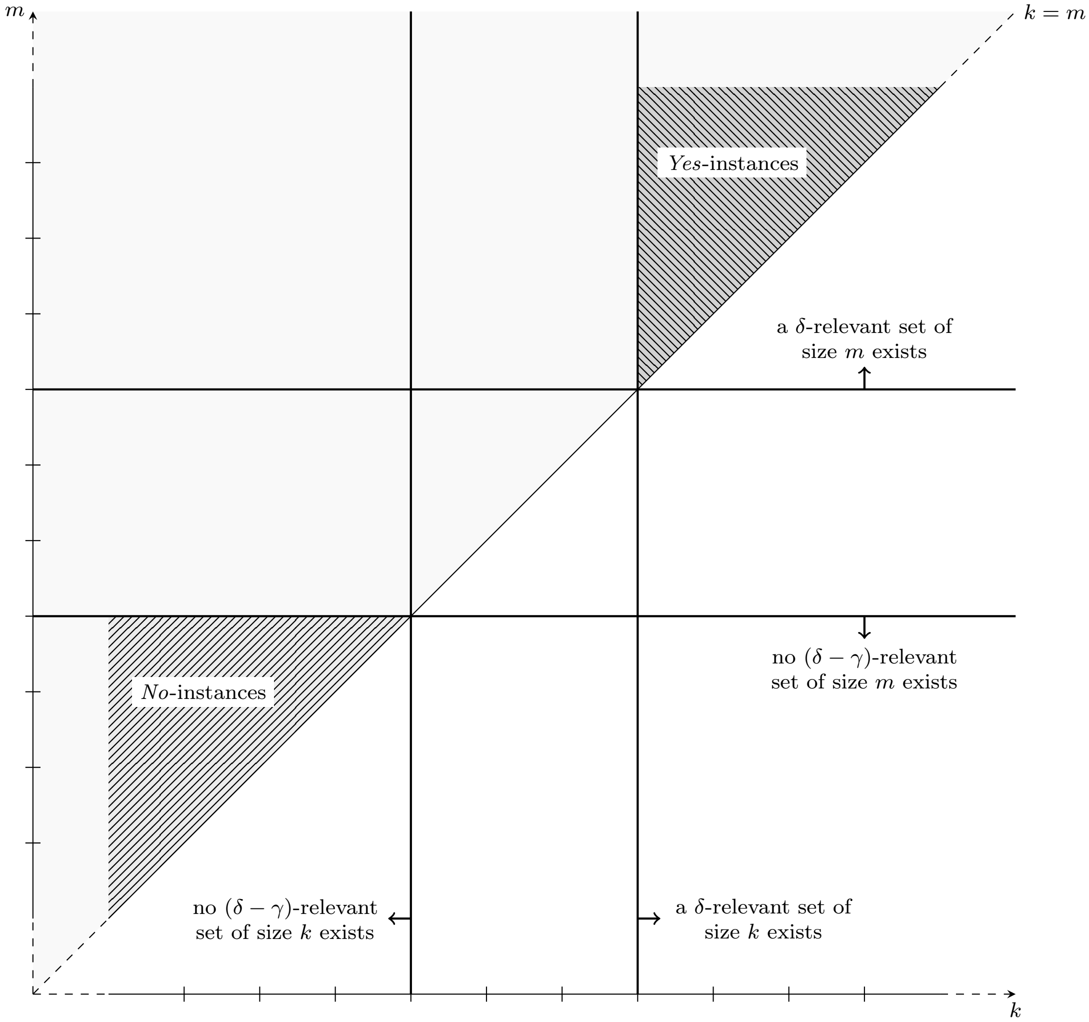
m
$k = m$
Yes-instances
a δ-relevant set of size m exists
no $(\delta - \gamma)$-relevant set of size m exists
No-instances
no $(\delta - \gamma)$-relevant set of size k exists
a δ-relevant set of size k exists
k

Figure 4: Visualization of the Intermediate Problem 3 for some fixed Φ and $\mathbf{x}$ and for various k and m. As before we do not expect an answer for this problem in the unmarked regions. The restriction to the diagonal $k = m$ corresponds to the γ-Gapped-δ-Relevant-Input problem (cf. Figure 3).

initially set to one. Fixing all $u_i^{(j)}$ effectively sets u_i to one. Randomising all $u_i^{(j)}$ effectively sets u_i to zero with high probability. If we now disjoin the resulting formula with a polynomially large conjunction of independent variables, i.e.

$$\Phi\left(\bigwedge_{j=1}^{q} u_1^{(j)}, \dots, \bigwedge_{j=1}^{q} u_d^{(j)}\right) \vee \left(\bigwedge_{i=1}^{M} v_i\right),$$

initially also set to one, then any satisfying assignment for Φ yields a δ-relevant set of size at most dq by effectively setting $\mathbf{u}$ to the satisfying assignment. On the other hand, if Φ is not satisfiable a $(\delta - \gamma)$-relevant set has to include almost all of the additional M variables. Choosing M sufficiently larger than dq results in the desired set size gap. We now make this argument formal.

Proof. Given a SAT instance in conjunctive normal form (CNF), let $\Phi\colon \{0,1\}^d \to \{0,1\}$ be the Boolean circuit representation corresponding to the CNF formula. From now on we will not distinguish between Φ and the CNF formula that it represents. We will construct $\{\Phi', \mathbf{x}', k', m'\}$ that is a *Yes*-instance for IP3 if and only if Φ is a *Yes*-instance for SAT. Let

$$q = \left\lceil \log_2\left(\frac{d}{1-\delta}\right) \right\rceil \quad \text{and} \quad p = \left\lfloor \log_2\left(\frac{1}{\delta-\gamma}\right) \right\rfloor + 1.$$

We set

- $k' = dq$,
- $m' \geq k'$ arbitrary but at most polynomial in d,
- $\Phi'\colon \{0,1\}^{d\times q} \times \{0,1\}^{m'+p} \to \{0,1\}$ with

  $$\Phi'(\mathbf{u}^{(1)}, ..., \mathbf{u}^{(q)}, \mathbf{v}) = \Phi\left(\bigwedge_{j=1}^{q} \mathbf{u}^{(j)}\right) \vee \left(\bigwedge_{i=1}^{m'+p} v_i\right),$$

  where each $\mathbf{u}^{(j)} \in \{0,1\}^d$ and the conjunction within Φ is understood component-wise,
- $\mathbf{x}' = \mathbf{1}_{dq+m'+p}$.

This is a polynomial time construction. By the choice of Φ' and $\mathbf{x}'$ we guarantee $\Phi'(\mathbf{x}') = 1$ regardless of the satisfiability of Φ.

Necessity: Let Φ be a *Yes*-instance for SAT. This means that there exists $\mathbf{x} \in \{0,1\}^d$ with $\Phi(\mathbf{x}) = 1$. Let $S = \{\, i \in [d] \,:\, x_i = 1 \,\}$ and $S' = S \times [q]$. Then, $|S'| \leq k'$. Denote

$$A(\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(q)}) = \bigwedge_{(i,j)\in S'} u_i^{(j)}.$$

Hence, S' is δ-relevant for Φ' and $\mathbf{x}'$ if $P\big(\Phi'(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(q)},\mathbf{v}) \,\big|\, A(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(q)})\big) \geq \delta$. We have

$$\begin{aligned} P\Big(\Phi'(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(q)},\mathbf{v}) \,\Big|\, A(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(q)})\Big) &\geq P\left(\Phi\left(\bigwedge_{j=1}^{q}\mathbf{u}^{(j)}\right) \,\middle|\, A(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(q)})\right) \\ &\geq P\left(\bigwedge_{j=1}^{q}\mathbf{u}^{(j)} = \mathbf{x} \,\middle|\, A(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(q)})\right). \end{aligned}$$

From this, with a union bound, we obtain

$$\begin{aligned} P\left(\bigwedge_{j=1}^{q}\mathbf{u}^{(j)} = \mathbf{x} \,\middle|\, A(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(q)})\right) &= 1 - P\left(\neg\bigwedge_{j=1}^{q}\mathbf{u}^{(j)} = \mathbf{x} \,\middle|\, A(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(q)})\right) \\ &= 1 - P\left(\exists i \in S^c \,:\, \bigwedge_{j=1}^{q} u_i^{(j)}\right) \\ &\geq 1 - |S^c| 2^{-q} \\ &\geq \delta, \end{aligned}$$

which shows that $\{\Phi', \mathbf{x}', k', m'\}$ is a *Yes*-instance for IP3.

Sufficiency: Now, conversely, let Φ be a *No*-instance for SAT. Then, for any subset $S' \subseteq [dq + m' + p]$ with $|S'| \leq m'$ we have

$$\begin{aligned} P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') = \Phi'(\mathbf{x}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{x}'_{S'}\big) &= P_{\mathbf{y}'}\big(\Phi'(\mathbf{y}') \,\big|\, \mathbf{y}'_{S'} = \mathbf{1}\big) \\ &= P_{(\mathbf{u}^{(1)},\dots,\mathbf{u}^{(q)},\mathbf{v})}\left(\bigwedge_{i=1}^{m'+p} v_i \,\middle|\, (\mathbf{u}^{(1)},\dots,\mathbf{u}^{(q)},\mathbf{v})_{S'} = \mathbf{1}\right) \\ &\leq 2^{-(m'+p-|S'|)} \\ &\leq 2^{-p} \\ &< \delta - \gamma. \end{aligned}$$

This shows that S' is not $(\delta-\gamma)$-relevant for Φ' and $\mathbf{x}'$, hence $\{\Phi', \mathbf{x}', k', m'\}$ is a *No*-instance for IP3. □

Recall the second main theorem of the paper which shows the inapproximability of the Min-γ-Gapped-δ-Relevant-Input problem.

Theorem 2.5. *Let $\delta \in (0,1)$ and $\gamma \in [0,\delta)$. Then, for any $\alpha \in (0,1)$ there is no polynomial time approximation algorithm for* Min-γ-Gapped-δ-Relevant-Input *with an approximation factor of $d^{1-\alpha}$ unless* $\mathsf{P} = \mathsf{NP}$.

The proof idea is to choose the m' in the previous proof large enough such that even an approximation algorithm that promises only a rough approximation factor could still be used to solve an NP-hard problem.

Proof. We prove this by showing that the existence of such an approximation algorithm would allow us to decide IP3 in polynomial time for certain instances. These can be chosen as in the proof of Lemma 4.6, which in turn implies that we could decide SAT in polynomial time. This is only possible if $\mathsf{P} = \mathsf{NP}$.

Given a SAT instance as a CNF formula, let $\Phi\colon \{0,1\}^d \to \{0,1\}$ be a Boolean circuit representation of the CNF formula and $\{\Phi', \mathbf{x}', k', m'\}$ an equivalent IP3 instance as in the proof of Lemma 4.6. As before, we will not further distinguish between the SAT formula in CNF and the circuit Φ representing it. We have seen that there is some freedom in the choice of m' as long as it satisfies $k' \leq m'$ and is at most polynomial in d. We will choose it in such a way, that any approximate solution k with approximation factor ${d'}^{1-\alpha}$ would allow us to decide $\{\Phi', \mathbf{x}', k', m'\}$ by checking whether $k < m'$ or $k > m'$. For this we set $m' = \left\lceil \max(2k'({k'}^{1-\alpha} + p^{1-\alpha}), (2k')^{\frac{1}{\alpha}} + 1)\right\rceil$ with $p = \left\lfloor \log_2\left(\frac{1}{\delta-\gamma}\right)\right\rfloor + 1$ as before. Recall that $k' = dq$ with $q = \left\lceil \log_2\left(\frac{d}{1-\delta}\right)\right\rceil$, so clearly m' is polynomial in d and $k' \leq m'$. Further, we have $m' > (2k')^{\frac{1}{\alpha}}$ so $1 - k'{m'}^{-\alpha} > \frac{1}{2}$ and therefore

$$m'(1 - k'{m'}^{-\alpha}) > \frac{m'}{2} \geq k'({k'}^{1-\alpha} + p^{1-\alpha}).$$

Now, let $d' = k' + m' + p$ denote the number of variables of Φ'. By the subadditivity of the map $z \mapsto z^{1-\alpha}$ we finally obtain

$$k'{d'}^{1-\alpha} = k'(k' + m' + p)^{1-\alpha} \leq k'\left({k'}^{1-\alpha} + {m'}^{1-\alpha} + p^{1-\alpha}\right) < m'.$$

It remains to show that an IP3 instance with $m' > k'{d'}^{1-\alpha}$ can be decided by an approximation algorithm for Min-γ-Gapped-δ-Relevant-Input with approximation factor ${d'}^{1-\alpha}$. Assume such an algorithm exists and let k be an approximate solution. Then, there exists a true solution $\widetilde{k}$ with $\widetilde{k} \leq k \leq {d'}^{1-\alpha}\widetilde{k}$.

Firstly, assume that $\{\Phi', \mathbf{x}', k', m'\}$ is a *Yes*-instance for IP3. Then, there is a δ-relevant set of size k'. However, no set smaller than $\widetilde{k}$ can be δ-relevant. This implies $\widetilde{k} \leq k'$ and therefore $k \leq {d'}^{1-\alpha}k' < m'$.

Secondly, assume that $\{\Phi', \mathbf{x}', k', m'\}$ is a *No*-instance for IP3. Then, all sets of size at most m' are not $(\delta - \gamma)$-relevant. However, there exists a $(\delta - \gamma)$-relevant set of size $\widetilde{k}$. This implies $k \geq \widetilde{k} > m'$.

Altogether, checking whether $k < m'$ or $k > m'$ decides $\{\Phi', \mathbf{x}', k', m'\}$. □

5. Discussion

We want to briefly discuss the scope of our analysis and the implications for algorithms that explain the predictions of classifiers such as neural networks. One could ask whether a solution set S to the Min-δ-Relevant-Input problem is in itself already a good explanation for a classifier prediction.

We are not arguing that a solution set alone is enough to fully explain the decision of a classifier to humans. The solution sets have some limitations that are discussed in the following subsection. We rather argue that any good explanation should contain a solution of the Min-δ-Relevant-Input problem. Given a good explanation for a classification prediction, we want to be able to conclude: If we fix these input variables then the classification will remain unchanged with high probability.

The evaluation methods for explanations of Samek et al. (2017a) and Fong and Vedaldi (2017) indicate that practitioners agree and design algorithms that should solve Min-δ-Relevant-Input in practice. Yet, our hardness results indicate that efficient methods cannot be proven to achieve this under all circumstances.

5.1 Stability and Uniqueness of δ-Relevant Sets

One should note that, like prime implicant explanations, the solution sets of the Min-δ-Relevant-Input problem are generally not unique. However, this behaviour is expected since an input can contain redundant information and each part of it alone can already be sufficient for the prediction. Even for instances with unique solution sets, these can depend sensitively on the probability threshold δ, i.e. slight changes in δ can lead to very different (and possibly not even overlapping) solution sets.

Further, the concept of δ-relevance is not monotone, in the sense that if S_1 is δ-relevant then $S_2 \supseteq S_1$ does not need to be δ-relevant as well. Again, this behaviour is expected, since there can be negative evidence in some of the variables. For example, think of a cat-vs-dog image classifier and an input containing both a cat and a dog. A set of variables including the cat will get less relevant for the prediction "cat" if we add more variables covering the dog to it. In fact, this non-monotonicity property was essential for the constructions used in our proof of the inapproximability theorem.

5.2 Binary vs Continuous

In our analysis, we considered Boolean circuit classifiers and binary input variables. As discussed in the introduction, the classifier is fixed in each problem instance, thus any class of classifiers that can efficiently describe Boolean circuits is also subject to our hardness results. This includes ReLU neural networks as well as Bayesian networks.[3]

Moreover, we only considered a binary partition into relevant and non-relevant variables analogous to the prime implicant explanation, even though many practical methods provide continuous relevance scores (in some cases even negative scores) (Samek et al., 2017b). The reason for this is two-fold. Firstly, there generally is no agreed upon interpretation of what continuous relevance scores mean. Therefore, we prefer to keep the clear meaning of a partition of the variables as in the prime implicants. Secondly, many of the most prominent applications of these relevance mappings rely on binarisations of the continuous relevance scores. A mask for relevant objects in the input, e.g. tumor cells in body tissue (Lyu & Haque, 2018) or expressive genes in a sequence (Vidovic et al., 2015), can be obtained by considering only the variables with a sufficiently large relevance score. The decision in the end is thus a binary one.

[3] The conditional probabilities describing the Bayesian network can represent truth tables of logical operators. This allows them to emulate Boolean circuits (Park, 2002).

5.3 Choice of Distribution

We restricted our analysis to the case of using the uniform distribution over the binary cube to randomise non-relevant variables. One could also consider more data-adapted or even empirical distributions. However, this may obscure insights about the classifiers reasoning. As an example, consider a faulty image classifier for boats that recognises water instead. If the data-adapted distribution only models images of boats on water, then marking the boat as relevant will always lead to random completions including water around the boat. Consequently, the prediction will remain unchanged even though the boat is not the true underlying reason for the classifier prediction. The classifier appears to work correctly, even though it will fail for images showing other objects on water. Instead, if the distribution used for the random completion is oblivious to the correlation between boats and water, the function output will only remain constant when the water is fixed, revealing the true relevant region.

6. Conclusion

There exists a wide variety of algorithms that aim to make modern machine learning methods interpretable. In turn, these algorithms themselves must be trusted. Thus it is clearly important to define the exact problem that the algorithms try to solve and to gain insights about the quality of their solution.

We discussed in this paper that our probabilistic version of prime implicant explanations is a crucial part of the problem that practitioners want to solve when they design interpretation algorithms. We showed that the task of identifying the relevant components of an input assignment to the variables of a Boolean circuit is complete for the complexity class $\mathsf{NP}^{\mathsf{PP}}$ and thus, for example, as difficult as planing under uncertainty (Drummond & Bresina, 1990).

Our paper furthermore investigates whether it is possible to reduce the complexity of the problem at the cost of getting only approximate solutions. We relaxed the problem by introducing the promise of a probability gap that allows for efficient bounding of the fraction of satisfying assignments. Furthermore we required that a solution set only approximates the optimal set up to any non-trivial approximation factor. Both these relaxations do not render this problem computationally feasible unless $\mathsf{P} = \mathsf{NP}$.

This makes practical guarantees for the interpretation of neural networks infeasible, as long as the networks are powerful enough to represent arbitrary logical functions. It is thus necessary to further restrict the problem setting. However, the hardness instances we construct can already be represented by neural networks with a fixed number of layers and bounded weights. Excluding these instances by further restricting these coarse hyperparameters would go against the idea of neural networks.

This only leaves the option of more subtle restrictions on the neural networks and the inputs that depend on the actual data structures on which the networks have been trained. These, however, are not yet well enough understood. As long as this is the case, we have to rely on heuristic solutions that are thoroughly evaluated numerically.

In a companion paper (Macdonald et al., 2019) we present a heuristic algorithm for a continuous (non-discrete) variant of the δ-Relevant-Input problem and classifier functions with compositional (layered) structure (such as neural networks). For several image classification tasks we demonstrate numerically that our algorithm approximates small relevant sets better than widely-used comparable methods.

Acknowledgments

The authors would like to thank Philipp Petersen for several fruitful discussions during the early stage of the project as well as Peter Bürgisser for helpful feedback on the first draft. J. M. and S. W. acknowledge support by DFG-GRK-2260 (BIOQIC). S. H. is grateful for support by CRC/TR 109 "Discretization in Geometry and Dynamics". G. K. acknowledges partial support by the Bundesministerium für Bildung und Forschung (BMBF) through the "Berliner Zentrum für Machinelles Lernen" (BZML), by the Deutsche Forschungsgemeinschaft (DFG) through Grants CRC 1114 "Scaling Cascades in Complex Systems", CRC/TR 109 "Discretization in Geometry and Dynamics", DFG-GRK-2433 (DAEDALUS), DFG-GRK-2260 (BIOQIC), SPP 1798 "Compressed Sensing in Information Processing" (CoSIP), by the Berlin Mathematics Research Centre MATH+, and by the Einstein Foundation Berlin.

Appendix A. Raising the Probability

We give constructive proofs of Lemmas 3.8 and 3.12, starting with the first.

Proof of Lemma 3.8: Let $\Phi\colon \{0,1\}^d \to \{0,1\}$ be arbitrary and $0 \le \delta_1 < \delta_2 < 1$. We will construct a monotone function $\Pi\colon \{0,1\}^n \to \{0,1\}$ such that

$$P_{\mathbf{y}}(\Phi(\mathbf{y})) > \delta_1 \quad\iff\quad P_{(\mathbf{y},\mathbf{r})}(\Phi(\mathbf{y}) \vee \Pi(\mathbf{r})) \ge \delta_2, \tag{3}$$

with

$$n \in \mathcal{O}\left(\left(d + \log_2\left(\frac{1-\delta_1}{1-\delta_2}\right)\right)^2\right).$$

In our context δ_1 and δ_2 are constant and therefore $n \in \mathcal{O}(d^2)$.

Denote $\Phi'\colon \{0,1\}^d \times \{0,1\}^n \to \{0,1\}\colon (\mathbf{y},\mathbf{r}) \mapsto \Phi(\mathbf{y}) \vee \Pi(\mathbf{r})$, then

$$P(\Phi') = P(\Phi) + (1 - P(\Phi))P(\Pi), \tag{4}$$

which is monotonically increasing in both $P(\Phi)$ and $P(\Pi)$. Thus, it suffices to consider the edge case when $P(\Phi)$ is close to δ_1. Since $P(\Phi)$ can only take values in $\left\{\frac{0}{2^d}, \frac{1}{2^d}, \dots, \frac{2^d}{2^d}\right\}$ we see that (3) is equivalent to the two conditions

$$P(\Phi) = \frac{\lfloor \delta_1 2^d \rfloor}{2^d} \qquad \Longrightarrow \qquad P(\Phi') < \delta_2,$$
$$P(\Phi) = \frac{\lfloor \delta_1 2^d \rfloor + 1}{2^d} \qquad \Longrightarrow \qquad P(\Phi') \ge \delta_2,$$

which together with (4) is equivalent to

$$\frac{\lfloor \delta_1 2^d \rfloor}{2^d} + \frac{2^d - \lfloor \delta_1 2^d \rfloor}{2^d} P(\Pi) < \delta_2 \tag{5}$$
$$\frac{\lfloor \delta_1 2^d \rfloor + 1}{2^d} + \frac{2^d - \lfloor \delta_1 2^d \rfloor - 1}{2^d} P(\Pi) \ge \delta_2. \tag{6}$$

In the case $\delta_1 < \delta_2 \leq \frac{\lfloor \delta_1 2^d \rfloor + 1}{2^d}$ these conditions are already fulfilled if we simply set $\Pi \equiv 0$. Otherwise, if $\delta_2 > \frac{\lfloor \delta_1 2^d \rfloor + 1}{2^d}$, rearranging (5) and (6) yields the bounds

$$a \leq P(\Pi) < b$$

on $P(\Pi)$, where

$$\begin{aligned} a &= \frac{\delta_2 2^d - \lfloor \delta_1 2^d \rfloor - 1}{2^d - \lfloor \delta_1 2^d \rfloor - 1}, \\ b &= \frac{\delta_2 2^d - \lfloor \delta_1 2^d \rfloor}{2^d - \lfloor \delta_1 2^d \rfloor}. \end{aligned}$$

It is not hard to check that indeed we have $0 \leq a < b \leq 1$.

In Appendix C we show for $\eta \in [0,1]$ and $\ell \in \mathbb{N}$ the existence of a monotone DNF-function $\Pi_{\eta,\ell} \colon \{0,1\}^n \to \{0,1\}$ such that $\Pi_{\eta,\ell}(\mathbf{0}_n) = 0$, $\Pi_{\eta,\ell}(\mathbf{1}_n) = 1$, and

$$|P(\Pi_{\eta,\ell}) - \eta| \leq 2^{-\ell}$$

with $n \leq \frac{\ell(\ell+3)}{2} \in \mathcal{O}(\ell^2)$. We conclude by choosing

$$\begin{aligned} \eta &= \frac{b+a}{2}, \\ \ell &= \left\lfloor -\log_2\left(\frac{b-a}{2}\right) \right\rfloor + 1 \in \mathcal{O}\bigg(d + \log_2\left(\frac{1-\delta_1}{1-\delta_2}\right)\bigg), \end{aligned}$$

and setting $\Pi = \Pi_{\eta,\ell}$. We get $n \in \mathcal{O}(\ell^2) = \mathcal{O}\Big(\Big(d + \log_2\Big(\frac{1-\delta_1}{1-\delta_2}\Big)\Big)^2\Big)$, which finishes the proof of Lemma 3.8.

Proof of Lemma 3.12: We proceed analogously to before. For $0 \leq \delta_1 \leq \delta_2 < 1$, we construct a monotone function $\Pi \colon \{0,1\}^n \to \{0,1\}$ such that

$$P_{\mathbf{y}}(\Phi(\mathbf{y})) \geq \delta_1 \quad \Longleftrightarrow \quad P_{(\mathbf{y},\mathbf{r})}(\Phi(\mathbf{y}) \vee \Pi(\mathbf{r})) > \delta_2,$$

with

$$n \in \mathcal{O}\bigg(\bigg(d + \log_2\left(\frac{1-\delta_1}{1-\delta_2}\right)\bigg)^2\bigg).$$

Again, δ_1 and δ_2 are constant in our setting and therefore $n \in \mathcal{O}\big(d^2\big)$.

Similar to before, in the case that $\delta_1 \leq \delta_2 < \frac{\lceil \delta_1 2^d \rceil}{2^d}$, we can simply set $\Pi \equiv 0$. Otherwise, we get the bounds

$$a < P(\Pi) \leq b$$

with

$$\begin{aligned} a &= \frac{\delta_2 2^d - \lceil \delta_1 2^d \rceil}{2^d - \lceil \delta_1 2^d \rceil}, \\ b &= \frac{\delta_2 2^d - \lceil \delta_1 2^d \rceil + 1}{2^d - \lceil \delta_1 2^d \rceil + 1}. \end{aligned}$$

Again, we can check that $0 \leq a < b \leq 1$, and set

$$\begin{aligned}\eta &= \frac{b+a}{2},\\ \ell &= \left\lfloor -\log_2\left(\frac{b-a}{2}\right)\right\rfloor + 1 \in \mathcal{O}\left(d + \log_2\left(\frac{1-\delta_1}{1-\delta_2}\right)\right),\end{aligned}$$

and $\Pi = \Pi_{\eta,\ell}$ with $n \in \mathcal{O}(\ell^2) = \mathcal{O}\Big(\Big(d + \log_2\Big(\frac{1-\delta_1}{1-\delta_2}\Big)\Big)^2\Big)$, which concludes the proof of Lemma 3.12.

Appendix B. Lowering the Probability

We give constructive proofs of Lemmas 3.9 and 3.13, starting with the first.

Proof of Lemma 3.9: Let $\Phi\colon \{0,1\}^d \to \{0,1\}$ be arbitrary and $0 < \delta_1 \leq \delta_2 \leq 1$. We will construct a monotone function $\Pi\colon \{0,1\}^n \to \{0,1\}$ such that

$$P_{\mathbf{y}}(\Phi(\mathbf{y})) > \delta_2 \quad \iff \quad P_{(\mathbf{y},\mathbf{r})}(\Phi(\mathbf{y}) \wedge \Pi(\mathbf{r})) \geq \delta_1, \tag{7}$$

with

$$n \in \mathcal{O}\left(\left(d + \log_2\left(\frac{\delta_2}{\delta_1}\right)\right)^2\right).$$

In our context δ_1 and δ_2 are constant and therefore $n \in \mathcal{O}(d^2)$.

Denote $\Phi'\colon \{0,1\}^d \times \{0,1\}^n \to \{0,1\}\colon (\mathbf{y},\mathbf{r}) \mapsto \Phi(\mathbf{y}) \wedge \Pi(\mathbf{r})$, then

$$P(\Phi') = P(\Phi)P(\Pi), \tag{8}$$

which is monotonically increasing in both $P(\Phi)$ and $P(\Pi)$. Thus, it suffices to consider the edge case when $P(\Phi)$ is close to δ_2. Since $P(\Phi)$ can only take values in $\left\{\frac{0}{2^d}, \frac{1}{2^d}, \ldots, \frac{2^d}{2^d}\right\}$ we see that (7) is equivalent to the two conditions

$$\begin{aligned} P(\Phi) &= \frac{\lfloor \delta_2 2^d \rfloor}{2^d} &&\implies && P(\Phi') < \delta_1,\\ P(\Phi) &= \frac{\lfloor \delta_2 2^d \rfloor + 1}{2^d} &&\implies && P(\Phi') \geq \delta_1,\end{aligned}$$

which together with (8) is equivalent to

$$\frac{\lfloor \delta_2 2^d \rfloor}{2^d} P(\Pi) < \delta_1 \tag{9}$$

$$\frac{\lfloor \delta_2 2^d \rfloor + 1}{2^d} P(\Pi) \geq \delta_1. \tag{10}$$

In the case $\frac{\lfloor \delta_2 2^d \rfloor}{2^d} < \delta_1 \leq \delta_2$ these conditions are already fulfilled if we simply set $\Pi \equiv 1$. Otherwise, if $\delta_1 \leq \frac{\lfloor \delta_2 2^d \rfloor}{2^d}$, rearranging (9) and (10) yields the bounds

$$a < P(\Pi) \leq b$$

on $P(\Pi)$, where

$$\begin{aligned} a &= \frac{\delta_1 2^d}{\lfloor \delta_2 2^d \rfloor + 1}, \\ b &= \frac{\delta_1 2^d}{\lfloor \delta_2 2^d \rfloor}. \end{aligned}$$

It is not hard to check that indeed we have $0 \leq a < b \leq 1$.

In Appendix C, we show for $\eta \in [0,1]$ and $\ell \in \mathbb{N}$ the existence of a monotone DNF-function $\Pi_{\eta,\ell} \colon \{0,1\}^n \to \{0,1\}$ such that $\Pi_{\eta,\ell}(\mathbf{0}_n) = 0$, $\Pi_{\eta,\ell}(\mathbf{1}_n) = 1$, and

$$|P(\Pi_{\eta,\ell}) - \eta| \leq 2^{-\ell}$$

with $n \leq \frac{\ell(\ell+3)}{2} \in \mathcal{O}(\ell^2)$. We conclude by choosing

$$\begin{aligned} \eta &= \frac{b+a}{2}, \\ \ell &= \left\lfloor -\log_2\left(\frac{b-a}{2}\right) \right\rfloor + 1 \in \mathcal{O}\left(d + \log_2\left(\frac{\delta_2}{\delta_1}\right)\right), \end{aligned}$$

and setting $\Pi = \Pi_{\eta,\ell}$. We get $n \in \mathcal{O}(\ell^2) = \mathcal{O}\left(\left(d + \log_2\left(\frac{\delta_2}{\delta_1}\right)\right)^2\right)$, which finishes the proof of Lemma 3.9.

Proof of Lemma 3.13: We proceed analogously to before. For $0 < \delta_1 < \delta_2 \leq 1$, we construct a monotone function $\Pi \colon \{0,1\}^n \to \{0,1\}$ such that

$$P_{\mathbf{y}}(\Phi(\mathbf{y})) \geq \delta_2 \quad \iff \quad P_{(\mathbf{y},\mathbf{r})}(\Phi(\mathbf{y}) \wedge \Pi(\mathbf{r})) > \delta_1,$$

with

$$n \in \mathcal{O}\left(\left(d + \log_2\left(\frac{\delta_2}{\delta_1}\right)\right)^2\right).$$

Again, δ_1 and δ_2 are constant in our setting and therefore $n \in \mathcal{O}\big(d^2\big)$.

Similar to before, in case that $\frac{\lceil \delta_2 2^d \rceil - 1}{2^d} \leq \delta_1 < \delta_2$, we can simply set $\Pi \equiv 1$. Otherwise, we get the bounds

$$a < P(\Pi) \leq b$$

with

$$\begin{aligned} a &= \frac{\delta_1 2^d}{\lceil \delta_2 2^d \rceil} \\ b &= \frac{\delta_1 2^d}{\lceil \delta_2 2^d \rceil - 1}. \end{aligned}$$

Again, we can check that $0 \leq a < b \leq 1$, and set

$$\begin{aligned} \eta &= \frac{b+a}{2}, \\ \ell &= \left\lfloor -\log_2\left(\frac{b-a}{2}\right) \right\rfloor + 1 \in \mathcal{O}\left(d + \log_2\left(\frac{\delta_2}{\delta_1}\right)\right), \end{aligned}$$

and $\Pi = \Pi_{\eta,\ell}$ with $n \in \mathcal{O}(\ell^2) = \mathcal{O}\Big(\Big(d + \log_2\Big(\frac{\delta_2}{\delta_1}\Big)\Big)^2\Big)$, which concludes the proof of Lemma 3.13.

Appendix C. Construction of the Functions $\Pi_{\eta,\ell}$

For $\eta \in [0,1]$ (the target probability) and $\ell \in \mathbb{N}$ (the accuracy) we construct a Boolean function $\Pi_{\eta,\ell} \colon \{0,1\}^n \to \{0,1\}$ in disjunctive normal form with $n \in \mathcal{O}(\ell^2)$, $\Pi_{\eta,\ell}(\mathbf{0}_n) = 0$, $\Pi_{\eta,\ell}(\mathbf{1}_n) = 1$, and

$$|\eta - P(\Pi_{\eta,\ell})| \leq 2^{-\ell}.$$

If $\eta \leq 2^{-\ell}$, we can simply choose $\Pi_{\eta,\ell}(x_1, \dots, x_\ell) = \bigwedge_{k=1}^{\ell} x_k$. So from now on assume $2^{-\ell} < \eta \leq 1$. In this case we construct a sequence of functions $\Pi_i \colon \{0,1\}^{n_i} \to \{0,1\}$ such that $p_i = P(\Pi_i)$ is monotonically increasing and converges to η from below. We proceed according to the following iterative procedure: Start with the constant function $\Pi_0 \equiv 0$. Given Π_i and p_i we can stop and set $\Pi_{\eta,\ell} = \Pi_i$ if $|\eta - p_i| \leq 2^{-\ell}$. Otherwise, we set $n_{i+1} = n_i + \Delta n_i$ with

$$\Delta n_i = \operatorname{argmin}\big\{\, n \in \mathbb{N} : p_i + (1 - p_i)2^{-n} \leq \eta \,\big\}, \tag{11}$$

and

$$\Pi_{i+1}(x_1, \dots, x_{n_{i+1}}) = \Pi_i(x_1, \dots, x_{n_i}) \vee \left(\bigwedge_{k=n_i+1}^{n_{i+1}} x_k \right).$$

Clearly, we obtain $p_{i+1} = p_i + (1 - p_i)2^{-\Delta n_i}$. We will see below that Δn_i can not be too large and thus (11) can be efficiently computed by sequential search.

Lemma C.1. *The sequence $(p_i)_{i \in \mathbb{N}}$ is monotonically increasing and we have*

$$|\eta - p_{i+1}| \leq \frac{1}{2}|\eta - p_i|$$

for all $i \in \mathbb{N}$. In particular $|\eta - p_i| \leq 2^{-i}$ and $p_i \to \eta$ as $i \to \infty$.

Proof. Since $0 = p_0 \leq \eta$ and by choice of Δn_i, we have $p_i \leq \eta$ for all $i \in \mathbb{N}$. Also from (11) we know that $p_i + (1 - p_i)2^{-(\Delta n_i - 1)} > \eta$ since otherwise Δn_i would be chosen smaller. Therefore,

$$\begin{aligned} \eta - p_{i+1} &= \eta - p_i - (1 - p_i)2^{-\Delta n_i} \\ &= \eta - \frac{1}{2}p_i - \frac{1}{2}\Big(p_i + (1 - p_i)2^{-(\Delta n_i - 1)}\Big) \\ &\leq \frac{1}{2}(\eta - p_i). \end{aligned}$$

The second part simply follows by repeatedly applying the above recursion i times and from the fact that $\eta - p_0 = \eta \leq 1$. □

We conclude that the desired accuracy is reached after at most ℓ iterations in which case we stop and set $\Pi_{\eta,\ell} = \Pi_\ell$. It remains to determine how many variables need to be used in total. We first bound how many variables are added in each step.

Lemma C.2. *For any $i \in \mathbb{N}$, we have $\Delta n_i < -\log_2(\eta - p_i) + 1$.*

Proof. As before we know $p_i + (1-p_i)2^{-(\Delta n_i - 1)} > \eta$ since otherwise Δn_i would be chosen smaller. This implies

$$2^{-(\Delta n_i - 1)} > \frac{\eta - p_i}{1 - p_i} \geq \eta - p_i,$$

and therefore $\Delta n_i < -\log_2(\eta - p_i) + 1$. □

This can finally be used to bound how many variables are used in total.

Lemma C.3. *The total number of variables for $\Pi_{\eta,\ell} = \Pi_\ell$ is*

$$n = n_\ell = \sum_{i=1}^{\ell} \Delta n_{i-1} \in \mathcal{O}\big(\ell^2\big).$$

Proof. From Lemma C.1 we get $\eta - p_i \geq 2(\eta - p_{i+1})$ and thus $\eta - p_i \geq 2^{\ell-1-i}(\eta - p_{\ell-1})$. Without loss of generality we can assume $\eta - p_{\ell-1} \geq 2^{-\ell}$ since otherwise we can stop the iterative construction of $\Pi_{\eta,\ell}$ at $\ell - 1$. Using Lemma C.2, this immediately results in

$$\begin{aligned} n &= \sum_{i=1}^{\ell} \Delta n_{i-1} \\ &\leq \sum_{i=1}^{\ell} -\log_2(\eta - p_{i-1}) + 1 \\ &\leq \sum_{i=1}^{\ell} -\log_2\Big(2^{\ell-i}(\eta - p_{\ell-1})\Big) + 1 \\ &\leq \sum_{i=1}^{\ell} -\log_2\big(2^{-i}\big) + 1 \\ &= \frac{\ell(\ell+1)}{2} + \ell \in \mathcal{O}\big(\ell^2\big). \end{aligned}$$

□

Appendix D. Neutral Operation

We provide a constructive proof of Lemma 3.10.

Proof of Lemma 3.10: Let $\Phi\colon \{0,1\}^d \to \{0,1\}$ be arbitrary and $0 < \delta < 1$. We will construct a function $\Gamma = \Gamma_{d,\delta}\colon \{0,1\}^r \to \{0,1\}$ so that for some $n_{d,\delta} \in \mathbb{N}$ we have

$$P_{\mathbf{y}}(\Phi(\mathbf{y})) \geq \delta \quad \Longleftrightarrow \quad P_{(\mathbf{y},\mathbf{r},\mathbf{t})}\left((\Phi(\mathbf{y}) \wedge \Gamma(\mathbf{r})) \vee \left(\bigwedge_{i=1}^{n} t_i\right)\right) \geq \delta, \tag{12}$$

for all $n \geq n_{d,\delta}$ and

$$r + n_{d,\delta} \in \mathcal{O}\bigg(\log\bigg(\frac{1}{\delta}\bigg) + d^2\bigg).$$

We introduce $\Phi' = \Phi \wedge \Gamma$ and $\Phi'' = \Phi' \vee (\bigwedge_{i=1}^{n} t_i)$. Let us distinguish three cases

$$\begin{aligned}
&\textbf{Case I:} && \delta \leq 2^{-d}, \\
&\textbf{Case II:} && \delta > 2^{-d} \quad \text{and} \quad \left\lceil \delta 2^d \right\rceil - \delta 2^d \geq \frac{2}{3}, \\
&\textbf{Case III:} && \delta > 2^{-d} \quad \text{and} \quad \left\lceil \delta 2^d \right\rceil - \delta 2^d < \frac{2}{3}.
\end{aligned}$$

Let us begin with the construction for the first case. Here, we see that $P(\Phi) < \delta$ is equivalent to $P(\Phi) = 0$. We simply set $\Gamma \equiv 1$ and $n_{d,\delta} = \lceil \log\left(\frac{1}{\delta}\right) \rceil + 1$. It is easy to check that this satisfies Equation (12).

Next, for the second case, we want to construct Γ such that

$$P(\Phi) = \frac{\lceil \delta 2^d \rceil}{2^d} \qquad \Longrightarrow \qquad P(\Phi') \geq \delta, \tag{13}$$

$$P(\Phi) = \frac{\lceil \delta 2^d \rceil - 1}{2^d} \qquad \Longrightarrow \qquad P(\Phi') < \delta - \frac{1}{3} 2^{-d}, \tag{14}$$

which results in the condition

$$a \leq P(\Gamma) < b$$

with

$$\begin{aligned}
a &= \frac{\delta 2^d}{\lceil \delta 2^d \rceil} \\
b &= \frac{\delta 2^d - \frac{1}{3}}{\lceil \delta 2^d \rceil - 1}.
\end{aligned}$$

Thus we can set $\Gamma = \Pi_{\eta,\ell}$ according to Appendix C with $\eta = \frac{b+a}{2}$ and $\ell = \lfloor \log\left(\frac{2}{b-a}\right) \rfloor + 1$. Using the fact that $\delta > 2^{-d}$ and hence

$$b - a = \frac{\delta 2^d - \frac{1}{3} \lceil \delta 2^d \rceil}{\lceil \delta 2^d \rceil \left(\lceil \delta 2^d \rceil - 1 \right)} \geq \frac{1}{3} 2^{-2d},$$

we obtain $\ell \leq 2d + \lfloor \log(6) \rfloor + 1 = 2d + 3$. In Appendix C we showed that $r \in \mathcal{O}(\ell^2)$ and thus $r \in \mathcal{O}(d^2)$. We continue to construct Φ'' by choosing $n_{d,\delta}$ such that

$$\begin{aligned}
P(\Phi') &\geq \delta & &\Longrightarrow & P(\Phi'') &\geq \delta, \\
P(\Phi') &< \delta - \frac{1}{3} 2^{-d} & &\Longrightarrow & P(\Phi'') &< \delta,
\end{aligned}$$

holds for all $n \geq n_{d,\delta}$. The first condition is automatically fulfilled. From

$$P(\Phi'') = P(\Phi') + (1 - P(\Phi')) 2^{-n},$$

as well as $(1 - P(\Phi')) \leq 1$ we observe that

$$2^{-n} < \frac{1}{3} 2^{-d}$$

is sufficient for the other condition. Thus, we choose $n_{d,\delta} = d + \lfloor \log(3) \rfloor + 1 = d + 2$.

Finally, for the third case, we again want to construct Γ so that (13) and (14) hold. Here, this is already satisfied by setting $\Gamma \equiv 1$. We continue analogously as in the second case and choose the same $n_{d,\delta}$.

References

Akers, S. B. (1978). Binary decision diagrams. *IEEE Transactions on computers*, *C-27*(6), 509–516.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, *10*(7), 1–46.

Brown, T. B., Mané, D., Roy, A., Abadi, M., & Gilmer, J. (2017). Adversarial patch. *CoRR*, *abs/1712.09665*.

Bryant, R. E. (1986). Graph-based algorithms for boolean function manipulation. *Computers, IEEE Transactions on*, *100*(8), 677–691.

Coudert, O., & Madre, J. C. (1992). Implicit and incremental computation of primes and essential primes of boolean functions.. In *DAC*, Vol. 92, pp. 36–39.

Darwiche, A. (2000). On the tractable counting of theory models and its application to belief revision and truth maintenance. *CoRR*, *cs.AI/0003044*.

Darwiche, A. (2011). Sdd: A new canonical representation of propositional knowledge bases. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

de Campos, C. P., & Ji, Q. (2008). Strategy selection in influence diagrams using imprecise probabilities. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI'08, pp. 121—128, Arlington, Virginia, USA. AUAI Press.

Deng, X., & Papadimitriou, C. H. (1994). On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, *19*(2), 257–266.

Drummond, M., & Bresina, J. (1990). *Anytime synthetic projection: Maximizing the probability of goal satisfaction*. NASA, Ames Research Center, Artificial Intelligence Research Branch.

Eiter, T., & Gottlob, G. (1995). The complexity of logic-based abduction. *Journal of the ACM (JACM)*, *42*(1), 3–42.

Erhan, D., Bengio, Y., Courville, A., & Vincent, P. (2009). Visualizing higher-layer features of a deep network. Tech. rep. 1341, University of Montreal. Also presented at the ICML 2009 Workshop on Learning Feature Hierarchies, Montréal, Canada.

Fatima, S. S., Wooldridge, M., & Jennings, N. R. (2008). A linear approximation method for the shapley value. *Artificial Intelligence*, *172*(14), 1673–1699.

Fong, R. C., & Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3429–3437.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, *36*(4), 193–202.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Graves, A., Mohamed, A.-R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649.

Grötschel, M., Lovász, L., & Schrijver, A. (1988). *Geometric Algorithms and Combinatorial Optimization*, Vol. 2 of *Algorithms and Combinatorics*. Springer.

Hoeffding, W. (1994). Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pp. 409–426. Springer.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, *4*(2), 251–257.

Ignatiev, A., Narodytska, N., & Marques-Silva, J. (2019). Abduction-based explanations for machine learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 1511–1519.

Kang, E., Min, J., & Ye, J. C. (2017). A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction. *Medical Physics*, *44*(10), e360–e375.

Khosravi, P., Liang, Y., Choi, Y., & Van den Broeck, G. (2019). What to expect of classifiers? reasoning about logistic regression with missing features. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 2716–2724. International Joint Conferences on Artificial Intelligence Organization.

Kononenko, I., et al. (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, *11*(Jan), 1–18.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc.

Littman, M. L., Goldsmith, J., & Mundhenk, M. (1998). The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research*, *9*, 1–36.

Littman, M. L., Majercik, S. M., & Pitassi, T. (2001). Stochastic boolean satisfiability. *Journal of Automated Reasoning*, *27*(3), 251–296.

Liu, X., Yang, H., Song, L., Li, H., & Chen, Y. (2018). Dpatch: Attacking object detectors with adversarial patches. *CoRR*, *abs/1806.02299*.

Lyu, B., & Haque, A. (2018). Deep learning based tumor type classification using gene expression data. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 89–96. ACM.

Macdonald, J., Wäldchen, S., Hauch, S., & Kutyniok, G. (2019). A rate-distortion framework for explaining neural network decisions. *CoRR*, *abs/1905.11092*.

Manquinho, V. M., Oliveira, A. L., & Marques-Silva, J. (1998). Models and algorithms for computing minimum-size prime implicants. In *Proceedings of the International Workshop on Boolean Problems*.

Marquis, P. (1991). Extending abduction from propositional to first-order logic. In *International Workshop on Fundamentals of Artificial Intelligence Research*, pp. 141–155. Springer.

Marquis, P. (2000). Consequence finding algorithms. In Kohlas, J., & Moral, S. (Eds.), *Handbook of Defeasible Reasoning and Uncertainty Management Systems: Algorithms for Uncertainty and Defeasible Reasoning*, pp. 41–145. Springer Netherlands, Dordrecht.

McBee, M. P., Awan, O. A., Colucci, A. T., Ghobadi, C. W., Kadom, N., Kansagra, A. P., Tridandapani, S., & Auffermann, W. F. (2018). Deep learning in radiology. *Academic Radiology*, *25*(11), 1472–1480.

Mukherjee, A., & Basu, A. (2017). Lower bounds over boolean inputs for deep neural networks with relu gates. *CoRR*, *abs/1711.03073*.

Nielsen, M. A. (2018). Neural networks and deep learning..

Parberry, I. (1996). *Circuit complexity and feedforward neural networks*. Hillsdale, NJ: Lawrence Erlbaum.

Park, J. D. (2002). Map complexity results and approximation methods. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pp. 388–396. Morgan Kaufmann Publishers Inc.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., & Müller, K.-R. (2017a). Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(11), 2660–2673.

Samek, W., Wiegand, T., & Müller, K. (2017b). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *CoRR*, *abs/1708.08296*.

Shapley, L. S. (1953). A value for n-person games. In Kuhn, H. W., & Tucker, A. W. (Eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, Princeton.

Shen, D., Wu, G., & Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, *19*(1), 221–248.

Shih, A., Choi, A., & Darwiche, A. (2018). A symbolic approach to explaining bayesian network classifiers. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pp. 5103—5111. AAAI Press.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In Bengio, Y., & LeCun, Y. (Eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Sun, Y., Wang, X., & Tang, X. (2013). Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3476–3483.

Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 26*, pp. 2553–2561. Curran Associates, Inc.

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708.

Toshev, A., & Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1653–1660.

Vidovic, M. M.-C., Görnitz, N., Müller, K.-R., Rätsch, G., & Kloft, M. (2015). Opening the black box: Revealing interpretable sequence motifs in kernel-based learning algorithms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 137–153. Springer.

Xie, J., Xu, L., & Chen, E. (2012). Image denoising and inpainting with deep neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 25*, pp. 341–349. Curran Associates, Inc.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In Fleet, D., Pajdla, T., Schiele, B., & Tuytelaars, T. (Eds.), *Computer Vision – ECCV 2014*, pp. 818–833, Cham. Springer International Publishing.