Synthesizing Pareto-Optimal Interpretations for Black-Box Models

Hazem Torfah[1], Shetal Shah[2], Supratik Chakraborty[2], S. Akshay[2], Sanjit A. Seshia[1]
[1]*University of California at Berkeley*
{torfah, sseshia}@berkeley.edu
[2]*Indian Institute of Technology Bombay*
{shetals, supratik, akshayss}@cse.iitb.ac.in

***Abstract*—We present a new multi-objective optimization approach for synthesizing interpretations that "explain" the behavior of black-box machine learning models. Constructing *human-understandable* interpretations for black-box models often requires balancing conflicting objectives. A simple interpretation may be easier to understand for humans while being less precise in its predictions vis-a-vis a complex interpretation. Existing methods for synthesizing interpretations use a single objective function and are often optimized for a single class of interpretations. In contrast, we provide a more general and multi-objective synthesis framework that allows users to choose (1) the class of syntactic templates from which an interpretation should be synthesized, and (2) quantitative measures on both the correctness and explainability of an interpretation. For a given black-box, our approach yields a set of Pareto-optimal interpretations with respect to the correctness and explainability measures. We show that the underlying multi-objective optimization problem can be solved via a reduction to quantitative constraint solving, such as weighted maximum satisfiability. To demonstrate the benefits of our approach, we have applied it to synthesize interpretations for black-box neural-network classifiers. Our experiments show that there often exists a rich and varied set of choices for interpretations that are missed by existing approaches.**

I. Introduction

Machine learning (ML) components, especially deep neural networks (DNNs), are increasingly being deployed in domains where trustworthiness and accountability are major concerns. Such domains include health care [5], automotive systems [28], finance [21], loans and mortgages [25], [33], and cyber-security [10] among others. For a system to be considered accountable and trustworthy, it is necessary to provide understandable explanations to (possibly expert) humans of why the system took specific actions/decisions in response to inputs of concern. This requires the availability of models that are human-understandable, and that also predict the outcome of different components of the system with reasonable accuracy. Laws and regulations, such as the General Data Protection Regulation (GDPR) in Europe [1], are already emerging with requirements on explainability of ML components in such systems. Unfortunately, the working of ML components like DNNs can be extremely complex to comprehend, and more so when the components are used as black boxes. Therefore, there is an urgent need for automated techniques that generate "easy-to-understand" and "targeted" interpretations of black-box ML components, with formal guarantees about tradeoffs between correctness and explainability.

Synthesizing a "good" interpretation of a black-box ML component often requires striking the right balance between correctness or accuracy of the interpretation (measured in terms of fidelity, misclassification rate of predictions etc.) and explainability or understandability (approximated by the size of the ML model – e.g., depth of decision tree/list/diagram, number and nature of predicates used, etc.). In most cases, the correctness and explainability measures are in direct conflict with each other. Thus, a simple interpretation that is easily understood by humans may disagree in its predictions with the output of a black-box ML component for many input instances, whereas an interpretation that correctly predicts the output for most input instances may be too large and unwieldy for human comprehension. This is not surprising since components like DNNs are often used to learn highly non-trivial functions for which simple models are not available. Therefore, *synthesis of interpretations for black-box ML components is inherently a multi-objective optimization problem with conflicting objectives, and Pareto optimality is the best we can hope for when synthesizing such interpretations.*

The literature contains a rich collection of techniques for synthesis of interpretations for black-box ML components (see, for example, recent surveys by [2] and [13]). Most of these approaches optimize a single correctness measure (e.g. misclassification rate on a set of samples) while systematically constraining some explainability measure (e.g. number of nodes or depth of a decision tree). Examples of such techniques include [19] wherein sparse logical formulae are synthesized, and also recent approaches to learning optimal decision trees using constraint programming [35]–[37], itemset/rulelist mining [3] and SAT-based techniques [6], [18], [27], among others. These approaches often allow efficient generation of a *single* interpretation with high correctness measure and satisfying user-provided explainability constraints. However, no formal guarantees of Pareto-optimality (w.r.t. correctness and explainability) are provided. Furthermore, these techniques do not compute the set of *all* Pareto-optimal interpretations, thereby constraining the choice of which interpretation to use for a given application.

In this paper, we present a novel multi-objective optimization approach for synthesizing Pareto-optimal interpretations of black-box ML components, using an off-the-shelf quantitative constraint solver (weighted MaxSAT solver in

https://doi.org/10.34727/2021/isbn.978-3-85448-046-4_24

our case). For each problem instance, our approach yields a set of interpretations that correspond to *all* Pareto-optimal combinations of correctness and explainability measures. This contrasts sharply with earlier approaches such as [3], [6], [18], [19], [27], [35]–[37] that always yield a single interpretation, leaving the user with no choice of exploring the trade-off between correctness and explainability of alternative interpretations. Similar to existing work, we use syntactic constraints to restrict the class of interpretations over which to search. Unlike earlier approaches, however, we do not combine quantitative correctness and explainability measures into a single optimization objective. Any such mapping of an inherently multi-dimensional optimization problem to the uni-dimensional case results in exclusion of some Pareto-optimal solutions in general. Given that quantitative explainability measures are often just approximations of subjective preferences of the end-user, we believe it is important to present the entire set of Pareto-optimal interpretations, and leave the choice of the "best" interpretation to the user. As our experiments show, there is significant diversity among Pareto-optimal interpretations, and a user aware of this diversity can make an informed choice for a specific application.

The syntactic constraints considered in this paper restrict the space of interpretations to decision diagrams (a generalization of decision trees) with specified bounds on the number of nodes, predicates and branching factors. For simplicity, we let the set of predicates be pre-determined but potentially large, and with possibly different relative preferences for different predicates. We focus on the setting where the black-box ML model can only be treated as an input-output oracle, i.e., given an input, we can observe its output and nothing else. Additionally, we do not have access to training or test data used to create the black-box component. Our correctness measure is therefore based on querying the black-box component with random samples chosen from its input space, where the sample set size is carefully chosen to provide statistical guarantees of near-optimality. Our explainability measure takes into account user preferences of predicates and also size of the interpretation, prefering smaller interpretations over larger ones. The overall framework is, however, general enough to admit other syntactic classes (beyond decision diagrams), and also other correctness and explainability measures.

We have implemented our approach in a prototype tool and applied it to synthesize Pareto-optimal interpretations for some black-box neural network classifiers. Our results exhibit the richness of choices available to the end-user in each case, none of which would be exposed by existing methods that generate only a single optimal interpretation. Indeed, we find that significant improvements in explainability can sometimes be achieved by only a marginal reduction of accuracy.

Our primary contributions can be summarized as follows:

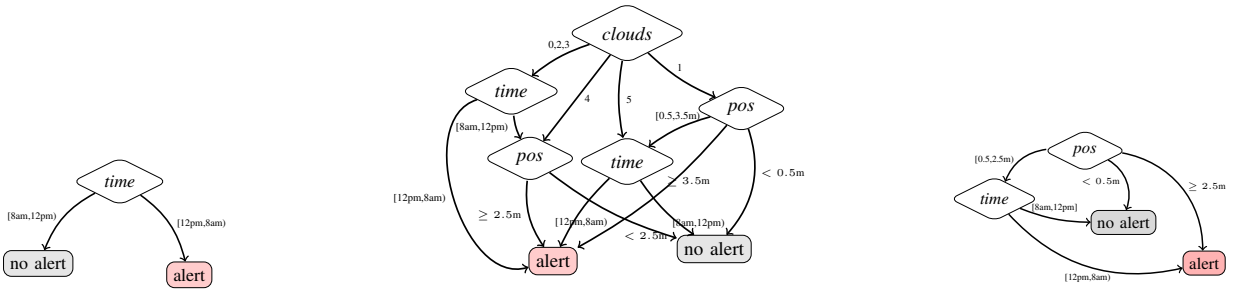
1) We formulate the Pareto-optimal interpretation synthesis problem for black-box ML components.
2) We show that finding a single Pareto-optimal interpretation can be formulated as a weighted MaxSAT problem, for some meaningful choices of correctness and explainability scores.
3) We present a divide-and-conquer algorithm for synthesizing interpretations for *all* Pareto-optimal combinations of correctness and explainability scores.
4) We provide formal guarantees of soundness, completeness and universality of our algorithm, and also statistical guarantees of near-optimality when only a subset of behaviors of a black-box component is sampled.
5) We build a prototype tool and apply it to a collection of black-box neural network classifiers: our results show that significant diversity exists among Pareto-optimal interpretations which earlier tools fail to discover.

II. Motivating Example

We start with an example, adapted from [11], that illustrates the diversity that exists among Pareto-optimal interpretations of black-box ML models. Consider a scenario where an airplane uses a neural network to autonomously taxi along a runway, relying on a camera sensor. Suppose the plane is expected to follow the runway centerline within a tolerance of 2.5 meters. The airplane is equipped with monitoring modules that decide under what circumstances certain learning-enabled components can be trusted to behave correctly. One of these monitoring modules decides under what conditions the camera-based perception module, that determines the distance to the centerline, can be trusted to deliver the right values. For example, the monitoring module may use the weather condition, time of day, and initial positioning of the airplane to decide whether the perception module's output is reliable. We wish to reason about this black-box monitoring module, and hence need an understandable interpretation for it.

Given a set of user-defined predicates (viz. clouds, time of day, and initial position of the plane), the user may favor certain predicates over others, and also favor concise interpretations. By giving favorability weights to each predicate, we can define an explainability score that is related to the number of nodes in the interpretation and also to the predicates used (this is detailed later). The prediction accuracy of an interpretation is measured w.r.t a set of examples sampled from the black box, and is represented by a correctness score. Our approach explores the space of interpretations, searching for concise interpretations that use more favored predicates and also have high accuracy. Clearly, to find a "good" interpretation that meets these conflicting goals, one must explore *all* Pareto-optimal interpretations w.r.t. the criteria above.

Figure 1 shows three of the many Pareto-optimal interpretations our approach synthesized for the monitoring black-box. Each of these has its own pros and cons, and is incomparable with the others. The user can now choose the interpretation that best suits the user's purpose. For example, if interpretation size is not of concern but accuracy is, then Figure 1(b) is the best choice. However, if the user wants concise models with favored predicates (related to time of day and initial position), then Figure 1(a) is the best choice. The user may also choose the interpretation in Figure 1(c), which is only



(a) Pareto-optimal interpretation with correctness measure $c = 0.61$ and explainability measure $e = 0.95$

(b) Pareto-optimal interpretation with correctness measure $c = 0.94$, explainability measure $e = 0.71$

(c) Pareto-optimal interpretation with correctness measure $c = 0.90$ and explainability measure $e = 0.89$

Fig. 1. Pareto-optimal decision diagram interpretations for the black-box monitoring component that decides based on time of day, cloud types, and initial position of an airplane whether to trust a perception module to help the plane track the centerline of a runway. The correctness score is given by the prediction accuracy w.r.t. to the used sample set. The explainability score is the normalized sum of weights of used predicates and unused nodes.

slightly less accurate than that in Figure 1(b), but has a higher explainability score. In fact, Figure 1(c) represents a healthy balance between accuracy and explainability. According to it, the perception module can be trusted only during morning hours if the plane starts no more than 2.5m from the centerline, or at any time if the plane starts within 0.5m of the centerline.

Tools that use a single-objective function to synthesize interpretations can only find one of these Pareto-optimal interpretations, depending on the relative weights given to accuracy and explainability. The rich diversity among Pareto-optimal interpretations is completely missed by such tools, effectively restricting the user's choice of a "good" interpretation.

III. Pareto-optimal Interpretation Synthesis

In this section, we formalize the Pareto-optimal interpretation synthesis problem and present a solution (for specific choices of correctness and explainability scores) using a quantitative constraint satisfaction engine. In our case, this engine is an off-the-shelf weighted maximum satisfiability solver. The key idea is that the user sets syntactic restrictions on the class of considered interpretations as well as quantitative objectives for evaluating the interpretations. The quantitative objectives are defined using two inherently incomparable measures – the explainability measure and the correctness measure. The explainability measure relates to "ease" of understanding of the interpretation by an end-user, while the correctness measure relates to how precisely the interpretation explains the behavior of the black-box model on a given set of samples. Examples of quantitative correctness measures include accuracy, recall, precision, F1-score [34], while examples of explainability measures include those that reward usage of concise interpretations and less complex predicates.

Since our access to the black-box model is only via input/output samples, the correctness measure referred to above is defined with respect to a set of samples, and not with respect to the black-box model in its entirety. While this may appear ad-hoc at first sight, we show in Section IV that rigorous statistical guarantees can indeed be provided with sufficiently many samples.

A. Formal problem definition

We now give a formal definition of the Pareto-optimal interpretation synthesis problem. An interpretation is simply a syntactic structure, viz. decision tree, decision diagram, linear model, etc. We will fix a class of interpretations $\mathcal{E}$ over an input domain $\mathcal{I}$ and output domain $\mathcal{O}$. For an interpretation $E \in \mathcal{E}$, we define $f_E \in (\mathcal{I} \to \mathcal{O})$ to be the semantic function that is computed by E. Note that different interpretations may compute the same semantic function.

Every interpretation $E \in \mathcal{E}$ is associated with a pair of real-valued measures (c, e), where c is the correctness measure and e is the explainability measure of E. We define a partial order $\preceq$ on such pairs as: $(c, e) \preceq (c', e')$ iff $c \leq c'$ and $e \leq e'$. Given a set X of (c, e) pairs, we define $\max^{\preceq} X$ to be the set of $\preceq$-maximal pairs in X. An interpretation E with the pair of measures (c, e) is said to be *Pareto-optimal* if (c, e) is maximal over pairs of measures of all interpretations.

Definition 1 (Pareto-optimal interpretation synthesis): Let $\mathcal{E}$ be a syntactic class of interpretations over inputs $\mathcal{I}$ and outputs $\mathcal{O}$. Further, let $\mathcal{S} \subseteq \mathcal{I} \times \mathcal{O}$ be a set of samples, $\Delta_{\mathcal{C}} \colon (\mathcal{I} \to \mathcal{O}) \times 2^{(\mathcal{I} \times \mathcal{O})} \to \mathbb{R}^{\geq 0}$ be a correctness measure, and $\Delta_{\mathcal{E}} \colon \mathcal{E} \to \mathbb{R}^{\geq 0}$ an explainability measure. The Pareto-optimal interpretation synthesis problem $\langle \mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}, \Delta_{\mathcal{E}} \rangle$ is the multi-objective problem of finding a Pareto-optimal interpretation $E \in \arg\max^{\preceq}_{E' \in \mathcal{E}} (\Delta_{\mathcal{C}}(f_{E'}, \mathcal{S}), \Delta_{\mathcal{E}}(E'))$.

We interpret $\Delta_{\mathcal{C}}(f_E, \mathcal{S})$ as a measure of closeness between the semantic function f_E of interpretation E and the semantic constraints defined by a set $\mathcal{S}$ of samples. An optimally correct interpretation is one with maximal closeness. An example of such a measure is the *prediction accuracy* $\frac{|\{(i,o) \in \mathcal{S} | f_E(i) = o\}|}{|\mathcal{S}|}$. The problem can also be defined in terms of the "distance" between an interpretation and the semantic constraints defined by $\mathcal{S}$, in which case, the optimization problem is one of minimization. An example of such a measure is the *misclassification rate*, which is one minus the prediction accuracy. Similarly, for $\Delta_{\mathcal{E}}(\cdot)$, we choose to define it as a reward function that we want to maximize, but it can also be dually defined as a cost function we want to minimize.

For each $\preceq$-maximal pair of measures, there can be multiple corresponding interpretations realizing the measures. We don't distinguish between them for purposes of this paper. The following definition is therefore relevant.

Definition 2 (Minimal representative set): A set Γ of Pareto-optimal interpretations is a minimal representative set for $\langle \mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}, \Delta_{\mathcal{E}} \rangle$ if for every $(c, e) \in \max^{\preceq}_{E \in \mathcal{E}}(\Delta_{\mathcal{C}}(f_E, \mathcal{S}), \Delta_{\mathcal{E}}(E))$, there is exactly one interpretation $E' \in \Gamma$ such that $(\Delta_{\mathcal{C}}(f_{E'}, \mathcal{S}), \Delta_{\mathcal{E}}(E')) = (c, e)$.
Our goal can therefore be stated as one of finding a minimal representative set of interpretations for a black-box model.

B. Synthesis via weighted maximum satisfiability

We now discuss how to synthesize one (of possibly many) Pareto-optimal interpretation for specific choices of $\mathcal{E}$, $\Delta_{\mathcal{C}}$ and $\Delta_{\mathcal{E}}$, by encoding the synthesis problem as a *weighted maximum satisfiability* problem (weighted MaxSAT). For purposes of our discussion, we choose $\mathcal{E}$ to be the class of *bounded multi-valued decision diagrams*, i.e., decision diagrams with multiple branching at each node, where the branching is governed by decision predicates, and with a bound on the number of decision nodes (see, e.g., diamond nodes in Figure 1). We use prediction accuracy as the correctness measure, and define the explainability measure with weights (denoting preferences) on the predicates and on the number of used nodes. The encoding for several other classes of interpretations, such as decision trees, decision rules, etc. and for other explainability and correctness measures can be done similarly.

We start by recalling the weighted MaxSAT problem. A Boolean formula φ over variables in a set X is said to be in conjunctive normal form (CNF) if φ is of the form $C_1 \wedge C_2 \wedge \cdots C_m$, where each C_i is a disjunction of literals (i.e. variables or negations of variables). An assignment $\sigma \colon X \to \{0, 1\}$ is an assignment of truth values to variables. If a clause C_i evaluates to 1 under σ, we say σ satisfies C_i, denoted by $\sigma \models C_i$.

Definition 3 (Weighted Maximum Satisfiability): Given a Boolean formula $\varphi = \bigwedge_{i=1}^{m} C_i$ in CNF and a weight function $w \colon \{C_1, \ldots C_m\} \to \mathbb{R}^{\geq 0}$ that assigns a non-negative real weight to each clause, the weighted MaxSAT problem is to find an assignment σ which maximizes $\sum_{\{C_i \mid \sigma \models C_i\}} w(C_i)$.
In a variant of the above definition, the clauses in φ are partitioned into *hard* and *soft* clauses. The problem now is to find an assignment σ that satisfies *all hard clauses* and maximizes the sum of weights of satisfied soft clauses. We use this variant for encoding our problem.

At a high level, for an instance $\langle \mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}, \Delta_{\mathcal{E}} \rangle$ of the Pareto-optimal interpretation synthesis problem, we define its encoding as a conjunction of four formulae. Specifically, $\phi_{\langle \mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}, \Delta_{\mathcal{E}} \rangle} = \phi_{\mathcal{E}} \wedge \phi_{\mathcal{S}} \wedge \phi_{\Delta_{\mathcal{C}}} \wedge \phi_{\Delta_{\mathcal{E}}}$ where, (i) $\phi_{\mathcal{E}}$ encodes the syntactic restrictions, i.e., bounded multi-valued decision diagrams with the permitted predicates (features and branchings) and labels; (ii) $\phi_{\mathcal{S}}$ encodes the semantic constraints, i.e., the relation between the samples in $\mathcal{S}$ and an interpretation satisfying $\phi_{\mathcal{E}}$; (iii) $\phi_{\Delta_{\mathcal{C}}}$ encodes the correctness measure, e.g., in case of prediction accuracy it encodes whether an interpretation agrees on a sample; and finally (iv) $\phi_{\Delta_{\mathcal{E}}}$ defines constraints that encode certain structural aspects of an interpretation, e.g., what predicates were chosen and whether a node was used. We discuss some details of these formulas below, leaving the full encoding to the long version of this paper at [31].

a) Encoding of the interpretation class ($\phi_{\mathcal{E}}$): We start by discussing the encoding for our interpretation class of bounded multi-valued decision diagrams over inputs $\mathcal{I}$ and outputs $\mathcal{O}$. These diagrams are restricted by a finite set of decision predicates, denoted by P. For example, in Figure 1(a), the initial node uses the "*time of day*" predicate with branchings: {[8am-12pm], [12pm-8am]}. Let L be a set of output labels, e.g., in Figure 1, we have two labels, "*alert*" and "*no alert*". An *interpretation* $E \in \mathcal{E}$ is a multi-valued decision diagram over a finite set of nodes $\mathcal{N}$, where each internal node corresponds to a decision predicate $p \in P$ and each leaf to an output label $\ell \in L$. Outgoing transitions of a node are labelled according to the branchings of the predicate corresponding to the node. We remark that features are distinct from inputs to the black-box. For example, in the decision diagrams in Figure 1 the feature "*pos*" uses the latitude and longitude inputs to compute the initial position of the plane. Furthermore, the same predicate may appear on different nodes in the decision diagram, but not more than once along a path. For a given P, L, and a bound n on the number of nodes $\mathcal{N}$ in the decision diagram, the formula $\phi_{\mathcal{E}}$ encodes an acyclic decision diagram of at most n-nodes over a set P of predicates, with leaves labeled by elements of L.

b) Encoding of the samples: The formula $\phi_{\mathcal{S}}$ encodes the relation between the samples and the interpretation $\phi_{\mathcal{E}}$. It uses an auxiliary variable $m_{(i,o)}$ for each sample (i, o) in the set $\mathcal{S}$. Logically, $m_{(i,o)}$ is set to true iff the interpretation given by a satisfying assignment of $\phi_{\mathcal{E}}$ produces the output label o when fed the input i. For decision diagrams, this is encoded by symbolically matching the input i to a decision path in the diagram, and by comparing the value of o with that of the label reached at the end of the decision path. Note that the number of these auxiliary variables grows linearly with the size of the sample set.

c) Encoding the correctness measure ($\phi_{\Delta_{\mathcal{C}}}$): To encode $\Delta_{\mathcal{C}}$, we add a unit soft clause (i.e., a clause with only one literal) $m_{(i,o)}$ for each sample (i, o). By assigning appropriate weights to these unit clauses and by maximizing the sum of weights of satisfied clauses (see Definition 3), we obtain an interpretation that maximizes $\Delta_{\mathcal{C}}$ with respect to the sample set $\mathcal{S}$. E.g., if $\Delta_{\mathcal{C}}$ represents the prediction accuracy, then assigning a weight of 1 to each unit clause $m_{(i,o)}$ gives us an interpretation that agrees on a maximal number of samples in $\mathcal{S}$. If the user is interested in interpretations that agree on certain types of samples, then higher weights should be given to these samples. More precisely, to define such measures $\Delta_{\mathcal{C}}$, the user can provide a function $w \colon \mathcal{I} \times \mathcal{O} \to \mathbb{R}$, that defines these weights. For example, in the case of prediction accuracy, w is the constant function 1.

d) Encoding the explainability measure ($\phi_{\Delta_{\mathcal{E}}}$): To encode $\Delta_{\mathcal{E}}$, we add a unit clause u_γ for each syntactic structure γ of an interpretation in $\mathcal{E}$ and give it a weight according to how

important γ is. For example, in the case of decision diagrams, using some predicates may be more favorable than others. To encode this, we add unit clauses $u_{(i,p)}$ that are set to true iff predicate p is used in node i, and assign higher weights for clauses representing favorable predicates. Moreover, predicates with fewer branches can be favored by using soft clauses with appropriate weights. To further reward the synthesis of decision diagrams with fewer nodes, we can also add unit soft clauses u_i for each node i that is set to true iff node i is not reachable from the root node in an interpretation satisfying $\phi_{\mathcal{E}}$, and give them positive weights. In this case, by maximizing the satisfaction of these clauses, we reward the synthesis of small decision diagrams.

In our weighted MaxSAT formulation, we require that all clauses resulting from a Tseitin encoding (i.e., a transformation into CNF) of the formula $\phi_{\langle \mathcal{E},\mathcal{S},\Delta_{\mathcal{C}},\Delta_{\mathcal{E}}\rangle}$, except the unit soft clauses mentioned above, be hard clauses. On feeding the above formula to a MaxSAT solver, it returns a satisfying assignment giving a concrete instantiation of the decision diagram template that maximizes the sum of weights of $m_{(i,o)}$ and u_γ clauses.

The encoding described above is specific to a particular choice of $\mathcal{E}$, $\Delta_{\mathcal{C}}$ and $\Delta_{\mathcal{E}}$. However, similar encoding can be done for a much wider class of interpretations, and explainability and correctness measures. In fact, most types of interpretation classes used in the literature, viz. decision trees, decision diagrams, decision lists and sets of bounded depth/size admit encoding as Boolean formulas. In addition, if the computation of explainability and correctness measures can be encoded using arithmetic circuits of bounded bit-width, the Pareto-optimal intepretation synthesis problem can be reduced to weighted MaxSAT by assigning appropriate weights to bits in the bit-vector representing the measures. The following theorem applies to our encoding, and to all other similar encodings referred to above.

Theorem 1 (Pareto-optimality): Every solution of the weighted MaxSAT problem $\phi_{\langle \mathcal{E},\mathcal{S},\Delta_{\mathcal{C}},\Delta_{\mathcal{E}}\rangle}$ gives a solution for the Pareto-optimal interpretation synthesis problem $\langle \mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}, \Delta_{\mathcal{E}}\rangle$.

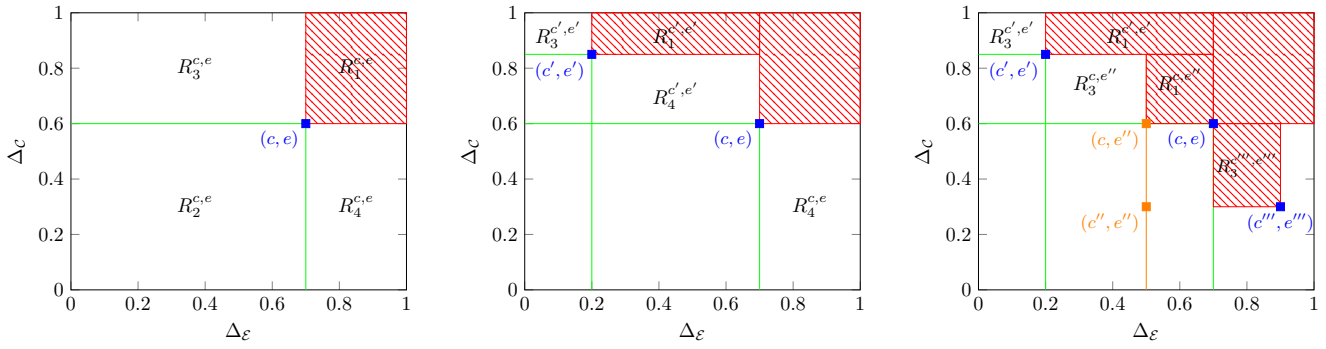
C. Exploring the set of Pareto-optimal interpretations

We now present an algorithm for computing a minimal representative set of Pareto-optimal interpretations. The algorithm is based on the key observation that every Pareto-optimal measure (c, e) splits the space of measures into four regions, depicted in Figure 2(a), (1) a region $R_1^{c,e}$ of measures for which there exists no solution, namely, all measures $(c', e') \neq (c, e)$ with $c' \geq c$ and $e' \geq e$, otherwise (c, e) would not be Pareto-optimal, (2) a region $R_2^{c,e}$ of measures that are not Pareto-optimal, namely, all points $(c', e') \neq (c, e)$ with $c' \leq c$ and $e' \leq e$, (3) a region $R_3^{c,e}$ with measures of potential Pareto-optimal interpretations with better correctness measures, i.e., those with measures (c', e') with $c' > c$ and $e' < e$, and lastly (4) a region $R_4^{c,e}$ with measures of potential Pareto-optimal interpretations with better explainability measures, i.e., points (c', e') with $c' < c$ and $e' > e$. By synthesizing a first Pareto-optimal interpretation using the procedure from last section, and then dividing the search space into corresponding regions (1)-(4), our algorithm proceeds by searching for further Pareto-optimal interpretations with better correctness in region (3) and better explainability in region (4). This process is repeated for every Pareto-optimal interpretation found by our algorithm, thus, directing the search into smaller and smaller regions until no new Pareto-optimal interpretation can be found.

This is detailed in Algorithm 1 and the exploration process it implements is illustrated in Figure 2. For $\mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}$, and $\Delta_{\mathcal{E}}$, Algorithm 1 returns a minimal representative set Γ of interpretations for all Pareto-optimal measures. To synthesize a Pareto-optimal interpretation within a given region of measures, Algorithm 1 relies on the procedure QuIntSynt which given $\mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}$, and $\Delta_{\mathcal{E}}$, in addition to a lower-bound $\delta_{\mathcal{E}}^l$ and upper-bound $\delta_{\mathcal{E}}^u$ on the explainability measure, returns a Pareto-optimal interpretation E with explainability measure e such that $\delta_{\mathcal{E}}^l \leq e \leq \delta_{\mathcal{E}}^u$. QuIntSynt effectively solves an extension of the weighted MaxSAT instance defined in the last section, in which we additionally require the explainability measure to satisfy the constraints given by the lower-bound $\delta_{\mathcal{E}}^l$ and upper-bound $\delta_{\mathcal{E}}^u$. This can be done by extending the formula ϕ in the last section with a fifth conjunct $\phi_{\delta_{\mathcal{E}}^l, \delta_{\mathcal{E}}^u}$. This conjunct is satisfied if the sum of weights of the used syntactic structures (e.g. in the case of decision diagrams, this will be sum of weights of the satisfied clauses $u_{(i,p)}$ and u_i) lies within the given bounds. We leave details of this encoding to [31], but intuitively, we encode a binary adder that sums up the weights of satisfied $u_{(i,p)}$ and u_i clauses and compare the results to binary encodings of the bounds. To fix the number of bits to encode both the adder and bounds, we normalize the weights to values between 0 and 1 up to a certain floating-point precision k. Now let us go further into Algorithm 1 while elaborating on why it suffices to only bound the explainability measure when exploring regions (3) and (4) depicted in Figure 2(a).

Initially, Algorithm 1 explores the entire set of Pareto-optimal solution space. To this end, the exploration set W is initialized with the point $(0, 1, 0)$ (line 2) defining a lower bound on the explainability measure, an upper-bound on the explainability measure, and a lower-bound on the correctness measure, respectively. For every point $(\delta_{\mathcal{E}}^l, \delta_{\mathcal{E}}^u, \delta_{\mathcal{C}})$ in W, QuIntSynt synthesizes a Pareto-optimal region within the explainability measure bounds defined by $\delta_{\mathcal{E}}^l$ and $\delta_{\mathcal{E}}^u$ (line 5). If an interpretation E is found with measures c and e, i.e., $E \neq \perp$ (line 6), the algorithm further divides the search space based on the following case distinction:

- if $c > \delta_{\mathcal{C}}$, then a new Pareto-optimal interpretation with measures (c, e) is found and the regions $R_3^{c,e}$ and $R_4^{c,e}$ defined by the points $(\delta_{\mathcal{E}}^l, \downarrow e, c)$ and $(\uparrow e, \delta_{\mathcal{E}}^u, \delta_{\mathcal{C}})$, respectively, are added to W (lines 9 and 10). The operators $\downarrow$ and $\uparrow$define the predecessor and successor value of the value e (we assume that the values are discrete and hence the predecessor and successor exist). For example, if the interpretation synthesized by QuIntSynt is one with measures c', e' as depicted in Figure 2(b), then the region



(a) First iteration: Exploring region defined by bounds $(0, 1, 0)$. Expand W with new regions $R_3^{c,e}$ and $R_4^{c,e}$ by adding the points $(0, \downarrow e_0, c_0)$ and $(\uparrow e_0, 1, 0)$. No Pareto-optimal points exist in the red region.

(b) Exploring the region $R_3^{c,e}$. A new Pareto-optimal interpretation is found with measures (c', e'). Add the points $(0, \downarrow e', c')$ and $(\uparrow e', \downarrow e, c)$ to W.

(c) Exploring region $R_4^{c',e'}$. Optimal interpretation had correctness measure $c'' < c$. Exclude region $R_1^{c,e''}$ and add new region defined by $(\uparrow e', \downarrow e'', c)$ to W. For another Pareto-optimal point (c''', e'''), no solution found when exploring its region $R_3^{c''',e'''}$.

Fig. 2. An illustration of Algorithm 1.

Algorithm 1 EXPLOREPOI

Input: $\mathcal{E}$, $\mathcal{S}$, $\Delta_{\mathcal{C}}$, $\Delta_{\mathcal{E}}$
Output: Minimal representative set Γ for $\langle \mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}, \Delta_{\mathcal{E}} \rangle$

1: $\Gamma := \emptyset$
2: $W := \{(0, 1, 0)\}$
3: **while** $W \neq \emptyset$ **do**
4: $\quad (\delta_{\mathcal{E}}^l, \delta_{\mathcal{E}}^u, \delta_{\mathcal{C}}) :=$ pop(W)
5: $\quad (E, (c, e)) =$ QUINTSYNT$(\mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}, \Delta_{\mathcal{E}}, \delta_{\mathcal{E}}^l, \delta_{\mathcal{E}}^u)$
6: $\quad$ **if** $E \neq \perp$ **then**
7: $\quad\quad$ **if** $c > \delta_{\mathcal{C}}$ **then**
8: $\quad\quad\quad \Gamma := \Gamma \cup \{(E, (c, e)\}$
9: $\quad\quad\quad$ push$(W, (\delta_{\mathcal{E}}^l, \downarrow e, c))$
10: $\quad\quad\quad$ push$(W, (\uparrow e, \delta_{\mathcal{E}}^u, \delta_{\mathcal{C}}))$
11: $\quad\quad$ **else**
12: $\quad\quad\quad$ push$(W, (\delta_{\mathcal{E}}^l, \downarrow e, \delta_{\mathcal{C}}))$
13: $\quad\quad$ **end if**
14: $\quad$ **end if**
15: **end while**
16: **return** Γ

$R_4^{c',e'}$ is be captured by the point $(\uparrow(e'), \downarrow(e), c)$. The region $R_3^{c',e'}$ is captured by $(0, \downarrow(e'), c')$. Notice that we do not need to include an upper bound on the correctness measure as it is already implicitly defined by the $R_1^{c,e}$ region of any Pareto-optimal point (c, e). For example, in Figure 2(b) the upper bound on the correctness for region $R_4^{c',e'}$ is already captured through the fact that no Pareto-optimal solutions exist in $R_1^{c',e'}$.

- if $c \leq \delta_{\mathcal{C}}$, then (c, e) cannot be Pareto-optimal, because we already know that there is a Pareto-optimal interpretation with measures $(\delta_{\mathcal{C}}, \uparrow \delta_{\mathcal{E}}^u)$. In this case, we can exclude the search in the region $R_1^{\delta_{\mathcal{C}},e}$, because if there was any Pareto-optimal interpretation with measures $(\hat{c}, \hat{e})$ in $R_1^{\delta_{\mathcal{C}},e}$, then QUINTSYNT would have found this interpretation. Thus, Algorithm 1 further prunes the search region to a smaller region defined by $(\delta_{\mathcal{E}}^l, \downarrow e, \delta_{\mathcal{C}})$ (line 12). For example, if Algorithm 1 used QUINTSYNT to synthesize an interpretation from $R_4^{c',e'}$, and returned a solution with measures (c'', e'') as depicted in Figure 2(c), then we can exclude the search in region $R_1^{c,e''}$ and add the region $R_3^{c,e''}$ to W.

Lastly, if QUINTSYNT returns no interpretation, then we can immediately exclude the searched region from further exploration and thus no new points are added to W in this case. For example, as shown in Figure 2(c), if QUINTSYNT found no Pareto-optimal interpretations in $R_3^{c''',e'''}$, then this region is excluded from the search and Algorithm 1 continues with the next available point in W.

Next we show some important properties of Algorithm 1.

Lemma 1 (Soundness): For an instance $\langle \mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}, \Delta_{\mathcal{E}} \rangle$ of the Pareto-optimal interpretation synthesis problem, if $(E, (c, e)) \in$ EXPLOREPOI$(\mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}, \Delta_{\mathcal{E}})$, then $(c, e) \in \max_{E' \in \mathcal{E}}^{\preceq} (\Delta_{\mathcal{C}}(f_{E'}, \mathcal{S}), \Delta_{\mathcal{E}}(E'))$.

In the rest of this section, we assume that each of the explainability measures has finitely many discrete values, as they are defined as floating points up to a certain precision. Thus, we obtain that the range of $\Delta_{\mathcal{E}}$ is finite, which allows us to obtain the following results.

Lemma 2 (Completeness): For an instance $\langle \mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}, \Delta_{\mathcal{E}} \rangle$ of the Pareto-optimal interpretation synthesis problem, if $(c, e) \in \max_{E' \in \mathcal{E}}^{\preceq} (\Delta_{\mathcal{C}}(f_{E'}, \mathcal{S}), \Delta_{\mathcal{E}}(E'))$, then there is an interpretation E with measures (c, e) such that $(E, (c, e)) \in$ EXPLOREPOI$(\mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}, \Delta_{\mathcal{E}})$.

We summarize the correctness result next which follows immediately from Lemmas 1 and 2.

Theorem 2 (Correctness of Algorithm 1): For a class of interpretations $\mathcal{E}$, a finite set of samples $\mathcal{S}$, and measures $\Delta_{\mathcal{C}}$ and $\Delta_{\mathcal{E}}$, the algorithm EXPLOREPOI terminates and returns a minimal representative set for $(\mathcal{E}, \mathcal{S}, \Delta_{\mathcal{C}}, \Delta_{\mathcal{E}})$.

Algorithm EXPLOREPOI solves the interpretation synthesis problem as a multi-objective optimization problem. If we were to solve the same problem using single-objective optimization, it would be necessary to combine the accuracy and explainability measures for every interpretation to yield a single hybrid measure. Let $\lambda : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a function that yields such a

measure. Since higher values of c and e always increase the desirability of an interpretation, we require λ to be *strictly increasing*, i.e., $(c,e) \prec (c',e') \implies \lambda(c,e) < \lambda(c',e')$. For example, $\lambda(c,e) = w_1 \cdot c + w_2 \cdot e$ is a strictly increasing function for every $w_1, w_2 > 0$. Then, for any (c,e) pair that is maximal wrt such a function λ, our algorithm can find an interpretation with this measure pair. Formally,

Theorem 3 (Universality): For every strictly increasing function $\lambda : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and every $\langle \mathcal{E}, \mathcal{S}, \Delta_\mathcal{C}, \Delta_\mathcal{E} \rangle$ if $E \in \arg\max_{E' \in \mathcal{E}}(\lambda(\Delta_\mathcal{C}(f_{E'}, \mathcal{S}), \Delta_\mathcal{E}(E')))$, then there exists an interpretation $E^\star \in \mathcal{E}$ such that (i) $\Delta_\mathcal{C}(f_E, \mathcal{S}) = \Delta_\mathcal{C}(f_{E^\star}, \mathcal{S})$, (ii) $\Delta_\mathcal{E}(E) = \Delta_\mathcal{E}(E^\star)$, and (iii) $(E^\star, (\Delta_\mathcal{C}(f_{E^\star}, \mathcal{S}), \Delta_\mathcal{E}(E^\star))) \in$ EXPLOREPOI$(\mathcal{E}, \mathcal{S}, \Delta_\mathcal{C}, \Delta_\mathcal{E})$.

We conclude the section with some remarks on Algorithm 1.

Remark 1: Algorithm 1 can also be applied interactively as a conversation between synthesizer and user. Given a Pareto-optimal interpretation, the user may guide the search to interpretations that are more explainable or to those with more accuracy, until the user has found an optimal interpretation.

Remark 2: Note that there might be multiple interpretations with the same pair (c,e). In this case, Algorithm 1 will add only one of them as a representative interpretation, since the others are indistinguishable wrt correctness and explainability.

Finally, we can also search for Pareto-optimal solutions based on regions solely bounded on the correctness measure. We choose to use bounds on the explainability measure, because the sample sets tend to be large and will result in much larger encodings.

IV. Statistical Guarantees for Black-box Models

In Section III, the correctness of an interpretation E, defined using a measure $\Delta_\mathcal{C}$, was determined with respect to a set of samples $\mathcal{S}$ obtained from the black-box model $\mathcal{B}$. Our approach guarantees that E is optimal for $\mathcal{S}$ and the measure $\Delta_\mathcal{C}$. Our ultimate goal, however, is to synthesize an interpretation E that is optimal with respect to the entire black-box model $\mathcal{B}$, i.e., w.r.t. the set $\mathcal{S}_\mathcal{B} = \{(i,o) \mid f_\mathcal{B}(i) = o, i \in \mathcal{I}\}$. Obtaining an exhaustive set of samples from a black-box model is often not practical. The question that we, therefore, raise in this section is: *how large must $\mathcal{S}$ be such that it is not* misleading, *i.e., optimal interpretations synthesized by our approach for $\mathcal{S}$ do not overfit the set, and thus the guarantees obtained over $\mathcal{S}$ can be adopted for $\mathcal{S}_\mathcal{B}$?*

The answer to the above question lies in the theory of *Probably Approximately Correct (PAC) Learnability* [32]. The notion of a *loss function*, ℓ, that must be minimized to obtain an optimal interpretation, is central to this discussion. For our purposes, the loss function may be viewed as $1 - \Delta_\mathcal{C}$, where the range of the (normalized) correctness measure $\Delta_\mathcal{C}$ is assumed to be $[0,1]$. Thus for every $(i,o) \in \mathcal{I} \times \mathcal{O}$, and $f \in \mathcal{I} \to \mathcal{O}$, we define $\ell(f,(i,o)) = 1 - \Delta_\mathcal{C}(f, \{(i,o)\})$. For technical reasons, we also assume that for every set $\mathcal{S}$ of (i,o) samples, we have $\Delta_\mathcal{C}(f,\mathcal{S}) = \frac{\sum_{(i,o)\in\mathcal{S}} \Delta_\mathcal{C}(f,\{(i,o)\})}{|\mathcal{S}|}$. This is true, for example, if $\Delta_\mathcal{C}$ is the prediction accuracy (the loss function being the misprediction rate in this case). Note that in this case, the loss function for the sample set $\mathcal{S}$ is given by $\frac{\sum_{(i,o)\in\mathcal{S}} \ell(f,(i,o))}{|\mathcal{S}|} = 1 - \Delta_\mathcal{C}(f,\mathcal{S})$.

A class of interpretations (or hypotheses) $\mathcal{E}$ over inputs $\mathcal{I}$ and outputs $\mathcal{O}$ is said to be PAC-learnable with respect to the set $Z = \mathcal{I} \times \mathcal{O}$ and a loss function $\ell\colon (\mathcal{I} \to \mathcal{O}) \times Z \to [0,1]$, if there exists a function $m_\mathcal{E}\colon (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property: For every ϵ, $\delta \in (0,1)$ and for every distribution D over Z, when running the learning algorithm on $m \geq m_\mathcal{E}(\epsilon,\delta)$ i.i.d. samples generated by D, the algorithm returns a hypothesis E such that, with probability (confidence) of at least $1-\delta$, $L_D(f_E) - \min_{E' \in \mathcal{E}} L_D(f_{E'}) \leq \epsilon$, where $L_D(f_E) = \mathbb{E}_{z\sim D}[\ell(f_E, z)]$. Furthermore, choosing an interpretation $E \in \mathcal{E}$ that minimizes $\frac{\sum_{z\in\mathcal{S}} \ell(f_E,z)}{|\mathcal{S}|}$ suffices for the learning algorithm in the above definition [32].

It is known that every finite class of interpretations is PAC-learnable due to the uniform convergence property [32]. In fact, the sample complexity, i.e., the function $m_\mathcal{E}$, can be determined in such cases in terms of $|\mathcal{E}|$, δ and ϵ. Under the standard *realizability assumption*, i.e assuming $\mathcal{E}$ includes an interpretation E such that f_E implements the semantic function $f_\mathcal{B}$ of the black-box, $m_\mathcal{E}$ is bounded above by $\lceil \frac{\log(|\mathcal{E}|/\delta)}{\epsilon} \rceil$. This bound increases to $\lceil \frac{2\log(2|\mathcal{E}|/\delta)}{\epsilon^2} \rceil$ if we do not make the realizability assumption [32].

From the results above, if we use the $m_\mathcal{E}$ bound for the sample size, we get interpretations that are very close to the optimal interpretation within the class $\mathcal{E}$ with high probability. Of course, sans the realizability assumption, this does not necessarily mean the obtained interpretation is very close to the black-box model. The latter depends highly on the class of interpretations. Note also that the price for the PAC guarantee is that we may have to work with an increased size of the sample set $\mathcal{S}$, as given by $m_\mathcal{E}$. In general, this affects the scalability of our synthesis procedure, since size of the weighted MAXSAT formula increases linearly with $|\mathcal{S}|$. This can limit how small δ and ϵ can be in practice. Nevertheless, as we show in Section V, we are able to use fairly small values of δ and ϵ in our experiments.

V. Evaluation

a) Benchmarks: We apply our approach to three black-box models: a *decision module* for predicting the performance of a perception module in an airplane (AP), a *bank loan predictor* (BL), and a *solvability predictor* (TP).

The decision module predicts, based on the time of day, the cloud types, and initial positioning of an airplane on a runway, whether a perception module used by the plane can be trusted to behave correctly. The decision module is an implementation of a decision tree that was trained on data collected from 200 simulations, using the XPlane (x-plane.org) simulator.

The bank loan predictor is a deep neural network that was trained on synthetic data that we created. The training set included 100000 entries chosen such that majority of people with age between 18 to 29 years, and those with age between 30 and 49 years but with income less than $6000, were denied the loan. The network has five dense fully connected hidden

layers with 200 ReLU's each, in addition to a softmax layer and the output layer comprised of two nodes.

The solvability predictor is a neural network built to predict the solvability of first-order formulas by a theorem prover with respect to percentage of unit clauses and average clause length in a formula. The network had three hidden dense fully connected layers each with 200 ReLU's. The data used to train the neural network can be found on the UCI machine learning repository [8]. We used the data for heuristic H1 from [8], thus predicting solvability for H1.

b) Experiments and setup: We conducted two types of experiments: (1) application of our exploration algorithm on the three benchmarks (2) performance evaluation of QUINTSYNT. The MaxSAT engine used an implementation of RC2 in PySAT [16], [17]. All experiments were conducted on a 2.4GHz Quad-core machine with 8GB of RAM. For additional details of the experiments and results, please see [31].

c) Exploring the Pareto-optimal space: We ran our approach on the three benchmarks mentioned above. We used confidence measure $\delta = 0.05$ and error margin $\epsilon = 0.05$ to determine the size of the sample set (as given in Table I) under the realizability assumption referred to in Section IV. Figures 3(a) to 3(c) show the measures of the Pareto-optimal interpretations found by our exploration algorithm. We used prediction accuracy for correctness (recall this satisfies the technical assumption mentioned in Section IV), and an explainability measure that favored decision diagrams of smaller size with predicates having a fewer number of branchings.

For all three benchmarks we found a variety of interpretations with interesting tradeoffs between the correctness and explainability measures, reflected by the blue squares in each plot. The exploration algorithm shows that searching for interpretations that are optimal only in size or in accuracy may result in unfavorable solutions. For example, in Figure 3(a) we see that the interpretation with highest accuracy has very low explainability. However, a very small tradeoff in accuracy resulted in significantly more explainable interpretations.

d) Performance: Table I presents our results on each benchmark and gives the confidence value δ, error rate ϵ and the number of samples $|S|$ used for each run. The number of Pareto-optimal points (PO), total number of points explored (TNP) and minimum, maximum and median times to find a Pareto-optimal interpretation are also shown. The number shown in parenthesis next to each benchmark is the number of predicates used. From Table I we can see that the number of Pareto-optimal (PO) points is considerably smaller than the total number of points explored (TNP). The minimum time taken to find an interpretation was less than 3 seconds for all benchmarks, but there were a few points in the Pareto-optimal space where finding an interpretation took considerably more time (see the maximum times). For most Pareto-optimal points though, the time taken to the find an interpretation was less than 20 seconds, as demonstrated by the median values. If an interpretation did not exist for a combination of correctness and explanability measures, the MaxSAT solver returned UNSAT in less than a second in all performance runs.

TABLE I
PERFORMANCE OF QUINTSYNT: EXPLORATION OF THE ENTIRE PARETO-OPTIMAL SPACE

Bench mark	δ,ϵ	$\lvert\mathcal{S}\rvert$	Explored (PO, TNP)	min time (s)	max time (s)	median time (s)	unsat time (s)
Theorem Prover (6)	0.05, 0.05	338	4, 20	0.767	3.392	1.138	< 1
	0.05, 0.03	703	3, 28	2.051	18.148	3.643	< 1
Air plane (3)	0.05, 0.05	333	7, 25	1.709	388.527	5.696	< 1
	0.05, 0.03	555	5, 26	2.513	616.520	11.222	< 1
Bank Loan (4)	0.05, 0.05	365	7, 27	1.927	387.599	8.975	< 1
	0.05, 0.03	608	4, 27	2.855	1299.196	17.998	< 1

As none of the other interpretation synthesis tools in the literature compute the set of all Pareto optimal interpretations, we omit comparison with other tools (any such comparison wouldn't be fair, especially when using different notions for explainability). However, to understand if the variation in running times is inherent to the problem, we performed a similar experiment with MinDS, a tool for learning decision sets [38]. In MinDS, correctness and explainability are combined in a single objective and the contribution of the explainability measure is governed by a parameter λ. We ran MinDS for 15 values of λ and found interpretations for all these values. We observed again (Table II) that the time taken to find interpretations for some λ was much more than others.

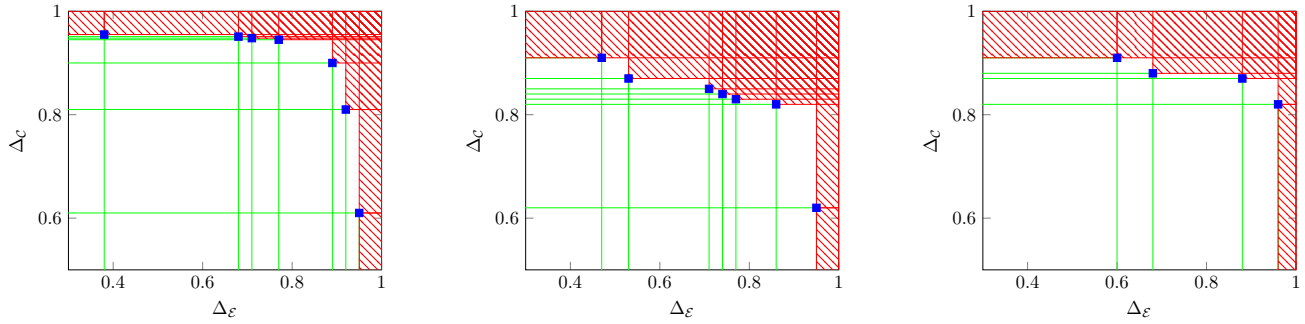
Note that unlike in our approach, running MinDS in this manner does not guarantee that the entire Pareto-optimal space of interpretations has been obtained. Finding all Pareto optimal points by varying the weights of explainability and correctness measures is also not feasible, since this requires trying out all (infinitely many) weight combinations. While some decision sets learned by MinDS were indeed semantically equivalent to some of the Pareto-optimal interpretations synthesized by our approach, some interpretations that our methods found did not have a decision set counterpart within the range of weights we experimented on. We emphasize that running approaches like MinDS that combine explainability and correctness measures into single objective function may result in the same interpretation being returned for different combinations of weights. This can be avoided using our exploration method.

TABLE II
ILLUSTRATING VARIATION IN RUNNING TIMES EVEN ON NON-EXHAUSTIVE PARETO SEARCH WITH MINDS

Bench mark	δ,ϵ	$\lvert\mathcal{S}\rvert$	min time (s)	max time (s)	median time (s)
Theorem Prover (6)	0.05, 0.05	338	0.707	0.813	0.719
	0.05, 0.03	703	0.687	0.798	0.725
Air plane (3)	0.05, 0.05	333	0.771	364.456	7.603
	0.05, 0.03	555	0.748	757.639	9.687
Bank Loan (4)	0.05, 0.05	365	0.744	25.819	1.165
	0.05, 0.03	608	0.738	52.388	0.841

VI. RELATED WORK

There is a large body of work on interpreting black-box models, where a dominant paradigm is to generate labeled data samples and obtain an interpretable model representation in terms of input features, some of which were discussed in the introduction. In some applications, the aim is to explain the



(a) Pareto-optimal solution for airplane perception module benchmark. Used decision diagrams of size 7 over 3 different predicates.

(b) Pareto-optimal solution for the bank loan predictor benchmark. Used decision diagrams of size 7 over 4 different predicates.

(c) Pareto-optimal solution for the theorem prover solvability benchmark. Used decision diagrams of size 7 over 6 different predicates.

Fig. 3. Exploring Pareto-optimal solutions for three benchmarks. The size of the sample sets used for constructing interpretations was computed based on confidence values $\delta = 0.05$ and error margin $\epsilon = 0.05$, as well as the size of the class of interpretation in each benchmark.

output of a black-box model in the neighbourhood of a specific input, and specialized techniques [12], [24], [29], [30], [39] give such local and robust explanations. Other applications use techniques like model distillation (in the form of decision trees [7], [9], [20], [22], [23]), counterfactual explanations [26] etc. For further information on these techniques, we refer to reader to the excellent surveys in [2], [13].

The work in [15], [38] comes closest to ours. In [38], the authors encode the problem of finding an interpretation as optimal decision sets (to a weighted MAXSAT formulation). They present two variants: (i) optimize on accuracy (100%) while constraining the explanability (number of literals), and (ii) directly minimize the size of decision sets at the cost of accuracy. In [15], sparse optimal decision trees are built using an objective function that combines misclassification rate and number of leaves. Solution approaches like these give a single point of the optimized function in the Pareto-optimal space and hence a single value for the correctness and explainability measures.

Our Pareto-optimal interpretation synthesis problem formulation can also be related to Structural Risk Minimization (SRM), which is well-studied in the literature. Like in SRM, we have two orthogonal measures – one that depends only on the structure/complexity of the hypothesis/interpretation, and the other that depends on how well the hypothesis/interpretation "explains" the given sample set. The SRM formulation (e.g., see [32], Section 7.2) effectively combines these two measures into one and treats the problem as a single-objective optimization problem. In contrast, our Pareto-optimal synthesis problem is inherently a multi-objective optimization problem. As mentioned in the introduction, such a multi-objective optimization problem cannot be reduced to a single-objective optimization problem in general, without potentially excluding some (possibly important) solutions.

Finally, we note that the idea of using SAT (and related) solvers for systematically searching for all Pareto-optimal points has been used in other settings earlier (see, for example, systems biology applications in [4], [14]). However, their use in finding Pareto-optimal interpretations for black-box ML components appears not to have been explored earlier.

VII. Conclusion and Future Work

We have presented a new approach to automatically generate a complete set of Pareto-optimal interpretations for black-box ML models, which works in the absence of training or test data sets. Our interpretations are obtained by instantiating user-provided decision diagram templates, and satisfy optimality conditions, while also providing formal guarantees on the tradeoff between accuracy and explainability. We have presented an empirical evaluation demonstrating that our approach produces compact, accurate explanatory interpretations for neural networks used for applications such as autonomous plane taxiing, predicting bank loans and classifying theorem-provers. The discovery of multiple Pareto-optimal interpretations, as opposed to a single one, demonstrates the value of the multi-objective approach.

The current work focuses on finite classes of possible interpretations, although we allow a class to be combinatorially large. The weighted MAXSAT encoding allows us to solve this problem symbolically by leveraging significant recent advances in MaxSAT solving that scale to very large solution spaces. Using a finite, yet large hypothesis class permits us to strike a balance between generality and practical efficiency of our approach. An interesting avenue for futurework would be to see if our approach can be extended to interpretation classes of infinite cardinality but finite Vapnik-Chervonenkis (VC) dimension. While the overall problem formulation, the notions of Pareto-optimality of explanations, and our algorithm for finding representative sets of explanations easily adapt to this setting, we would need to go beyond the current weighted MAXSAT formulation to find individual Pareto-optimal interpretations. Using an optimization modulo theories (OMT) encoding is a promising direction for such a generalization.

Acknowledgments. This work is partially supported by NSF grants 1545126 (VeHICaL), 1646208 and 1837132, by the DARPA contracts FA8750-18-C-0101 (AA) and FA8750-20-C-0156 (SDCPS), by Berkeley Deep Drive, and by Toyota under the iCyPhy center. We would also like to express our gratitude to the anonymous reviewers for their in-depth reviews, constructive suggestions and various pointers.

REFERENCES

[1] General Data Protection Regulation (GDPR). https://gdpr.eu/, 2018.

[2] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[3] Gaël Aglin, Siegfried Nijssen, and Pierre Schaus. Learning Optimal Decision Trees Using Caching Branch-and-Bound Search. In *AAAI 2020*, pages 3146–3153. AAAI Press, 2020.

[4] S. Akshay, Sukanya Basu, Supratik Chakraborty, Rangapriya Sundararajan, and Prasanna Venkatraman. Functional Significance Checking in Noisy Gene Regulatory Networks. In *Principles and Practice of Constraint Programming*, pages 767–785, 2019.

[5] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology*, 2015.

[6] Florent Avellaneda. Efficient Inference of Optimal Decision Trees. In *AAAI 2020*, pages 3195–3202. AAAI Press, 2020.

[7] Olcay Boz. Extracting Decision Trees from Trained Neural Networks. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, New York, NY, USA, 2002. Association for Computing Machinery.

[8] James P. Bridge, Sean B. Holden, and Lawrence C. Paulson. Machine Learning for First-Order Theorem Proving - Learning to Select a Good Heuristic. *J. Autom. Reasoning*, 53(2):141–172, 2014. https://archive.ics.uci.edu/ml/datasets/First-order+theorem+proving.

[9] Mark W. Craven and Jude W. Shavlik. Extracting Tree-Structured Representations of Trained Networks. In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, NIPS'95, page 24–30, Cambridge, MA, USA, 1995. MIT Press.

[10] George E Dahl, Jack W Stokes, Li Deng, and Dong Yu. Large-scale malware classification using random projections and neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3422–3426. IEEE, 2013.

[11] Daniel J. Fremont, Johnathan Chiu, Dragos D. Margineantu, Denis Osipychev, and Sanjit A. Seshia. Formal analysis and redesign of a neural network-based aircraft taxiing system with VerifAI. In *32nd International Conference on Computer Aided Verification (CAV)*, July 2020.

[12] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local Rule-Based Explanations of Black Box Decision Systems. *CoRR*, abs/1805.10820, 2018.

[13] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.*, 51(5), August 2018.

[14] Friedman A. M. He L. and Bailey-Kellogg C. A divide-and-conquer approach to determine the Pareto frontier for optimization of protein engineering. *Proteins*, 80(3):790–806, 2012.

[15] Xiyang Hu, Cynthia Rudin, and Margo Seltzer. Optimal Sparse Decision Trees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[16] Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. PySAT: A Python toolkit for prototyping with SAT oracles. In *SAT*, pages 428–437, 2018.

[17] Alexey Ignatiev, António Morgado, and João Marques-Silva. RC2: an efficient MaxSAT solver. *J. Satisf. Boolean Model. Comput.*, 11(1):53–64, 2019.

[18] Mikolás Janota and António Morgado. SAT-Based Encodings for Optimal Decision Trees with Explicit Paths. In Luca Pulina and Martina Seidl, editors, *Theory and Applications of Satisfiability Testing - SAT 2020*, volume 12178 of *Lecture Notes in Computer Science*, pages 501–518. Springer, 2020.

[19] Susmit Jha, Tuhin Sahai, Vasumathi Raman, Alessandro Pinto, and Michael Francis. Explaining AI Decisions Using Efficient Methods for Learning Sparse Boolean Formulae. *J. Autom. Reasoning*, 63(4):1055–1075, 2019.

[20] U. Johansson and L. Niklasson. Evolving decision trees using oracle guides. In *2009 IEEE Symposium on Computational Intelligence and Data Mining*, pages 238–244, 2009.

[21] Eric Knorr. How PayPal beats the bad guys with machine learning. http://www.infoworld.com/article/2907877/machine-learning/how-paypal-reduces-fraud-with-machine-learning.html, 2015.

[22] R. Krishnan, G. Sivakumar, and P. Bhattacharya. Extracting decision trees from trained neural networks. *Pattern Recognition*, 32(12):1999 – 2009, 1999.

[23] Sanjay Krishnan and Eugene Wu. PALM: Machine learning explanations for iterative debugging. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, HILDA'17, New York, NY, USA, 2017. Association for Computing Machinery.

[24] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[25] Douglas Merrill. AI is coming to take your mortgage woes away. https://www.forbes.com/sites/douglasmerrill/2019/04/04/ai-is-coming-to-take-your-mortgage-woes-away/, April 2019.

[26] Christoph Molnar. *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/.

[27] Nina Narodytska, Alexey Ignatiev, Filipe Pereira, and João Marques-Silva. Learning Optimal Decision Trees with SAT. In Jérôme Lang, editor, *International Joint Conference on Artificial Intelligence, IJCAI 2018*. ijcai.org, 2018.

[28] NVIDIA. Nvidia tegra drive px: Self-driving car computer, 2015.

[29] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Knowledge Discovery and Data Mining*, KDD '16. Association for Computing Machinery, 2016.

[30] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-Precision Model-Agnostic Explanations. In *AAAI Conference on Artificial Intelligence*, 2018.

[31] Hazem Torfah Shetal Shah, Supratik Chakraborty, S. Akshay, and Sanjit A. Seshia. Synthesizing pareto-optimal interpretations for black-box models. *CoRR arXiv*, abs/2108.07307, 2021.

[32] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014.

[33] Justin Sirignano, Apaar Sadhwani, and Kay Giesecke. Deep learning for mortgage risk, 2016.

[34] Pang-Ning Tan, Michael S. Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[35] Hélène Verhaeghe, Siegfried Nijssen, Gilles Pesant, Claude-Guy Quimper, and Pierre Schaus. Learning Optimal Decision Trees using Constraint Programming (extended abstract). In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4765–4769. ijcai.org, 2020.

[36] Sicco Verwer and Yingqian Zhang. Learning Decision Trees with Flexible Constraints and Objectives Using Integer Optimization. In Domenico Salvagnin and Michele Lombardi, editors, *Integration of AI and OR Techniques in Constraint Programming*, pages 94–103, Cham, 2017. Springer International Publishing.

[37] Sicco Verwer and Yingqian Zhang. Learning Optimal Classification Trees Using a Binary Linear Program Formulation. In *AAAI 2019*, pages 1625–1632. AAAI Press, 2019.

[38] Jinqiang Yu, Alexey Ignatiev, Peter J. Stuckey, and Pierre Le Bodic. Computing Optimal Decision Sets with SAT. In *Principles and Practice of Constraint Programming*, pages 952–970, Cham, 2020. Springer International Publishing.

[39] Xin Zhang, Armando Solar-Lezama, and Rishabh Singh. Interpreting Neural Network Judgments via Minimal, Stable, and Symbolic Corrections. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4874–4885. Curran Associates, Inc., 2018.