Safe Partial Diagnosis from Normal Observations

Roni Stern[1] and **Brendan Juba**[2]
[1] Dept. of Software and Information Systems Engineering, Ben Gurion University of the Negev, Israel
e-mail: roni.stern@gmail.com
[2]Dept. of Computer Science and Engineering, Washington University in St. Louis, USA
e-mail: bjuba@wustl.edu

Abstract

Model-based diagnosis (MBD) is difficult to use in practice because it requires a model of the diagnosed system, which is very hard to obtain. We explore theoretically how observing the system when it is in a normal state can provide sufficient information about the system to learn a partial system model that allows automated diagnosis. We analyze the number of observations needed to learn a model capable of finding faulty components in most cases. Then, we explore how knowing the system topology can help us to learn a useful model from the normal observations for settings in which many of the internal system variables cannot be observed. Unlike other data-driven methods, our learned model is safe, in the sense that subsystems identified as faulty are guaranteed to truly be faulty.

1 Introduction

Model-based diagnosis (MBD) is an approach for automated diagnosis in which a *model* that describes the diagnosed system's expected behavior is assumed to be given as input. This model is used to infer possible explanations – *diagnoses* – for an observed abnormal system behavior. MBD is a well-established, principled, approach for automated diagnosis that has been studied in the Artificial Intelligence literature for decades, and has been applied in practice [1; 2; 3]. However, a key inhibitor to its widespread use is that obtaining a model of the diagnosed system is often impossible or prohibitively expensive.

Spectrum-based fault localization (SFL) [4; 5] provides an alternative approach for diagnosis that does not require modeling the system behavior. SFL considers the observations collected at the time when the system has failed, and finds diagnoses by "discovering statistical coincidences between system failures and the activity of the different parts of a system" [6]. While SFL is lightweight, its diagnostic accuracy is inherently limited compared to model-based approaches. Moreover, SFL approaches do not improve over time, and the number of required observations needed to obtain accurate diagnoses is currently not fully understood.

Learning approaches for diagnosis [7; 8; 9] learn from observations of past failures how to map various observed features to the correct diagnosis. These methods usually rely on having a set of abnormal observations, each associated with its correct diagnosis. Since most systems function properly most of the time, obtaining a large set of abnormal observations and their diagnosis is often very difficult. Moreover, given that such observations are inherently abnormal, the kinds of assumptions underlying data-driven techniques, such as stationarity of the data, are questionable for diagnosis. By contrast, normal observations, i.e., observations in which the system is behaving as planned, are easily obtained in practice. Indeed, systems often collect an abundance of data while operating, to be used for monitoring and optimization purposes. Such data contains valuable information about the (mostly) normal behavior of the monitored system. In this work we explore theoretically how such data can be used for diagnostic purposes.

Our first contribution is a method for learning a partial model from normal observations that can be used for diagnosis. In particular, this partial model enables identifying some components as *surely faulty* and other components as *probably normal*, where surely faulty components must be faulty and probably normal components cannot be in any (minimal) diagnosis. Our learning method is *safe*, in the sense that "surely faulty" components are indeed faulty, in contrast to standard data-driven methods that cannot provide such a guarantee. As can be expected, the effectiveness of this learned model strongly depends on how visible the internal variables of the systems are, and on the number of available normal observations. **Our second contribution** is a statistical analysis of the number of normal observations needed to learn an effective model. This novel form of statistical analysis draws from the well-known probably approximately correct (PAC) framework for analyzing learning algorithms [10; 11; 12]. Our main theoretical result is that under several assumptions, the number of observations needed to learn, with high probability, a model that will identify faults in most cases is only quasi-linear in the number of components. **Our third contribution** is a method for using knowledge about the system topology, in addition to the set of normal observations, to learn a useful and more informed partial model of the system. Here too, we provide theoretical bounds relating the number of observations and properties of the system topology to the probability of learning a model that can identify faulty components in abnormal observations with high probability.

We demonstrate our learning approach on a Boolean circuit from the standard ISCAS '85 benchmark [13], showing that with a single probe and only 256 observations, we are able to find a faulty component in 28% of the cases, and this percentage increases rapidly. Importantly, our ap-

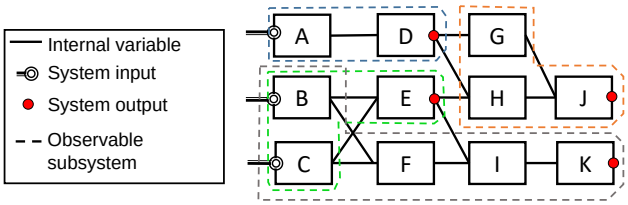


Figure 1: An illustration of a system. Rectangles are components, lines connect outputs to inputs (going from left to right), and circles are the system inputs and outputs. The minimal observable subsystem of each system output is marked with a dashed rectangle.

proach is not specific to Boolean circuits, and it applies to any discrete-valued system, and thus applies to the wide range of systems that can be qualitatively modeled.

2 Background: Model-based Diagnosis

There are three key entities in MBD: the set of system components that comprise the diagnosed system (COMPS), a model of the system (SD), and the observed system behavior (OBS). At a given point in time, each component $c \in$ COMPS is either *healthy* or *faulty*, represented by the predicate $h(c)$, which is true iff c is currently healthy. We focus on systems in which the components perform some function, i.e., each component accepts a set of values as input and produces a set of values as output. The relation between the input values and the output values is referred to as the *behavior* of the component. The normal behavior of c, i.e., the behavior when c is healthy, is denoted by $\varphi_h(c)$ and assumed to be deterministic, i.e., for the same inputs we expect the same outputs. We briefly discuss how to relax this assumption in Section 7.

The *system's topology* is a graph whose nodes are the components and there is an edge from component c_i to c_j iff at least one output of c_i is an input of c_j. We say that a system model is *behavioral* if it contains information about the behavior of the components in addition to the system topology. A common behavior system model in the MBD literature is the *weak fault model* (WFM), in which we have rules capturing the normal behavior ($\varphi_h(\cdot)$) of all of the components. The WFM system model SD_{WFM} is given by:

$$\text{SD}_{\text{WFM}} = \bigwedge_{c \in \text{COMPS}} h(c) \rightarrow \varphi_h(c) \qquad (1)$$

The union of all the components' inputs and outputs represent the set of *system variables*. The values of these variables may change over time. The *system inputs*, denoted SYSINS, are all the components' inputs that are not the output of any component in the system. Thus, the values of the system inputs are set externally by whoever is using the system. The system's *internal variables* are all the components' inputs that are not system inputs, i.e., the inputs whose values are outputs of the system's components. The *system outputs*, denoted SYSOUTS, are the components' outputs that are visible from outside the system. The internal variables and the system outputs are not necessarily disjoint, as there may be internal variables that are also exposed outside of the system. Such internal variables are sometimes called "probes", and are especially useful for diagnosis. An *observation* is an assignment of values to SYSINS and SYSOUTS. The vector of the values of SYSINS and SYSOUTS in an observation obs are denoted by $in(obs)$ and $out(obs)$, respectively. The inputs and outputs of a component c in obs are denoted by $in(c, obs)$ and $out(c, obs)$, respectively.

Figure 1 illustrates a system with 11 components, represented by rectangles, 3 system inputs (the inputs of A, B, and C), represented by empty circles, and 4 system outputs (the outputs of D, E, J, and K), represented by red circles. A line connecting the two components represents that an output of the component on the left is an input to the component on the right. An observation obs is thus an assignment of values to $in(A, obs)$, $in(B, obs)$, $in(C, obs)$, $out(D, obs)$, $out(E, obs)$, $out(J, obs)$, and $out(K, obs)$. The outputs of components D and E are *probes*, since they are system outputs and provide inputs to other components.

A diagnosis problem arises when there is an observation obs_{ab} that indicates the system is behaving abnormally. The task in consistency-based diagnosis is to find a hypothesis about which components are faulty that is consistent with the observation and with our knowledge of the system [14]. Such a hypothesis is called a *diagnosis*, and is defined as follows:

Definition 1 (Diagnosis)**.** *A set of components ω is a diagnosis iff*

$$\bigwedge_{c \in \text{COMPS} \setminus \omega} h(c) \wedge obs_{ab} \wedge \text{SD} \not\models \bot$$

A diagnosis is minimal iff no proper subset of ω is a diagnosis.

MBD algorithms accept SD, COMPS and obs_{ab} as input and output one or more diagnoses. Minimal diagnoses are often preferred, following Occam's razor and due to the potentially exponential number of possible diagnoses.[1]

Definition 2 (Conflict)**.** *A* conflict *is a set of components γ such that*

$$\bigwedge_{c \in \gamma} h(c) \wedge obs_{ab} \wedge SD \models \bot$$

Conflicts are useful in diagnosis because every diagnosis is a hitting set of all conflicts [14]. Conflict-directed MBD algorithms find diagnoses by searching in the space of possible hitting sets of the found conflicts [16; 17; 14]. Roughly speaking, finding conflicts, and in particular minimal ones, helps to speed up conflict-directed MBD algorithms as it reduces the size of their search space.

3 Model-based Diagnosis without a Model

In this work, we study a setting in which the system model SD_{WFM} is not available to the diagnostician. Instead, a set of *normal observations* is available, collected when the system was working properly. Thus, they represent observations of the system when all of its components were healthy. We now show how these observations can help us find faulty components, and provide a statistical analysis of how likely this approach is to succeed.

[1] Definition 1 follows the *consistency-based* approach to diagnosis. In *abductive* diagnosis, the diagnosis is *derived* from (as opposed to only being consistent with) obs_{ab} and SD [15], and is beyond our scope.

3.1 Full Probing

The first setting we analyze is where the output of every component is also a system output, i.e., $\forall c \in$ COMPS we have that $out(c) \in$ SYSOUTS. We refer to this setting as *full probing*, since it corresponds to putting a *probe* after every component. Probing components may be very costly and impractical in many scenarios, but we analyze this setting first as a step towards the partial probing settings analyzed later in this paper.

Let OKS denote the given set of normal observations. The partial model we can learn in the full probing setting from OKS is a conjunction of statements about how each component is expected to behave when healthy. To formalize this, we introduce the predicate $f(c, i, o)$, which represents the following statement: if the input vector to component c is i then its output vector must be o. The partial model we can learn in the full probing setting, denoted SD_{FP}, is given by

$$\bigwedge_{\substack{obs_{OK} \in \text{OKS}, \\ c \in \text{COMPS}}} \Big(h(c) \rightarrow f\big(c, in(c, obs_{OK}), out(c, obs_{OK})\big) \Big) \tag{2}$$

SD_{FP} states that if a component is healthy and it receives the same input values as it had in one of the normal observations, then we expect it to output the same value as it had in that observation. SD_{FP} can be viewed as an *incomplete* version of SD_{WFM}, as follows.

Proposition 1. $SD_{\text{WFM}} \models SD_{FP}$

Proof. SD_{FP} describes the normal behavior of all components for a subset of their possible input values, while SD_{WFM} describes the normal behavior of all components for all possible input values. Thus, for every observation obs and normal observation $obs_{OK} \in$ OKS it holds that $\varphi_h(c) \rightarrow f(c, in(c, obs_{OK}), out(c, obs_{OK}))$. Thus, (1) entails (2) as required. □

One can compute diagnoses and conflicts with respect to this partial model (see Definitions 1 and 2), but will these diagnoses and conflicts be useful? The following general observation, which follows from the definitions of diagnoses and conflicts (Def. 1 and 2), describes the relationship between conflicts and diagnosis of SD_{WFM} and SD_{FP}.

Proposition 2. *For every pair of system models SD and SD' and an abnormal observation obs_{ab}, if $SD \models SD'$ then (1) a conflict w.r.t. SD' is a conflict w.r.t. SD, and (2) a diagnosis w.r.t. SD is a diagnosis w.r.t. SD'.*

A diagnosis w.r.t. SD_{FP} may not be a diagnosis for SD_{WFM}. For example, if we do not have any normal observations, then SD_{FP} is empty, and consequently assuming all components are healthy is a diagnosis w.r.t. SD_{FP} in this case. Clearly, this is not desirable. In general, in this work we consider SD_{WFM} as a reference, and unless stated otherwise use the term diagnosis to refer to diagnoses w.r.t. SD_{WFM}.

Definition 3 (Surely Faulty and Normally Behaving)**.** *A component c has a* surely normal input *in obs_{ab} if $\exists obs \in$ OKS such that $in(c, obs) = in(c, obs_{ab})$. A component c with a surely normal input in obs_{ab} is called* surely faulty *if $out(c, obs_{ab}) \neq out(c, obs)$ and is called* normally behaving *otherwise.*

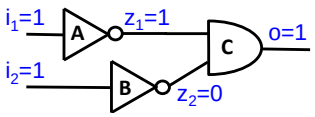


Figure 2: A simple Boolean circuit demonstrating Prop. 3.

That is, a surely faulty component is one that has the same inputs in the abnormal observation obs_{ab} as in some other normal observation $obs \in$ OKS, but has a different output. Since we assume components are deterministic when they are healthy, a surely faulty component c cannot be healthy. Thus, c represents a conflict w.r.t. SD_{FP} and, due to Proposition 2, also w.r.t. SD_{WFM}.

Normally behaving components serve a different purpose. All we know about these components is that they behaved in obs_{ab} just like they did in a normal observation. Faulty components can behave intermittently, i.e., behave as healthy components in some observations [18]. Thus, a normally behaving component may actually be faulty. However, since it followed its normal behavior in obs_{ab}, assuming it is faulty does not help explaining the abnormal system behavior. The following observation summarizes the above discussion.

Proposition 3. *Every diagnosis must contain all of the surely faulty components, and every diagnosis that contains a normally behaving component is not minimal.*

Proposition 4 (At least one surely faulty)**.** *If there is $obs \in$ OKS such that $in(obs) = in(obs_{ab})$ then we will find at least one component that is faulty.*

Proof. Since obs and obs_{ab} have exactly the same input, obs_{ab} is abnormal, and we observe every input and output of every component, then there must be at least one component that has a surely normal input and its output is different from its normal output for that input. Thus, this component is surely faulty. □

Example. Consider the simple Boolean circuit depicted in Figure 2. Now, assume we are given one normal observation $obs = \{i_1 = 1, i_2 = 1, z_1 = 0, z_2 = 0, o = 0\}$ and an abnormal observation $obs_{ab} = \{i_1 = 1, i_2 = 1, z_1 = 1, z_2 = 0, o = 1\}$. The only diagnosis w.r.t. SD_{WFM} is $\{A, C\}$. With the learned model, we can infer that A is surely faulty and B is normally behaving, since both have surely normal inputs. C does not have a surely normal input, so we cannot classify it as either surely faulty or normally behaving. This highlights that while the set of surely faulty components must be faulty, it is not necessarily a diagnosis.

A direct implication of Proposition 4 is that given a sufficient number of normal observations, we will be able to find a faulty component for *every* diagnosis problem.

However, in non-trivial systems, the set of all possible combinations of system input values is extremely large. A key question is how many normal observations are needed to be able to find a faulty component in *most* diagnosis problems. We analyze this in the next section.

Distribution of Normal System Inputs

In the following analysis we focus on faults that are caused by malfunctions in the components themselves, and not due to abnormal system inputs. That is, we assume that the system inputs of obs_{ab} are standard system inputs. Another assumption we make is that the system inputs in each of the

given observations – normal and abnormal – are drawn independently from the same distribution, denoted by D.

In the normal observation, this distribution over the system inputs induces a distribution over the input values for every component c, denoted D_c. Let $P(c, i)$ be the probability of component c getting input i in a normal observation according to D_c. We denote the domain of possible input values for component c by $dom(c, in)$ and the largest domain size by V, i.e., $V = \max_{c \in \text{COMPS}} |dom(c, in)|$.

Lemma 1 (Common Inputs)**.** *For every system with n components and every $\epsilon, \delta \in (0, 1)$, after observing*

$$m \geq \frac{1}{\epsilon} \ln \frac{n \cdot V}{\delta}$$

normal observations, then with probability at least $1 - \delta$ we have observed every input value i for every component c such that $P(c, i) \geq \epsilon$.

Proof. Consider an input value i for a component c in a normal observation such that $P(c, i) \geq \epsilon$. Since each observation is independent, the probability that we do not observe i in any of the m observations is at most

$$(1 - \epsilon)^m \leq (1 - \epsilon)^{\frac{1}{\epsilon} \ln \frac{n \cdot V}{\delta}} \leq e^{-\ln \frac{n \cdot V}{\delta}} = \frac{\delta}{n \cdot V} \qquad (3)$$

By a union bound over the at most $n \cdot V$ pairs of components and input values, we have that with probability $1 - \delta$, every component c and input value i for which $P(c, i) \geq \epsilon$ will be observed in one of the m normal observations □

Theorem 1. *Given an abnormal observation obs_{ab} and m normal observations, where $m \geq \frac{1}{\epsilon} \ln \frac{n \cdot V}{\delta}$, then with probability at least $1 - \delta$, the learned model SD_{FP} is sufficient to identify at least one surely faulty component with probability at least $1 - \epsilon$.*

Proof. The *scope* of a component c, denoted scope(c), is the set of components that affect its input, i.e., have a path to c in the system's topology. Since obs_{ab} is an abnormal observation and we assumed the system inputs are normal, then there is at least one faulty component c_f such that its scope contains only healthy components ($h(c')$ for every $c' \in scope(c_f)$), and (2) its output is different from its normal output, i.e., $\varphi_h(c_f) \wedge in(c_f, obs_{ab}) \wedge out(c_f, obs_{ab}) \models \bot$.

If c_f has a surely normal input in obs_{ab}, then it must be surely faulty (Definition 3), and thus using SD_{FP} we can immediately identify that c_f is faulty. If c_f does *not* have a surely normal input, then there is a component $c \in scope(c_f)$ for which there is no normal observation obs where $in(c, obs) = in(c, obs_{ab})$. Following Lemma 1, this means $P(c, in(c, obs_{ab})) < \epsilon$, i.e., this event occurs with probability smaller than ϵ. □

Note that the number of normal observations required grows only logarithmically with the number of system components (and linearly with the fan-in). For example, consider a system that consists of 100 Boolean gates, each having a fan-in of 2. Assume we wish to learn, with probability at least 0.9, sufficient information to allow us to identify a faulty component in at least 0.9 of the observations. This corresponds to setting $V = 2^2 = 4$, $n = 100$, $\delta = 0.1$, and $\epsilon = 0.1$. Thus, the number of normal observations required is $\frac{1}{0.1} \cdot \ln \frac{400}{0.1} \approx 83$ normal observations. Increasing the system size from 100 components to 1,000 components requires 106 normal observations, which is slightly more than 20% more observations than the result for 100 components.

3.2 Partial and No Probing

Next, we analyze the setting where we do not observe the outputs of all of the components. Consider first the extreme case where we have no internal probes, i.e., there is no system output that is also an input to some other component. We refer to this as the *no probing* setting. To determine whether a component is surely faulty or surely normal for a given observation, we must be able to observe both its inputs and some of its outputs. We refer to such a component as a *observed* component. There are no observed components in a no probing setting. Between full probing and no probing is the *partial probing* setting, in which only a subset of the components are observed. The system description SD_{FP} in the partial probing setting is defined as in Eq. 2, but the conjunction is only over the observed components. Thus, in the no probing setting SD_{FP} is empty, and we cannot use it to infer diagnoses.

In the partial probing setting and given enough normal observations, we may be able to identify the observed components as surely faulty or normally behaving in a given abnormal observation. However, it is not clear how likely is it for such identification to occur. The statistical analysis given for the full probing setting, however, does not transfer nicely to the partial probing setting. Lemma 1 by itself is still true in partial probing: given a sufficient number of examples ($m \geq \frac{1}{\epsilon} \ln \frac{n \cdot V}{\delta}$), with high probability ($\geq$1-δ) we observe every common ($P(c, i) \geq \epsilon$) input value of each component.

However, since we do not observe the inputs of all of the components, we cannot estimate how likely we are to be able to identify a surely faulty component in an abnormal observation (Theorem 1). The key difference between the full and partial probing settings is that in partial probing we may not have a component that outputs an abnormal value and has a surely normal input, since some of the components in a partial probing setting are not observed.

4 Using Topology and Normal Observations

So far, we have ignored an important and commonly available source of information about the system: its topology. Lamperti and Zanella [19] explored how system topology can be used to perform consistency-based diagnosis and how it relates to MBD with a behavioral model. Here we extend their discussion to consider diagnosis with normal observations and a topology.

The system topology allows us to extend the notion of a *observed component* to an *observed subsystem*. A *subsystem* is a connected induced subgraph of the system topology G. The inputs of a subsystem S are the set of inputs of the components in S that are either system inputs or are outputs of components that are not in S. The subsystem outputs are defined in a similar way.

Definition 4 (Observed Subsystem)**.** *An observed subsystem is a subsystem in which all its inputs and at least one of its outputs are either system inputs or system outputs.*

Note that an input of an observed subsystem can be a system output. The key property of an observed subsystem is that all the inputs and at least one of its outputs must be *observable* (i.e., given in an observation).

Figure 1 shows several of the observed subsystems that exist in the depicted system. For example, the set of components $\{G, H, J\}$ forms an observed subsystem with inputs from D E and outputs from J.

Let $\mathcal{S}$ be the set of all observed subsystems of the diagnosed system. This set allows us to extend the model we can learn from normal observations described in Eq. 2, to the following model, denoted by $SD_{\mathcal{S}}$:

$$\bigwedge_{\substack{S\in\mathcal{S},\\ obs\in \text{OKs}}} \Big(\big(\bigwedge_{c\in S} h(c) \big) \rightarrow f\big(S, in(S, obs_{OK}), out(S, obs_{OK})\big) \Big) \tag{4}$$

where $in(S, obs_{OK})$ and $out(S, obs_{OK})$, are the input and output values of S, respectively, in obs_{OK}, and f is generalized in the natural way from components to subsystems.

$SD_{\mathcal{S}}$ is more powerful than SD_{FP}, in the sense that $SD_{\mathcal{S}} \models SD_{FP}$ since every observed component is, in fact, an observed subsystem where the subsystem contains a single component. Since $SD_{\mathcal{S}}$ still only describes the normal behavior of the components, it is still weaker than SD_{WFM}, in the sense that $SD_{\text{WFM}} \rightarrow SD_{\mathcal{S}}$.

The concepts of surely faulty components and normally behaving components can be adapted to the subsystem level: a surely faulty *subsystem* and a normally behaving *subsystem* are subsystems that have a surely normal input in obs_{ab} and whose outputs are different and the same, respectively.

Proposition 5. *Every diagnosis must contain at least one member from every surely faulty subsystem.*

Proof outline: A surely faulty subsystem is a conflict w.r.t. $SD_{\mathcal{S}}$, as it follows by induction: if all subsystems are healthy and all inputs are normal, then the output should also be normal; contrapositively, if the output is abnormal but the inputs are normal, then there must be a faulty component.

4.1 Minimal Observed Subsystem

The set of all observed subsystems ($\mathcal{S}$) is never empty, as every system output induces at least one observed subsystem. In fact, in a full probing setting $\mathcal{S}$ can even be exponential in the number of components. Thus, we focus on the set of *minimal* observed subsystems, where an observed subsystem S is *minimal* if no proper subset of S is also an observed subsystem. We denote this set of subsystems by $\mathcal{S}_{min}$. Every system output o has a corresponding minimal observed subsystem, defined by going backwards on the system topology until reaching either a system output or the system inputs. In fact, $\mathcal{S}_{min}$ is exactly the set of all these subsystems. Figure 1 shows, in dashed lines, all the subsystems in $\mathcal{S}_{min}$ for the example system. Note that $\mathcal{S}_{min}$ can be found in polynomial time as described above, and it contains at most $|\text{SysOuts}|$ subsystems. Next, we show that this set is sufficient for diagnostic purposes and there is no need to find all of the observed subsystems.

Let $SD_{\mathcal{S}_{min}}$ be the subset of the system model $SD_{\mathcal{S}}$ that uses only the minimal observed subset in $\mathcal{S}_{min}$.

Proposition 6. $SD_{\mathcal{S}_{min}} \equiv SD_{\mathcal{S}}$

Proof outline: Since $\mathcal{S}_{min} \subseteq \mathcal{S}$, then $SD_{\mathcal{S}} \models SD_{\mathcal{S}_{min}}$. To prove the other direction, we observe that a non-minimal observed subsystem S is equivalent to a union of the set of minimal observed subsystems that correspond to the outputs of S. Thus, the statement in $SD_{\mathcal{S}}$ added due to S can be derived from the set of statements added to $SD_{\mathcal{S}_{min}}$ due to S's constituent minimal observed subsystems.

m\P	1	4	8	16	Full
4	0.01 (0.1)	0.07 (0.1)	0.10 (0.3)	0.24 (0.5)	0.68 (0.0)
16	0.04 (0.6)	0.10 (0.4)	0.21 (1.2)	0.25 (0.6)	0.91 (0.0)
64	0.11(0.8)	0.22 (1.8)	0.36 (1.9)	0.60 (2.6)	0.98 (0.0)
256	0.28(3.9)	0.37 (3.6)	0.47 (3.9)	0.62 (3.2)	1.00 (0.0)

Table 1: Fraction out of 100 trials for which the learned model for system 74181 was able to identify the failed component as the number of probes P and training examples m varies. In parentheses, we show the average number of components in the same observable subsystem as the faulty component.

The set of observable subsystems that a component is a part of provides additional, useful information:

Proposition 7. *If every observable subsystem that a component c is a part of is normally behaving, then any diagnosis that contains c is not minimal.*

The proof is omitted due to space constraints.

4.2 Statistical Analysis

The system model $SD_{\mathcal{S}_{min}}$ allows us to extend Theorem 1 to consider the system topology, making it also applicable for any probing setting (full, partial, or none).

Theorem 2. *Let $V_{\mathcal{S}_{min}}$ denote the size of the largest domain for the input of a minimal observable subsystem. Given*

$$m \geq \frac{1}{\epsilon} \ln \frac{|\text{SysOuts}| \cdot V_{\mathcal{S}_{min}}}{\delta}$$

normal observations, with probability at least $1-\delta$, the learned model $SD_{\mathcal{S}_{min}}$ is sufficient to identify at least one surely faulty minimal observed subsystem with probability at least $1-\epsilon$.

The proof of this theorem follows the proof of Theorem 1 and Lemma 1 exactly, where the only change needed is that instead of the number of components n we have the number of system outputs $|\text{SysOuts}|$, and instead of the size of the largest domain for a component's input V we have $V_{\mathcal{S}_{min}}$.

While $|\text{SysOuts}|$ cannot be larger than n, the largest domain for a subsystem's input can be much larger than V. For example, in a system that consists only of Boolean gates with fan-in 2, V is equal to 2^2. However, if the inputs of an observed subsystem consist of the inputs of three fan-in 2 Boolean gates, then $V_{\mathcal{S}_{min}}$ is $(2^2)^3$. Still, $\ln V_{\mathcal{S}_{min}} \leq n \ln V$ so the required number of examples is at most linear in n.

5 Experimental Results

As a proof-of-concept, we implemented our diagnosis algorithm and ran it on system 74181 from the ISCAS 85 Boolean circuit benchmark [13]. This system has 65 components, 14 inputs, and 8 outputs. We created training sets by setting uniform-random inputs to the system. In each trial, we injected a single fault, ran our diagnosis algorithm, and checked if the injected fault was identified as part of a surely faulty subsystem. We varied the size of the training set (4, 16, 64, and 256) and the number of probes (1, 4, 8, 16, and full probing). Table 1 shows the proportion of trials out of 100 where the learned model was able to identify the failed component. We show the average number of other components that were in the same observable subsystem as the faulty component in parentheses. The size of the training set varies across rows, and the number of probes

varies across columns. Our results are incomparable to that of MBD algorithms that know the system's model. However, as expected, more probes and larger training sets increase the likelihood that our algorithm will detect the fault. Although the system has 14 inputs (2^{14} possible input vectors), with full probing and only 256 training instances we always found exactly the faulty component. Even with only 8 probes, our algorithm found the faulty component in almost half (47%) of the cases. Due to the system topology, in 28% of the problems even one probe is sufficient to identify a set of fewer than five components (on average) that contain the faulty component.

6 Related Work

Niggemann et al. [20] proposed a method for learning behavioral models of a system described by a hybrid timed automaton, but the learned model was used for fault detection and not for diagnosis (fault isolation). Feldman et al. [21] proposed an algorithm for learning how to construct a hybrid model from a set of models of different fidelity. Preliminary work by Sadov et al. [22] studied how to test the system in order to efficiently learn a model for diagnosis. We do not assume the ability to test the system, and only infer a partial model from a given set of observations. Also, we exploit the system topology, which they did not.

Approximate MBD has been studied in the Fault Detection and Isolation (FDI) community, based on Fuzzy models [23; 24; 25] and other ways to capture how the system description can be inaccurate (e.g., robust FDI) [26; 27]. We do not assume a-priori knowledge about the inaccuracy of the model, and, unlike these prior works, propose a framework that supports logic-based reasoning over the model. Juba [28] proposed a method for PAC-learning of abductive reasoning that could in principle be used for diagnosis, extended to partial observations by Juba et al. [29]. But, this approach requires that the training set contains both normal and abnormal observations. In contrast, we do not need to train on abnormal observations.

The notion of a *surely faulty* subsystem is related to the discussion of *partial diagnoses* by Shchekotykin et al. [30]. A partial diagnosis is a diagnosis that is created by a hitting set of a subset of all minimal conflicts. Thus, a partial diagnosis may or may not be correct. By contrast, a surely faulty component must be faulty, but it may not be a diagnosis in the sense that it may not explain all of the abnormal behavior we currently observe. Deriving partial diagnoses from normal observations is a topic for future work.

Grouping components into a set of minimal observable subsystems bears some resemblance to the system abstractions [31; 32; 33; 34]. The set of minimal observed subsystems, however, does not create a disjoint partition over the components, i.e., there are components that are part of more than one minimal observed subsystem (e.g., B in Figure 1).

7 Conclusion and Discussion

We provide a theoretical foundation for diagnosis using only normal observations, system topology, and probes if available. We showed how to learn a partial system model from this information, and how to use it to for diagnosis. Our approach is unique in that the learned model is *safe*, in the sense that conflicts extracted from it are conflicts that would have been extracted if we knew the system's model. The number of observations needed to obtain an effective partial model is analyzed, and a small experimental proof of concept is reported, suggesting that our method is applicable.

In these experiments only a single fault was injected. However, our approach can detect more than a single fault. This occurs when multiple observed subsystems are detected as surely faulty, and these observed subsystems have no overlap. However, the analysis in Theorems 1 and 2 only guarantees that we find one surely faulty observed subsystem. Analyzing the number of observations and probing needed to identify a complete multiple fault diagnosis is a topic for future work. In addition, we made several significant assumptions in this paper. We now discuss approaches for relaxing them.

Deterministic Normal Behavior. If a component may output different values in two different observations even if it received the same inputs and it is healthy, it has a *non-deterministic* normal behavior. To capture such a nondeterministic behavior formally, for every component c we define a relation R_c that is the set of all the input-output pairs that are normal. If we assume that the normal behavior of the system components is *stationary*, i.e., the choice of output for a given input follows some stationary distribution, then we can adapt our theoretical results by replacing the size of the input domain (V in Theorem 1) with the size of the relation R_c.

We stress that we can only diagnose faults with respect to a nondeterministic specification, not a *probabilistic* specification. As an extreme case, suppose that one of our components is supposed to output an unbiased random bit, but it outputs 0 with probability 1. It is faulty with respect to our specification, but we cannot diagnose this from a single observation of the system, even with full probing. Many observations of this faulty behavior are needed to have any hope of detecting it.

Faults due to Abnormal Inputs. Faults may occur due to abnormal or illegal system inputs, and not faulty components. This means the correct diagnosis may be that a system input was illegal. We can model this setting by considering every system input as a "buffer" component: when healthy, it outputs normal system inputs, and when faulty it does not. Thus, our analysis is fundamentally the same.

Cyclic System Topology. In some systems there is no clear notion of components' inputs and outputs, and components may affect each other in both directions. Thus, a proper analysis of the system's behavior requires a temporal aspect, to represent the propagation of values in the system [35]. Learning a model from normal observations over time is an exciting direction for future work.

Components with a Continuous Domain. The analysis in Theorems 1 and 2 implicitly assumes that the components' input domain is discrete, as otherwise it has an infinite number of possible values. However, various methods have been proposed to learn functions over continuous-valued inputs (and with continuous-valued outputs). For example, strong generalization bounds have been proved for the Rademacher complexity [36], and analyses of the Rademacher complexity for many common kinds of regression functions have been given by Kakade et al. [37]. Thus, it may be possible to derive polynomial sample complexity bounds for our problem from these works given some a-priori knowledge about the range of normal behaviors ($\varphi_h(c)$), e.g., some parametric model for the healthy components.

Acknowledgments

Brendan Juba was supported by an AFOSR Young Investigator Award and NSF award CCF-1718380. Roni Stern was supported by the Cyber Security Research Center at Ben Gurion University of the Negev.

References

[1] Brian C. Williams and P. Pandurang Nayak. A model-based approach to reactive self-configuring systems. In *AAAI*, pages 971–978, 1996.

[2] Peter Struss and Chris Price. Model-based systems in the automotive industry. *AI magazine*, 24(4):17–34, 2003.

[3] Alexander Feldman, Helena Vicente de Castro, Arjan van Gemund, and Gregory Provan. Model-based diagnostic decision-support system for satellites. In *Aerospace Conference, IEEE*, pages 1–14, 2013.

[4] Rui Abreu, Peter Zoeteweij, and Arjan JC Van Gemund. Spectrum-based multiple fault localization. In *IEEE/ACM International Conference on Automated Software Engineering*, pages 88–99, 2009.

[5] Shekhar Gupta, Rui Abreu, Johan de Kleer, and Arjan JC van Gemund. Automatic systems diagnosis without behavioral models. In *IEEE Aerospace Conference*, pages 1–8, 2014.

[6] Rui Abreu, Peter Zoeteweij, Rob Golsteijn, and Arjan JC Van Gemund. A practical evaluation of spectrum-based fault localization. *Journal of Systems and Software*, 82(11):1780–1792, 2009.

[7] S Joe Qin. Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, 36(2):220–234, 2012.

[8] Betty Keren, Meir Kalech, and Lior Rokach. Model-based diagnosis with multi-label classification. In *International workshop on principles of diagnosis (DX)*, 2011.

[9] Wassim Hafez, Timothy Ross, and David Gadd. Machine learning for model-based diagnosis. In *American Control Conference*, volume 1, pages 42–46. IEEE, 1997.

[10] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[11] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[12] David Haussler. Overview of the probably approximately correct (PAC) learning framework. *Information and Computation*, 100(1):78–150, 1992.

[13] F. Brglez, D. Bryan, and K. Kozminski. Combinatorial profiles of sequential benchmark circuits. In *IEEE International Symposium on Circuits and Systems*, pages 1929–1934, 1989.

[14] Raymond Reiter. A theory of diagnosis from first principles. *AIJ*, 32(1):57–95, 1987.

[15] Luca Console and Pietro Torasso. A spectrum of logical definitions of model-based diagnosis. *Computational intelligence*, 7(3):133–141, 1991.

[16] Brian C Williams and Robert J Ragno. Conflict-directed A* and its role in model-based embedded systems. *Discrete Applied Mathematics*, 155(12):1562–1595, 2007.

[17] Johan de Kleer and Brian C Williams. Diagnosing multiple faults. *AIJ*, 32(1):97–130, 1987.

[18] Johan de Kleer. Diagnosing multiple persistent and intermittent faults. In *IJCAI*, pages 733–738, 2009.

[19] Gianfranco Lamperti and Marina Zanella. Consistency-based diagnosis: a topological approach. In *International workshop on the principles of diagnosis (DX)*, pages 207–214, 2012.

[20] Oliver Niggemann, Benno Stein, Asmir Vodencarevic, Alexander Maier, and Hans Kleine Büning. Learning behavior models for hybrid timed systems. In *AAAI*, volume 2, pages 1083–1090, 2012.

[21] Alexander Feldman, Gregory Provan, Rui Abreu, and Johan de Kleer. Learning diagnosis models using variable-fidelity component model libraries. *International Federation of Automatic Control (IFAC)*, 48(21):428 – 433, 2015.

[22] Vladimir Sadov, Eliahu Khalastchi, Meir Kalech, and Gal A Kaminka. Towards partial (and useful) model identification for model-based diagnosis. In *the Workshop on Principles of Diagnosis (DX)*, 2010.

[23] Arthur L Dexter and Mourad Benouarets. Model-based fault diagnosis using fuzzy matching. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 27(5):673–682, 1997.

[24] LF Mendonça, JMG Sá da Costa, and JM Sousa. Fault detection and diagnosis using fuzzy models. In *European Control Conference (ECC)*, pages 647–652. IEEE, 2003.

[25] Luís F Mendonça, JMC Sousa, and JMG Sá da Costa. An architecture for fault detection and isolation based on fuzzy methods. *Expert systems with applications*, 36(2):1092–1104, 2009.

[26] Jie Chen and Ron J Patton. *Robust model-based fault diagnosis for dynamic systems*, volume 3. Springer Science & Business Media, 2012.

[27] Paul M Frank and Xianchun Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of process control*, 7(6):403–424, 1997.

[28] Brendan Juba. Learning abductive reasoning using random examples. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 999–1007, 2016.

[29] Brendan Juba, Zongyi Li, and Evan Miller. Learning abduction under partial observability. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 1888–1896, 2018.

[30] Kostyantyn M Shchekotykhin, Thomas Schmitz, and Dietmar Jannach. Efficient sequential model-based fault-localization with partial diagnoses. In *IJCAI*, pages 1251–1257, 2016.

[31] Roni Stern, Meir Kalech, and Orel Elimelech. Hierarchical diagnosis in strong fault models. In *International workshop on the principles of diagnosis (DX)*, 2014.

[32] Sajjad Ahmed Siddiqi and Jinbo Huang. Sequential diagnosis by abstraction. *Journal of Artificial Intelligence Research*, 41:329–365, 2011.

[33] Martin Sachenbacher and Peter Struss. Task-dependent qualitative domain abstraction. *AIJ*, 162(1-2):121–143, 2005.

[34] Gianluca Torta and Pietro Torasso. Automatic abstraction in component-based diagnosis driven by system observability. In *IJCAI*, pages 394–402, 2003.

[35] Alexander Feldman and Johan de Kleer. Diagnosing multiple faults in sequential logic. In *the International Workshop on Principles of Diagnosis (DX)*, pages 2–8, 2017.

[36] Peter L Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[37] Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *NIPS*, pages 793–800, 2009.