HFMV: Hybridizing Formal Methods and Machine Learning for Verification of Analog and Mixed-Signal Circuits

Hanbin Hu, Qingran Zheng, Ya Wang, and Peng Li
Department of ECE, Texas A&M University, College Station, TX
{hanbinhu,pli}@tamu.edu

ABSTRACT

With increasing design complexity and robustness requirement, analog and mixed-signal (AMS) verification manifests itself as a key bottleneck. While formal methods and machine learning have been proposed for AMS verification, these two techniques suffer from their own limitations, with the former being specifically limited by scalability and the latter by the inherent uncertainty in learning-based models. We present a new direction in AMS verification by proposing a hybrid formal/machine-learning verification technique (HFMV) to combine the best of the two worlds. HFMV adds formalism on the top of a probabilistic learning model while providing a sense of coverage for extremely rare failure detection. HFMV intelligently and iteratively reduces uncertainty of the learning model by a proposed formally-guided active learning strategy and discovers potential rare failure regions in complex high-dimensional parameter spaces. It leads to reliable failure prediction in the case of a failing circuit, or a high-confidence pass decision in the case of a good circuit. We demonstrate that HFMV is able to employ a modest amount of data to identify hard-to-find rare failures which are completely missed by state-of-the-art sampling methods even with high volume sampling data.

1 INTRODUCTION

With increasing design complexity and rising robustness requirement, analog and mixed-signal (AMS) verification manifests itself as a key bottleneck [1]. For instance, automotive electronics may have an extremely low failure rate requirement, e.g. 1 DPPM (defective parts per million) or less, making failure detection and design verification very challenging. On one hand, formal verification is appealing as it provides a provable "yes/no" answer w.r.t the specifications under check. However, performing formal verification directly on top of a detailed low-level (nonlinear) SPICE circuit netlist or model (e.g. a DAE or hybrid automation) severely limits scalability. To date, formal techniques are only feasible for small analog blocks described by idealistic models, falling behind the practical industrial needs [3, 6]. On the other hand, one may employ machine learning for AMS verification [4] with the advantages being data-driven, incremental, and much more scalable, particularly when a sufficient amount of training data can be collected through simulation or silicon measurement. However, learning-based models do not provide a formal answer, and come with inherent model uncertainty and noise [4, 9]. Furthermore, the state-of-the-art smart statistical sampling techniques (e.g. [11]) are not specifically targeted for providing a guarantee for rare failure detection and can miss rare failures especially under a limited sampling data budget.

This work presents a new perspective in AMS verification by proposing a hybrid formal/machine-learning verification (HFMV) framework that simultaneously exploits formal and machine learning techniques. In its most abstract form, HFMV comprises two key elements: a *probabilistic machine learning model* and *formal verification* that acts on top of the machine learning model. It is the interactions between the two elements that form the promise of HFMV. The probabilistic model is trained from limited simulation/measurement data and comes with a measure of uncertainty for each prediction of the circuit performance under verification. Given a bounded verification space of design or uncertainty parameters, e.g. process variations or operating conditions, formal verification is applied with respect to a symbolic formula derived from the posterior prediction of the probabilistic learning model to check if the targeted specification is met across the *entire* verification space with a sufficiently high confidence.

HFMV has the best of the two worlds: it adds a degree of formalism on top of learning-based models by utilizing satisfiability modulo theories (SMT) [7, 10] formal techniques; and it is much more scalable than pure formal techniques at the same time. Furthermore, to circumvent the inherent uncertainty of machine learning, the proposed framework "formally" bounds learning model uncertainty and practically verifies design properties over a high-dimensional space of design uncertainty. HFMV presents several key contributions to AMS verification:

- Bridges the gap between design complexity and scalability of verification by integrating formal and machine-learning techniques into a general hybrid verification framework;
- Builds a degree of formalism into machine-learning based verification to safeguard detection of extremely rare failure under a limited data budget;
- Explores novel formally-guided active learning to iteratively reduce learning model uncertainty towards rare failure detection;
- Significantly accelerates formal solutions by efficient one-time preprocessing of SMT formulas to be checked.

Experimental studies have demonstrated that HFMV can reliably verify AMS design specifications and identify extremely rare failures under complex high-dimensional parametric uncertainties for which state-of-the-art smart statistical sampling techniques fail.

2 PROBABILISTIC MODEL-BASED FAILURE PREDICTION

We propose a notion of probabilistic model-based failure prediction before presenting HFMV. Given a bounded D-dimensional parameter space $\Omega \subseteq \mathbb{R}^D$, the true performance $y(\boldsymbol{x})$ at a particular point $\boldsymbol{x} \in \Omega$ of the design under verification (DUV) can be determined either by simulation or measurement. Without loss of generality, a point $\boldsymbol{x} \in \Omega$ is considered as a (true) failure if $y(\boldsymbol{x}) \geq T$, where T is the targeted specification (assuming greater the value, worse the performance). Verifying a highly robust design

DAC '18, June 24–29, 2018, San Francisco, CA, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5700-5/18/06…$15.00
https://doi.org/10.1145/3195970.3196059

for which failures are extremely rare, finding a failure point in a high-dimensional space can be extremely challenging and expensive in terms of numbers of measurements and simulation samples needed. We propose to leverage an efficient probabilistic machine learning model to replace direct measurements and simulations.

2.1 Probabilistic Machine Learning Model

HFMV exploits a large body of popular probabilistic machine learning models where each inference is probabilistic such as Bayesian additive regression trees [2], relevance vector machine (RVM) [12], and sparse relevance kernel machine (SRKM) [5] . The last two fall under the broad family of Gaussian processes.

Generally, each prediction from a probabilistic model with model parameters $\boldsymbol{\theta}$ is based on a posterior predictive distribution, whose cumulative distribution function (CDF) $F_Y(y;\boldsymbol{\theta})$ specifies the probability for true performance $y(\boldsymbol{x})$ to fall in the range of $[a,b]$:

$$Prob\left\{a \le y(\boldsymbol{x}) \le b\right\} = F_Y(b;\boldsymbol{\theta}) - F_Y(a;\boldsymbol{\theta}). \tag{1}$$

We define the **P-Prediction** $\hat{y}(\boldsymbol{x},P;\boldsymbol{\theta})$ associated with a probability value P for a certain point $\boldsymbol{x}$ as:

Definition 2.1. $\hat{y}(\boldsymbol{x},P;\boldsymbol{\theta}) = F_Y^{-1}(1-P;\boldsymbol{\theta})$.

According to the definition above, it is straightforward to show that the probability for true performance $y(\boldsymbol{x})$ to be no less than $\hat{y}(\boldsymbol{x},P;\boldsymbol{\theta})$ is P:

$$Prob\left\{y(\boldsymbol{x}) \ge \hat{y}(\boldsymbol{x},P;\boldsymbol{\theta})\right\} = P. \tag{2}$$

Consider SRKM as an example, which is an extension to the relevance vector machine (RVM) [12] and offers improved accuracy and the appealing probabilistic feature weighting capability [5]. A trained SRKM model has a posterior Gaussian predictive prediction with mean $\hat{y}_{est}(\boldsymbol{x})$ and variance $\hat{\sigma}_{est}(\boldsymbol{x})$ at a point $\boldsymbol{x} \in \Omega$ as:

$$\begin{aligned} y &\sim \mathcal{N}\left(\hat{y}_{est}(\boldsymbol{x}), \hat{\sigma}^2_{est}(\boldsymbol{x})\right) && (3)\\ \hat{y}_{est}(\boldsymbol{x}) &= \bar{\boldsymbol{v}}^T \boldsymbol{K}(\boldsymbol{x}) && (4)\\ \hat{\sigma}_{est}(\boldsymbol{x}) &= \sqrt{\sigma^2 + \boldsymbol{K}(\boldsymbol{x})^T \Sigma_{\boldsymbol{v}} \boldsymbol{K}(\boldsymbol{x})}, && (5) \end{aligned}$$

where σ^2 is the estimated intrinsic noise, $\bar{\boldsymbol{v}}$ and $\Sigma_{\boldsymbol{v}}$ are the posterior $D\times 1$ expectation and $D\times D$ covariance matrix of the feature weights, respectively, and $\boldsymbol{K}(\boldsymbol{x})$ is the $D \times 1$ design vector based on a chosen kernel function, which will be further discussed in Section 5. The detailed expressions for the above prediction can be found from [5]. The P-Prediction for SRKM is given by:

$$\hat{y}_{SRKM}(\boldsymbol{x},P;\boldsymbol{\theta}) = \hat{y}_{est}(\boldsymbol{x}) - \Phi^{-1}(P) \cdot \hat{\sigma}_{est}(\boldsymbol{x}). \tag{6}$$

where $\Phi^{-1}(\cdot)$ is the inverse function of CDF of a standard normal distribution.

2.2 Probabilistic Failure Detection

We leverage a probabilistic model for failure detection. However, instead of using the optimal posterior performance estimator, which is the mean $\hat{y}_{est}(\boldsymbol{x})$ of the posterior predictive distribution, we make use of the P-Prediction to cope with uncertainty of machine learning and check if $\boldsymbol{x}$ is a failure at a given confidence level by:

$$\hat{y}(\boldsymbol{x},P;\boldsymbol{\theta}) \ge T. \tag{7}$$

If (7) holds true, it is easy to see the following based on the monotonicity of the CDF:

$$Prob\left\{y(\boldsymbol{x}) \ge T\right\} \ge Prob\left\{y(\boldsymbol{x}) \ge \hat{y}(\boldsymbol{x},P;\boldsymbol{\theta})\right\} = P. \tag{8}$$

Therefore, if a point $\boldsymbol{x}$ satisfies $\hat{y}(\boldsymbol{x},P;\boldsymbol{\theta}) \ge T$, $\boldsymbol{x}$ is a (true) failure with a probability at least P. If P is large enough, $\boldsymbol{x}$ can be identified as a true failure (red cross) with a high confidence as shown in Fig. 1. Conversely, if the P-Prediction $\hat{y}(\boldsymbol{x},P;\boldsymbol{\theta}) < T$, then

$$Prob\left\{y(\boldsymbol{x}) < T\right\} \ge 1 - Prob\left\{y(\boldsymbol{x}) \ge \hat{y}(\boldsymbol{x},P;\boldsymbol{\theta})\right\} = 1 - P. \tag{9}$$

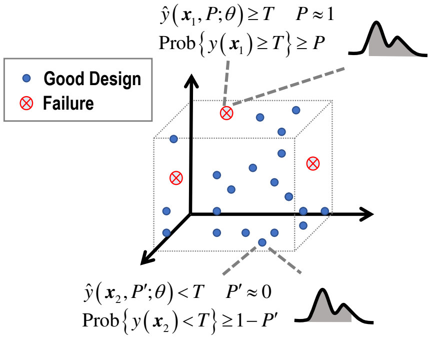
$\hat{y}(\boldsymbol{x}_1,P;\theta) \ge T \quad P \approx 1$
$\text{Prob}\left\{y(\boldsymbol{x}_1) \ge T\right\} \ge P$
● Good Design
⊗ Failure
$\hat{y}(\boldsymbol{x}_2,P';\theta) < T \quad P' \approx 0$
$\text{Prob}\left\{y(\boldsymbol{x}_2) < T\right\} \ge 1 - P'$

Figure 1: Probability to be a failure/good point by a probabilistic machine learning model.

Therefore, $\boldsymbol{x}$ is a good design point with a probability at least $1-P$. When P is small enough, $\boldsymbol{x}$ may be identified as a good point (blue dot) with high confidence as shown in Fig. 1. Based on the satisfiability of (7), $\boldsymbol{x}$ can be classified as a failure/good point with certain model belief determined by P as summarized in Table 1.

Table 1: Model beliefs based on statisfiability of (7).

	$P \approx 1.0$	$P \approx 0.0$
SAT	Failure (strong belief)	Failure (weak belief)
Non-SAT	Good (weak belief)	Good (strong belief)

3 FORMAL PROBLEM FORMULATION

We apply formal verification on top of a trained probabilistic machine learning model to provide a degree of **coverage** for failure detection, which is accomplished by exhaustively proving or disproving a given specification T at an adaptively chosen confidence level P in the entirety of the parameter space Ω. To do so, we make use of the recent advances in satisfiability modulo theories (SMT) solvers.

SMT solvers are extensions to Boolean satisfiability (SAT) counterparts which check the satisfiability of formulas defined on Boolean variables and operations. SMT solvers come with added expressiveness of uninterpreted function symbols, equality, quantifiers, and various operations such as arithmetic, datatype and array operations [8]. While originally developed in 1970s, SMT technology has undergone significant improvements lately. A number of efficient SMT solvers have emerged, for example Z3[7] and iSAT3 [10].

3.1 Two Hybrid Verification Problems

In HFMV, *two hybrid verification problems are defined.* To exhaustively check the existence of any failure point according to the machine learning model belief in the entire parameter space, we define an SMT-based problem called **Failure Detection Problem** using (7):

$$\exists \boldsymbol{x} \in \Omega \text{ s.t. } \hat{y}(\boldsymbol{x},P;\boldsymbol{\theta}) \ge T, P \approx 1.0. \tag{10}$$

A SAT solution with P close to one returned by the SMT solver is very likely to be a true failure. If this is verified to be a true failure by a single simulation/measurement, a “Fail” conclusion is immediately drawn for the verification task, and additional failures may be obtained by finding more SAT solutions if desired.

We define the **Design Certification Problem** which is checked when one attempts to verify that the targeted specification is met across the entire parameter space:

$$\exists \boldsymbol{x} \in \Omega \text{ s.t. } \hat{y}\left(\boldsymbol{x}, P; \boldsymbol{\theta}\right) \geq T, P \approx 0. \tag{11}$$

A Non-SAT solution from the SMT solver indicates that all points in Ω are believed to be good by the model at a high confidence level. In practice, we draw a "Pass" conclusion of verification only when both P and the model uncertainty (measured by the prediction variance) are sufficiently low, the latter of which is achieved during the active-learning guided iterative model re-training process described in Section 4. By tuning the confidence level P and monitoring the model uncertainty while operating on the two problems, we direct the SMT solver towards solving either the *Failure Detection Problem* or the *Design Certification Problem.*

Note again the HFMV framework can be built upon any probabilistic machine learning model as long as a posterior predictive distribution $F_Y\left(y; \boldsymbol{\theta}\right)$ is provided. Using SRKM as the underlying machine learning model as an example, assume Ω is a D-dimensional bounding box with $\underline{x\left(i\right)} \leq x\left(i\right) \leq \overline{x\left(i\right)}$ along each parameter dimension i, the SMT form of the *Failure Detection Problem* or *Design Certification Problem* at a properly chosen P is:

$$\begin{aligned} &\exists \boldsymbol{x} \in \mathbb{R}^D \\ \text{s.t. } &\left\{\hat{y}_{est}\left(\mathbf{x}\right) - \Phi^{-1}\left(P\right) \cdot \hat{\sigma}_{est}\left(\mathbf{x}\right) \geq T\right\} \\ &\wedge \left\{\underline{x\left(i\right)} \leq x\left(i\right) \leq \overline{x\left(i\right)}\right\}, i = \left[1, D\right]. \end{aligned} \tag{12}$$

4 PROPOSED ACTIVE LEARNING

Based on the fact that failures are extremely rare and hard to detect, a small initial training dataset may not contain any failure, which results in an initial "lousy" probabilistic model as illustrated in Fig. 2. To address this challenge, we propose to iteratively improve the model accuracy through active learning with the goal of approaching rare failure regions under a limited data budget. Active learning selects optimal sampling locations on-the-fly and directs re-training of the machine learning model across multiple iterations. We explore two active learning approaches: **1)** max variance learning to reduce model uncertainty based on max variance values of model prediction, and **2)** a novel formally-guided approach aiming at discovery of rare failure regions.

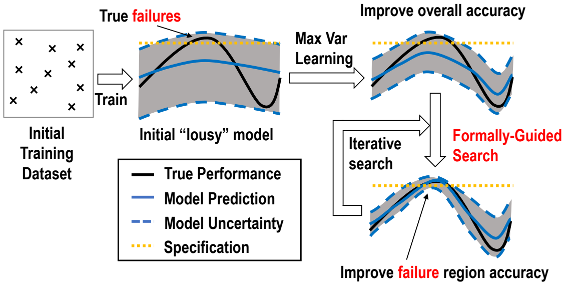

Figure 2: Proposed active learning.

4.1 Max Variance Learning

The posterior predictive distribution $F_Y\left(y; \boldsymbol{\theta}\right)$ reveals the essential information of model uncertainty. In particular, regions with large prediction variance $Var\left(y; \boldsymbol{\theta}\right)$ correspond to locations where model uncertainty is high. Additional sampling can be performed at points with the largest variance to improve the overall model accuracy:

$$\underset{\boldsymbol{x}}{\operatorname{argmax}}\, Var\left(y; \boldsymbol{\theta}\right), \text{ s.t. } \boldsymbol{x} \in \Omega. \tag{13}$$

Since the above *Max Variance Learning* phase takes place early on in the active learning process as shown in Fig. 2, the optimization needs not to be done exactly. Instead, we efficiently evaluate the model variance at a large number of randomly chosen points in Ω, and pick the top N_{var} locations for additional simulation or measurement. Then, the model is retained using the larger training dataset. Experimentally, performing one or two such iterations is sufficient.

4.2 Formally-Guided Active Learning

Finding extremely-rare failures can be very challenging for designs with stringent failure-rate requirements. Improving just the overall machine learning model accuracy as typically done in a standard active learning strategy is far from addressing the rare-failure detection challenge. Our key idea is to propose a novel formally-guided active learning approach, where the main objective is to search for the most-probable failure locations in the entire high-dimensional parameter space as shown in Fig. 3.

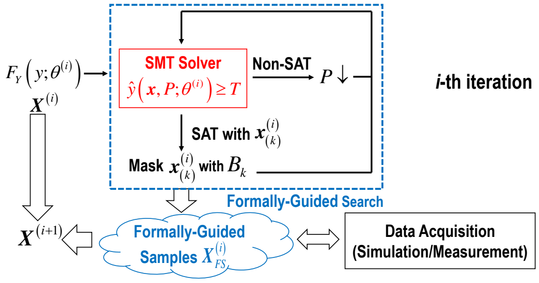
$F_Y\left(y;\theta^{(i)}\right)$
$X^{(i)}$
SMT Solver
$\hat{y}\left(x,P;\theta^{(i)}\right) \geq T$
Non-SAT
$P \downarrow$
SAT with $x_{(k)}^{(i)}$
Mask $x_{(k)}^{(i)}$ with B_k
i-th iteration
Formally-Guided Search
$X^{(i+1)}$
Formally-Guided Samples $X_{FS}^{(i)}$
Data Acquisition (Simulation/Measurement)

Figure 3: Formally-guided active learning.

For the i-th iteration of the proposed formally-guided approach, denote the parameters and the posterior predictive distribution of the present model by $\boldsymbol{\theta}^{(i)}$ and $F_Y\left(y; \boldsymbol{\theta}^{(i)}\right)$, respectively, which are trained on the current dataset $\boldsymbol{X}^{(i)}$. The corresponding P-Prediction is denoted by $\hat{y}\left(\boldsymbol{x}, P; \boldsymbol{\theta}^{(i)}\right)$. The *Failure Detection Problem* of (10) is solved to find the most-probable failure locations with $P \approx 1$. It is entirely possible that a Non-SAT solution is returned, indicating no points can be identified as a failure with a strong model belief. In this case, P is reduced with a small step gradually to allow for finding candidate failure points with reduced model confidence. Otherwise, a returned SAT solution $x_{(k)}^{(i)}$ satisfying (10) will be included as the kth sampling point in the i-th iteration of active learning. We repeatedly solve the following SMT instance to get the $(k+1)$-th point while avoiding getting the same solutions returned before:

$$\begin{aligned} &\exists \boldsymbol{x}_{(k+1)}^{(i)} \in \Omega \backslash \cup \left\{B_1, B_2, \ldots, B_k\right\} \\ &\text{s.t. } \hat{y}\left(\boldsymbol{x}_{(k+1)}^{(i)}, P; \boldsymbol{\theta}^{(i)}\right) \geq T, \end{aligned} \tag{14}$$

where each $B_j \left(j = \left[1, k\right]\right)$ is a D-dimensional bounding box, i.e. $B_j = \left\{\boldsymbol{x} \in \Omega \middle| \left\|x\left(p\right) - x_{(j)}^{(i)}\left(p\right)\right\| \leq d_0, p = \left[1, D\right]\right\}$ is a hyper-cube enclosing $x_{(j)}^{(i)}$ at its center with a length of $2d_0$ along each dimension. Assume at each i-th active learning iteration, a user-defined N_i number of formally guided samples are selected: $\boldsymbol{X}_{FS}^{(i)} =$

$\left\{x_{(1)}^{(i)}, x_{(2)}^{(i)}, \ldots, x_{(N_i)}^{(i)}\right\}$. All points in $X_{FS}^{(i)}$ are queried using either circuit simulation or measurement to obtain the corresponding true performance values. Adding these training samples to the dataset used in the i-th iteration gives a larger dataset: $X^{(i+1)} = X^{(i)} \cup X_{FS}^{(i)}$, which is used to re-train the model and update the predictive distribution $F_Y\left(y; \boldsymbol{\theta}^{(i+1)}\right)$.

"Actively" finding out the most-probable failure locations in the high-dimensional parameter space is instrumental for extremely-rare failure detection under limited data budgets. The proposed active learning process terminates when reaching the set data limit, when a large percentage of formally determined points are verified to be true failures, or when a targeted number of true failures have been found. In the event of no true failure detection during the active learning process, we then attempt to solve the *Design Certification Problem* of (11) by which we certify the circuit to be good if both P and model uncertainty are sufficiently low.

5 ACCELERATION OF SMT SOLUTIONS

A key computational component of the proposed HFMV framework to solve variants of (10). To significantly boost runtime efficiency, a promising solution is to simplify the nonlinear SMT formula through novel equivalent transformations or approximations that can be much more efficiently solved. We propose two numerical preprocessing schemes: *input space re-mapping* and *linear approximation* under the context of SRKM based probabilistic model. These two techniques only present negligible one-time preprocessing overhead for each SMT instance but have been shown to speed up SMT solving by a few orders of magnitude.

5.1 Input Space Re-Mapping

SRKM employs a vector kernel function $\boldsymbol{K}(\boldsymbol{x}_*) \in \mathbb{R}^{D\times 1}$ of the following form to compute the similarity between M training samples and the input vector $\boldsymbol{x}_*$ at which a prediction shall be made over D parameter dimensions:

$$\boldsymbol{K}(\boldsymbol{x}_*)(k) = \sum_{j=1}^{M} \omega(j) \cdot K_k(\boldsymbol{x}_*(k), \boldsymbol{X}(j)(k)), k = [1, D], \tag{15}$$

where $\omega(j)$ is the posterior mean estimation for the j-th sample weight, $\boldsymbol{X}(j)(k)$ is the k-th feature of the j-th training sample from the training dataset $\boldsymbol{X}$, and $K_k(\cdot,\cdot)$ is a scalar kernel function measuring the similarity of two input vectors over the k-th dimension, which can be chosen arbitrarily by the user to be, for example, a radial basis function (RBF) kernel or polynomial kernel.

Important to note that in a trained model, $\boldsymbol{x}_*$ is the only symbolic vector variable in (15) and other terms are known constants. Furthermore, $\boldsymbol{K}(\boldsymbol{x}_*)(k)$ is only symbolically dependent on the k-th dimension (parameter) $\boldsymbol{x}_*(k)$ of the input vector $\boldsymbol{x}_*$, allowing defining new symbolic variables $\boldsymbol{a} = [a(1), a(2), \cdots, a(D)]^T$:

$$a(k) = g(\boldsymbol{x}_*(k)) = K(\boldsymbol{x}_*)(k), \tag{16}$$

where function $g(\cdot)$ is introduced to signify the fact that $a(k)$ only depends on $\boldsymbol{x}_*(k)$. This allows to re-map the input vector from the original X-space to the new A-space. The minimum $\underline{a(k)}$ and maximum $\overline{a(k)}$ of $a(k)$ for each dimension can be obtained through a trivial one-dimensional optimization:

$$\underline{a(k)} = \min_{\underline{x_*(k)} \le x_*(k) \le \overline{x_*(k)}} K(\boldsymbol{x}_*)(k), \tag{17}$$

$$\overline{a(k)} = \max_{\underline{x_*(k)} \le x_*(k) \le \overline{x_*(k)}} K(\boldsymbol{x}_*)(k), \tag{18}$$

where $\underline{x_*(k)}$ and $\overline{x_*(k)}$ specify the bounds of the k-th component of the input vector in the original parameter space Ω.

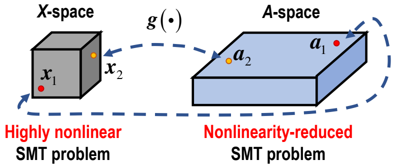

Figure 4: Input space re-mapping.

Instead of operating in the the original X-space, the SMT problem can be reformulated in the A-space as illustrated in Fig. 4:

$$\begin{aligned} &\exists \boldsymbol{a} \in \mathbb{R}^D \\ &\text{s.t. } \left\{\bar{\boldsymbol{v}}^T \boldsymbol{a} - \Phi^{-1}(P) \cdot \sqrt{\sigma^2 + \boldsymbol{a}^T \Sigma_{\boldsymbol{v}} \boldsymbol{a}} \ge T\right\} \\ &\wedge \left\{\underline{a(k)} \le a(k) \le \overline{a(k)}\right\}, k = [1, D]. \end{aligned} \tag{19}$$

The new SMT instance of (19) is equivalent to the original problem while having no strong nonlinearity introduced by the nonlinear kernel function, and hence can be more efficiently solved. A solution obtained in the A-space is easily mapped back to the X-space numerically.

5.2 Linear Approximation

Note that the formula of (19) is nonlinear due to the square root computation for the model variance and the quadratic term $\boldsymbol{a}^T \Sigma_{\boldsymbol{v}} \boldsymbol{a}$. We propose to find a close linear approximation of (19) such that a state-of-the-art fast linear SMT solver such as Z3 [7] can be applied.

Since Σ_v is a positive semidefinite matrix, the lower bound of $\boldsymbol{a}^T \Sigma_{\boldsymbol{v}} \boldsymbol{a}$ term can be efficiently found by a one-time convex quadratic minimization within the bounded hyper-cube:

$$l_a = \min_{\boldsymbol{a}} \boldsymbol{a}^T \Sigma_v \boldsymbol{a}, \text{ with } \underline{a(k)} \le a(k) \le \overline{a(k)}, k = [1, D]. \tag{20}$$

Inserting the this lower bound into (19) leads a safe linear approximation to (21):

$$\begin{aligned} &\exists \boldsymbol{a} \in \mathbb{R}^D \\ &\text{s.t. } \left\{\bar{\boldsymbol{v}}^T \boldsymbol{a} - \Phi^{-1}(P) \cdot \sqrt{\sigma^2 + l_a} \ge T\right\} \\ &\wedge \left\{\underline{a(k)} \le a(k) \le \overline{a(k)}\right\}, k = [1, D]. \end{aligned} \tag{21}$$

Our results show that this linear formula is reasonably accurate and can be very efficiently solved. The identified candidate failure points are further checked by the exact formula to filter out false solutions. Z3 can solve 1,000 SMT instances of (21) in 2 CPU minutes while solving one instance of (19) may take around 9 minutes.

6 EXPERIMENTAL RESULTS

We test the proposed HFMV on three analog circuits and compare its performance with the Monte Carlo (MC) method and Scaled-Sigma Sampling algorithm (SSS) [11] , a state-of-the-art smart statistical sampling technique that has demonstrated excellent performance for analog yield estimation. To maximize the possibility of hitting rare failures based on MC, uniform sampling is adopted in each bounded parameter space. The prototyped HFMV tool was developed using SRKM as the underlying probabilistic machine learning model in C++.

The three test circuits are: a differential amplifier (Amp), a low-dropout voltage regulator (LDO) (Fig. 5) , and a DC-DC converter (DCDC) (Fig. 6) , all designed using a commercial 90nm CMOS

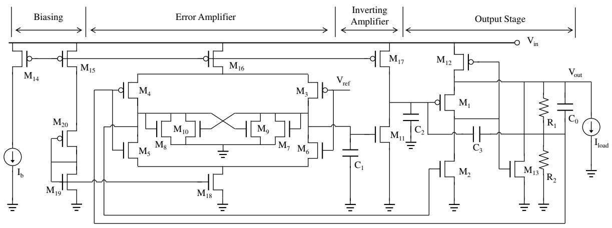

Figure 5: A LDO with 60 transistor-level variations.

technology design kit. Simulation data is collected using Synopsys HSPICE (for Amp and LDO) and Cadence Spectre (for DCDC). A few specifications for each of the following 10 performances are chosen as verification targets: GBW, gain and CMRR for the amplifier, OA (output accuracy), OS (overshoot), RS (ripple size) and PE (power efficiency) for the DC-DC converter, and QC (quiescent current), US (undershoot) and LR (load regulation) for the LDO. Three types of transistor-level variations are considered for each transistor in the amplifier and LDO: channel length, threshold voltage, and gate oxide thickness, resulting in a 15-dimensional and 60-dimensional verification problem, respectively. Channel length and width variations are considered for each transistor in the DC-DC converter, resulting in a 44-dimensional verification problem.

Two 15-D hyper-cubes covering $\pm4\sigma$ and $\pm8\sigma$ variation of each device parameter around the mean, respectively, are set up as the bounded parameter space for verification of the amplifier. Similarly, 44-D and 60-D hyper-cubes are set up for $\pm4\sigma$ and $\pm8\sigma$ verification of the DC-DC converter and LDO, respectively. Based upon the Gaussian distribution used to model all process variations (not required by HFMV though), the probability masses outside each hyper-cube are 0.1%, 0.3% and 0.4% respectively for the amplifier, DC-DC converter and LDO in the $\pm4\sigma$ case, and $2 \times 10^{-12}\%$, $6 \times 10^{-12}\%$, and $8 \times 10^{-12}\%$ for the three designs in the $\pm8\sigma$ case.

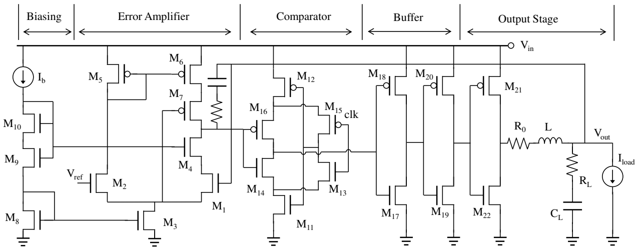

Figure 6: A DC-DC converter with 44 transistor variations.

6.1 Active-Learning Guided Failure Discovery

By interfacing with HSPICE or Spectre, HFMV starts with 200 initial simulation samples uniformly sampled in the bounded parameter space. After that, the proposed active learning strategy performs two rounds of sampling to collect 400 data points with max variance of model prediction, and then directs failure discovery by collecting around $N_i \approx 350$ samples for each formally-guided learning iteration. During this process, we record the number of samples taken for finding the first true design failure. The process moves on to find additional failures until reaching a user-defined target or sample size limit. In case of finding no failure, a specification is considered satisfied across the entire bounded parameter space if a non-SAT solution is returned for the SMT formula that checks the nonexistence of any point at which the performance meets the specification with a probability lower than some user-specified probability approximating to 100%.

We illustrate the active-learning guided failure discovery process for the challenging task of $\pm8\sigma$ CMRR verification of the amplifier in Fig. 7. The process starts off with samples having max SRKM model variance to improve the overall model accuracy. The proposed active learning then directs the sampling process towards rare failure points in the 15-D bounded parameter space. The effectiveness of the active learning can be observed by the discovery of a true failure point early on in the process and then many other failure points later on, which are very rare.

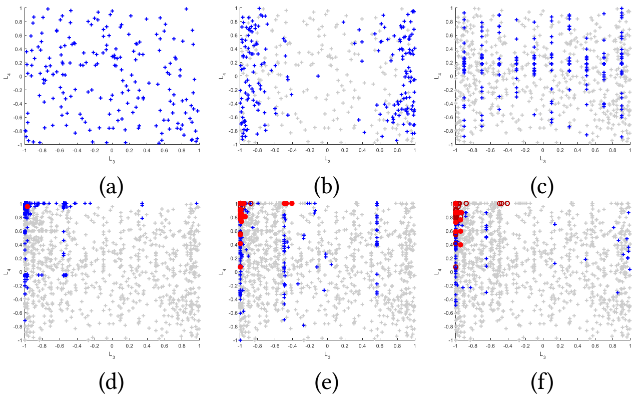
(a) (b) (c) (d) (e) (f)

Figure 7: Selected 6 iterations of proposed active learning projected onto a 2D space spanned by two device variables for $\pm8\sigma$ failure detection of amplifier CMRR. Gray crosses: samples from the previous iteration; blue crosses: samples selected in current iteration; red circle: sampled true failures. (a) Initial dataset; (b) max variance learning; (c) first iteration of formally-guided active learning; (d) first true failure found; (e)(f) large numbers of failures found.

6.2 Rare Failure Detection

All three methods are applied to $\pm4\sigma$ verification of 10 specifications of the three designs as in Table 2. Only HFMV and SSS are applied to $\pm8\sigma$ verification as shown in Table 3 as it is almost completely meaningless to even try MC within such wide-ranges of parameter variations for finding any extremely rare failure. HFMV mainly targets at extremely rare failure detection, where the samples are very expensive to collect. The listed runtimes for MC in Table 2 attempt to demonstrate that even for relatively small circuits, the simulation cost is huge when requiring a large number of samples. When facing a fairly large circuit, the simulation time can easily dominate the overall HFMV runtime. For example, the overall HFMV runtime for output accuracy of the DCDC converter cost around 14 hours, and around 10 hours were consumed by simulation for the pre-layout schematic. Hence, the number of simulation runs shall be minimized as much as possible. As seen from Table 2 and Table 3, the numbers of simulation samples used by HFMV are significantly lower than SSS and MC. HFMV can hit the first true failure point using 600 to about 1,500 samples, which are about **10x and up to 1,000x lower than used by SSS and MC**, respectively. Yet, both MC and SSS cannot find any true failure in the bounded parameter space. While SSS is one of the state-of-the-art statistical sampling technique and has been shown to produce excellent results for yield estimation of analog circuits [11] , it lacks mechanisms specifically targeting for finding extremely rare failure locations in high-dimensional parameter spaces.

Fig. 8 and Fig. 9 report the worst-case performances normalized with respect to the corresponding specifications found by each method. It can be observed that since both MC and SSS fail to find any true failure for all targeted performances, they produce misleading outcomes for verification. In contrast, HFMV is able

Table 2: Comparison on ±4σ failure detection. # Samp: # of training (simulation) samples used by each method; # 1st Fail Hit: # of samples used for finding the first true failure by HFMV; # Failure: # of failures found in the bounded parameter space.

Spec.		Target	HFMV			SSS		MC		
			# Samp	# 1st Fail Hit	# Failure	# Samp	# Failure	# Samp	# Failure	Time
Amp	GBW	22MHz	1,307	600	227	6,000	0	600,000	0	228:10:48
	Gain	2.5dB	2,307	1,507	155	6,000	0	600,000	0	
	CMRR	10dB	1,400	1,000	437	6,000	0	600,000	0	
DCDC	OA	5.50%	1,200	600	334	4,000	0	45,000	0	699:16:48
	OS	0.94%	1,000	600	319	4,000	0	45,000	0	
	RS	0.598mV	1,000	600	162	4,000	0	45,000	0	
	PE	83.20%	900	600	198	4,000	0	45,000	0	
LDO	QC	16mA	2,486	600	287	6,000	0	649,000	0	160:25:12
	US	60%	1,800	1,000	319	6,000	0	649,000	0	
	LR	55%	1,897	898	435	6,000	0	649,000	0	

Table 3: Comparison on ±8σ failure detection. Variables defined as in Table 2.

Spec.		Target	HFMV			SSS	
			# Samp	# 1st Fail Hit	# Failure	# Samp	# Failure
Amp	GBW	5MHz	1,000	600	396	9,000	0
DCDC	OA	10.0%	1,300	600	312	9,000	0
	OS	1.00%	600	600	85	9,000	0
	RS	0.6mV	600	600	87	9,000	0
	PE	80.00%	1,000	600	275	9,000	0
LDO	QC	20mA	896	600	140	9,000	0
	US	100%	1,897	897	381	9,000	0
	LR	80%	1,618	599	382	9,000	0

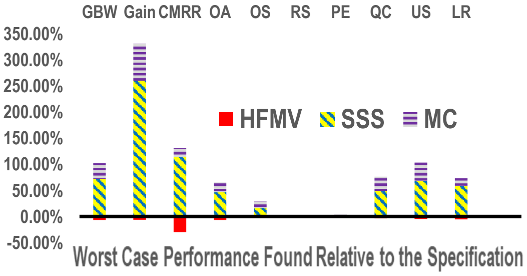

Figure 8: Worst-case performances relative to the specifications found by each method in the ±4σ region.

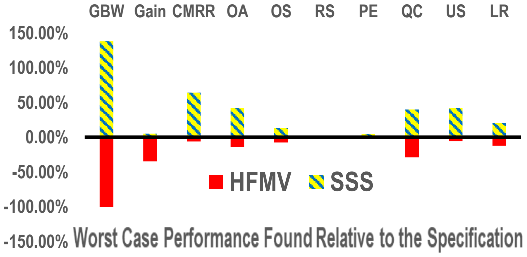

Figure 9: Worst-case performances relative to the specifications found by HFMV and SSS in the ±8σ region.

to find many specification violations (true failures). The identified worst-case performance values can be significantly *worse* than the corresponding specifications.

7 CONCLUSION

A novel hybrid approach, namely HFMV, has been presented to address rare failure detection challenges associated with AMS verification. HFMV combines the key benefits of formal verification and machine-learning based approaches while circumventing their key limitations in terms of scalability and model uncertainty. It has been demonstrated that HFMV can provide reliable verification of AMS performance specifications in high-dimensional parameter spaces for which both Monte Carlo and a state-of-the-art sampling technique lead to misleading results.

ACKNOWLEDGMENTS

This material is based upon work supported by the Semiconductor Research Corporation (SRC) through Texas Analog Center of Excellence at the University of Texas at Dallas (Task ID:2712.004).

REFERENCES

[1] H. Chang and K. Kundert. 2007. Verification of Complex Analog and RF IC Designs. *Proc. IEEE* 95, 3 (March 2007), 622–639.

[2] H. A. Chipman, E. I. George, and R. E. Mcculloch. 2010. BART: Bayesian Additive Regression Trees. *Annals of Applied Statistics* 4, 1 (2010), 266–298.

[3] W. Denman, B. Akbarpour, S. Tahar, M. H. Zaki, and L. C. Paulson. 2009. Formal verification of analog designs using MetiTarski. In *Formal Methods in Computer-Aided Design (FMCAD)*. 93–100.

[4] J. A. Kumar, S. N. Ahmadyan, and S. Vasudevan. 2014. Efficient Statistical Model Checking of Hardware Circuits With Multiple Failure Regions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 33, 6 (June 2014), 945–958.

[5] H. Lin, A. M. Khan, and P. Li. 2017. Statistical circuit performance dependency analysis via sparse relevance kernel machine. In *IEEE International Conference on IC Design and Technology (ICICDT)*. 1–4.

[6] M. Miller and F. Brewer. 2013. Formal verification of analog circuit parameters across variation utilizing SAT. In *Design, Automation Test in Europe Conference Exhibition (DATE)*. 1442–1447.

[7] L. D. Moura and N. Bjørner. 2008. Z3: An Efficient SMT Solver. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. 337–340.

[8] L. D. Moura, B. Dutertre, and N. Shankar. 2007. A Tutorial on Satisfiability Modulo Theories. In *Proceedings of the International Conference on Computer Aided Verification (CAV)*. 20–36.

[9] M. Rana, R. Canal, J. Han, and B. Cockburn. 2016. SRAM memory margin probability failure estimation using Gaussian Process regression. In *IEEE 34th International Conference on Computer Design (ICCD)*. 448–451.

[10] K. Scheibler, F. Neubauer, A. Mahdi, M. Fränzle, T. Teige, T. Bienmüller, D. Fehrer, and B. Becker. 2016. Accurate ICP-based floating-point reasoning. In *Formal Methods in Computer-Aided Design (FMCAD)*. 177–184.

[11] S. Sun, X. Li, H. Liu, K. Luo, and B. Gu. 2015. Fast Statistical Analysis of Rare Circuit Failure Events via Scaled-Sigma Sampling for High-Dimensional Variation Space. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34, 7 (July 2015), 1096–1109.

[12] M. E. Tipping. 2001. Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research* 1 (June 2001), 211–244.