IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 2, 2021, accepted May 10, 2021, date of publication May 14, 2021, date of current version May 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3080294

Cross-Corner Delay Variation Model for Standard Cell Libraries

KWANGSU KIM[1], (Graduate Student Member, IEEE), BYUNGHA JOO[2], YOUNG MIN PARK[1], TAEYANG JEONG[1], KI TAE KIM[1], AND EUI-YOUNG CHUNG[1], (Member, IEEE)

[1]Department of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea
[2]Rangduru, San Jose, CA 95134, USA

Corresponding author: Eui-Young Chung (eychung@yonsei.ac.kr)

This work was supported in part by the Samsung Science and Technology Foundation under Grant 2020-11-1656, in part by the Ministry of Trade, Industry and Energy (MOTIE) under Grant 10080722 and Grant 10080590, and in part by the Korea Semiconductor Research Consortium Support Program (KSRC) for the development of the future semiconductor device.

ABSTRACT For timing closure of logic circuits, circuit designers must perform sign-offs on a variety of *process, voltage, and temperature (PVT)* conditions. Designs of advanced logic circuits involve a multitude of voltage islands and operating modes, each of which requires delay characterizations at nearby PVT corners. Furthermore, advanced technologies nodes suffer from *corner explosion*: while the impact of PVT variations is being exacerbated, process variations are also diversifying, increasing the number of operating conditions exponentially. This paper revisits the importance of cross-corner timing estimations and proposes a delay variation model to mitigate such corner explosion. Our objective is to reduce PVT corner characterization effort for timely static timing analysis on an exploding number of operating conditions. Our proposed *Decomposed Propagation Vector Variation Model (DPVVM)* decomposes propagation delay and timing constraints into the driving by the receiver cell and its driver cell; by scaling them separately, delay characterization effort is reduced to a fraction of time while realizing accurate timing estimations. We also propose *Multi-Dimension Recomposition (MDR)* scheme, which exploits a multitude of pre-characterized corners to further improve the consistency of cross-corner timing estimations. As a result, with only 8.0% of a corner characterization effort compared to the full-characterization, DPVVM combined with MDR achieves overall cross-corner timing estimation errors of 4.8% and 5.6% for single cells and complex logic circuits—or improvements of 69% and 61% over the conventional derating method, respectively. Our proposed method's characterization overhead is 11% over the conventional derating method; the overhead is marginal, accounting for only 0.76% of a full-characterization time.

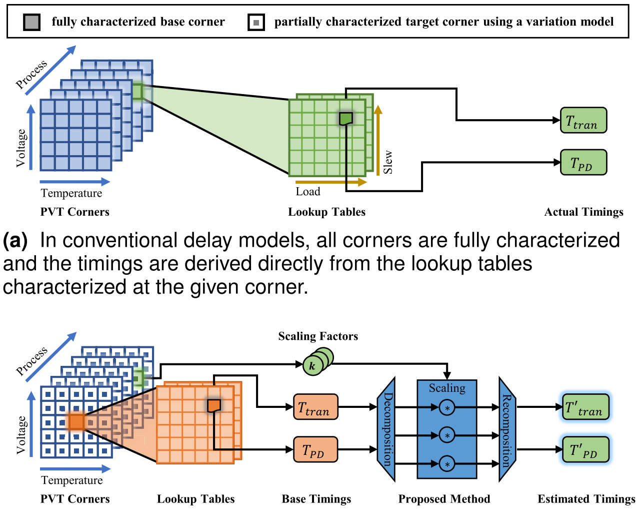
INDEX TERMS CMOS technology, design automation, design tools, integrated circuit synthesis, logic circuits, semiconductor devices.

I. INTRODUCTION

Timing variability is of significant concern when it comes to timing sign-off of modern integrated circuits. The shrinking of process technology often implies a diversity of physical effects to take into account, making timing estimation at various operating conditions non-linear and hence unpredictable [1]. At the same time, the importance of the validation under an increasing number of *process(P), voltage(V), and temperature(T) (PVT)* conditions is growing, causing *corner explosion* [2].

Design automation tools must validate given designs under a vast range of PVT corners, both timely and accurately. Failure to provide accurate timing estimation can either lead to design *pessimism*—over-estimating timing violations; the circuit ends up occupying unnecessarily large area or providing limited clock frequency—or *optimism*—under-estimating timing violations; the circuit ends up either being unstable or non-functional. Sign-off with *multi-corner multi-mode (MCMM)* analysis [3] requires sign-offs on a vast number of operating conditions, but the effort to provide such conditions is often overlooked. Furthermore, if given the ability to perform timing closures on more PVT conditions, designers can better assess a given circuits' sensitivity to PVT variations, resulting in more efficient circuit designs in performance,

The associate editor coordinating the review of this manuscript and approving it for publication was Christian Pilato.



(a) In conventional delay models, all corners are fully characterized and the timings are derived directly from the lookup tables characterized at the given corner.

(b) In DPVVM, only a base corner is fully characterized and scaling factors—derived from partial characterizations—are used to estimate timings by scaling from the base corner.

FIGURE 1. Comparison of approaches to derive timing under desired PVT corner.

power, or area. Fig. 1(a) presents an overview on how propagation delay (T_{PD}) and transition time (T_{tran}) of a gate cell are extracted from lookup tables. Figuratively, thousands of lookup tables for hundreds of cells need to be generated for each PVT corner; multiplying the number to hundreds of corners yields millions of timing arcs to be simulated. The idea of characterizing libraries on hundreds or even tens of thousands of PVT corners [4] is rapidly becoming unpractical at best, if ever feasible.

EDA vendors and foundries have been providing delay variation estimation methods to alleviate the characterization problem. However, these methods fall short of accuracy because propagation delay measures the propagation made by both the receiver cell and its driver cell as a single lump-sum, failing to capture a receiver cell's actual propagation from its input threshold voltage. In a few cases, the output is driven before the input reaches $\frac{1}{2}V_{DD}$, causing *negative delay*; the conventional derating method does not model this phenomenon and causes significant errors. We observed that the variability in timing components of a receiver cell and its driving cell—together composing propagation delay—differ and should be scaled separately.

We hereby propose *Decomposed Propagation Vector Variation Model (DPVVM)*, an accurate delay variation model to estimate timing in non-characterized *target corners* from pre-characterized *base corners* with minimal SPICE simulations. The flow of our work is highlighted in Fig. 1(b). The propagation delay or the timing constraint of a receiver cell is *decomposed* into propagation vectors: the driver propagation—driven by the driver cell before the receiver reaches its threshold voltage—and the receiver propagation—driven by the receiver cell itself after the driver propagation. These components are *scaled* independently using scaling factors and then *recomposed* back to yield the timing at the desired PVT corner.

The novelties of this paper are as follows:

1) We propose a novel method to decompose, scale, and recompose T_{PD} to achieve improved accuracy in cross-corner timing estimation over the conventional derating method. Extra characterization effort is marginal in comparison to the conventional derating.
2) We also propose *Multiple-Dimension Recomposition (MDR)*—a method applicable to both derating and DPVVM—to perform timing estimation from multiple base corners to further improve accuracy for the similar characterization effort.
3) DPVVM combined with MDR—or *DPVVM-MDR*—provides accuracy level comparable to state-of-the-arts variation models based on machine learning; DPVVM-MDR is the most effortless variation model, requiring the least computation overheads, covering sequential cells, and being independent of process technologies.

The characteristics of DPVVM-MDR can be highlighted as follows:

1) **Delay estimation with high accuracy**: with its ability to capture monotonic propagation characteristics by distinguishing different sources of propagation, DPVVM-MDR showcases average cross-corner delay estimation errors of 4.8% and 5.6%—or improvements of 69% and 61% over the conventional derating method—for single-cells and complex circuits, respectively.
2) **Low sensitivity to sampling policies**: DPVVM-MDR exhibits consistent accuracy across different sampling methods; different data sampling point indices within a lookup table yielded consistent error rates within 2.7%p range as opposed to 17%p of the baseline derating method. By sampling from 4 points, the characterization effort of a corner could be reduced by 92%.
3) **Application to timing constraints**: DPVVM-MDR can also decompose and scale timing constraints from clock and data signals. We achieved an average timing estimation error of 15%—or an improvement of 59% over the conventional derating method for timing constraints.
4) **Compatibility to conventional delay models**: DPVVM-MDR scales T_{PD} and T_{tran} information—the most basic timing information required by commercial timing analysis tools—from base corners and is thus compatible with most delay models, i.e., *NDLM*, *CCS*, and *ECSM*.

We organized the following sections as follows. Section II includes preliminaries, the industrial context where cross-corner timing estimation has become critical, and the previous works related to cross-corner timing estimation. Then, in Section III, we demonstrate a motivating example of the conventional timing estimation method and discuss its shortcomings. In Section IV, we describe our proposed approaches in detail. Section V contains our experimental setup and timing estimation results, as well as in-depth discussions on the results. Section VI includes comparisons of our methods to state-of-the-art approaches in several aspects. Finally, Section VII concludes the topic with proposals of future work.

II. BACKGROUND

A. DELAY MODELS AND INPUT THRESHOLD VOLTAGE

A *delay model* is a timing estimation model where propagation time from an input pin to an output pin is estimated. Delay models are the main components in *static timing analysis (STA)* to enable timely and accurate timing estimations since dynamic timing analysis using slow device-level circuit simulators such as *HSPICE* is infeasible in large-scale circuits. Several delay models have been proposed [5]–[10] and those prominently deployed in the industry use the identical framework of table lookup as presented in [8]. For instance, NLDM, the most representative model, contains propagation delay T_{PD} in function of the *effective load capacitance* C_l and the *input transition time* T_{tran}—the toggling time from one logical state to an opposite state. Waveform-based models such as CCS and ECSM differ from NLDM in that entries of lookup tables are waveform data instead of decimal values. Nonetheless, the acquired waveform is further processed to obtain T_{PD} and T_{tran}, and thus STA procedure is identical for most prominent delay models.

T_{PD}, as defined in the JEDEC standard [11], is defined as follows:

$$T_{PD} := T_{out}(\frac{1}{2}V_{DD}) - T_{in}(\frac{1}{2}V_{DD}), \tag{1}$$

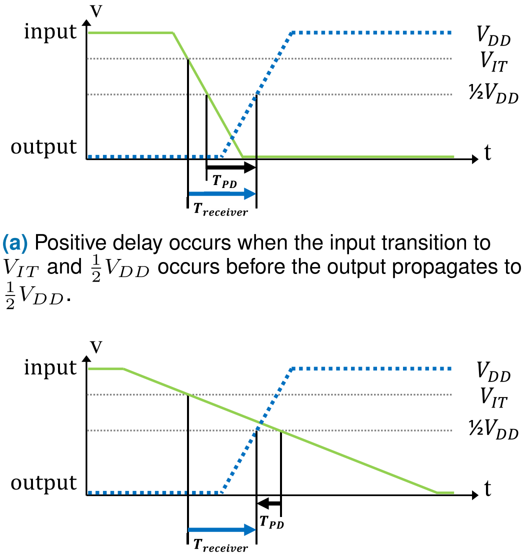
where $T_{in}(v)$ and $T_{out}(v)$ denote the input and output time of a given cell—referred to as a *receiver cell*—respectively, when the voltage equals v. T_{tran} is defined as follows, in consideration of both rise and fall transitions:

$$T_{tran} := |T_{in}(V_{upper}) - T_{in}(V_{lower})|, \tag{2}$$

where V_{upper} and V_{lower} represent upper and lower reference voltage level—typically expressed in proportion to V_{DD}—respectively.

The rationale for choosing $\frac{1}{2}V_{DD}$ as the reference voltage point is to be able to accumulate the delay of individual cells into the delay of a whole path. However, the actual driving of the receiver cell does not commence at $\frac{1}{2}V_{DD}$.

Each cell has a unique *input threshold voltage* (V_{IT}), the input voltage level at which the output propagation occurs [12]–[16]. To illustrate the importance of V_{IT} in precise delay representation, Fig. 2 plots examples of input and output waveforms of a rise delay. Here, we define V_{IT} as the input voltage at which the output voltage is $\frac{1}{2}V_{DD}$—that is, if we wait enough time after the input reaches V_{IT}, the output would eventually propagate to $\frac{1}{2}V_{DD}$. In Fig. 2(a), the output propagation occurs at V_{IT}; in the example, the input transition occurs rapidly and the difference between T_{PD} and propagation time from input V_{IT} to output $\frac{1}{2}V_{DD}$—denoted as $T_{receiver}$—seems insignificant. However, in Fig. 2(b), the input transition occurs slowly and the output propagation to $\frac{1}{2}V_{DD}$ occurs even before the input reaches $\frac{1}{2}V_{DD}$; this phenomenon is referred to as *negative delay*.



(a) Positive delay occurs when the input transition to V_{IT} and $\frac{1}{2}V_{DD}$ occurs before the output propagates to $\frac{1}{2}V_{DD}$.

(b) Negative delay occurs when the input transition to $\frac{1}{2}V_{DD}$ occurs after the output propagates to $\frac{1}{2}V_{DD}$.

FIGURE 2. Examples of an inverting signal propagation for a rise delay.

The notion of input threshold voltages is not something new and is found in earlier works, although its definition varies in the literature [12]–[16]. However, the scopes of the works are limited to conceptual introductions to V_{IT} and do not propose a method to apply it to existing STA flows, especially for path delay computations. In this paper, we adopt the concept of V_{IT} to establish a reliable delay variation model which applies to the conventional STA and thus to path delays.

It is worth noting that the driver cell's driving strength determines the driving of the input pin. As a result, T_{PD}, a combination of the driving of the driver cell and the receiver cell, is non-monotonic and unpredictable across T_{tran}, C_l, and especially across PVT corners, as will be discussed in Section III. Our proposed cross-corner variation model, DPVVM, decomposes propagation delay by its driving sources and independently scales them to provide accurate and consistent delay estimates.

B. CORNER EXPLOSION AND TIMING CLOSURE

In the past, foundries offered scaling-based cross-corner timing estimation methods—within proprietary documents—with empirically derived *derating factors* to reflect timing variability due to PVT variations. The timing closure was performed on the nominal corner, and the derating factors are used to estimate the worst operating condition and the timings therein.

Then, due to the advancement in process technologies and the emergence of design technologies such as power gating, voltage islands, and fine-grained dynamic voltage frequency scaling [17]–[19], timing variability due to process, voltage, and temperature variations became critical to the timing closure efficiency of logic circuits. The industry adopted *multi-corner multi-mode (MCMM)* timing analysis and transitioned from the derating to performing sign-off at every operating condition [1], [20], [21]. MCMM analysis requires standard cell libraries characterized at a set of PVT corners for each of the given operating voltage modes.

The number of PVT corners was not a concern until recently. However, in sub-micron era, timing closure problem began to suffer from a phenomenon referred to as the *corner explosion*, where the timing variability increases due to the process variations [1], [2], [22], voltage variations [23], and temperature variations [1], [24]–[26]. To further exacerbate the corner explosion, new dimensions in process variations—e.g., metal layers and lithography–are adding up in advanced process technologies, increasing the number of corners exponentially [2]. In a decade-old 28nm node, for instance, a combination of 5 base layer process corners—*slow-slow, slow-fast, fast-slow, fast-fast, normal-normal*—5 metal process corners—*Cmax, Cmin, typical, RCmax, RCmin*—5 voltage corners, and 5 temperature corners produces 625 corners, for the nominal operating mode alone.

In short, under the increasing influence of PVT variations and recent process and design technologies, the preparation and the updating of the required standard cell libraries on an overwhelming number of PVT corners are becoming problematic [2]. We claim that it is time to revisit the derating methods and seek a method to improve them; we propose an accurate delay variation model inspired by the conventional derating model to estimate timing information across corners.

C. CROSS-CORNER VARIATION MODELS

In this paper, the term *cross-corner variation models* refers to methods to estimate propagation delay $T_{PD}{}'$ under a target corner OC^t from delay T_{PD} under pre-characterized base corner OC^b.

In academia, there are a few analytical approaches to reflect and estimate PVT variations [24]–[27]. While these variation models claim accurate estimation results, they do not cover all of the process, voltage, and temperature variations; these models typically target specific physical aspects of devices. Besides, analytical models must be often revisited and verified upon the emergence of new process technologies. Furthermore, most works omit the application on sequential cells; cross-corner timing estimation of timing constraints is crucial since their pass-fail characterization is time-consuming.

State-of-the-art variation models are based on deep learning or machine learning [4], [28]–[31]. These models claim accurate variation estimations across various operating conditions. However, the gain in characterization effort can be overshadowed by computational efforts such as network configuration, validation overheads, training, and inference computations. We shall further discuss the details of these works in Section VI along with the comparisons to our work.

In the industry, foundries and EDA vendors have been providing derating-based methods, which are still being deployed and studied [26], [32]–[34]. The derating method is more practical than the analytical models in that they are transparent to specific physical effects but are less capable in terms of accuracy since the derating is applied to the lump-sum T_{PD}, as mentioned earlier in Section II-A

One of the most primitive examples of derating found in the literature is from [35]; where derating factors—also denoted as *scaling factors* or *k-factors* in the literature—are given for each of P, V, and T variations without specifying exact operating conditions. However, it is more common to consider unique derating factors for each OC^t, for each cell within each OC^t, or for each timing arc within each cell. In this paper we refer to derating factors as being unique for each of T_{PD} and T_{tran}, cell c, and timing arc a as in [33]. For T_{PD} under given OC^b, c, a, T_{tran}, and C_l condition, the derating factor for OC^t is applied as follows:

$$T'_{PD}(OC^t, c, a, T_{tran}, C_l) = k_{PD}(OC^t, c, a) * T_{PD}(OC^b, c, a, T_{tran}, C_l), \quad (3)$$

where k_{PD} is the derating factor for T_{PD} which is typically derived as follows:

$$k_{PD}(OC^t, c, a) = central(\frac{T_{PD}(OC^t, c, a)}{T_{PD}(OC^b, c, a)}), \quad (4)$$

where $central(T)$ represents an arbitrary central tendency—e.g., mean, median, or mode—from data samples T. The same principles apply for T_{tran} and its derating factor, k_{tran}; while T'_{PD} is accumulated to estimate path delays, T'_{tran} becomes the input of the next cells. This straightforward variation model is inaccurate, as will be discussed in the following sections.

Analytical models are impractical since they are bound to specific physical effects; the derating method falls short of accurate results since it scales the lump-sum of delay components with separate driving sources. However, if we overcome the conventional derating method's limitation, the method can be transparently applied to any operating conditions and most prominent delay models, with minimal characterization effort.

III. MOTIVATION

As a motivating example, we observed the accuracy of the derating method by case-studying T_{PD} under voltage variations, using SPICE simulations. Fig. 3 plots the rise delay—`t_pd`—of a NAND4 cell for different T_{tran} at the nominal supply voltage condition—0.8V—and at a target condition—1.0V—postfixed as `@0.8V` and `@1.0V`, respectively. Then the estimated delay—`t_pd_derate` was acquired by scaling the delay measured at 0.8V `t_pd@0.8V`; k_{PD} was obtained by using all the acquired data, and the mean value was chosen as a central tendency. By comparing `t_pd@0.8V` to `t_pd@1.0V`, what we observe are that these two curves are not inter-scalable; they have different tipping points, and even worse, the delay at 1.0V exhibits

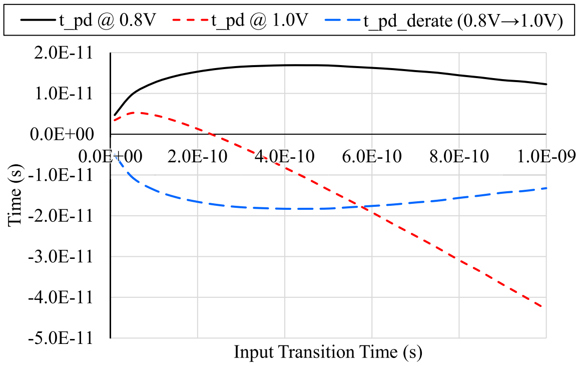
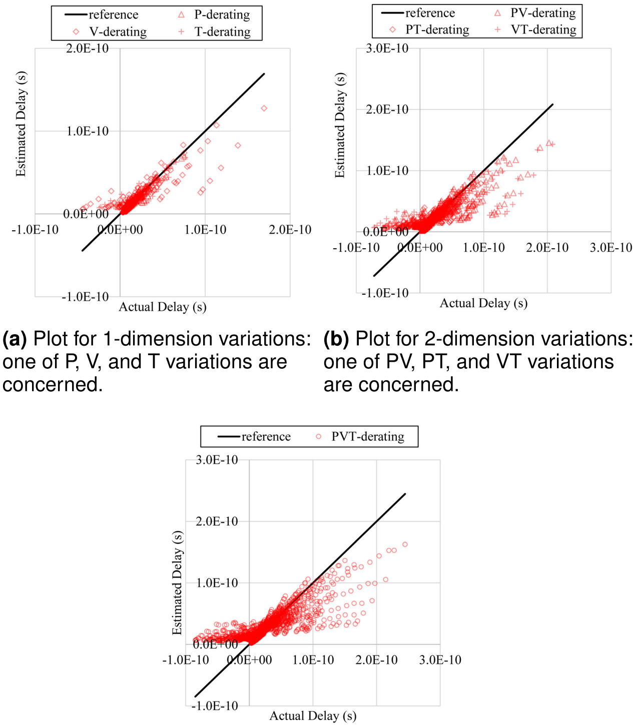


FIGURE 3. NAND4 rise delay at 0.8V, 1.0V, and derated from 0.8V to 1.0V.

negative delay, whereas the delay is always positive at 0.8V under the given T_{tran} range. Negative delay is a well-known phenomenon under slow transition; the implication behind the negative delay is that the driving of the cell commences before the input reaching $\frac{1}{2}V_{DD}$. In the given example, k_{PD} happened to be approximately -1.0 due to the dominance of negative delay and the resulting curve of `t_pd'` appears to be a flat mirror image as shown in Fig. 3. It becomes clear that T_{PD} is not suitable for derating over PVT variations—at least in extreme cases.

To further assess the accuracy of the conventional cross-corner timing estimation method, we performed the derating of NAND4 rise delay over all possible operating conditions. The setup including T_{PD} and T_{tran} are as specified in Section V-B. Fig. 4 compares the estimated delay value against the actual SPICE measurement at the target corners, under a different number of dimensions of variations—out of the P, V, and T domain. The reference curve $y = x$ is where estimated values match the actual values, so the closer a point is to the curve, the better the estimation is. The results were inaccurate; we observed a few points far-fetched from the reference curve, and the sign of the estimated delay was mismatched when the actual delay was negative. The error worsened when multiple variation dimensions are combined, manifesting the most inconsistency when variation occurs in all three dimensions of P, V, and T domains, as shown in Fig. 4(c).

It is important to note that *mean absolute percentage error (MAPE)*—the most general error metric—is inadequate to represent the accuracy of delay models and can be misleading in some cases. MAPE is infamous for boosting its value from outliers when actual values are close to zero; *absolute percentage error (APE)* diverges to infinity near 0 due to division-by-zero [36]. In cross-corner timing estimations, a few outliers with significant errors in the near-zero delay region may overshadow an entire dataset's accuracy. For each dataset shown in each of Fig. 4(a), (b), and (c), we observed MAPEs of 130%, 82%, and 100%, respectively and maximum absolute percentage errors of whopping high 23,000%, 13,000%, and 41,000%, respectively. The figures are misleading because the data in Fig. 4(c) seem more diffused. To alleviate the overwhelming effect of near-zero outliers with significant errors, we tried excluding the data points with a sub-picosecond delay, which accounted for approximately 1.0% of the entire dataset. We then observed MAPEs of 42%, 49%, and 54%, respectively, and maximum absolute percentage errors of 1,400%, 2,100%, and 2,000%, respectively; the figures and their relative magnitude drastically differ from the previous ones, implying the inconsistency of MAPE due to near-zero outliers.



(a) Plot for 1-dimension variations: one of P, V, and T variations are concerned.

(b) Plot for 2-dimension variations: one of PV, PT, and VT variations are concerned.

(c) Plot for 3-dimension variations: PVT variations are all concerned.

FIGURE 4. Estimated delay plotted against actual SPICE measurements, across different number of PVT dimensions.

The derating method's inaccuracy is due to the error in the timing from which the conventional delay models measure the delay. Fig. 5 shows the concept of our proposed variation model, where we decompose the delay propagation into the propagation time caused by the receiver cell itself—denoted as `t_receiver`—and the propagation time caused by its driver cell—denoted as `t_driver`. We observed that `t_receiver` trends are monotonically increasing and positive throughout input transition time as opposed to `t_pd`. In contrast to `t_pd@0.8V` and `t_pd@1.0V`, `t_receiver@0.8V` and `t_receiver@1.0V` are highly correlated; `t_receiver'`— $T'_{receiver}$ scaled from `t_receiver@0.8V`—resulted in an MAPE of 4% when compared to `t_receiver@1.0V`. `t_receiver'` is then added to `t_driver` to yield `t_pd'`, an estimation of `t_pd@1.0V`. With accurately fitted `t_pd'`, we achieved an MAPE of 14% as compared to 300% using conventional derating as depicted in Fig. 3. Note the jump in MAPE from

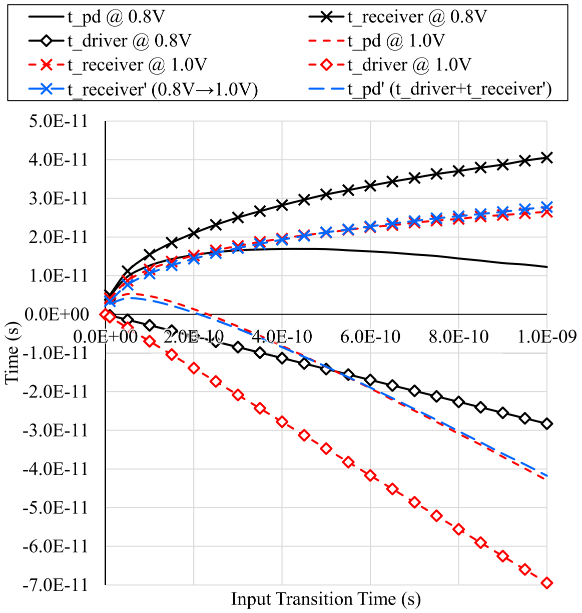


FIGURE 5. NAND4 rise delay at 0.8V, 1.0V, $T'_{receiver}$ scaled from 0.8V to 1.0V, and T'_{driver} derived from input T_{tran}.

4% to 14% after recomposition; again, this is owing to the error-boosting near-zero outliers.

This section showcased how decomposing T_{PD} by the driving source effectively mitigates negative delay and yields higher accuracy in cross-corner timing estimations. We shall discuss the details on the decomposition and recomposition methodology further in the following section.

IV. DECOMPOSED PROPAGATION VECTOR VARIATION MODEL

A. PROPAGATION VECTOR AND DECOMPOSITION

This section establishes the relationship between the timing and voltage propagation of logic gates by vectorizing them. We first define *propagation vector* as follows:

$$\vec{P} := \Delta v \cdot \hat{V} + \Delta t \cdot \hat{T}, \tag{5}$$

where $\hat{V}$ and $\hat{T}$ denote unit vectors in voltage axis and time axis, respectively, and Δv and Δt denote the propagation in voltage and time domain, respectively. Using the notation, we vectorize the propagation delay—defined in Equation (1)—as follows:

$$\vec{P}_{delay} := (\frac{1}{2}V_{DD} - \frac{1}{2}V_{DD}) \cdot \hat{V} + T_{PD} \cdot \hat{T} = T_{PD} \cdot \hat{T} \tag{6}$$

Then we introduce the concept of V_{IT} and use it as a reference point to decompose $\vec{P}_{delay}$ into a driver-driven portion—$\vec{P}_{driver}$— and a receiver-driven portion—$\vec{P}_{receiver}$. In DPVVM, we define V_{IT} as the input voltage at which the output voltage reaches $\frac{1}{2}V_{DD}$. If a driver cell is to drive a receiver cell having the same V_{IT}, it seems more reasonable to define V_{IT} as the crossing voltage where the input voltage matches the output voltage, defining T_{PD} as the delay from input V_{IT} to output V_{IT} [13]. However, this is unrealistic; every pin in every cell has its unique input voltage threshold, and therefore, the model must comply with different cells with different V_{IT}. In this perspective, fixing the output voltage to $\frac{1}{2}V_{DD}$ facilitates the recomposition of T_{PD} for STA, as will be shown shortly.

Now that the reference voltage point—V_{IT}—is defined, we define *driver propagation vector*—propagation of a receiver cell in input pin, driven by its driver cell:

$$\vec{P}_{driver} := V_{driver} \cdot \hat{V} + T_{driver} \cdot \hat{T}, \tag{7}$$

where

$$V_{driver} = V_{IT} - \frac{1}{2}V_{DD} \tag{8}$$

$$T_{driver} = T_{in}(V_{IT}) - T_{in}(V_{DD}) \tag{9}$$

Likewise, we define *receiver propagation vector*—propagation of the receiver cell from its input pin to output pin, driven by the receiver cell itself:

$$\vec{P}_{receiver} := V_{receiver} \cdot \hat{V} + T_{receiver} \cdot \hat{T}, \tag{10}$$

where

$$V_{receiver} = \frac{1}{2}V_{DD} - V_{IT} \tag{11}$$

$$T_{receiver} = T_{out}(\frac{1}{2}V_{DD}) - T_{in}(V_{IT}) \tag{12}$$

Combining Equation (7) and Equation (10) with Equation (1) yields:

$$\vec{P}_{driver} + \vec{P}_{receiver} = \vec{P}_{delay} \tag{13}$$

$$\therefore T_{driver} + T_{receiver} = T_{PD} \tag{14}$$

By decomposing $\vec{P}_{delay}$ into $\vec{P}_{driver}$ and $\vec{P}_{receiver}$, we aim to scale them separately then combine them back to recompose $\vec{P}_{delay}$ on a non-characterized PVT corner.; we derive $\vec{P}_{driver}$ with linear scaling of the input T_{tran} and we compute $\vec{P}_{receiver}$ by scaling from the reference PVT corner.

Before further discussions, it is essential to note that $\vec{T}_{receiver}$ should be, in theory, always positive when the output voltage waveform is a monotonic function near $\frac{1}{2}V_{DD}$. As defined earlier in this section, V_{IT} is the input voltage level at which the output voltage is $\frac{1}{2}V_{DD}$; by the very definition of V_{IT}, the input must preemptively reach V_{IT} before the output reaches $\frac{1}{2}V_{DD}$. Our objective is to obtain the trend of $\vec{T}_{receiver}$ as being inter-scalable across corners, so it is desirable that $\vec{T}_{receiver}$ waveforms exhibit positive values and monotonic trends to input T_{tran} and C_l, as seen in Fig. 5. In fact, under an exhaustive examinations on all $(OC^t, c, a, T_{tran}, C_l)$ combinations specified in Section V-B, we always observed positive values for $\vec{T}_{receiver}$. Such consistency, with reliable experimental results, proves that our tentative definition of V_{IT} was appropriate for our purpose.

To understand the relationship between the propagation vectors and the implications under each vector's direction, Fig. 6 depicts possible propagation scenarios of an inverting delay categorized by propagation conditions: rise/fall delay,

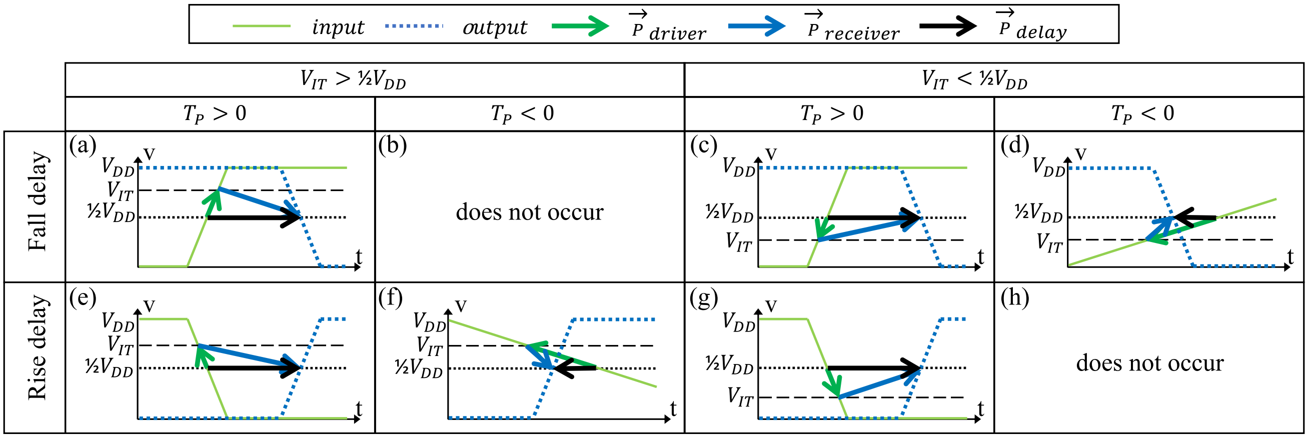


FIGURE 6. Possible combinations of delay propagation with various timing arc, input transition time, and V_{IT} conditions.

the superiority of V_{IT} over $\frac{1}{2}V_{DD}$, and the sign of T_{PD}. By definition, T_{driver} is positive when $T_{in}(V_{IT}) > T_{in}(\frac{1}{2}V_{DD})$—as are the cases in Fig. 6(a), (b), (g), and (h)—and negative otherwise—as are the cases in Fig. 6(c), (d), (e), and (f). In Fig. 6(a) and (g), T_{driver} and $T_{receiver}$ are both positive, thus T_{PD} is always positive. On the contrary, in Fig. 6(c), (d), (e), and (f), T_{driver} is negative; if its magnitude is smaller than that of $T_{receiver}$, T_{PD} is still positive—as in Fig. 6(c) and (e). Negative delays occur if and only if T_{driver} is negative and its magnitude outmatches that of $T_{receiver}$—as shown in Fig. 6(d) and (f). The cases in Fig. 6(b) and (h) do not occur since negative delay is not possible when T_{driver} is positive.

In short, the effect of $\vec{P}_{driver}$ may obscure the effect of $\vec{P}_{receiver}$ causing inconsistent or even negative T_{PD} in some conditions.

B. VARIATION MODEL CHARACTERIZATION

In this subsection, we discuss how DPVVM is characterized. Our DPVVM libraries contain V_{IT} for both OC^b and OC^t—denoted as V_{IT}^b and V_{IT}^t, respectively, in this paper.

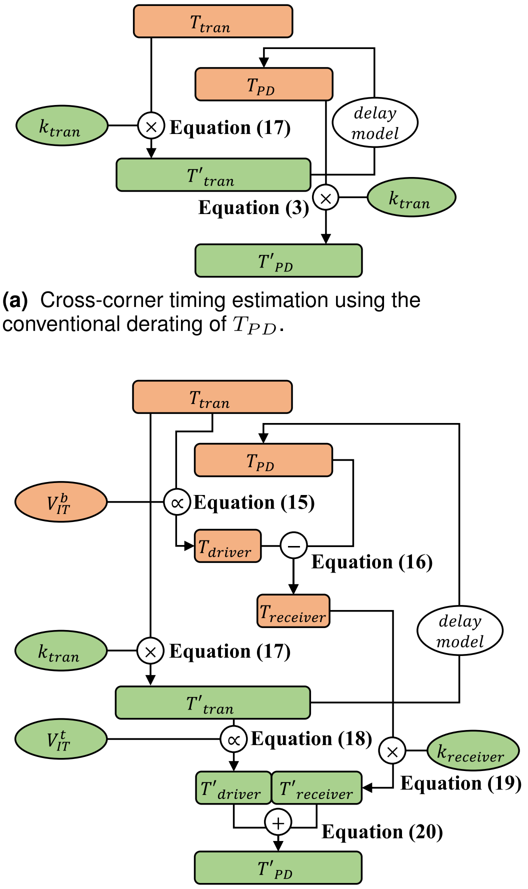
V_{IT} acts as a reference voltage point which allows the derivation of T_{driver} and $T_{receiver}$ using T_{tran} and T_{PD}, respectively. This property makes our variation model applicable to mainstream delay models—namely NLDM, CCS, and ECSM—as they all eventually generate T_{tran} and T_{PD} during STA, the data to which DPVVM is applied. For a given T_{tran}, T_{driver} can be computed using proportionality as follows:

$$T_{driver} = \frac{V_{driver} * T_{tran}}{V_{tran}} = \frac{(V_{IT}^b - \frac{1}{2}V_{DD}) * T_{tran}}{V_{tran}}, \qquad (15)$$

where $V_{tran} = V_{upper} - V_{lower}$. Then for given for OC^b and OC^t, $T_{receiver}$ can be derived from T_{driver} and T_{PD} as follows:

$$T_{receiver} = T_{PD} - T_{driver} \qquad (16)$$

Now that we have $T_{receiver}$ for OC^b and OC^t, we can derive $k_{receiver}$ for OC^t, in a way equivalent to Equation (4). The derating method and DPVVM also require k_{tran}, a derating factor for T_{tran}.



(a) Cross-corner timing estimation using the conventional derating of T_{PD}.

(b) Cross-corner timing estimation using DPVVM.

FIGURE 7. Workflow for deriving intermediate variables and cross-corner scaling.

C. CROSS-CORNER TIMING ESTIMATION

In STA, our goal is to derive T'_{PD} by estimating T'_{driver} and $T'_{receiver}$ at OC^t; its overall flow is summarized in Fig. 7 with corresponding equation numbers. As depicted in Fig. 7(a), cross-corner timing estimation flow in the conventional derating involves the scaling of T_{tran} and T_{PD}. It should be noted that T_{PD} is derived from T'_{tran} using a given delay model; the process is represented by *delay model* in the figure. The estimation flow in our proposed DPVVM is depicted in Fig. 7(b). The most significant difference between these approaches is that DPVVM scales $T_{receiver}$ instead of T_{PD}; the rest of the steps involves the derivation of T_{driver} to retrieve $T_{receiver}$, scaling, then recomposing into T'_{PD}.

It is possible to determine T'_{driver} using proportionality, so T_{driver} does not require a derating factor other than k_{tran} from the previous cell, which is used to yield T'_{tran}:

$$T'_{tran} := k_{tran} * T_{tran} \tag{17}$$

T'_{driver} is derived in a way similar to Equation (15):

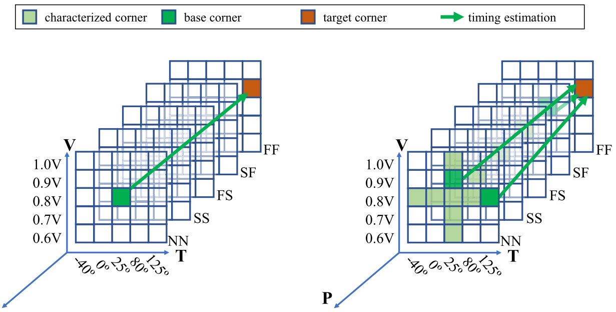
$$T'_{driver} = \frac{(V^t_{IT} - \frac{1}{2}V_{DD}) * T'_{tran}}{V_{tran}} \tag{18}$$

Then, $T'_{receiver}$ is estimated by derating $T_{receiver}$ with $k_{receiver}$:

$$T'_{receiver} = k_{receiver} * (T_{PD} - T_{driver}) \tag{19}$$

It is important to note that T_{driver}—not T'_{driver}—is used in Equation (19) since the decomposition is performed to $T_{receiver}$ in OC^b. Finally, we obatin T'_{PD}:

$$T'_{PD} := T'_{driver} + T'_{receiver} \tag{20}$$



(a) Cross-corner estimation from a single base corner.

(b) Cross-corner estimation from multiple base corners using MDR.

FIGURE 8. **Cross-corner timing estimations from base corner(s) to a desired target corner, plotted on PVT dimension.**

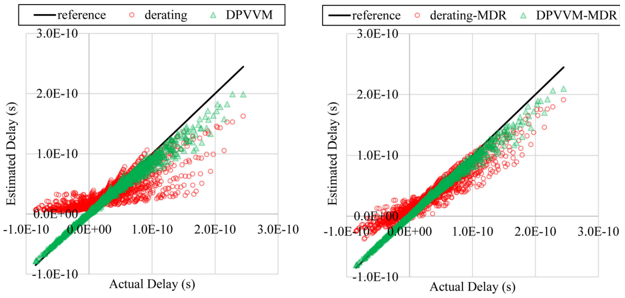
D. RECOMPOSITION FROM MULTIPLE DIMENSIONS

In the previous subsections, we assumed the timing estimation from a single base corner. The concept is visualized in Fig. 8(a), where $OC^b = (NN/0.8V/25\,°C)$ and $OC^t = (FF/0.9V/125\,°C)$. To further improve the estimation accuracy, we propose full-characterizations—if not already done—on single-dimension PVT variation conditions—conditions where only one of P, V, or T variation occurs. Then, upon multi-dimension variation, we estimate timings from more than one pre-characterized base corner; the proposed concept is visualized in Fig. 8(b), where pre-characterized libraries exist for each of P, V, and T variation. In the example, cross-corner timing estimation is done from each base corners denoted as follow: $OC^{b,P} = (FF/0.8V/25\,°C)$, $OC^{b,V} = (NN/0.9V/25\,°C)$, and $OC^{b,T} = (NN/0.8V/125\,°C)$.

We apply a simple averaging of timing from applicable base corners as follows:

$$T'_{PD} = average(T'^{P}_{PD}, T'^{V}_{PD}, T'^{T}_{PD}), \tag{21}$$

for applicable dimensions where $T'^{cond.}_{PD}$ refers to T_{PD} estimated from $OC^{b,cond.}$ to OC^t using unique derating factors for each of $OC^{b,cond.}$ and OC^t combinations. We shall denote the method—applicable to both derating and DPVVM—as *Multi-Dimension Recomposition (MDR)*.



(a) Cross-corner estimation without MDR, from a single base corner without variation.

(b) Cross-corner estimation of MDR, from multiple base corners with 1-dimention variations.

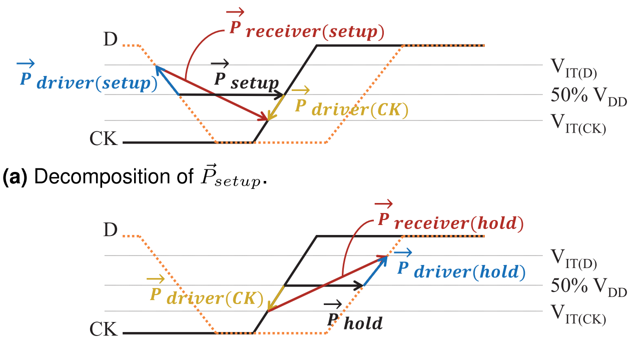
FIGURE 9. **Estimated delay plotted against actual HSPICE measurements with and without MDR, under mult-dimension variations.**

Fig. 9(a) and (b) plots the accuracy of estimated delay from a single OC^b and using MDR, respectively, for multi-dimensional cross-corner estimation. First, in Fig. 9(a), we observed remarkable accuracy gain with `DPVVM` alone, allowing minimal error even for the negative delay. As for the impact of MDR, as shown in Fig. 9(b), we observed significant improvements to `derating-MDR` over `derating` even estimating negative delay to some extent. However, the accuracy of the best-performing `DPVVM-MDR`, by bringing the best of both worlds, fit the reference curve better than `derating-MDR`.

The rationale of characterizing multiple base corners under single-dimension variations is as follows. First, the number of base corners increases linearly with the introduction of new variation parameters, whereas the number of total corners increases exponentially; the characterization effort of base corners is always disproportionately lower. Second, we expect that most users would choose to characterize most single-dimension corners, if not already present. In most cases, MDR can exploit existing pre-characterized corners to enhance estimation accuracy without further overheads. Finally, we demonstrate in Section V-D that DPVVM-MDR is more accurate than DPVVM under comparable characterization effort; it is far more beneficial to increase the number of base corners instead of the number of data samples.

E. APPLICATION TO TIMING CONSTRAINTS

Characterization of timing constraints such as setup and hold time is especially problematic in cell characterization due to substantial computational effort. This subsection extends our



(a) Decomposition of $\vec{P}_{setup}$.

(b) Decomposition of $\vec{P}_{hold}$.

FIGURE 10. Decomposed vectors of a flip-flop; acquisition of $\vec{P}_{setup}$ and $\vec{P}_{hold}$ slightly differ.

notion of propagation vectors into sequential cells to enable cross-corner estimations of timing constraints.

A timing constraint of cell is measured as the minimum or maximum propagation time between a data pin and the clock pin; like for propagation delay, it is important to decompose the driving time of its input cells—clock and data—to obtain meaningful timing components that are scalable across corners. Decompositions of setup and hold time are depicted in Fig. 10; we define signal propagations involving setup and hold time, $\vec{P}_{setup}$ and $\vec{P}_{hold}$, respectively, each of which is decomposed into three vectors as follows:

$$\vec{P}_{setup} = \vec{P}_{driver(setup)} - \vec{P}_{driver(CK)} + \vec{P}_{receiver(setup)} \tag{22}$$

$$\vec{P}_{hold} = \vec{P}_{driver(CK)} - \vec{P}_{driver(hold)} + \vec{P}_{receiver(hold)}, \tag{23}$$

where $\vec{P}_{receiver(setup)}$ and $\vec{P}_{receiver(hold)}$ each denotes the receiver propagation vector for setup time and hold time, respectively; $\vec{P}_{driver(setup)}$ and $\vec{P}_{driver(hold)}$ each denotes the driver propagation vector for setup time and hold time of data pin, respectively; finally, $\vec{P}_{driver(CK)}$ denotes driver propagation vector of clock pin.

The scaling of the timing constraints are, in principle, similar to that of propagation delays:

$$\begin{aligned} T'_{setup} = {} & T'_{driver(setup)} - T'_{driver(CK)} \\ & + k_{receiver(setup)} * (T_{setup} - T_{driver(setup)} + T_{driver(CK)}) \end{aligned} \tag{24}$$

$$\begin{aligned} T'_{hold} = {} & T'_{driver(CK)} - T'_{driver(hold)} \\ & + k_{receiver(hold)} * (T_{hold} - T_{driver(CK)} + T_{driver(hold)}) \end{aligned} \tag{25}$$

For instance, for setup time, $T_{receiver(setup)}$ is scaled using $k_{receiver(setup)}$; $T_{driver(setup)}$ and $T_{driver(CK)}$ are derived in proportion to T_{tran} of data and clock pin, respectively, using Equation (19). The scaling accuracy of timing constraints will be discussed shortly in Section V.

F. COMPATIBILITY WITH CONVENTIONAL DELAY MODELS

There are mainly two types of variation models: separate variation models which induce modifications to STA—e.g., [29], [31] OCV, AOCV and POCV [1]—and library generation methods—e.g., [4], [28]— which generate generic libraries with estimated timings. In contrast, our proposed model may fall into both categories. DPVVM is a separate variation model independent of delay models since the decomposition and scaling occur at STA-stage after T_{PD} and T_{tran} are computed. It is thus compatible with many prominent delay models, namely NLDM, CCS, and ECSM. Our model can also generate NLDM libraries at non-base corners by applying the scaling to readily available T_{PD} and T_{tran} in the lookup tables.

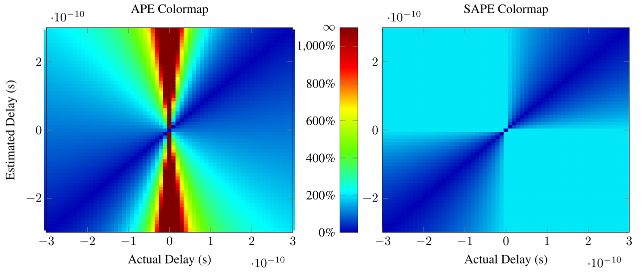


FIGURE 11. Colormaps of APE(left) and SAPE(right); APE diverges to infinity when the actual value reaches 0.

Furthermore, the concept of decomposition itself may find its place in state-of-the-art variation models based on machine learning or deep learning such as [4], [28]–[31], although the impact is to be assessed in future works. Decomposition of T_{PD} yields monotonic and predictable timing components; this process is equivalent to pre-processing input data in machine learning. For instance, *Aadam* [31] is a aging-aware variation model which applies deep learning to STA flow. In its timing characterization phase, Aadam characterizes *aging-aware cell delay dataset*; the model may be combined with our decomposition approach to tentatively improve its accuracy.

V. EXPERIMENTAL RESULTS

A. ERROR METRIC

As discussed in Section III, the conventional error metric—MAPE—is inadequate for delay estimations due to outliers or an undefined value in the near-zero region. As our error metric, we adopt *asymmetric mean absolute percentage error (SMAPE)*, which is the mean of *symmetric absolute percentage error (SAPE)* [37]; we shall use a variant of SAPE which is widely used in academia [38]–[42] in order to mitigate division-by-zero problem defined as follows:

$$SAPE = \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}, \tag{26}$$

where A_t and F_t each denotes actual value—equivalent to actual delay in this paper—and forecast value—equivalent to estimated delay in this paper—respectively, for an arbitrary datapoint t. Compared to MAPE, SMAPE is less prone to outliers for SAPEs are upper-bounded to 200%. To highlight the characteristics of APE and SAPE, Fig. 11 plots their error colormaps for different actual and estimated delay values.

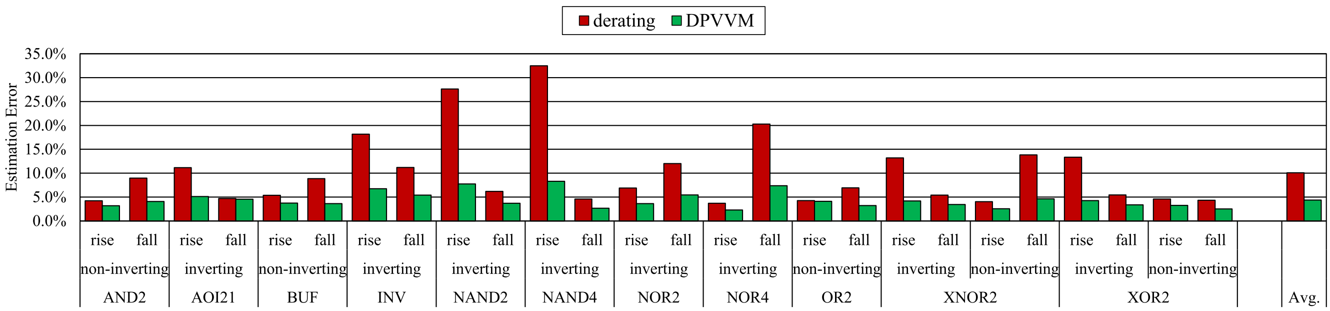


FIGURE 12. Cross-corner timing estimation errors of combinational logic cells under single-dimension PVT variations.

SAPE provides an upper-bound error of 200% for negative error, and the map is symmetrical to $F_t = A_t$, whereas APE diverges to infinity when A_t is near zero and the map is asymmetrical; the error sensitivity differs to the sign difference of F_t offset. It should be noted that in this paper, undefined error—due to zero denominator—is mitigated by adding a tiny constant(1e-30) to the denominator of SAPE.

B. EXPERIMENT SETUP

To highlight the accuracy of DPVVM, we performed a series of comparisons to the baseline derating method by performing cross-corner timing estimations on various operating conditions. We used the *Synopsys Finesim P-2019.06-SP2* simulator for SPICE simulation of the pre-layout standard cells coupled with a custom *predictive technology model (PTM)* fitted to *Intel 14nm* process technology acquired from [43].

Types of parameters and their respective configurations are described in Table 1. As variation models, the baseline derating and our proposed DPVVM are compared.

TABLE 1. Applicable parameters and respective configurations in our experiments.

Parameter	Configuration
Variation Models	derating, derating-MDR, DPVVM, DPVVM-MDR
Cell Type	AND2, AOI21, BUF, INV, NAND2, NAND4, NOR2, NOR4, OR2, XNOR2, XOR2
Input T_{tran}	8ps - 1ns
C_l	fan-out 1 - 40 (effective capacitance of an inverter)
Process Variation	SS, SF, NN, FS, FF
Voltage Variation	0.6V, 0.7V, 0.8V, 0.9V, 1.0V
Temperature Variation	$-40\,°C$, $0\,°C$, $25\,°C$, $80\,°C$, $125\,°C$

For both methods, derating factors—k_{PD} for derating and $k_{receiver}$ for DPVVM;k_{tran} for both—are uniquely derived for each of $(OC^t, c, a, T_{tran}, C_l)$ condition. Both are configured both with and without MDR to distinguish its impact on the accuracy from DPVVM.

We performed simulations over a variety of standard cells; both inverting and non-inverting timing arcs are also considered if applicable. We also evaluated 4-input configurations for NAND and NOR cells.

Lookup tables consist of combinations of 8 input T_{tran} and 5 C_l values. We validated the variation models on 124 non-base corners—12 corners with 1-dimension variations, 48 corners with 2-dimension variations, and 64 corners with 3-dimension variations. In MDR, 12 corners with 1-dimension variations become base corners; we validated the models on 112 non-base corners.

Unless otherwise stated, we neglected the effect of data sampling by using all data points in the lookup tables. For setup/hold timing constraints, a DFF cell is used with 8 data and 8 clock transition time values.

C. TIMING ESTIMATION RESULTS

1) ESTIMATION ACCURACY ACROSS LOGIC GATES

We first examine the estimation accuracy across a variety of logic gates. Fig. 12 shows the accuracy of single-dimension cross-corner timing estimations compared to the baseline derating. Derating had an estimation error of 10%; the rise error rate was overwhelmingly high for `INV` and `NAND` cells. The rate is due to the high driving strength of the PMOS transistor; the rise transition in `INV` and `NAND` involves the transition of a single PMOS device. The high error rate primarily originates from fast transition errors. In comparison, the driving of a single NMOS device, represented by the `NOR` cell's fall transition, is less vulnerable to error. Furthermore, transitions involving multiple transistors showed relatively low error for derating.

In contrast to the derating, DPVVM successfully estimated the timing variety in a consistent manner, outperforming the baseline derating in every cell. DPVVM achieved an average SMAPE of 4.3%, an improvement of 56% over the baseline 10%.

2) ESTIMATION ACCURACY ACROSS SOURCES OF VARIATIONS

We now examine how the estimation error is affected by various sources of variations by dissecting the estimation error into each of the PVT corners; the results are shown in Fig. 13.

For process variations, we observed that the derating errors were relatively high for `FS` and `SF` corners, which are heavily impacted by negative delays; DPVVM was able to successfully estimate negative delays. Derating errors were relatively low for `FF` and `SS` corners with marginal improvements by

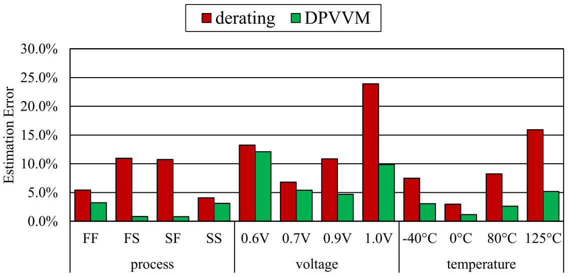
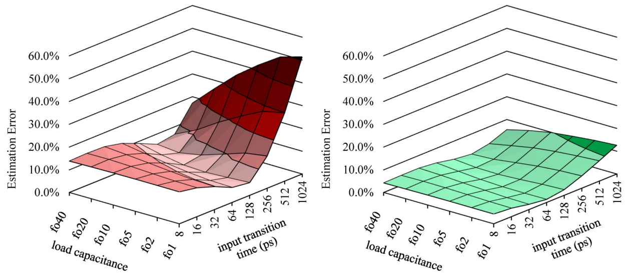


FIGURE 13. Cross-corner timing estimation errors for different sources of variations under single-dimension PVT variations.

DPVVM. Overall, for process variations, DPVVM achieved an average error rate of 2.0%, which is an improvement of 74% over the baseline of 7.8%.

For voltage variations, we observed higher derating error for higher voltages, caused by faster transitions and thus negative delays. DPVVM error appeared to be proportional to voltage offset; DPVVM suffered the most under voltage variations, probably due to the large gap in V_{IT}. Overall, for voltage variations, DPVVM achieved an average SMAPE of 8.0%, which is an improvement of 42% over the baseline of 13%.

For temperature variation, DPVVM provided consistently accurate estimation results; we achieved an average SMAPE of 3.0%, which is an improvement of 65% over the baseline of 8.7%.



(a) Derating error surface curve **(b)** DPVVM error surface curve

FIGURE 14. Cross-corner timing estimation errors plotted across different T_{tran} and C_l conditions, under single-dimension PVT variations.

3) SENSITIVITY TO SAMPLING POLICIES

To show the impact of sampling, we begin by comparing the accuracy error across different T_{tran} and C_l pairs, as shown in Fig. 14. The larger the error gap, the more sensitive the error is to the choice of samples since scaling factors are derived in a way that minimizes the scaling error of the chosen sample. In effect, the choice sampling policy affects the location of saddle points—the points with the lowest error rate—of the surface curves, although its impact on the actual shape itself is marginal. The surface curve in Fig. 14(a) cannot become as flat as in Fig. 14(b) no matter what derating factor is chosen.

Derating—represented by Fig. 14(a)—shows a U-shaped curve, peaking errors in high T_{tran} and low C_l region; selecting samples in the high-error region would reduce those peaks, but the error would increase in any other region. On the contrary, DPVVM—represented by Fig. 14(b)—shows a relatively flat curve, implying less sensitivity to sampling.

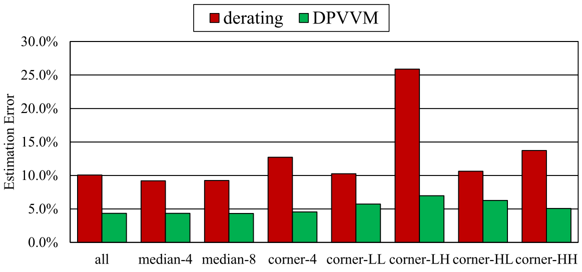


FIGURE 15. Cross-corner timing estimation errors for different sampling policies.

To prove the observations, we compared different sampling approaches to derive the derating factors, as shown in Fig. 15. Here, `all` is the baseline without sampling. `median-4` and `median-8` refer to the choice of 4 and 8 samples from the median, respectively. `corner-4` refers to the choice of 1 sample from each corner of the lookup table. Finally, `corner-XX` refers to the choice of 4 samples from an extreme corner; `LL` denotes low-C_l/low-T_{tran}, `LH` denotes low-C_l/high-T_{tran}, `HL` denotes high-C_l/low-T_{tran}, and `HH` denotes high-C_l/high-T_{tran} corner. SMAPE of derating fluctuated and ranged from 9.2% to 26%, resulting in high error-susceptibility to sampling policies. SMAPE of DPVVM, on the contrary, was very consistent throughout various sampling methods and ranged from 4.3% to 7.0%. These results align with the curves shown in Fig. 14, where low-C_l/high-T_{tran} corner struggled the most from inaccuracies.

4) ESTIMATION ACCURACY UNDER MULTI-DIMENSIONAL VARIATIONS

Until this point, the simulations covered single-dimension PVT variations only, assuming one of P, V, or T variations at a time. Fig. 16 shows the timing estimation error of both derating and DPVVM, with and without MDR. Through the application of DPVVM, we observed an average SMAPE of 7.7%, which is an improvement of 50% over the baseline 15%. It is interesting to note that although the estimation errors under multi-dimension variations were consistently more considerable than the errors under single-dimension variation, as shown in Fig. 12, the results in both conditions were highly correlated. Applying MDR to both derating and DPVVM yielded average error rates of 10% and 4.8%, respectively, which are improvements of 32% and 37$, respectively, over derating and DPVVM, respectively.

With MDR combined with DPVVM, our cross-corner timing estimation error was 4.8% on average, which is an improvement of 69% over the baseline derating method.

5) ESTIMATION ACCURACY OF TIMING CONSTRAINTS

We also validated DPVVM for the timing constraints of a DFF sequential cell, as depicted in Fig. 17 *Pass-fail*

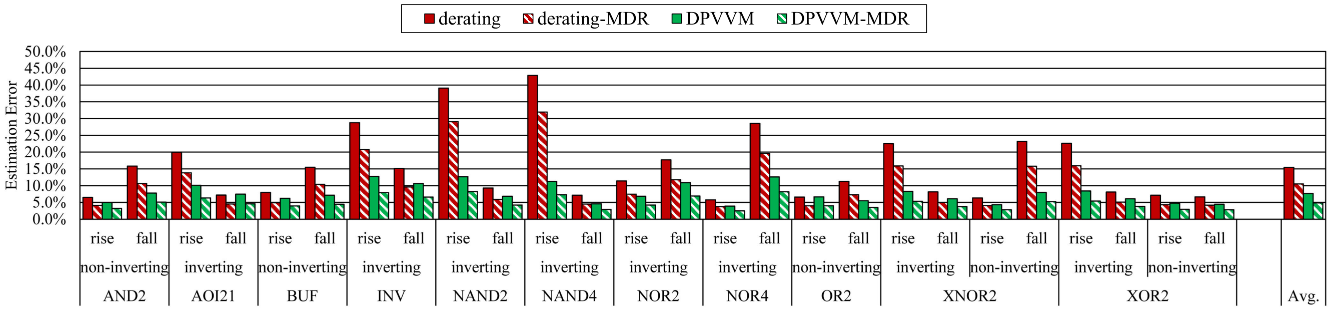
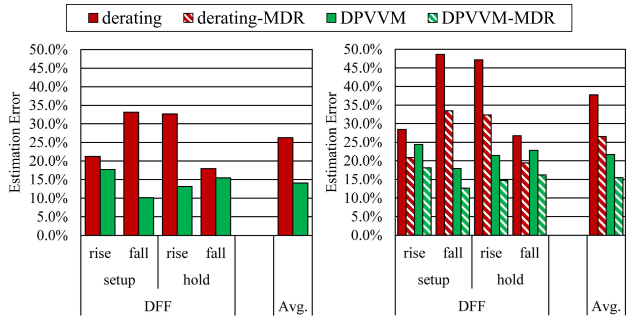


FIGURE 16. Cross-corner timing estimation errors of combinational logic cells under multi-dimension PVT variations.



(a) Single-dimension PVT variations

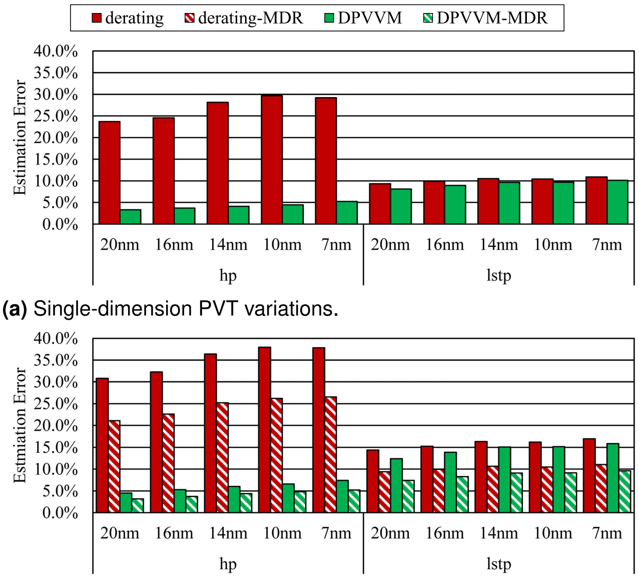
(b) Multi-dimension PVT variations

FIGURE 17. Cross-corner timing estimation errors of a sequential logic cell under single and multi-dimension PVT variations.

model is used as constraint style—which defines criteria for the judgment of simulations [44]—but primitive evaluations revealed similar tendencies with *delay-degradation* and *slew-degradation* style. Estimations of timing constraints rely on binary search for a pre-defined number of iterations; the reference SPICE timings are already rough estimations, limiting the cross-corner timing estimation accuracy.

Fig. 17(a) shows the timing estimation error under single-dimension variations. We observed a relatively high error rate using the derating method, especially for setup/fall and hold/rise combinations. For these combinations, negative constraints occurred for slow clock transition time; as repeatedly discussed in the previous sections, the derating was prone to sign flipping across corners, whereas DPVVM estimated the cases more accurately. For timing constraints under single-dimension variations, we achieved an average error rate of 14%, which is an improvement of 46% over the baseline 26%.

The trends in multi-dimension variations—as shown in Fig. 17(b)—again resembled that of single-dimension variations but with higher error rates. Upon applying MDR to derating and DPVVM, the error rate mirrored that of single-dimension variations, the observation of which is similar to that of propagation delay estimations. Applying MDR to both derating and DPVVM yielded average SMAPEs of 27% and 15%, respectively, which are improvements of 43% and 42$, respectively, over derating and DPVVM, respectively. With MDR combined with DPVVM, our cross-corner timing estimation error was 15% on average, which is an improvement of 59% over the baseline derating method.



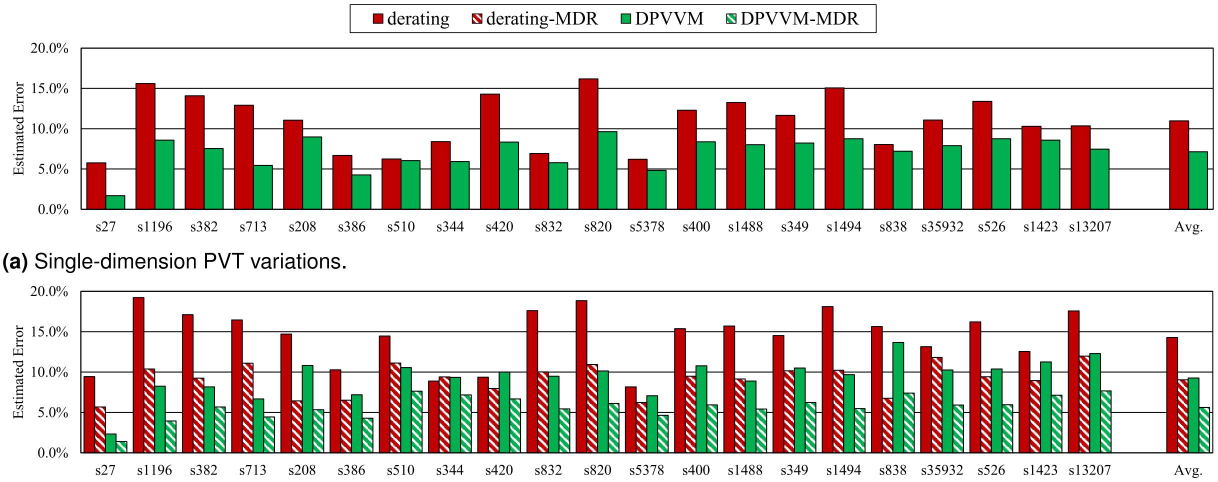
(a) Single-dimension PVT variations.

(b) Multi-dimension PVT variations.

FIGURE 18. Cross-corner timing estimation errors of NAND2 cell across technology nodes and device types.

6) ESTIMATION ACCURACY ACROSS TECHNOLOGY NODES

We also performed an analysis on how the accuracy of cross-corner timing estimation methods is affected by process technology. Fig. 18 shows average estimation error across different process nodes from *PTM-MG*, the latest PTM models [45], [46]. `hp` and `lstp` are device types and each represents *high-performance* and *low standby power* device technology, respectively. Overall, we observed a slight increase in estimation error for more advanced nodes, but the difference seemed relatively insignificant. However, it should be noted that PTM-MG assumes the same technologies; the same parameters were tuned differently to match the characteristics of node shrinking [46].



(a) Single-dimension PVT variations.

(b) Multi-dimension PVT variations.

FIGURE 19. Cross-corner timing estimation errors of ISCAS'89 benchmark circuits under single and multi-dimension PVT variations.

However, there were differences between `hp` and `lstp`. We observed that the estimation accuracy of `hp` is improved by applying DPVVM, whereas the improvements of `lstp` depended more on MDR. To understand the phenomenon, we further examined `NAND2` cell under single-dimension variations and observed that `hp` devices were much faster and thus had more negative delays: in overall, 15% of timings were negative, and we observed 7.6% sign mismatch upon derating. In contrast, for `lstp` devices, only 1.3% of timings were negative, and we observed 0.21% sign mismatch upon derating. For both `hp` and `lstp`, DPVVM-MDR had 0.28% and 0.23% sign mismatch and thus more consistent results.

7) ESTIMATION ACCURACY ON COMPLEX LOGIC CIRCUITS

Finally, we validate the accuracy of our proposed method in complex logic circuits. We used *ISCAS'89* benchmark circuits [47] and performed STA on *Synopsys PrimeTime K-2015.12-SP2*. The results are shown in Fig. 19; the circuits are presented in path length order, ranging from 5 to 139 gates. Under single-dimension variations, as shown in Fig. 19(a), DPVVM achieved an error rate of 7.1%, an improvement of 35% over the baseline 11% Under multi-dimension variations, DPVVM and MDR both contributed to yield approximately 9.3% and 9.0% estimation error, respectively; together, DPVVM-MDR achieved an SMAPE of 5.6%, an improvement of 61% over the baseline 14%.

D. CHARACTERIZATION EFFORT

To demonstrate our proposed method's efficiency, we analyze our variation model's characterization effort and compare it to the full-characterization without delay variation models. For both pre-characterized base corners and non-base corners, the characterization of V_{IT} through DC analysis is required, which is comparable to a single transient analysis and is negligible to timing constraints. The derivation of derating factors—$k_{receiver}$ and k_{tran}—is as simple as performing transient analysis on sample data points; the analysis is identical for the deviation of k_{PD} and k_{tran} for derating. Theoretically, without considering the imbalance in simulation time and DC analysis overheads, sampling 4 points out of 6 C_l and 8 T_{tran} pairs would require 8.3% of the simulation time. For timing constraints, sampling 4 points out of 8 T_{tran} values for each of the data and clock pin would require 7.1% simulation time. Real-time SPICE simulations required 9.8% and 5.1% of full simulation for propagation delay and timing constraints, respectively, including DC analysis, which was an increase by 11% over derating.

For the characterization of the standard cell library from *45nm Open Cell Library* [48], the characterization in 125 PVT corners—consisting of 1136 delay and 49 timing constraints lookup tables from the total of 135 cells—would require approximately 3200 hours. For DPVVM, the characterization time can be reduced to 280 hours or by 91%. For DPVVM-MDR, the characterization time can be reduced to 570 hours or by 82%.

It is worth mentioning that, as discussed in Section IV-D, we project that our variation model will be utilized to estimate timings at less-frequent PVT corners. To characterize the variation model for an extra PVT corner, our proposed methods—both DPVVM and DPVVM-MDR—only require 8.0% of full-characterization time—2.1 hours as compared to 26 hours for a full-characterization.

We also plotted the characterization effort of our methods against estimation error to discuss Pareto efficiency—trade-off between characterization effort and accuracy—of each method by sweeping the characterization effort[1] required to yield 125 corners. The plot is shown in Fig. 20; the closer

[1] Here, we swept the number of samples to yield the scaling factors; 100% characterization effort means all data points are computed, thus requiring no timing estimation.

TABLE 2. Comparison of estimation approaches of state-of-the-art models and our proposed models; our proposed simple derating-based models showcase the fastest time-to-market with low estimation overhead and technology dependency.

Prior Arts and the Proposed Models	DPVVM	DPVVM-MDR	E. Naswali et al. [29]	F. Klemme et al. [4]	P. Cao et al. [30]
Estimation Method	derating-based		deep learning	machine learning	machine learning
Computational Overheads	small		high	moderate	moderate
Technology Dependency	independent		unknown	dependent	unknown
Base/Target Corners	1/124	13/112	7/10	875/20,000	5/40
Improvement in Characterization Effort	91%	82%	40%	96%	unspecified
Sequential Cells	yes		no	no	no
Validation Scope	cell & path		cell only	cell & path	path only

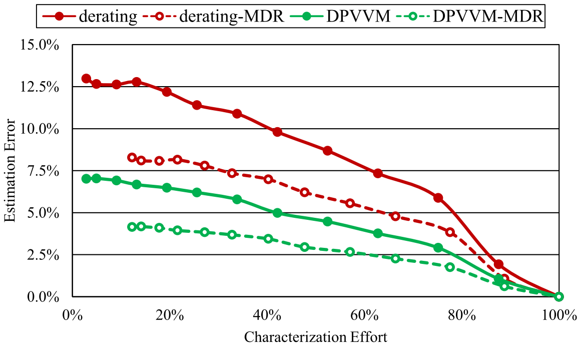


FIGURE 20. Cross-corner timing estimation errors for different characterization efforts—reflecting the number of samples and base corners—to generate a full library.

the curve is to the lower-left corner, the better its trade-off relationship is. Upon the adoption of DPVVM, characterization overhead is similar under the same sampling ratio, with significant accuracy improvement; Pareto improvement is obtained. As for the adoption of MDR, the choice is made between the number of samples versus more base corners for the same characterization effort. However, the accuracy of MDR was always superior than non-MDR models in all cases; again, Pareto improvement is obtained. We therefore recommend combining DPVVM and MDR for Pareto optimality, achieving the best trade-off relationship between the accuracy and characterization effort.

In this subsection, we discussed the characterization effort of DPVVM-MDR under the context of full-library characterization and an extra corner characterization. We also explored the trade-off between the accuracy and characterization effort—i.e., the number of samples and base corners—and showed that DPVVM-MDR was Pareto optimal, achieving the most accurate timing estimation in all cases.

VI. COMPARISON TO STATE-OF-THE-ART APPROACHES

In this part, we compare our proposed delay variation models to the state-of-the-art variation models [4], [28]–[30]. First, we compare various aspects of these works as summarized in Table 2. Our proposed models can be considered derating-based, with marginal computational overheads of DC analysis and library generation. Our models are also transparent to sources of variations, making them independent of process technologies. Also, our work is the only work in the group that considers sequential cells.

The model in [28] uses a convolutional neural network (CNN) to estimate timing in both inter-corner and intra-corner manner. This 40-layer fully connected network has the highest computational overheads of network configuration, training, and inference. Its technology dependence is at large, but it seems clear that even if the network can be reused on different process technologies, training and validation is obligatory. The model was validated on a fairly low number of target corners resulting in low improvement in characterization effort. Next, the authors in [4] use machine learning to generate libraries for design technology co-optimization (DTCO). Although the computational—i.e. training—overheads of their model are significant, the authors show that the inference effort is negligible compared to library compilation time. For its DTCO flow—sweeping of process parameters—the number of target corners was the highest in the comparison group; the improvement in characterization effort was 96%. Finally, the work from [29], [30] uses feature extractions from CNN; the inference overhead should be moderate in comparison to a fully connected neural network. The sampling rate for cross-corner timing estimation is unspecified.

We also compared the estimation accuracy of each method as shown in Table 3. We matched our experimental conditions to these works as fairly as possible, computed our models' error according to the given metric, and compared the value extracted from the papers. Note that the comparison should only be interpreted as rough estimates to show how all of these models are comparable in terms of claimed accuracy since the differences in the environments—i.e., target cells, validation methods, paths, and corners—are significant. For instance, the variations in [4], [31] differ from global PVT variations in our work. However, we decided to include [4] to compare its DTCO parameter variations with the process and voltage variations. We see that all the presented works easily outperform the conventional derating method in order of magnitude. In contrast, when compared to DPVVM-MDR, they yielded comparable results. This infers that both DPVVM-MDR and learning-based models were capable of estimating timing tendency across various operating conditions. The only significant difference was when compared to [29]. In effect, the model was compared against results

TABLE 3. Comparison of estimation accuracy of state-of-the-art models and DPVVM-MDR; our proposed model showcases comparable results as compared to the complex learning-based models. Note that the figures should only be interpreted as rough estimates owing to the differences in experimental environments.

Prior Arts	**E. Naswali et al. [29]**	**F. Klemme et al. [4]**		**P. Cao et al. [30]**
Variations	unspecified	P,V		P(FF, SS, TT), V, T
Error Metric	occurrence rate of >5% error	$1 - R^2$ [a]		normalized RMSE
Validation Scope	cell	cell	path	path
Error in the Work	1.3%	1.0%	1.3%	4.2%
Error with Derating	14% [b]	6.4% [c]	5.0% [c]	19% [d]
Error Relative to Derating	0.093	0.15	0.27	0.22
Error with DPVVM-MDR	1.2% [b]	0.61% [c]	1.3% [c]	1.5% [d]
Error Relative to DPVVM-MDR	1.1	1.6	1.0	2.8

[a] The accuracy metric used in [4] was originally R^2 but it was converted here to represent the degree of error, not accuracy.

[b] Experimental conditions matched to [29] by adjusting input transition time to match output delay range from [29] and by adjusting the number of target corners for equivalent characterization effort.

[c] Experimental conditions matched to [4] by omitting temperature variations.

[d] Experimental conditions matched to [30], [31] by omitting SF and FS corners and by considering 2-dimension variations only; the comparison in [30], [31] is done against HSPICE results, whereas our comparison is done against PrimeTime results.

from *Synopsys HSPICE* as opposed to *Synopsys PrimeTime* in our work; the accuracy gap is marginal considering the difference in comparison methods.

In this section, we discussed that the accuracy of our proposed models was comparable to sophisticated state-of-the-art variation models. Then, DPVVM and DPVVM-MDR shine in that it is independent to process technologies, simple to deploy with very low computational overhead. Then again, the concept of decomposition into $T_{receiver}$ and T_{driver} in DPVVM can be adopted to any of these models, as well as aging-aware delay models such as [31]; the effectiveness of the decomposing remains to be assessed in our future works.

VII. CONCLUSION

The explosion of PVT conditions is becoming problematic when combined with emerging low-power design technologies. Furthermore, through the technology shrinking, the timing variability due to PVT variations is becoming exacerbated. This paper demonstrated that propagation delay and timing constraints could be decomposed into propagation made by a receiver cell and by its driver cell(s) and be independently scaled to achieve a very reliable cross-corner timing estimation. Our proposed global variation model, DPVVM, can be applied to any prominent delay models and can be applied to combinational and sequential logic gates alike. We achieved average timing estimation accuracy of 4.8% and 5.6%, respectively, on single cells and complex logic circuits, respectively; these are improvements of 69% and 61% in comparison to the conventional derating method. The characterization effort to model a PVT corner is reduced by 92%, compared to the full-characterization of a standard cell library, with 11% characterization overhead over the derating method due to DC analysis, but accounted for only 0.76% of full-characterization time.

We believe that not only does DPVVM enable cross-corner timing estimations with its high accuracy, its concept of decomposition can also be embraced by various fields of timing closure—e.g., power estimation, noise estimation, and on-chip variation models(e.g., *OCV*, *AOCV*, and *POCV*)—and even to the bleeding edge variation models based on machine learning. The study of DPVVM's efficiency in these areas remains to be assessed in the future, in conjunction with the actual fabrication statistics.

ACKNOWLEDGMENT

The EDA tool was supported by the IC Design Education Center (IDEC), South Korea.

REFERENCES

[1] S. Saurabh, H. Shah, and S. Singh, "Timing closure problem: Review of challenges at advanced process nodes and solutions," *IETE Tech. Rev.*, vol. 36, no. 6, pp. 580–593, Nov. 2019.

[2] A. Dalton. (2012). *How to Close Timing With Hundreds of Multi-Mode/Multi-Corner Views*. [Online]. Available: https://www.eejournal.com/article/20121206-cadence/

[3] P. Srinivas, A. Srinivasan, and S. Krishnamoorthy, "Method and apparatus for the analysis and optimization of variability in nanometer technologies," U.S. Patent 7 092 838, Aug. 15, 2006. [Online]. Available: https://www.google.com/patents/US7092838

[4] F. Klemme, Y. Chauhan, J. Henkel, and H. Amrouch, "Cell library characterization using machine learning for design technology co-optimization," in *Proc. 39th Int. Conf. Comput.-Aided Design*, Nov. 2020, pp. 1–9.

[5] J. Rubinstein, P. Penfield, and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. CAD-2, no. 3, pp. 202–211, Jul. 1983.

[6] R. Putatunda, "Auto-delay: A program for automatic calculation of delay in LSI/VLSI chips," in *Proc. 19th Design Autom. Conf.*, Dec. 1982, pp. 616–621.

[7] R. W. Phelps, "Advanced library characterization for high-performance ASIC," in *Proc. 4th Annu. IEEE Int. ASIC Conf. Exhibit*, Sep. 1991, pp. P15-1–P15-3.

[8] E.-Y. Chung, B.-H. Joo, Y.-K. Lee, K.-H. Kim, and S.-H. Lee, "Advanced delay analysis method for submicron ASIC technology," in *Proc. 5th Annu. IEEE Int. ASIC Conf. Exhibit*, Sep. 1992, pp. 471–474.

[9] S. Dutta, S. S. M. Shetti, and S. L. Lusky, "A comprehensive delay model for CMOS inverters," *IEEE J. Solid-State Circuits*, vol. 30, no. 8, pp. 864–871, Aug. 1995.

[10] *CCS Timing White Paper Version 2.0*, Synop., Mountain View, CA, USA, 2006.

[11] *Terms, Definitions, and Letter Symbols for Microelectronic Devices*, Standard JEDEC99C, 2012.

[12] F.-C. Chang, C.-F. Chen, and P. Subramaniam, "An accurate and efficient gate level delay calculator for MOS circuits," in *Proc. 25th ACM/IEEE Design Autom. Conf.*, Jun. 1988, pp. 282–287.

[13] D. Auvergne, N. Azemard, D. Deschacht, and M. Robert, "Input waveform slope effects in CMOS delays," *IEEE J. Solid-State Circuits*, vol. 25, no. 6, pp. 1588–1590, Dec. 1990.

[14] V. Chandramouli and K. A. Sakallah, "Selection of voltage thresholds for delay measurement," in *Analog Design Issues in Digital VLSI Circuits and Systems*. New York, NY, USA: Springer, 1997, pp. 9–28.

[15] D. Maheshwari, "Dual threshold delay measurement/scaling scheme to avoid negative and non-monotonic delay parameters in timing analysis/characterization of circuit blocks," U.S. Patent 6 405 353, Jun. 11, 2002.

[16] P. Bhatnagar and S. Garg, "Dynamic threshold delay characterization model for improved static timing analysis," *J. Electron. Test.*, vol. 30, no. 5, pp. 495–504, Oct. 2014.

[17] K. Kaur and A. Noor, "Strategies & methodologies for low power VLSI designs: A review," *Int. J. Adv. Eng. Technol.*, vol. 1, no. 2, p. 159, 2011.

[18] A. B. Kahng, S. Kang, R. Kumar, and J. Sartori, "Enhancing the efficiency of energy-constrained DVFS designs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 10, pp. 1769–1782, Oct. 2013.

[19] V. Peluso, A. Calimera, E. Macii, and M. Aliotoy, "Ultra-fine grain vdd-hopping for energy-efficient multi-processor SoCs," in *Proc. IFIP/IEEE Int. Conf. Very Large Scale Integr. (VLSI-SoC)*, Sep. 2016, pp. 1–6.

[20] S. R. Nassif, "Design for variability in DSM technologies [deep submicron technologies]," in *Proc. IEEE 1st Int. Symp. Qual. Electron. Design*, Mar. 2000, pp. 451–454.

[21] M. S. Abrishami, M. Pedram, and S. Nazarian, "CSM-NN: Current source model based logic circuit simulation–a neural network approach," in *Proc. IEEE 37th Int. Conf. Comput. Design (ICCD)*, Nov. 2019, pp. 393–400.

[22] P. McGuinness, "Variations, margins, and statistics," in *Proc. Int. Symp. Phys. Design (ISPD)*, 2008, pp. 60–67.

[23] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming Moore's law through energy efficient integrated circuits," *Proc. IEEE*, vol. 98, no. 2, pp. 253–266, Feb. 2010.

[24] A. Dasdan and I. Hom, "Handling inverted temperature dependence in static timing analysis," *ACM Trans. Design Autom. Electron. Syst.*, vol. 11, no. 2, pp. 306–324, Apr. 2006.

[25] B. Kaur, N. Alam, S. K. Manhas, and B. Anand, "Efficient ECSM characterization considering voltage, temperature, and mechanical stress variability," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 12, pp. 3407–3415, Dec. 2014.

[26] B. Lasbouygues, R. Wilson, N. Azemard, and P. Maurine, "Temperature-and voltage-aware timing analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 4, pp. 801–815, Apr. 2007.

[27] T.-B. Chan, W.-T.-J. Chan, and A. B. Kahng, "On aging-aware signoff for circuits with adaptive voltage scaling," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 10, pp. 2920–2930, Oct. 2014.

[28] E. Naswali, A. C. Quiros, and P. Chandran, "DNNLibGen: Deep neural network based fast library generator," in *Proc. 26th IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Nov. 2019, pp. 574–577.

[29] P. Cao, W. Bao, and J. Guo, "An accurate and efficient timing prediction framework for wide supply voltage design based on learning method," *Electronics*, vol. 9, no. 4, p. 580, Mar. 2020.

[30] W. Bao, P. Cao, H. Cai, and A. Bu, "A learning-based timing prediction framework for wide supply voltage design," in *Proc. Great Lakes Symp. VLSI*, Sep. 2020, pp. 309–314.

[31] S. M. Ebrahimipour, B. Ghavami, H. Mousavi, M. Raji, Z. Fang, and L. Shannon, "Aadam: A fast, accurate, and versatile aging-aware cell library delay model using feed-forward neural network," in *Proc. 39th Int. Conf. Comput.-Aided Design*, Nov. 2020, pp. 1–9.

[32] C. Lutkemeyer. (2015). *A Practical Model to Reduce Margin Pessimism for Multi-Input Switching in Static Timing Analysis of Digital CMOS Circuits*. [Online]. Available: http://www.tauworkshop.com/2015/slides/Lutkemeyer_TAU15_PPT.pdf

[33] T. C. Muller, J.-L. Nagel, M. Pons, D. Severac, K. Hashiba, S. Sawada, K. Miyatake, S. Emery, and A. Burg, "PVT compensation in Mie Fujitsu 55 nm DDC: A standard-cell library based comparison," in *Proc. IEEE SOI-3D-Subthreshold Microelectron. Technol. Unified Conf. (S3S)*, Oct. 2017, pp. 1–2.

[34] (2019). *Multiple Input Switching (MIS) Effects in Timing 2019, ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*. [Online]. Available: http://www.tauworkshop.com/2019/slides/TAU19_MIS_Panel_Final.pdf

[35] H. Bhatnagar, "Synopsys technology library," in *Advanced ASIC Chip Synthesis Using Synopsys Design Compiler Physical Compiler and PrimeTime*. Boston, MA, USA: Springer, 2002, pp. 63–80.

[36] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *Int. J. Forecasting*, vol. 32, no. 3, pp. 669–679, Jul. 2016.

[37] S. Makridakis, "Accuracy measures: Theoretical and practical concerns," *Int. J. Forecasting*, vol. 9, no. 4, pp. 527–529, Dec. 1993.

[38] V. Kreinovich, H. T. Nguyen, and R. Ouncharoen, "How to estimate forecasting quality: A system-motivated derivation of symmetric mean absolute percentage error (SMAPE) and other similar characteristics," Univ. Texas El Paso, El Paso, TX, USA, Tech. Rep. UTEP-CS-14-53, 2014.

[39] S. Panigrahi and H. S. Behera, "A hybrid ETS–ANN model for time series forecasting," *Eng. Appl. Artif. Intell.*, vol. 66, pp. 49–59, Nov. 2017.

[40] F. Martínez, M. P. Frías, M. D. Pérez-Godoy, and A. J. Rivera, "Dealing with seasonality by narrowing the training set in time series forecasting with K NN," *Expert Syst. Appl.*, vol. 103, pp. 38–48, Aug. 2018.

[41] H. Abbasimehr and M. Shabani, "A new framework for predicting customer behavior in terms of RFM by considering the temporal aspect based on time series techniques," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 515–531, Jan. 2021.

[42] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Comput. Ind. Eng.*, vol. 143, May 2020, Art. no. 106435.

[43] Y. Yang, H. Jeong, S. C. Song, J. Wang, G. Yeap, and S.-O. Jung, "Single bit-line 7T SRAM cell for near-threshold voltage operation with enhanced performance and energy in 14 nm FinFET technology," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 7, pp. 1023–1032, Jul. 2016.

[44] *Siliconsmart User Guide Version p-2019.06, June 2019*, Synop., Mountain View, CA, USA, 2016.

[45] *Predictive Technology Model*. Accessed: Mar. 2, 2021. [Online]. Available: http://ptm.asu.edu/

[46] S. Sinha, G. Yeric, V. Chandra, B. Cline, and Y. Cao, "Exploring sub-20 nm FinFET design with predictive technology models," in *Proc. 49th Annu. Design Autom. Conf. (DAC)*, Jun. 2012, pp. 283–288.

[47] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1989, pp. 1929–1934.

[48] Silicon Integration Initiative, Inc. (2011). *15 nm Open-Cell Library and 45 nm Freepdk*. [Online]. Available: https://si2.org/open-cell-library/

[49] P. Feldmann, S. Abbaspour, D. Sinha, G. Schaeffer, R. Banerji, and H. Gupta, "Driver waveform computation for timing analysis with multiple voltage threshold driver models," in *Proc. 45th Annu. Conf. Design Autom. (DAC)*, Jun. 2008, pp. 425–428.

[50] D. Patel, "CHARMS: Characterization and modeling system for accurate delay prediction of ASIC designs," in *Proc. IEEE Proc. Custom Integr. Circuits Conf.*, May 1990, pp. 5–9.

[51] O. Coudert, "Gate sizing: A general purpose optimization approach," in *Proc. ED TC Eur. Design Test Conf.*, Mar. 1996, pp. 214–218.

[52] I. Keller, K. H. Tam, and V. Kariat, "Challenges in gate level modeling for delay and Si at 65 nm and below," in *Proc. 45th Annu. Conf. Design Autom. (DAC)*, Jun. 2008, pp. 468–473.

[53] A. B. Kahng, "New game, new goal posts: A recent history of timing closure," in *Proc. 52nd Annu. Design Autom. Conf.*, Jun. 2015, p. 4.

[54] J. Hu, S. K. Raghunathan, D. Sinha, and V. P. Zolotov, "Statistical timing using macro-model considering statistical timing value entry," U.S. Patent 14 800 059, Jul. 15, 2015.

[55] J. M. Daga, E. Ottaviano, and D. Auvergne, "Temperature effect on delay for low voltage applications [CMOS ICs]," in *Proc. Design, Autom. Test Eur.*, Feb. 1998, pp. 680–685.

[56] V. Gerousis, "Design and modeling challenges for 90 nm and 50 nm," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2003, pp. 353–360.

[57] C. Forzan and D. Pandini, "Why we need statistical static timing analysis," in *Proc. 25th Int. Conf. Comput. Design*, Oct. 2007, pp. 91–96.

[58] A. Bellaouar, A. Fridi, M. J. Elmasry, and K. Itoh, "Supply voltage scaling for temperature insensitive CMOS circuit operation," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 45, no. 3, pp. 415–417, Mar. 1998.

[59] K.-I. Shinkai, M. Hashimoto, and T. Onoye, "A gate-delay model focusing on current fluctuation over wide range of process–voltage–temperature variations," *Integration*, vol. 46, no. 4, pp. 345–358, Sep. 2013.

[60] P. Gupta and E. Papadopoulou, "Yield analysis and optimization," in *Handbook of Algorithms for Physical Design Automation*, C. J. Alpert, D. P. Mehta, and S. S. Sapatnekar, Eds. Boca Raton, FL, USA: CRC Press, 2008, ch. 37, pp. 771–787.

[61] *Primetime User Guide Version L-2016.06, June 2016*, Synop., Mountain View, CA, USA, 2016.

[62] R. G. Rizzo, V. Peluso, A. Calimera, and J. Zhou, "On the efficiency of early bird sampling (EBS) an error detection-correction scheme for data-driven voltage over-scaling," in *Proc. IFIP/IEEE Int. Conf. Very Large Scale Integr.-Syst. Chip*. Cham, Switzerland: Springer, 2017, pp. 153–177.

[63] C. Knoth, U. Schlichtmann, B. Li, M. Zhang, M. Olbrich, E. Acar, U. Eichler, J. Haase, A. Lange, and M. Pronath, "Methods of parameter variations," in *Process Variations and Probabilistic Integrated Circuit Design*. New York, NY, USA: Springer, 2012, pp. 91–179.

KWANGSU KIM (Graduate Student Member, IEEE) received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2014, where he is currently pursuing the Ph.D. degree in electrical and electronic engineering.

In 2012, he was an Intern with Qualcomm Korea, Seoul. His current research interests include memory hierarchy, near-data processing, and memory management in operating systems.

BYUNGHA JOO was born in Busan, South Korea, in 1965. He received the B.Sc. degree from Yonsei University, Seoul, South Korea, in 1988, and the M.Sc. degree in electrical and computer engineering from Oklahoma State University, Stillwater, OK, USA, in 1996. His major field of study is cell library architecture and design automation.

He worked at major semiconductor companies, such as an Associate Staff Researcher with Samsung Electronics, Seoul, South Korea, a Senior Design Engineer with Intel, Folsom, CA, USA, and Intel, Santa Clara, CA, USA, a Staff Design Engineer with Virtual Silicon, Sunnyvale, CA, USA, and a Technical Manager with TSMC, Hsinchu, Taiwan. He is currently a Technical Consultant with Rangduru, San Jose, CA, USA.

YOUNG MIN PARK received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree in electrical and electronic engineering.

His research interests include solid-state disk system architecture and CAD flow on near threshold voltage.

TAEYANG JEONG received the B.S. degree from Yonsei University, Seoul, South Korea, in 2017, where he is currently pursuing the Ph.D. degree in electrical and electronic engineering.

His current research interests include hybrid memory systems and system software for processing-in-memory.

KI TAE KIM received the B.S. degree from Yonsei University, Seoul, South Korea, in 2017, where he is currently pursuing the Ph.D. degree in electrical and electronic engineering.

His current research interest includes system software and hardware architecture for processing-in-memory.

EUI-YOUNG CHUNG (Member, IEEE) received the B.S. and M.S. degrees in electronics and computer engineering from Korea University, Seoul, South Korea, in 1988 and 1990, respectively, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, in 2002.

From 1990 to 2005, he was a Principal Engineer with the SoC Research and Development Center, Samsung Electronics, Yongin, South Korea. He is currently a Professor with the School of Electrical and Electronics Engineering, Yonsei University, Seoul. His research interests include system architecture, bio-computing, and VLSI design, including all aspects of computer-aided design with the special emphasis on low-power applications, and flash memory applications.