Exploiting GPU Undervoltage to Improve the Energy Efficiency of Deep Learning Applications

Rafael Gil
Instituto Superior Técnico
Lisboa, Portugal
Email: rafael.gil@tecnico.ulisboa.pt

***Abstract*—The success of deep learning applications, within machine learning and artificial intelligence, is pushing further this area's development. However, the increasing performance and accuracy needs are usually met with higher computational requirements, whose efficiency is, more often than not, disregarded. General Purpose Graphics Processing Units (GPGPUs), being the state-of-the-art accelerators for these applications, play a significant role in making deep learning models widely available. However, the large power consumption increases operational costs and eschews resource-constrained environments from using such devices. To mitigate this problem, the present work proposes an approach to study the potential energy savings of reducing the supply voltage of those devices, using an AMD Radeon Vega Frontier Edition GPGPU. This endeavor is first applied to synthetic benchmarks to characterize the device's voltage guardband and then to current deep learning models to provide an insight into their behavior under minimum supply voltage. Results show deep learning models can achieve energy savings of up to $24.79\,\%$ (average of $15.35\,\%$) and still guarantee their initial accuracy. Nonetheless, the energy savings can be further increased up to $30.16\,\%$ (average of $18.37\,\%$) at the expense of the model's accuracy. Deep learning applications experienced an accuracy droop up to $61.52\,\%$ (average of $10.61\,\%$) when working at near failure supply voltage.**

I. Introduction

Modern applications leveraged by deep learning technologies are feasible due to hardware accelerators. In particular, General Purpose Graphics Processing Unit (GPGPU) devices are widely used in this context given their unique characteristics. They are easier to configure taking advantage of high-level programming languages, there is also an extensive user base community behind these devices, and additionally, they are largely available in data centers and cloud-based services. Consequently, GPGPU devices are considered the main accelerator for deep learning applications [1].

However, there is a high demand for performance improvements in deep learning applications. But, more often than not, the performance enhancement is met with increasingly higher computational requirements. As a result, there is a trade-off between deep learning application's performance and the available computing capabilities of GPGPU accelerators.

The literature on deep learning improvement techniques for GPGPU devices is also rapidly growing. However, the majority of these works neglect the power consumption impact of the proposed solutions. The increasing power consumption is a major challenge on the GPGPU device's usage, especially when considering resource-constrained environments. As a result, there is a need for performance improvements to be weighed against their energy overhead, and for a deeper dive on lower precision techniques that impose a trade-off between accuracy and efficiency [2].

GPGPU manufacturers design their processing units bearing in mind a given performance standard. Naturally, each architecture will have its minimum required supply voltage for the device to work properly. However, taking into account phenomena that can negatively impact the device, manufacturers add an extra safety margin on top. This opens up a window to improve the GPGPU devices energy efficiency, given that the power consumption is deeply related to the supply voltage [3].

This thesis proposes an approach to study the potential energy savings of GPGPU devices on modern deep learning applications by reducing the voltage guardband margin imposed by the manufacturers.

However, due to the limitations of NVIDIA devices for the prosecution of this work, it will focus on the Advanced Micro Devices, Inc. (AMD) Radeon Vega Frontier Edition GPGPU device.

The main goal of this thesis is to achieve energy efficiency in deep learning applications. To accomplish this, the following objectives are proposed:

- Characterization of AMD Radeon Vega Frontier Edition GPGPU card voltage guardband. Motivated by the potential energy saving results and the lack of literature featuring AMD devices.
- In-depth performance and energy efficiency evaluation regarding the proposed approach.
- Evaluation of the accuracy impact of the voltage guardband reduction approach on modern deep learning applications.

II. Computing power performance

GPGPU devices are the main accelerators for deep learning models, and it is widely used across such applications. At the same time, there is a high demand for performance improvements on those applications. However, the performance enhancement of deep learning applications is met with increasingly higher computational requirements. Consequently, the improvements in the deep learning field can outpace the improvements on the GPGPU devices [2].

The literature on deep learning improvement techniques for GPGPU devices is rapidly growing. However, the majority of these works neglect the power consumption impact of the proposed solutions. The increasing power consumption is a major challenge on the GPGPU device's usage, especially when considering resource-constrained environments. There is a need for performance improvements to be weighed against their energy overhead, and for a deeper dive on lower precision techniques that impose a trade-off between accuracy and efficiency [2].

A. Dynamic Voltage and Frequency Scaling Techniques

Dynamic Voltage and Frequency Scaling (DVFS) is a technique used to improve a given processing unit power management. It consists of dynamically updating the processing unit working frequency: Reducing the working frequency will consequently trigger a supply voltage reduction.

Equation 1 translates the relationship between the processing unit's frequency f, supply voltage V, and power consumption P.

$$P \propto f \times V^2 \tag{1}$$

Since the processing unit's frequency is proportional to its supply voltage, the relationship on the equation 1 strongly encourages DVFS when seeking energy efficiency.

First of all, note that DVFS is transversal to any processing unit, including both Central Processing Unit (CPU)s and GPGPU. Secondly, reducing the operating frequency has the advantage of reducing the supply voltage and consequently improving the overall energy consumption. However, this is possible at the expense of execution performance, since a lower frequency leads to longer execution timings [4].

Jiao et al. [5] studied how frequency scaling impacts the performance and power consumption of a GTX 280 GPGPU. To achieve that, they used distinct sets of applications categorized into three different groups: compute-intensive, memory-intensive, and hybrid. They observed that the DVFS impacts on energy efficiency are dependent on the application itself. More specifically, it is dependent on the relationship between global memory transactions and computation instructions. Finally, based on the application properties, both the memory and the cores frequency of the GPGPU device would be adjusted.

Guerreiro et al. [6] proposed a new model to evaluate the potential performance and power consumption improvements of applying DVFS to a given application. They used the profiling result of a set of synthetic benchmarks to train a machine learning model classifier. With the trained classifier it is possible to characterize any kind of GPGPU application performance-wise and power consumption-wise when submitted to DVFS. Their results show that the classifier can successfully predict the optimal frequency for each application, obtaining an average energy saving improvement of $16\,\%$ with a maximum of $36\,\%$.

B. Voltage operating limit

Even though DVFS techniques do achieve high energy saving potentials, its implementations focus mostly on frequency scaling to achieve energy savings. The voltage reduction is usually a byproduct of the frequency scaling since lower frequencies require lower supply voltages. In other words, the device's supply voltage, within common DVFS implementations, is not actively managed.

However, processing unit manufacturers when designing a processor must account for phenomenons that might negatively impact its performance. When doing so, they usually add a voltage guardband on top of the minimum supply voltage required for the device to function properly. It is estimated that the voltage guardband is approximately $20\,\%$ of the recommended voltage by the manufacturer, i.e. $20\,\%$ of the nominal voltage [3], [7].

Leng et al. [7] study the benefits of exploiting the voltage guardband using a set of commercial GPGPU devices from Nvidia. Their results show energy-saving results ranging up to $25\,\%$. Furthermore, they conclude that the GPGPU device's minimum operating voltage is dependent on the running application.

Additionally, based on each application's performance counters they build a model to predict a given application's minimum operating voltage. The prediction error of their model ranges up to $3\,\%$ with an average of $0.5\,\%$. As the authors suggest, their accurate model opens up possibilities to a dynamic voltage guardband scheme with potential energy savings ahead.

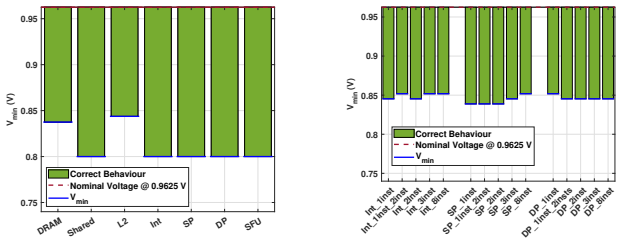
III. Voltage guardband characterization

This section focus on characterizing an AMD Radeon Vega Frontier GPGPU under a progressively lower supply voltage using two sets of synthetic benchmarks.

Benchmarks are designed to exclusively stress a given component of the GPGPU architecture, to explain to which degree each architecture component impacts the voltage guardband. The first set of synthetic benchmarks targets the GPGPU's memory unit – the Dynamic Random-Access Memory (DRAM), the L2 cache, and the shared memory – and the functional unit – integer, single precision, double precision, and special function unit operations. The second set of benchmarks extends the functional unit's benchmarks from the first set, but the instructions are organized in a way that forces dependencies. Different degrees of dependencies were used, ranging from 1 instruction dependency to 8 instruction dependency for each stressed architecture component.

A. Minimum operating voltage

For each benchmark, the GPGPU's supply voltage is reduced starting at the manufacturer recommend voltage, $V_{nominal}$, down to the lowest supply voltage that still guarantees the program's correctness, V_{min}. An execution is considered correct when its result exactly matches the result obtained from the same execution at $V_{nominal}$. Nevertheless, the voltage can be reduced further below V_{min} leading to execution errors,



(a) Synthetic benchmarks. (b) Dependencies benchmarks.

Fig. 1. Obtained V_{min} for the benchmark applications.

namely Silent Data Corruption (SDC), run-time errors, system crashes, and indefinitely long executions. SDC occurs when the execution finishes and no warning or error message is triggered, but the final result is not correct, i.e. apparently no error has occurred but the device wrote erroneous data into memory [8]. Run-time errors, on the other hand, occur when the program execution fails during run-time due to memory access faults. System crashes and indefinitely long executions require a manual system reboot to restore the GPGPU's normal state.

Hence, V_{crash} is defined as the threshold voltage at which the program still completes the execution, potentially with corrupted data due to SDC, but further reducing the supply voltage produces either a run-time error, a system crash, or an indefinitely long execution.

AMD GPU Tool (AGT) is used to control the GPGPU's supply voltage with a minimum granularity of $6.25\,\mathrm{mV}$. Thus, with each step of $6.25\,\mathrm{mV}$, the benchmark is executed after a system reboot and the execution correctness is validated against the reference execution at $V_{nominal}$. To minimize the impact of external variables, no other process was running in the GPGPU during the applications' execution, and AGT's fan speed control was used to stabilize the temperature throughout the experiment.

1) Synthetic benchmarks: Figure 1a plots the measured V_{min} for the studied synthetic benchmarks. SDC errors did not occur meaning V_{crash} matches V_{min} results for the synthetic benchmarks.

The first empirical conclusion is that multiple V_{min} results were obtained across the different benchmarks. The group of benchmarks that target the Arithmetic Logic Unit (ALU), which includes `Int`, `SP`, `DP` and `SFU`, all obtained along with `Shared` a V_{min} of $0.8\,\mathrm{V}$ at $1028.57\,\mathrm{MHz}$. `DRAM` and `L2` benchmarks obtained a V_{min} of $0.8375\,\mathrm{V}$ and $0.843\,75\,\mathrm{V}$, respectively.

Furthermore, there are different V_{min} results across different applications, which is aligned with the expectations that V_{min} depends mostly on the application itself [7].

One key difference between the `DRAM` benchmark and the remaining ones is that the first executes multiple writes to the DRAM memory whilst the latter only write once into DRAM at the end of their execution. According to the results, writing to the DRAM has a high impact on V_{min}.

Further analyzing the results presented in figure 1a, one can infer that a significant voltage guardband is obtained for all the benchmarks ranging from $12.34\,\%$ to $16.88\,\%$ with an average of $15.68\,\%$ regarding nominal voltage of $0.9625\,\mathrm{V}$. These values are slightly below the $20\,\%$ range obtained in previous works [7], [9], [10].

2) Dependencies benchmarks: There are 4 types of data dependencies for each application. Each of these data dependency types is identified by suffixing the name of the application with `1_inst`, `1_inst_2_inst`, `3_inst` and `8_inst`. These suffixes are self-explanatory. For instance, `1_inst` means each issued instruction has a dependency with the last issued instruction, i.e. 1 instruction dependency. Analogously, `8_inst` means each issued instruction has a dependency with the eighth last issued instruction, i.e. 8 instructions dependency. `1_inst_2_inst` means there is simultaneously a dependency with the last and the second last issued instruction.

Empirically, a larger adjacency between dependent instructions implies a higher computational cost, since more stalls have to be introduced by the processor.With that in mind, it is expected for V_{min} to be lower on `1_inst` suffixed applications, when compared with `8_inst` suffixed applications, as an example. However, that conclusion cannot be inferred from the results presented in figure 1b since mixed results are obtained: for `DP` instructions the expected behavior is observed, yet, for `SP` instructions, it is the opposite.

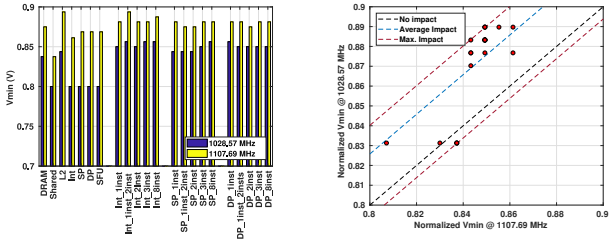
Finally, even though the V_{min} variation is not as steep as expected within the dependencies benchmark, when these are compared, as a whole, to the synthetic benchmarks one can observe a few differences. For `Int`, `SP` and `DP`, V_{min} is on average $0.853\,75\,\mathrm{V}$, $0.8475\,\mathrm{V}$ and $0.851\,25\,\mathrm{V}$ on the dependencies benchmarks, respectively. These values correspond to an undervoltage of $11.30\,\%$, $11.95\,\%$ and $11.55\,\%$. The obtained undervoltage for the dependency benchmarks is lower than the one benchmarks obtained for the synthetic benchmarks, which is aligned with the expectations.

B. Impacts of voltage guardband and frequency optimizations

Results show that V_{min} varies depending on the program that is executed, ranging from $0.8\,\mathrm{V}$ to $0.86\,\mathrm{V}$, corresponding to a variability of $0.06\,\mathrm{V}$ or $6.2\,\%$ when compared to $V_{nominal}$. The goal of this subsection is to study which variables are the main contributors to such variability, namely operating frequency, temperature, aging, process variation, and voltage noise.

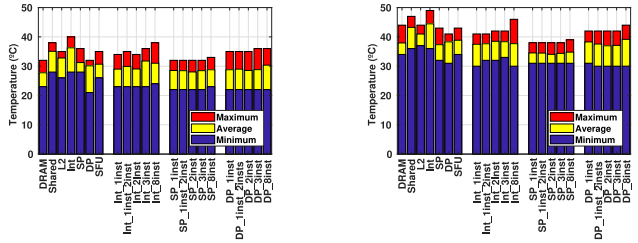
1) Operating Frequency: To evaluate the impact the frequency has upon V_{min} variability, the voltage guardband experiment was repeated under different constraints. All the controlled variables, such as fan speed, were kept and the GPGPU cores' frequency was increased to a higher level using AGT. Figure 2a shows the obtained V_{min} readings at the two studied frequency rates: $1028.57\,\mathrm{MHz}$ and $1107.69\,\mathrm{MHz}$.

Naturally, a higher core's frequency requires a higher $V_{nominal}$, which is $0.9625\,\mathrm{V}$ and $1.0375\,\mathrm{V}$ at $1028.57\,\mathrm{MHz}$ $1107.69\,\mathrm{MHz}$, respectively.



(a) (b)

Fig. 2. V_{min} obtained for the benchmarks at 1028.57 MHz and 1107.69 MHz. Benchmark's normalized V_{min} at 1028.57 MHz plotted against their normalized V_{min} at 1107.69 MHz



(a) Low-temperature (b) High-temperature

Fig. 3. Minimum, average and maximum temperature variation for both synthetic and dependency benchmarks.

Benchmarks that attained a higher V_{min} result at 1028.57 MHz, also obtained a higher V_{min} reading at 1107.69 MHz. Additionally, the V_{min} variability on the higher frequency experiment matches exactly the variability of the lower frequency experiment. The difference between the minimum and maximum obtained V_{min} is 0.06 V in both cases.

Figure 2b establishes a common ground of comparison between both frequency experiments, by normalizing V_{min} against the nominal voltage using equation $V_{normalized} = V_{min}/V_{nominal}$. There is an observable proclivity for higher V_{min} readings on lower core frequency settings, which translates into a lower voltage guardband on lower frequencies.

The linear approximations on figure 2b show that the gap between $V_{nominal}$ and V_{min} is negatively impacted up to 4.62 %, with an average of 2.58 %, on the higher frequency experiment when compared to its low frequency counterpart. This impact has a maximum

In conclusion, results show that the frequency has virtually no impact on the V_{min} variability since both experiments achieved the same 0.06 V V_{min} variability. Nonetheless, programs running at a higher frequency tend to have a slightly lower voltage guardband.

2) Temperature: To study how temperature impacts V_{min}, the benchmarks were executed, once again, using two different temperature levels at a fixed frequency. The temperature was regulated using AGT's fan medium and high intensity control, producing high and low frequency experiments, respectively. Figure 3a and 3b depict the minimum, average, and maximum temperature levels for each of the benchmarks on both experiments.

Tests using low fan intensity led the program execution to lag indefinitely and in some cases to system failures, thus preventing tests to reach a wider temperature range. Despite the limitations of the temperature enforcement method, results show an average difference of 6.8 °C between experiments.

Figure 4 plots the V_{min} results from the low-temperature experiment against the high-temperature experiment in an attempt to study the temperature's impact on the voltage guardband. V_{min} tends to be higher with higher temperature values.

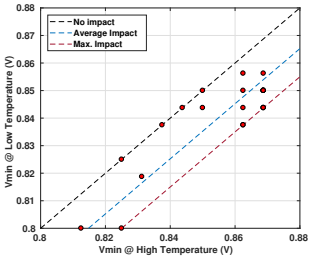


Fig. 4. Benchmark's V_{min} at different temperatures. The distance between the dashed lines represents the impact of the device's temperature on V_{min}.

Naturally, points plotted coincidentally on the $y = x$ line represent those benchmarks where the obtained V_{min} is the same in both the low-temperature and the high-temperature experiments. From the data points linear regression, one can infer that the temperature has an average impact of 0.0148 V in the V_{min} variability. Furthermore, the temperature had a maximum impact of 0.0250 V on V_{min}. More specifically, V_{min} was at most 0.0250 V lower on the low-temperature experiment.

In conclusion, the impact temperature has on the V_{min} variability is not enough to explain the whole magnitude of the V_{min} variability observed, which is aligned with [7].

3) Aging: AMD Vega Frontier is built using 12.5 billion 14 nm transistors using Low Power Plus (LPP) FinFET process technology [11]. Within the semiconductor industry, there is an issue, known for more than 50 years, called Negative Bias Temperature Instability (NBTI) and Positive Bias Temperature Instability (PBTI) that negatively impacts the reliability of the Metal Oxide Semiconductor (MOS) transistor technology [12].

NBTI/PBTI consists in the accumulation of positive/negative charges at the transistor's gate insulator due to a negative/positive bias voltage, V_g, at the transistor's gate. This process is aggravated by temperature, hence its name. The accumulated charges partially cancel out the gate's applied voltage. Consequently, the source to drain current flow is reduced ultimately leading to the transistor's performance loss.

Regarding the current study, all the test executions on the GPGPU device were completed during a period of six months,

which is relatively low when compared to the typical lifetime of such devices of ten years [13].

During the period of this study's tests, no evidence of V_{min} variability due to the aging process of the GPGPU device was found. Furthermore, related work in this field shows a performance impact of up to $2\,\%$ under real-use conditions on International Business Machines Corporation (IBM) microprocessors [14]. In conclusion, it is unlikely that aging factors alone could explain the V_{min} variability observed in this study.

4) Process Variation: Production processes are liable to degrees of variation that impact the quality of the final product. The quality of the products themselves can be used to describe the process quality, through which they were produced, using two variables: accuracy and precision.

The transistor manufacturing process is also exposed to such quality variables. These precision issues are called Process Variation (PV) and refer to the variability of the device's parameters, such as the transistor's gate width, the channel length, or the oxide thickness, from their nominal specifications.

PV has become increasingly more severe due to the increasing difficulty to precisely control the fabrication process as the transistor size became smaller. The chip yield i.e., the fraction of fully working chips within the wafer where they are produced, was reduced from $90\,\%$ to $50\,\%$ and then $30\,\%$ when the transistor size scaled from $350\,\mathrm{nm}$ to $90\,\mathrm{nm}$ and then to $30\,\mathrm{nm}$ respectively [15].

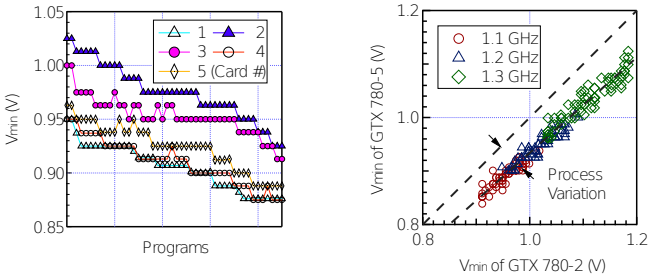
Due to constraints, during this study, there was only a single AMD Radeon Vega Frontier Edition available. Thus, an experimental study of the PV effects on the V_{min} could not be conducted.

Nonetheless, Leng et al. [7] briefly studied the impact of PV in the voltage guardband of Nvidea GPGPU devices. They used only five different Nvidia GTX 780 GPGPU devices, which, as the authors themselves suggest, is not statistically robust. But, the results provide insights into the PV effect on such endeavors that are useful to comprehend the root cause of the V_{min} variability.

They observed a constant offset of V_{min} between each of the tested GPGPU devices, with deviations on a few programs as shown by figure 5a. This offset means that a given program experiences V^1_{min} and V^2_{min} on GPGPU device 1 and 2, respectively, but $V^1_{min} \neq V^2_{min}$. More precisely, Leng et al. [7] observed a $0.07\,\mathrm{V}$ maximum offset between the five GPGPU devices.

On top of the constant offset, they also observed random V_{min} deviations from each GPGPU device to the other. These random deviations happened in different programs at each device.

The observed V_{min} variability is caused by PV, which is known for creating both systemic and random variances on the device's building blocks parameters [16]. A slight variation on the digital circuit components has the potential of changing the circuit's critical path. Therefore, programs that do not rely on the critical path in one device, might do rely on it on other devices, thus explaining random V_{min} deviations from the constant offset.



(a) Applications' V_{min} results on five Nvidia GTX 780 General Purpose Graphics Processing Unit devices (from [7]). A constant and a random offset is observed between each device's V_{min}.

(b) Applications' V_{min} from General Purpose Graphics Processing Unit device 2 and 5 plotted against each other at different frequencies (from [7]). PV impact on V_{min} is shown as the difference from the data points to the line $y = x$.

Fig. 5. Leng et al. [7] results of PV impact on V_{min}.

Notwithstanding, there was V_{min} variability on the tested programs across each GPGPU device. Additionally, that variability has approximately the same magnitude, on all devices. However, there were indeed differences in the absolute values of V_{min}, which can be attributed to PV, but PV itself cannot explain the observed variability.

5) Voltage Noise: The voltage signals that powers a digital circuit is not steady as one might expect because there is a degree of fluctuation associated with it. This fluctuation is called voltage noise. With this in mind, digital circuit manufacturers increase the supply voltage above the circuit's intrinsic voltage. Thus creating the mentioned voltage guardband which acts as a safe margin usually greater than $20\,\%$ [10].

The voltage guardband is also added as a protection against phenomenons such as temperature, aging, PV, and voltage noise. As discussed in the previous subsections temperature, aging and PV does not explain, by themselves, the whole extent of the V_{min} variability. Thus, by exclusion, voltage noise is the main contributor towards the V_{min} variability [7].

The voltage at a given point, A, in an electrical circuit is given by the equation 2, meaning that two factors impact the voltage: the current draw and the current draw's increasing rate.

$$V_A = V_{DD} - I \cdot R - L \cdot \frac{di}{dt} \tag{2}$$

Leng et al. [7] studied further how each of those factors impacts the voltage at the same point A. Given the power equation $P = R \cdot I^2$ differentiated from Ohm's law, they tested the hypothesis of $I \cdot R$ being the dominant factor in the voltage from equation 2. If this hypothesis holds, a program with a high power consumption would have a high voltage noise and consequently a higher V_{min}.

Using power consumption measurements, and also the Instructions Per Clock (IPC) as a predictor for power consump-

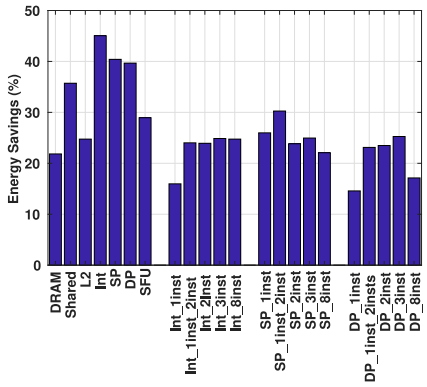


Fig. 6. Energy savings attained by operating at V_{min} voltage on both synthetic benchmarks and benchmarks with dependencies.

tion, no evidence of a correlation between those properties and V_{min} was found. Thus, $I \cdot R$ is not the dominant factor of the voltage noise on equation 2, and consequently di/dt is.

Leng et al. [7] proceed even further to study which program activity is responsible for generating the most di/dt droop. The study included Compute Unified Device Architecture (CUDA) runtime activities, inter-kernel activities, initial-kernel activities, and intra-kernel activities.

Their conclusion is that the greatest driver of di/dt droop and consequently the driver o voltage noise is the intra-kernel activities. These are related to the nature of the kernel itself, which varies from application to application. They also identify cache misses and pipeline stalls as a driver for di/dt droop.

Focusing on the current study, it is possible to conclude that the results match the established knowledge. Dependency benchmarks obtained significantly higher V_{min} results when compared to their synthetic counterpart. Whilst recovering from pipeline stalls, the GPGPU moves away from an idle state creating a sudden increase of the drawn current, which in its turn produces a voltage spike.

C. Energy gains

Energy savings are measured by comparing the consumed energy when the GPGPU is operating at V_{min} against the same metric when the device is running at $V_{nominal}$. With everything set up, gpowerSAMPLER [17] is used to obtain the energy consumption metrics, depicted in figure 6.

Results show an average energy saving of $26.4\,\%$, ranging between the lowest energy saving percentage of $14.58\,\%$ to a peak of $45.05\,\%$.

It is observable that the first set of benchmarks present better energy-savings compared to the second set. More precisely, the first set shows an average energy saving of $33.77\,\%$ which is significantly higher than the $22.95\,\%$ average energy saving obtained by dependencies benchmarks.

This difference between the energy savings obtained by both sets is deeply connected with the V_{min} differences obtained. In fact, the energy consumption per unit of time i.e., power P, has the following relationship with the supply voltage V: $P \propto f \times V^2$. Thus, a small variation in the supply voltage of a GPGPU device can greatly increase the device's energy efficiency.

TABLE I
DEEP LEARNING MODELS INFERENCE ACCURACY AT $V_{nominal}$.

DEEP LEARNING MODEL	ACCURACY
AlexNet	$83.0\,\%$
VGG-16	$89.8\,\%$
VGG-19	$89.8\,\%$
Inception V4	$95.2\,\%$
ResNet V2	$94.1\,\%$
Inception-ResNet	$95.3\,\%$
Skip-Thoughts	$71.2\,\%$
Sentiment	$73.0\,\%$
ReactionRNN	$61.3\,\%$
BERT	$89.3\,\%$

IV. VOLTAGE GUARDBAND IN DEEP LEARNING APPLICATIONS

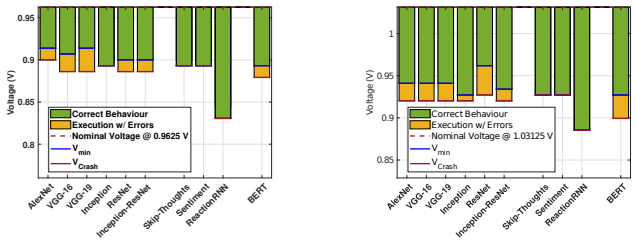
This section transposes the previous methodology into deep learning applications and evaluates the energy-saving potential against the precision loss that the voltage reduction process might impose. It uses a heterogeneous set of pre-trained deep learning applications, each with its own architecture and purpose, executing inference on a given set of test data. The focus is to evaluate the degree to which the inference accuracy is impacted by varying the GPGPU's supply voltage.

There was an effort to standardize the accuracy extraction method. All Convolution Neural Network (CNN) models, focused on image recognition, are evaluated based on their performance on the ImageNet dataset. Whilst Recurrent Neural Network (RNN) models and Bidirectional Encoder Representations from Transformers (BERT), focused on Natural Language Processing (NLP), are evaluated based on their performance on Microsoft Research Paraphrase Corpus (MRPC) dataset.

Table I depicts the obtained accuracy for all the deep learning models, at their respective tasks, at $V_{nominal}$.

Synthetic benchmarks showed that the application itself is the main responsible for the V_{min} variation. With this in mind, and bearing that deep learning applications are highly demanding of computing resources, it is expected for V_{min} to be substantially higher than the benchmarks, which translates into a lower energy-saving potential.

Also, synthetic benchmarks did not display any SDC errors, i.e. all benchmarks either performed correctly or didn't perform at all at each voltage level. This was explained based on the application's lower complexity. However, SDC is expected to occur for deep learning models, allowing for precision to energy efficiency trade-off. Therefore, the program's correctness threshold must also be updated. Each deep learning model execution is considered correct if its accuracy is no more than $0.1\,\%$ deviated from the reference at $V_{nominal}$. Conversely,



(a) 1028.57 MHz (b) 1107.69 MHz

Fig. 7. Obtained V_{min} and V_{crash} for deep learning models.

an accuracy deviation greater than 0.1 % is considered as an accuracy loss.

A. Voltage Guardband Accuracy Impact

Figure 7 depicts the V_{min} and V_{crash} absolute values obtained at two different GPGPU core frequencies, 1028.57 MHz, on the figure 7a, and 1107.69 MHz, on the figure 7b, respectively.

All deep learning models excluding ReactionRNN show a similar accuracy sensitivity to the voltage guardband reduction. The average V_{min} is 0.900 06 V and 0.941 25 V on the lower and higher frequency experiment respectively. Normalizing V_{min} results from both frequencies based on the nominal value, further confirms the obtained the results on subsection III-B1. Lower frequencies achieve higher V_{min} readings, thus resulting in a lower voltage guardband. At 1028.57 MHz, the nominal voltage set by the GPGPU manufacturer is 0.962 50 V resulting in an average normalized V_{min} of 0.935 71. Conversely, at 1107.69 MHz, the nominal voltage set by the GPGPU manufacturer is 1.031 25 V resulting in an average normalized V_{min} of 0.912 73. This corresponds to an undervoltage of 6.43 % and 8.73 % when compared to the nominal voltage of each frequency, 0.9625 V and 1.031 25 V, respectively.

The standard deviation of the V_{min} readings is 0.021 V and 0.017 V for the lower and higher GPGPU core's frequency respectively. By excluding ReactionRNN results, the standard deviation calculation decreases significantly to 0.008 V and 0.010 V respectively. This is caused by the smaller architecture of ReactionRNN that features substantially fewer parameters than its counterparts.

It is interesting to note that none of the RNN models showed an accuracy loss with the voltage supply reduction. All successful executions of these models did not incur in accuracy loss, meaning $V_{crash} = V_{min}$.

On the other hand, CNN models along with BERT displayed SDC occurrences resulting in an accuracy loss, hence $V_{crash} < V_{min}$. These models allowed further decreasing of the supply voltage by an average of 0.005 09 V and 0.013 13 V on each frequency experiment, respectively.

In sum, the results at 1028.57 MHz reveal that there is a voltage guardband, that can be safely reduced on deep learning applications, ranging from 4.55 % up to 12.33 % with an average of 6.43 %. The voltage guardband can be further reduced, up to 2.60 %, thus allowing accuracy loss. Likewise, at 1107.69 MHz the voltage guardband can be safely reduced from 6.06 % up to 12.73 % with an average of 8.75 %. By allowing accuracy loss, the voltage guardband can be further decreased up to 3.03 % on some of the tested models.

TABLE II
DEEP LEARNING MODELS INFERENCE ACCURACY AT V_{crash} AT 1028.57 MHz AND 1107.69 MHz, INCLUDING THE ACCURACY LOSS WHEN COMPARED TO THE REFERENCE ACCURACY FROM TABLE I

DEEP LEARNING MODEL	1028.57 MHz	1107.69 MHz
AlexNet	81.68 % (1.32 %)	76.58 % (6.42 %)
VGG-16	38.57 % (51.23 %)	76.24 % (13.56 %)
VGG-19	28.28 % (61.52 %)	65.31 % (24.49 %)
Inception V4	95.12 % (0.08 %)	95.07 % (0.13 %)
ResNet V2	89.14 % (4.96 %)	86.77 % (7.33 %)
Inception-ResNet	93.18 % (2.12 %)	94.44 % (0.86 %)
Skip-Thoughts	71.19 % (0.01 %)	71.19 % (0.01 %)
Sentiment	72.98 % (0.02 %)	73.00 % (0.00 %)
ReactionRNN	61.30 % (0.00 %)	61.30 % (0.00 %)
BERT	84.61 % (4.69 %)	55.95 % (33.35 %)

B. Accuracy loss

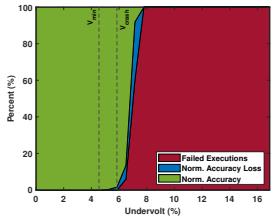
Table II depicts the obtained accuracy at V_{crash} for all deep learning models, at their respective tasks, at both studied operating frequencies. The table also includes, within brackets, the accuracy loss when compared to the nominal accuracy whose values are depicted in the table I above.

Results, when operating at the low frequency, 1028.57 MHz, show accuracy droops up to 61.52 % with an average of 12.59 % when working at V_{crash} supply voltage. Conversely, when operating at the high frequency, 1107.69 MHz, the accuracy droop achieved only a maximum of 33.35 % with an average of 8.62 %.

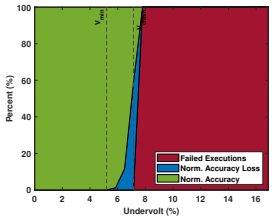
It is also important to note that there are models that did not experience any accuracy loss at all (Skip-Thoughts, Sentiment, and ReactionRNN) and a few more that had only a residual impact on the accuracy when working near failure supply voltages (InceptionV4, and Inception-ResNet).

1) Convolutional Neural Networks: According to the previous subsection, CNN models displayed a gap between V_{min} a V_{crash}. This means there is a voltage band where all executions are completed successfully but there is at least a 0.1 % accuracy loss.

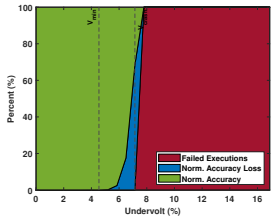
To further characterize this issue, the execution of each CNN model is displayed in figure 8. Due to space constraints, the high-frequency graphs were omitted. In these graphs, the y axis presents the percentage of failed executions out of a total of 10. While the x axis presents the undervolt percentage. The green zone corresponds to a condition where all executions presented an error lower than 0.1 %. The blue zone corresponds to a case of complete execution, but with accuracy losses; and the red zone represent cases of incomplete/failed executions.



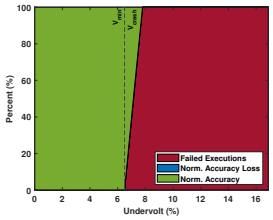
(a) AlexNet



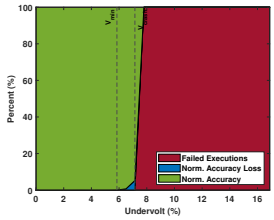
(b) VGG-16



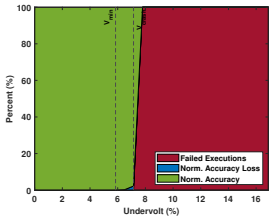
(c) VGG-19



(d) Inception

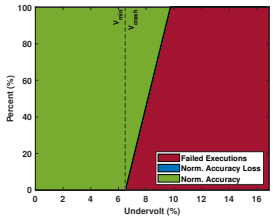


(e) ResNet

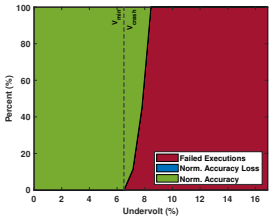


(f) Inception-ResNet

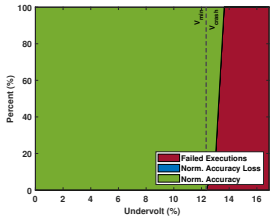
Fig. 8. CNN's inference accuracy distribution across an increasing undervolt percentage @ 1028.57 MHz.



(a) Skip-Thoughts

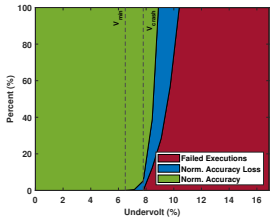


(b) Sentiment

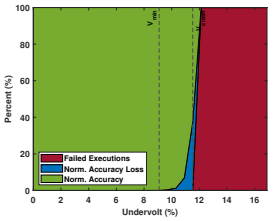


(c) ReactionRNN

Fig. 9. RNN's inference accuracy distribution across an increasing undervolt percentage @ 1028.57 MHz.



(a) BERT @ 1028.57 MHz



(b) BERT @ 1107.69 MHz

Fig. 10. BERT inference accuracy distribution across an increasing undervolt percentage.

Empirically, one can verify from figure 8 that decreasing the GPGPU device's supply voltage further below V_{crash} undervoltage, still produces successful executions. Unfortunately, some of those executions fail due to run-time errors, system crashes, or indefinitely long executions.

The models with the most data corruption impact at V_{crash} are VGG-16 and VGG-19. In both frequency experiments, these models have the greatest amount of accuracy loss. The VGG models are characterized by their high computationally requirements. These are the models with the most naively stacked convolutions layers featured in this study. On the other end of the spectrum, Inception has no data corruption on the lower frequency experiment, having $V_{min} = V_{crash}$.

There is also an apparent tendency for lower frequencies to display higher data corruption rates. VGG-16, VGG-19, Inception, and ResNet all have a higher precision loss on their lower frequency experiment.

2) Recurrent Neural Networks: Figure 9 depicts the accuracy loss evolution with an increasing undervoltage percentage for RNN models.

None of the RNN models had SDC occurrences even when the GPGPU device's supply voltage was below V_{crash}. Also, each model has a different evolution towards failed execution. Skip-thoughts had a less steep progression, meaning that progressively more and more executions were failing with the increased undervoltage. This contrasts with Sentiment and ReactionRNN that had an abrupt change in that regard.

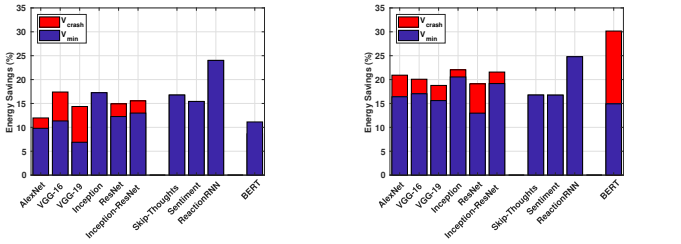
3) Other Neural Networks: Figure 10 depicts BERT's accuracy loss progression with an increasing undervoltage percentage.

Unlike the CNN models, BERT has higher data corruption readings on a higher frequency. It is also the only model to have drastically different progression patterns on both frequencies studied.

On one hand, the low-frequency experiment had most of the data corruption incidents occurring below V_{crash} with the amount of failed executions progressively increasing after that point. On the other hand, the high-frequency experiment had considerably more data corruption at V_{crash} and the amount of failed executions increased abruptly after that point. The accuracy loss at V_{crash} with 1028.57 MHz is 0.41 % compared to a 33.35 % when the device's core frequency is 1107.69 MHz.

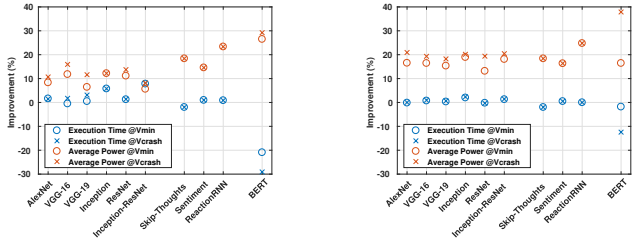
V. Voltage Guardband Energy Impact

Having V_{min} and V_{crash} results for each deep learning model, it is now possible to evaluate the energy efficiency



(a) 1028.57 MHz (b) 1107.69 MHz

Fig. 11. Energy savings attained by operating at V_{min} and V_{crash} voltage on deep learning models.



(a) 1028.57 MHz (b) 1107.69 MHz

Fig. 12. Execution time and average power consumption improvements at V_{min} and V_{crash} over the same metrics at the nominal voltage.

attained. Knowing the average power consumption and also execution time we can infer the energy consumption for each execution through the equation 3.

$$E = \frac{P}{\Delta t} \tag{3}$$

Figure 11 depicts the obtained energy saving results for the deep learning models. Results on low frequency experiment, guaranteeing the execution correctness, show an energy-saving potential ranging from $6.88\,\%$ up to $24.01\,\%$ with an average of $13.79\,\%$. The average energy-saving potential can be further increased by an average of $1.84\,\%$ by allowing the model's accuracy to drop.

Analogously, on the high frequency experiment, the energy saving potential ranges from $12.97\,\%$ up to $24.79\,\%$ with an average of $17.50\,\%$. This potential can be further increased from $3.77\,\%$ to $5.38\,\%$ with an average of $3.60\,\%$ by allowing the model's accuracy to drop.

Naturally, given V_{min} results, ReactionRNN achieved the highest energy consumption improvement at V_{min}. This improvement is only surpassed at the high frequency by the BERT model when the supply voltage is at V_{crash}.

By splitting the energy saving results across the deep learning models architectures: CNN models alone achieved an energy consumption improvement ranging from $12.97\,\%$ up to $20.55\,\%$ with an average of $17.50\,\%$. Conversely, RNN models achieved an energy consumption improvement ranging from $16.75\,\%$ up to $24.79\,\%$ with an average of $19.43\,\%$. While these results are retrieved from the high-frequency experiment, similar conclusions can be inferred from the lower frequency experiment. Also, CNN models achieved an average energy consumption improvement slightly below RNN. However, it is notable that it did so with approximately half the standard deviation error. This metric is $2.68\,\%$ for the CNN architecture and $4.64\,\%$ for the RNN architecture.

Two variables have an impact on the overall energy consumption: execution time and average power consumption. To further comprehend how each of these metrics had an impact on the results shown in figure 11, the graphs in figure 12 are presented. These graphs portray the improvement from both execution time and average power consumption metrics compared to the reference execution.

Results also confirm the initial expectation that the average power consumption decreases with the undervolting process, given the known relationship between power (P) and the supply voltage (V) represented in the equation 1, where f represents the GPGPU device's operating frequency. In particular, the power consumption has an average improvement of $13.83\,\%$ and $17.44\,\%$ on the low and high frequency experiment, respectively.

Additionally, the execution time does remain stable at a lower supply voltage compared to the reference execution at the nominal voltage. There were executions slightly faster than the reference and executions slightly slower than the reference, translating into close to no impact on the energy savings.

However, the BERT results on the low-frequency experiment are worthy of notice. At $1028.57\,\text{MHz}$, the execution speed drastically decreased thus trumping the improvements obtained from the lower power required. At V_{crash} levels, the execution speed dropped even lower making the V_{crash} energy-consumption higher than the result at V_{min}. This explains why the BERT model did obtain a V_{crash} lower than V_{min} during inference, but there were no energy-saving improvements displayed in figure 11.

Overall, voltage guardband exploitation provides relevant improvements in the context of energy efficiency maximization for deep learning applications.

VI. Conclusions

The present work proposes an approach to study the energy savings potential of modern deep learning applications on modern GPGPU devices, using a AMD Radeon Vega Frontier Edition GPGPU as a case study.

First of all, the GPGPU device's voltage guardband is characterized using benchmarks.

Results show an undervoltage potential ranging from ranging from $16.9\,\%$ to $20.7\,\%$ with an average of $15.68\,\%$ on the synthetic benchmarks set, and ranging from ranging from $11.04\,\%$ to $12.34\,\%$ with an average of $11.60\,\%$ on the synthetic benchmarks set. Thus, confirming the expectations that dependency benchmarks would obtain higher V_{min} readings, i.e. a lower voltage guardband.

When operating at V_{min}, the GPGPU device achieved a energy efficiency ranging from $14.58\,\%$ up to $45.05\,\%$ with an average of $26.4\,\%$.

The benchmark results have also shown a V_{min} variability of $0.06\,\mathrm{V}$ corresponding to $6.2\,\%$ when compared to the nominal voltage. An analysis, bearing in mind potential causes for the V_{min} variability, concluded that it is deeply connected with the application itself. The device's operating frequency, temperature, aging, process variation, and inter-kernel executions all rendered an insufficient V_{min} variability to explain the variability magnitude observed in the benchmarks.

Knowing the application itself is the root cause of the V_{min} variability, deep learning models were introduced where the same endeavor was repeated.

Results showed deep learning models can achieve energy savings of up to $24.79\,\%$ with an average of $15.35\,\%$ whilst guaranteeing the nominal accuracy. Furthermore, by working at V_{crash}, energy efficiency can be increased by an average of $2.72\,\%$ at the expense of the model's accuracy. When the GPGPU is set to work at near failure supply voltages, V_{crash}, the observed accuracy droop achieved an average of $10.61\,\%$ and a maximum of $61.52\,\%$.

Additionally, there were executions further below V_{crash} that achieved even higher energy-saving results. Obviously, given V_{crash} definition, there were also failed executions at those voltage levels, hence those executions were disregarded. Nonetheless, this means V_{crash} is not the ultimate limit on the voltage guardband reduction approach when seeking lower power consumption.

References

[1] K. Hazelwood, S. Bird, D. Brooks, S. Chintala, U. Diril, D. Dzhulgakov, M. Fawzy, B. Jia, Y. Jia, A. Kalro, *et al.*, "Applied machine learning at facebook: A datacenter infrastructure perspective," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, IEEE, 2018, pp. 620–629.

[2] S. Mittal and S. Vaishay, "A survey of techniques for optimizing deep learning on gpus," *Journal of Systems Architecture*, vol. 99, p. 101 635, 2019.

[3] J. Leng, Y. Zu, and V. J. Reddi, "Gpu voltage noise: Characterization and hierarchical smoothing of spatial and temporal voltage noise interference in gpu architectures," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, IEEE, 2015, pp. 161–173.

[4] S. Mittal and J. S. Vetter, "A survey of methods for analyzing and improving gpu energy efficiency," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, pp. 1–23, 2014.

[5] Y. Jiao, H. Lin, P. Balaji, and W.-c. Feng, "Power and performance characterization of computational kernels on the gpu," in *2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, IEEE, 2010, pp. 221–228.

[6] J. Guerreiro, A. Ilic, N. Roma, and P. Tomás, "Dvfs-aware application classification to improve gpgpus energy efficiency," *Parallel Computing*, vol. 83, pp. 93–117, 2019.

[7] J. Leng, A. Buyuktosunoglu, R. Bertran, P. Bose, and V. J. Reddi, "Safe limits on voltage reduction efficiency in gpus: A direct measurement approach," in *2015 48th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, IEEE, 2015, pp. 294–307.

[8] C. Constantinescu, I. Parulkar, R. Harper, and S. Michalak, "Silent data corruption—myth or reality?" In *2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN)*, IEEE, 2008, pp. 108–109.

[9] V. J. Reddi, M. S. Gupta, K. K. Rangan, S. Campanoni, G. Holloway, M. D. Smith, G.-Y. Wei, and D. Brooks, "Voltage noise: Why it's bad, and what to do about it," in *5th IEEE Workshop on Silicon Errors in Logic-System Effects (SELSE), Palo Alto, CA*, Citeseer, 2009.

[10] V. J. Reddi, S. Kanev, W. Kim, S. Campanoni, M. D. Smith, G.-Y. Wei, and D. Brooks, "Voltage smoothing: Characterizing and mitigating voltage noise in production processors via software-guided thread scheduling," in *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, IEEE, 2010, pp. 77–88.

[11] AMD, "Radeon's next-generation Vega architecture," Advanced Micro Devices, Inc., Tech. Rep., 2017.

[12] J. H. Stathis, S. Mahapatra, and T. Grasser, "Controversial issues in negative bias temperature instability," *Microelectronics Reliability*, vol. 81, pp. 244–251, 2018.

[13] E. Cai, D. Stamoulis, and D. Marculescu, "Exploring aging deceleration in finfet-based multi-core systems," in *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, IEEE, 2016, pp. 1–8.

[14] P.-F. Lu, K. A. Jenkins, T. Webel, O. Marquardt, and B. Schubert, "Long-term nbti degradation under real-use conditions in ibm microprocessors," *Microelectronics Reliability*, vol. 54, no. 11, pp. 2371–2377, 2014.

[15] S. Mittal, "A survey of architectural techniques for managing process variation," *ACM Computing Surveys (CSUR)*, vol. 48, no. 4, pp. 1–29, 2016.

[16] K. A. Bowman, S. G. Duvall, and J. D. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE Journal of solid-state circuits*, vol. 37, no. 2, pp. 183–190, 2002.

[17] J. Guerreiro, A. Ilic, N. Roma, and P. Tomas, "Gpgpu power modeling for multi-domain voltage-frequency scaling," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, IEEE, 2018, pp. 789–800.