A Study on Power and Energy Measurement of NVIDIA Jetson Embedded GPUs Using Built-in Sensor

Büşra Aslan
Computer Engineering Department
Istanbul Technical University
MGEO, AselsanN Inc.
Ankara, Türkiye
baslan@aselsan.com.tr

Ayse Yilmazer-Metin
Computer Engineering Department
Istanbul Technical University
Istanbul, Türkiye
yilmazerayse@itu.edu.tr

2022 7th International Conference on Computer Science and Engineering (UBMK) | 978-1-6654-7010-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/UBMK55850.2022.9919522

***Abstract*—Artificial intelligence (AI) has been shifted to the embedded devices known as edge devices. Component-level power is very important for the design and optimization of applications on edge devices to estimate energy consumption. Thus, accurate power measurements are needed for battery-powered systems. However, it is not straightforward. Because the behavior of a GPU is rather complex and not well documented. In this work, we report challenges getting power measurements using the built-in power sensor for an NVIDIA Jetson GPU device. We provide a method for true power and energy measurements of the kernels running on NVIDIA Jetson family GPUs.**

***Keywords* —dynamic power, edge devices, energy calculation, experimentation, internal power sensor, power monitoring, power profiling**

I. Introduction

GPU-based accelerators can afford greater computing performance and better energy efficiency than multicore CPUs [1]. Thus, the low-end GPUs have become the perfect deployment choices in handheld and embedded devices. Battery life is very crucial in all types of low-end embedded devices. Accurate power measurements are critical for determining bottlenecks of programs executing on GPUs and optimizing energy successfully. The power consumption of many processor units has been investigated from the past to the present [2], [3], [4].Power measurement methods for GPUs are generally performed using external power meters and statistical or predictive models [5], [6]. Internal power sensors are finding widespread use to provide power measurements because they enable convenient profiling. The new generation NVIDIA GPUs have built-in power sensors to enable querying the power consumption of CPU, GPU, or SoC. In [7], Burtscher and Zecena performed a study to measure the power consumption of a K20 GPU using a built-in sensor. In their study, they witnessed some unexpected behaviors and presented an approach to obtain correct power and energy consumption. This work has shown that we should understand the output of built-in sensors to make use of their readings for power monitoring. They worked on an NVIDIA's Kepler GPU which is NVIDIA's first microarchitecture focused on energy efficiency.

Neural network-based applications have become popular to be run on abroad range of devices from mobile devices to workstations. Performance, power, and energy consumption are important parameters for neural network applications. The preference for artificial intelligence applications to run on edge devices is increasing because of latency and privacy requirements [8]. In this study, we wanted to look at the power and energy consumption of the AI applications on NVIDIA's Jetson family embedded GPUs.[1] Similar to Burtscher and Zecena's work, we met similar behaviors when measuring power using Jetson's built-in sensor. Therefore we have decided to apply and validate their methodology to measure power and energy in our experiments. While we validate some of their findings, we found some differences, because of the different GPU architectures. When we apply their true power formula using these values, we observe large spikes in power profiles. We examine the reasons and add an intermediate process to obtain square-wave power profiles for Jetson GPUs.

In summary, we examine the power profile of an embedded GPU using built-in sensor data. We perform our experiments on an NVIDIA Jetson TX2 GPU and use CUDA environment. We make important observations about the power profile of CUDA kernels. We demonstrate an additional process to compute the true power and energy consumption when using the proposed methodology in [7].

We report related works in the literature in section II. We provide details on the hardware and software components of our experimental setup in section III. We analyze the result of our GPU's built-in sensor measurements in section IV. Then, we explain our approach to obtain the correct power profile in section V. We report the validation results using a Jetson AGX Xavier GPU in section VI. Finally, we discuss future works and conclude our study in section VII.

[1]Jetson TX2 and Jetson Xavier are power-efficient embedded AI computing devices from Pascal and Volta microarchitecture, respectively.

II. Related Work

Power monitoring provides an opportunity to evaluate and improve the energy efficiency of codes. In the literature, there are several studies about the energy consumption of GPUs [9], [10]. As distinct from them, our study analyzes the power and energy measurements of NVIDIA Jetson embedded GPUs.

There are two techniques to measure power consumption of a hardware component as direct measurement and indirect measurement. Direct measurements use internal or external hardware sensors to read power periodically during a time interval. External power meters provide accurate measurement but require physical system access and probing [6], [11], [12]. Internal onboard power sensors allow component-level power monitoring when they are available. They are easier to use and do not need additional hardware. On the other hand, indirect measurements use a model that estimates power consumption using hardware performance counters. Most of the published works in this area are about power estimation models [13], [14], [15].

In [7], the authors analyzed the power consumption measured by NVIDIA Management Library (NVML) based on the built-in power sensor, and proposed a methodology to compute the instant power and energy consumption of K20 GPUs.[2] Coplin studied the effects of the core and memory clock frequencies on energy consumption, instant power and runtime for a K20c GPU using the built-in power sensor [1]. Ikram proposed an experimental methodology for measuring peak and average power, energy, and run-time of any program executing on NVIDIA Tesla K40c GPU by using NVIDIA System Management Interface (NVSMI) to read power sensors on the GPU [17].[3] Sen and Imam examined four different GPU power profiling techniques and discussed the effects of moving-average filters on the measured power profiles [19]. Holly and Wendt provided a power measurement method that uses a built-in power sensor for CNNs for generating estimation models on NVIDIA Jetson Nano [20].

III. Experimental Setup

A. GPU Description

Jetson is a series of embedded computing boards from NVIDIA. Our target hardware is NVIDIA Jetson TX2 [21]. Jetson TX2 is a fast and power-efficient embedded AI computing device. It includes two NVIDIA Denver 2 CPU cores and four ARM Cortex-A57 CPU cores to offer different power and performance options. It is built around an NVIDIA Pascal-family GPU. It consists of 256 CUDA cores and 2 Streaming Multi-Processors (SMs). The Jetson TX2 module is equipped with an onboard TI-INA3221x power sensor chip [22].

B. Microbenchmarks

We designed and implemented a microbenchmark set to study the power characteristics of NVIDIA Jetson family GPUs in detail. While we study various microbenchmarks with different computation and memory access characteristics that we developed, here we only focus on one microbenchmark to explain the methodology.[4] It is a simple kernel that performs vector multiplication in a loop that we refer it as *MUL* kernel throughout this paper. Using *MUL* kernel, we stress arithmetic logic units (ALU) of the GPU. Power data are obtained from the built-in sensor while running the microbenchmarking kernel on the GPUs at MAX-N power mode.[5]

C. Power Monitoring

Unlike previous studies [24], [25], [13], [26], NVML is not supported on Jetson platforms. The power sensors can either be read automatically with the *tegrastats* [27] application or manually by reading *sysfs* nodes in the Jetson TX2 module. We developed a CUDA based program to collect the power consumption. We use two threads. The first thread launches the kernel while the second one reads the power and takes timestamps. The power monitoring thread running on the CPU reads the power in milliwatts through the sys-file when a kernel running on the GPU.

We preferred manual reading to achieve a higher time resolution and flexibility. During the measurements, we connected the device via an Ethernet port. It is used for communication over SSH. We disabled the graphical user interface to reduce the number of background processes running during our experiments. We observed that the average time resolution is approximately 422 µs when we run the power monitoring thread in a loop without delay.

IV. Measurements and Observations

We measure the power consumption before, during, and after the execution of the kernel in a loop without delay. We allocate memory for input and output vectors and initialize them at random. Then, we transfer input vectors from the host to GPU memory.

We start with measuring the power consumption of very-short running *MUL* kernel on the NVIDIA Jetson TX2 GPU when running eight times.[6] *MUL* kernel starts at time `t1` and ends at time `t2`. The total run-time for this kernel is approximately 160 ms. Fig. 1 displays the power profile of this execution. We observe edges in the power profile. The increase and decrease of power consumption is in the form of steps.[7] In Fig. 1, we observe the idle power as 152 mW. The last step on 229 mW takes some more time before reaching

[2] NVML [16] is an API that queries the onboard sensors to obtain the instantaneous power consumption of the GPU device.

[3] NVSMI [18] is a profiling tool based on NVML to access the built-in sensors.

[4] We observe similar behaviours with other microbenchmarks that we studied.

[5] Kernel launch configuration is critical to obtain a good occupancy and Streaming Multiprocessor (SM) activity level [23] on the GPU. In our experiments, we set the block and grid sizes accordingly to obtain a good occupancy and SM activity.

[6] Unless we state otherwise, our experiments run on the NVIDIA Jetson TX2 GPU.

[7] We run two other microbenchmarks: *MEM* and *FSIN* Kernel. With these kernels, we stress memory units and special function units (SFU) of the GPU. The run-time of each of these kernels is approximately equal to the very-short running *MUL* Kernel. We obtain a similar shape of power profile, but different peak power level.

to the idle power level. We believe that GPU waits to launch a new kernel before stabilizing at the idle level.

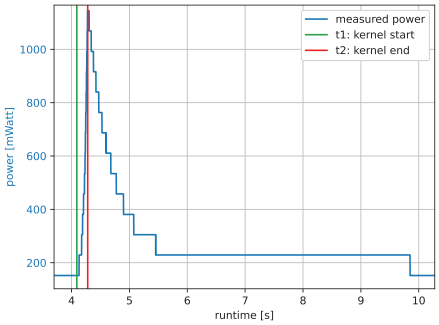

Fig. 1. Power profile of *MUL* kernel running 8 times.

When the execution of the kernel starts, the computational activity on the SMs immediately rises, remains constant during the execution, and then immediately is set to zero [7]. Therefore, the expected power profile is similar to the one that is shown in Fig. 2. However, the power profile that we observed is in the shape with a gradual increase at the start of the kernel and a gradual decrease at the end of the kernel. The power consumption has a *delay* at the beginning of the execution and a *tail* after the end of the execution.

Fig. 2 highlights this difference between the observed power profile and the kernel activity.[8] Reaching the peak level takes approximately 2,5 seconds in the observed power profile. **The difference between the kernel activity and power profile causes a problem to obtain instant power and energy consumption of kernels.** To calculate the energy of a kernel, we can simply integrate the power over the run-time. **But the integral from `t1` to `t2` does not correspond to true energy consumption due to the delay and the tail in power profile.**

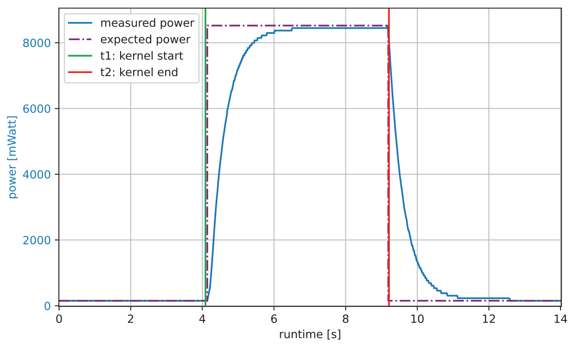

Fig. 2. Expected and measured power profiles of *MUL* kernel.

Next, we increase the run-time of *MUL* kernel by increasing and then doubling the loop count in the kernel. We aim to see whether the behavior of power consumption (in terms of increase and decrease) will change over time. First, the run-time of the kernel is around 5 seconds, and the second time it is around 10 seconds. Fig. 3 displays the power profile of the kernel when running with increased loop counts. There are still steps on the power profile but it looks like exponentially increasing and decreasing in the macroscopic scenario. We expect that the energy should also double, but there is a discrepancy. This behavior is consistent with the findings that are given in [7]. Here, we again observe the sharp profiles with manual readings compared to the NVML data given in [7]. In [7], Burtscher and Zecena used the NVML's function to read power consumption, but we do not use any library function. We think that edges occur due to this sampling dissimilarity.

[8]The figure here shows the power profile of *MUL* kernel at a higher scale.

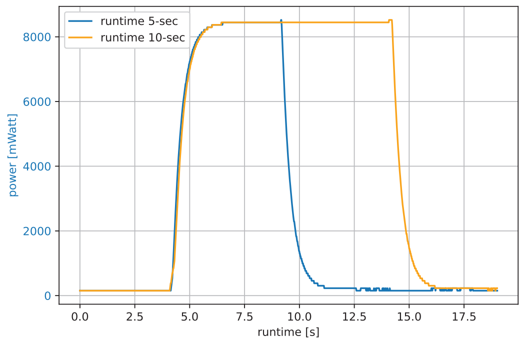

Fig. 3. Power profile of *MUL* kernel with increased run-time (running around 5 seconds and 10 seconds).

Then, we reduce the run-time of *MUL* kernel by reducing the loop count by five times in the kernel so that we obtain the run-time of the kernel around 1 second. We add a one-second delay in between two invocations of the kernel. Fig. 4 displays the power profile of two runs with a one-second delay. The second execution starts at a higher power level and reaches to a higher peak power level. Therefore, the energy consumption is more for the second execution.

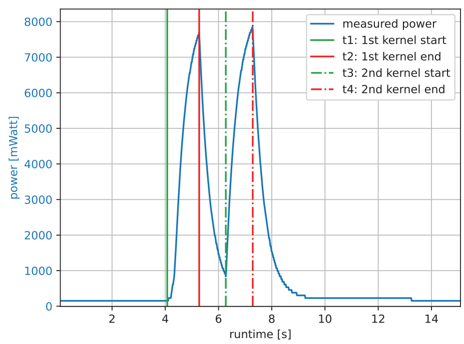

Fig. 4. Power profile when executing *MUL* kernel with a run-time of 1 second twice with a one-second delay.

Fig. 5 shows the sampling durations between consecutive power samples before, during, and after the execution of *MUL* kernel when sampling the power without delay. We make the following observations from this experiment. Firstly, interval lengths are not equal. Secondly, the patterns of interval lengths before and after the execution look similar. Rapid and consistent fluctuations occur around 495 µs in these phases. During the execution, average interval length decreases insignificantly, as shown in Fig. 5. Unlike Burtscher and Zecena's finding [7], the interval lengths in our experiments fluctuate less during the execution of the kernel.

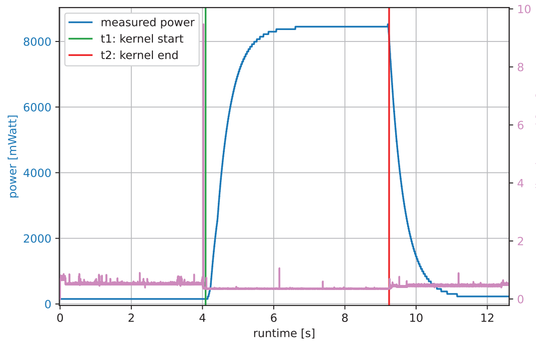

Fig. 5. Elapsed times between samples (right axis, violet line) and power profile (left axis, blue line) of *MUL* Kernel when sampling the power without delay.

In Fig. 5, we see that there is a large spike at around 9,47 ms, immediately before `t1`. Kernel launch occurs about this time. We think that driver activity causes this large delay around the power measurement request. Unlike [7], we cannot say that the interval length increases when the built-in sensor taking a new measurement. It is clearly seen in Fig. 6.

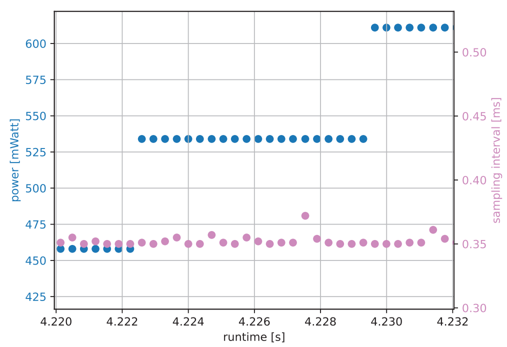

Fig. 6. Zoomed look of Fig. 5 showing the power samples (blue points, left axis) and the corresponding interval lengths (violet points, right axis).

Edges in the power profile decrease when we add a delay between power sampling. But we cannot increase the sampling delay further because the granularity of the sampling interval is important. In other words, we need an acceptable resolution to correctly measure the energy consumption of kernels with very-short runtimes by using the built-in sensor.

In the end, when we implement the proposed approach in [7] to calculate true power and energy, we observe a noticeable discrepancy on Jetson TX2. We measure power consumption with a constant sampling period of 14 ms as determined in [7]. In their approach, Burtscher and Zecena used the function in (1) to eliminate the curve between the times `t1` and `t2` as shown in Fig. 2. The constant value (C) in (1) compensates the differences between the measured and expected values. It should be determined experimentally for a target GPU. Similar to Burtscher and Zecena's study [7], we found the constant value (τ or C) 415 ms for Jetson TX2.

$$P_{true}(t_i) = P_{meas}(t_i) + C \times (P_{meas}(t_{i+1}) - P_{meas}(t_{i-1}))/(t_{i+1} - t_{i-1}) \quad (1)$$

However, we cannot obtain a rectangular profile like illustrated in [7]. There are notable errors in the corrected profile due to edges as seen in Fig. 7.

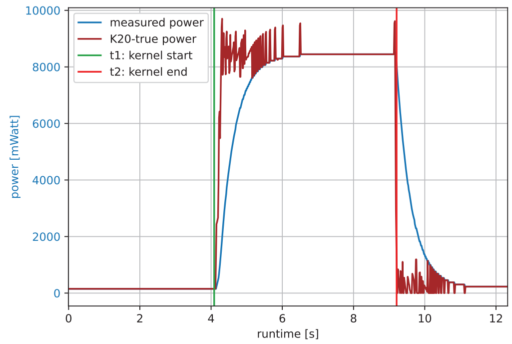

Fig. 7. Corrected power profile when we apply the proposed approach (in Burtscher and Zecena's paper) to the raw power measurements of *MUL* kernel.

We observe that the difference between power measurements is almost always 75 mW on Jetson TX2. This behaviour causes edges on the power profile. The difference is occasionally 1 mW. This behaviour causes wobbles on the power profile. Wobbles are related to the resolution of the power sensor. These errors can be negligible because of averaging of values. But edges are quite important and should be handled to obtain true measurements.

V. Our Proposed Approach

The moving average filter (MAF) is a simple and practical method to reduce white noise and provide smoothness for many applications [28]. It also has applications in machine learning, digital signal processing, and processing measured power profiles [19].

$$y[i] = \frac{1}{L} \sum_{j=-(L-1)/2}^{(L-1)/2} x[i+j] \quad (2)$$

The formula of the moving average filter is shown in (2). The array of x is the input signal, the array of y is the output signal, and L is the window size. The input points in the window is chosen symmetrically around the output point.

In Fig. 8, we apply a 9-point moving average filter to all power measurements to obtain the non-sharp curves for the power profile. As a result, we obtained the rectangle shape by filtering the measured power profile using the 9-point moving average filter and calculating the corrected values for all filtered power samples with (1) for C = 415 ms. The peak power level is about 8500 mW. We calculate the energy consumption of the kernel by integrating power values from t1 to t2. The energy consumption is calculated as 42,449 J using the measured power values, whereas 42,467 J using the corrected power values. There is a %0,04 difference.

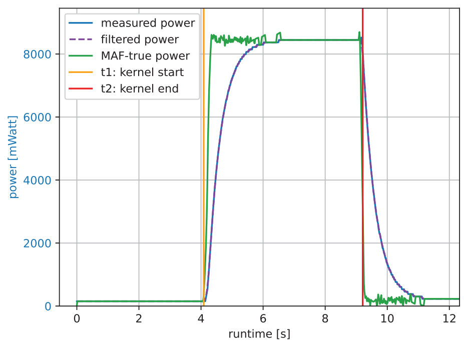

Fig. 8. Corrected power profile of *MUL* kernel running on the Jetson TX2 GPU.

Based on our observations, we propose the following approach for obtaining the true power and energy consumption of a kernel running on a Jetson GPU:

1) Read power manually with 14 ms sampling intervals during execution via a built-in sensor and include time stamps.
2) Apply moving average filter on raw power measurements.
3) Compute power consumption with (1) using filtered power measurements.
4) Compute energy consumption by accumulating the products of the corrected power and the elapsed time values during the kernel execution.

VI. Validation of Results

Fig. 9 displays that our approach resolves the swelling of power and energy consumption of the second kernel if there is a short time between two executions. The corrected profiles of the two runs are at a similar power level. The power is nearly 229 mW between the executions.

A. Jetson AGX Xavier

We repeated our experiments on a different Jetson family GPU using our *MUL* kernel. We observe that the maximum power level is higher than Jetson TX2 GPU but the time to reach the peak level is close. It is approximately 2 seconds. Fig. 10 demonstrates that the constant value (C) is appropriate to calculate true power and energy consumption for Jetson AGX Xavier GPU. The corrected power level is about 15800 mW. The energy consumption is calculated as 97,142 J using the measured power values, whereas 97,311 J using the corrected power values. There is a %0,17 difference.

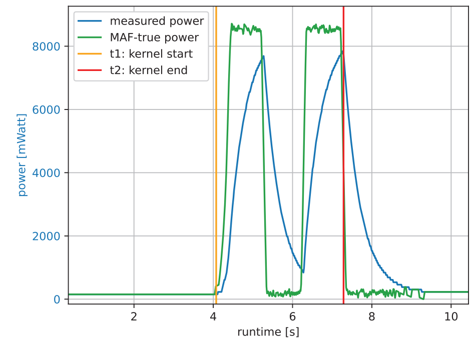

Fig. 9. Corrected power profile of *MUL* kernel running two times with a 1-second delay between invocations.

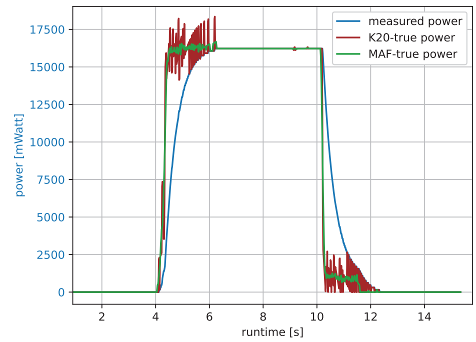

Fig. 10. Corrected power profile of *MUL* kernel running on the Jetson AGX Xavier GPU.

B. Applying our Approach to a GNN Benchmark on Jetson TX2 GPU

Finally, we repeated the experiments using a Graph Neural Network model, named StemGNN [29], on the Jetson TX2 GPU. We measured the power during the inference of the model. We used *ECG5000* dataset provided by [29]. Fig. 11 displays the corrected power profile after application of our approach. The corrected power level is about 6100 mW. It seems that the power level drops to 4750 mW shortly after the kernel starts. Then it rises again to 6100 mW. Some SMs may stay idle when the code is irregular. Therefore, power consumption may decrease depending on the decrease in utilization.

VII. Conclusion and Future Work

Researchers are still working on power models for GPUs because a modern GPU is a black box with complicated and not well documented power behavior. We have analyzed the behavior of power consumption of Jetson family of NVIDIA

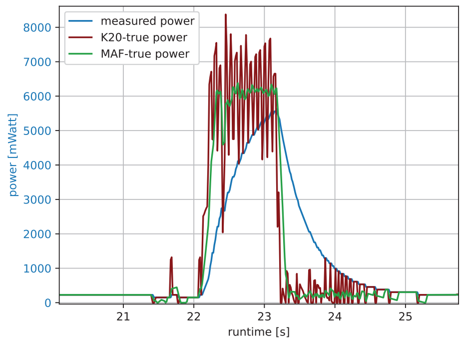

Fig. 11. Corrected power profile of StemGNN application running on the Jetson TX2 GPU.

GPUs. We have observed sharp profiles with manual power readings compared to NVML data. As a result, we have provided a reliable and simple method using a moving average filter to correctly compute power and energy consumption of GPU kernels based on built-in power sensors. We would like to evaluate and validate our approach using more sophisticated real world applications and develop a performance counter based power model. We plan to use our power measuring approach when collecting power data.

References

[1] J. Coplin and M. Burtscher, "Energy, power, and performance characterization of gpgpu benchmark programs," in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2016, pp. 1190–1199.

[2] J. Chen, B. Li, Y. Zhang, L. Peng, and J.-k. Peir, "Statistical gpu power analysis using tree-based methods," in *2011 International Green Computing Conference and Workshops*. IEEE, 2011, pp. 1–6.

[3] L. Senn, E. Senn, and C. Samoyeau, "Modelling the power and energy consumption of nios ii softcores on fpga," in *2012 IEEE International Conference on Cluster Computing Workshops*. IEEE, 2012, pp. 179–183.

[4] Y. S. Shao and D. Brooks, "Energy characterization and instruction-level energy model of intel's xeon phi processor," in *International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2013, pp. 389–394.

[5] S. Collange, D. Defour, and A. Tisserand, "Power consumption of gpus from a software perspective," in *International Conference on Computational Science*. Springer, 2009, pp. 914–923.

[6] X. Ma, M. Dong, L. Zhong, and Z. Deng, "Statistical power consumption analysis and modeling for gpu-based computing," in *Proceeding of ACM SOSP Workshop on Power Aware Computing and Systems (HotPower)*, vol. 1. Citeseer, 2009.

[7] M. Burtscher, I. Zecena, and Z. Zong, "Measuring gpu power with the k20 built-in sensor," in *Proceedings of Workshop on General Purpose Processing Using GPUs*, 2014, pp. 28–36.

[8] H. Bouzidi, H. Ouarnoughi, S. Niar, and A. A. E. Cadi, "Performance prediction for convolutional neural networks in edge devices," *arXiv preprint arXiv:2010.11297*, 2020.

[9] R. A. Bridges, N. Imam, and T. M. Mintz, "Understanding gpu power: A survey of profiling, modeling, and simulation methods," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, pp. 1–27, 2016.

[10] M. Fahad, A. Shahid, R. R. Manumachu, and A. Lastovetsky, "A comparative study of methods for measurement of energy of computing," *Energies*, vol. 12, no. 11, p. 2204, 2019.

[11] H. Nagasaka, N. Maruyama, A. Nukada, T. Endo, and S. Matsuoka, "Statistical power modeling of gpu kernels using performance counters," in *International conference on green computing*. IEEE, 2010, pp. 115–122.

[12] X. Ma, M. Rincon, and Z. Deng, "Improving energy efficiency of gpu based general-purpose scientific computing through automated selection of near optimal configurations," *University of Houston2011*, 2011.

[13] S. Song, C. Su, B. Rountree, and K. W. Cameron, "A simplified and accurate model of power-performance efficiency on emergent gpu architectures," in *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*. IEEE, 2013, pp. 673–686.

[14] B. Goel and S. A. McKee, "A methodology for modeling dynamic and static power consumption for multicore processors," in *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2016, pp. 273–282.

[15] M. J. Walker, S. Diestelhorst, A. Hansson, A. K. Das, S. Yang, B. M. Al-Hashimi, and G. V. Merrett, "Accurate and stable run-time power modeling for mobile and embedded cpus," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 1, pp. 106–119, 2016.

[16] "Nvlm api reference guide." NVIDIA Corporation, 2022. [Online]. Available: https://docs.nvidia.com/deploy/nvml-api/index.html

[17] M. J. Ikram, O. A. Abulnaja, M. E. Saleh, and M. A. Al-Hashimi, "Measuring power and energy consumption of programs running on kepler gpus," in *2017 Intl Conf on Advanced Control Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT)*. IEEE, 2017, pp. 18–25.

[18] "nvidia-smi 367.38." NVIDIA Corporation, 2016. [Online]. Available: https://developer.download.nvidia.com/compute/DCGM/docs/nvidia-smi-367.38.pdf

[19] S. Sen, N. Imam, and C.-H. Hsu, "Quality assessment of gpu power profiling mechanisms," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2018, pp. 702–711.

[20] S. Holly, A. Wendt, and M. Lechner, "Profiling energy consumption of deep neural networks on nvidia jetson nano," in *2020 11th International Green and Sustainable Computing Workshops (IGSC)*. IEEE, 2020, pp. 1–6.

[21] "Harness ai at the edge with the jetson tx2 developer kit." NVIDIA Corporation, 2018. [Online]. Available: developer.nvidia.com/embedded/jetson-tx2-developer-kit

[22] "Ina3221 datasheet." Texas Instruments, 2020. [Online]. Available: https://www.ti.com/lit/ds/symlink/ina3221.pdf

[23] "Profiler user's guide." NVIDIA Corporation, 2022. [Online]. Available: https://docs.nvidia.com/cuda/pdf/CUDA_Profiler_Users_Guide.pdf

[24] V. Kandiah, S. Peverelle, M. Khairy, J. Pan, A. Manjunath, T. G. Rogers, T. M. Aamodt, and N. Hardavellas, "Accelwattch: A power modeling framework for modern gpus," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, 2021, pp. 738–753.

[25] Y. Arafa, A. ElWazir, A. ElKanishy, Y. Aly, A. Elsayed, A.-H. Badawy, G. Chennupati, S. Eidenbenz, and N. Santhi, "Verified instruction-level energy consumption measurement for nvidia gpus," in *Proceedings of the 17th ACM International Conference on Computing Frontiers*, 2020, pp. 60–70.

[26] V. Adhinarayanan, B. Subramaniam, and W.-c. Feng, "Online power estimation of graphics processing units," in *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*. IEEE, 2016, pp. 245–254.

[27] "Tegrastats utility." NVIDIA Corporation, 2019. [Online]. Available: https://docs.nvidia.com/cuda/profiler-users-guide/index.html

[28] S. W. Smith, in *The Scientist and Engineer's Guide to Digital Signal Processing*, 2016, p. Chapter 15.

[29] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong *et al.*, "Spectral temporal graph neural network for multivariate time-series forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 766–17 778, 2020.