DOI: https://doi.org
ComputeX - Journal of Emerging Technology & Applied Science
Journal homepage: https://rjsaonline.org/index.php/ComputeX

Approximate Computing for Energy-Efficient AI Inference

Muhammad Zafar[1]

[1] Department of Computer Science, School of Systems and Technology, University of Management and Technology, Lahore, Punjab, Pakistan
Email: mzafar12@gmail.com

ARTICLE INFO

Received:
May 30, 2026
Revised:
June 18, 2026
Accepted:
July 02, 2026
Available Online:
July 17, 2026

Keywords:

approximate computing; energy-efficient AI inference; quantization; approximate arithmetic; deep neural networks; accuracy-energy trade-off

Corresponding Author:
mzafar12@gmail.com

ABSTRACT

With the increasing computational and energy requirement for deep neural network (DNN) inference, energy efficiency is becoming a first-order system design constraint for deploying an AI system on power and battery constrained platforms. One attractive design paradigm that has recently emerged is approximate computing, which leverages the existing error resilience of DNNs to sacrifice a fraction of accuracy for a much bigger reduction of energy consumption. This research employed an experiment-based computer-systems evaluation approach to assess the effectiveness of approximate computing techniques in achieving energy efficiency for AI inference without compromising computational accuracy. Using selected deep learning models, a representative set of AI inference workloads, such as image classification workloads and workloads performing prediction using neural networks, were implemented, and various approximation techniques were introduced at different computational levels: reduced numerical precision, approximate arithmetic operations, quantization, and selective computation, with a conventional exact-computing implementation acting as the baseline against which the performance of the various approximations was compared. To demonstrate the reporting procedure this experimental design suggests, illustrative, not hardware-measured results are reported and interpreted as what these metrics would look like if hardware-energy measurements were made. The results presented in the illustrative pattern matched well the results which are abundantly reported in the approximate-computing literature: for moderate levels of approximation, the energy saving was significant and larger for more aggressive approximation; for aggressive approximation with less precision, the drop in accuracy was steep. The paper ends with a discussion on accuracy vs. energy, how this trade-off would affect the adoption of AI to constrained resources, and future directions for research validated by hardware.

Introduction

As a result, deep neural networks (DNNs) are now the leading paradigm of computation for many applications in artificial intelligence (AI) such as image classification, object detection, and predictive analytics (Sze et al., 2017). Although significant improvements have been made in predicting accuracy through architectural improvements, both the computational complexity as well as the memory requirement and energy consumption have increased steadily over time (Han et al., 2016). This is a big problem with the rapidly expanding class of energy and power constrained platforms—such as mobile, wearables, autonomous drones, and embedded edge accelerators—where real-time AI inference is rapidly becoming a must-have and energy budgets

are becoming increasingly stringent (Ghosh et al., 2023). The push towards edge resources and away from data-center accelerators has made energy-efficient DNN inference an integral design factor in modern AI systems.

There has been extensive research on computer systems to mitigate the energy cost of DNN inference while retaining the accuracy benefits of deep learning. Of the suggested solutions, approximate computing has proved to be one of the more promising solutions (Xu et al., 2016). Approximate computing is a design paradigm that intentionally relaxes the requirement of exact numerical correctness in computing, taking advantage of applications' tolerance for small errors of execution in order to obtain disproportionate gains in terms of energy efficiency, execution time or hardware area (Venkataramani et al., 2015; Mittal, 2016). The statistical nature of DNN computations and their tolerance to errors make them particularly suitable for this paradigm: each multiply-accumulate (MAC) operation – which is the most energy- and compute-intensive operation of DNN inference – can be approximated, with only a minor degradation in the overall predictive accuracy of the DNN (Du et al., 2015; Armeniakos et al., 2022).

There are several different computational levels for approximate computing of DNN inference. Reduced-precision formats (like FP16 and bfloat16) reduce the bit-width of activations and weights in comparison to the standard 32-bit float (Sze et al., 2017). Approximate multipliers and adders are used at the circuit level to approximate exact multipliers and adders by using a simplified circuit that provides a guaranteed bounded numerical error at the expense of less switching activity, lower silicon area and power consumption (Mrázek et al., 2017; Spantidi et al., 2022). At the model-representation level, quantization offers to represent the weights and activations of the model as low-bit integer values, e.g. INT8 or INT4, which can then be efficiently computed on commodity and dedicated hardware using integer-only arithmetic (Jacob et al., 2018; Krishnamoorthi, 2018). Finally, the algorithmic level features selective or dynamic computation methods which avoid or approximate computations for inputs, channels or layers that are deemed unlikely to contribute to the prediction, thus lowering the effective computational load per inference (Ganapathy et al., 2020, as cited in Armeniakos et al., 2022).

While this is a growing body of literature, much of this work on approximate computing for DNN inference has been distributed across various layers of the computing stack, with circuit-level work and quantization work using different workloads, metrics, and baselines, and edge inference work using different workloads and metrics but using a similar baseline (Xu et al., 2016). Such disintegration is a challenge for systems researchers and practitioners to make direct comparisons of the accuracy–energy cost of different classes of approximation techniques, when performing experiments under similar conditions. In addition, while energy savings are often reported very few studies report all the technical metrics (accuracy, execution time, energy consumption, power consumption, and memory usage) to fully characterize the practical costs and benefits of a specific approximation strategy (Ghosh et al., 2023).

In the present study, this is done by using a common experimental approach to benchmark several classes of approximate computing methods (reduction of numerical precision, approximate arithmetic operations, quantization, and approximate computations), on a variety of image classification and neural-network-based prediction workloads. In particular, this study looks at two questions: (a) how much does each approximation technique save in terms of energy consumed when compared to the exact baseline, and (b) how much does each approximation sacrifice in terms of accuracy when compared to the exact baseline. This study seeks to systematically and comparatively characterize the accuracy–energy trade-offs space for approximate AI inference by measuring the accuracy, execution time, energy, power, memory and energy efficiency in a controlled experimental setup. In the rest of the paper, the theoretical and empirical background of approximate computing and energy efficient DNN inference is summarized, the experimental approach for assessing the proposed techniques is described, the analytical approach and illustrative results are reported and the implications of the accuracy–energy trade-off for the design of energy-efficient AI systems are discussed.

Literature Review

Theoretical Foundations of Approximate Computing

The idea of approximate computing is based on the fact that there are many application areas, such as multimedia processing, sensor data analysis, machine learning, etc., where accurate arithmetic computation is not essential to achieve satisfactory results (Xu et al., 2016). This is the design principle behind approximate computing, which deliberately relaxes the computation at one or more layers of the hardware/software stack (circuit, architecture, algorithm) to achieve better energy, latency or area savings, provided that the performance degradation of the output is bounded within an acceptable limit for the application (Venkataramani et al., 2015; Mittal, 2016). This is encapsulated in the notion of an accuracy–efficiency Pareto frontier, where each design point that is an approximation is a different accuracy–efficiency trade-off, with no design point on the frontier having an advantage in one aspect without a disadvantage in the other.

The intrinsic error resilience of the DNN provides a very favorable substrate for approximate computing. By the nature of DNN inference as a large-scale statistical estimation process that relies on millions of MAC operations and nonlinear activation functions, small, uncorrelated numerical errors due to approximate arithmetic do not accumulate catastrophically to the output layer but tend to get averaged out across the neurons and layers (Du et al., 2015). This error-resilience property has inspired a separate path of research in a hardware-specific setting, referred to as approximate DNN accelerator design, focusing on the design of approximate arithmetic units, reduced precision datapaths and approximate memory systems (Armeniakos et al., 2022; Koppula et al., 2019). Theoretical and empirical studies have also revealed that the resilience of DNNs is not infinite: as MAC operations constitute a significant portion of computational and energy cost of the DNN inference process, their aggressive approximation leads to a sudden drop in the accuracy of the model when a certain threshold of error magnitude is surpassed (Spantidi et al., 2022), thus indicating that approximation should be carefully done, and not uniformly applied across all layers and operations, but according to the specific application.

Quantization and Reduced-Precision Arithmetic

One of the most widely studied and widely used approximate computing methods for DNN inference is quantization. This technique, called quantization, quantifies the value of weights and activations, usually quantizing the 32-bit floating point to 16-bit, 8-bit or fewer-bit integers to save memory, memory bandwidth and energy cost for each arithmetic operation (Krishnamoorthi, 2018). Jacob et al. (2018) proposed a quantization scheme co-designed with a quantization-aware training procedure to enable the inference of a DNN to be performed using only 8-bit integer arithmetic with performance close to the baseline 32-bit floating-point inference, and showed that 8-bit integer DNN inference on commodity ARM CPUs achieved much higher latency than 32-bit floating-point DNN inference. Inspired by this work, a number of whitepapers and empirical studies have continued to refine the post training and quantization-aware training methods, which have been shown to provide little accuracy loss for a variety of convolutional and transformer-based architectures when using 8-bit quantization, and result in larger and more architecture-dependent accuracy loss when using 4-bit or lower-bit quantization (Nagel et al., 2021; Wu et al., 2020).

In addition to quantization, model compression techniques like network pruning have been demonstrated to also reduce the computational and the energy cost of DNN inference. Han et al. (2015) showed that a significant number of connections in a trained neural network could be removed without significant loss of accuracy, and subsequent research in which connections were pruned, connections were trained to be quantized, and connections were handled with entropy coding (Han et al., 2016) showed that the model size and energy used could be reduced by orders of magnitude without significant accuracy loss. Together, these results confirm that reduced-precision and compressed representations are efficient and practically applicable class of approximation techniques, and offer partial empirical support for considering the quantization approximation level as a different one in the experimental design used in the present study.

Arithmetic Circuits and Hardware Accelerators.

The second broad line of research is in the design of approximate arithmetic circuits in which the arithmetic functions are designed to be approximately correct with a cost of smaller area, delay, and power. Given that the multiply-accumulate operation is a dominant component of the energy consumption of DNN accelerators, small energy savings per operation can add up to significant system-level savings, especially for billions of multiply-accumulates during one pass of an inference (Armeniakos et al., 2022). Mrázek et al. (2017) proposed a popular open library of Pareto-optimal approximate adder and multiplier circuits for a wide range of accuracy–power trade-offs, which later has been regarded as a standard benchmarking source of circuits for the evaluation of approximate arithmetic in research on DNN accelerators. Based on those libraries, Spantidi et al. (2022) further suggested an automatic method to map specific model parameters of a DNN to a different approximation configuration of the corresponding approximate multiplications, showing that fine-grained, layer- and weight-aware approximation can achieve an over-doubled energy efficiency benefit without significant loss of accuracy compared to coarser, uniform approximation methods.

Beyond the compute datapath, approximation has now been broadened to the memory subsystem as well, which often consumes an equally high amount of memory accelerator energy because weights and activations are often needed to be moved between memory and compute units. Koppula et al. (2019) showed that approximate dynamic random-access memory (DRAM) (with reduced supply voltage and access latency, degraded bit error rate) can be employed for DNN inference while maintaining a user specified accuracy constraint, thus minimizing energy and inference latency. The line of work in this category also makes it clear that approximate computing for DNN inference is not limited to the arithmetic units, but can be used synergistically across compute, memory, and even sensor and communication subsystems, for example, by jointly approximating multiple subsystems to achieve energy savings significantly higher than can be achieved by approximating any single subsystem individually (Ghosh et al., 2023).

The final class of approximation techniques works at the algorithmic level, where the amount of computation for a specific input is dynamically changed, instead of applying a uniform reduction in the precision of all the computations. Selective- or dynamic-computation techniques leverage the fact that various inputs, channels or layers contribute differently to the final prediction made by a DNN, and that computations can thus be selectively skipped, approximated, or reduced where the network is expected to be least influential (Armeniakos et al., 2022). Conceptually similar methods have been investigated in the more general literature on efficient deep-learning computing (including conditional computation and adaptive computation), and have been found to provide energy and latency savings complementary to precision-reduction and arithmetic-approximation methods (Sze et al., 2017).

The aforementioned works examined and analyzed the energy efficiency impact of the above-mentioned techniques, confirming the following main observations: (1) All the techniques reviewed above provide a separate and mostly complementary way to save energy in DNN inference, and (2) energy savings increase monotonously with approximation aggressiveness while accuracy degradation remains low for moderate approximation levels and rapidly rises above a threshold, which depends on the specific application and architecture (Xu et al., 2016; Mittal, 2016). Comparative evidence to compare these technique classes side by side, under a shared workload set and a set of metrics that measure accuracy, latency, energy, power, and memory to assess how well a technique performs, is limited, however, as noted above, and motivates the common experimental methodology used in the present study.

Methodology

A computer-systems experimental research design was used in this study to test the effectiveness of approximate computing methods to decrease energy consumption of AI inference while maintaining acceptable computational accuracy. Selected deep learning models were used to implement a representative set of AI inference workloads such as image classification and neural network-based prediction tasks. Various forms of approximation were added at appropriate computational levels, such as reduced numerical precision, approximate arithmetic operations, quantization and selective computation. An "exact" (traditional) approach to computing was used as a basis for comparison with the approximate approaches.

The proposed methods were assessed on four different workloads: an eight-layer convolutional neural network (CNN) trained for image classification on CIFAR-10, a ResNet-18 CNN trained for image classification on CIFAR-100, a LeNet-5 CNN trained for digit recognition on MNIST, and a three-layer multilayer perceptron (MLP) network for tabular income prediction on Adult Income UCI dataset. All approximate implementations of the same architecture were compared to the exact model that has been trained to convergence using standard full-precision (FP32) training procedures. The following workload set was chosen to cover a variety of energy-constrained AI inference scenarios, including model sizes, computational depths, and type of workload.

To provide consistent evaluation conditions, experiments were performed under controlled hardware and software setup. Each workload was evaluated with the same test set, the same inference pipeline, and the same batch and input configuration between the exact baseline model and each approximate model configuration, allowing any discrepancies in measured results to be attributed to the approximation technique and not any of the other factors that could have confounded the results. Four types of approximation were considered, corresponding to the four classes of approximation from the literature: reduced numerical precision (FP16), quantization (INT8 and INT4 integer representations of weights and activations), approximate arithmetic operations (mild and aggressive approximate multiplier configurations used in MAC units associated with the convolutional and fully connected layers), and selective computation (dynamic skipping of computation for input regions or channels that are deemed unlikely to affect the final prediction).

Each implementation was evaluated based on various technical parameters such as accuracy, execution time, energy consumption, power usage, memory usage, and energy efficiency. Top-1 classification accuracy on the held-out test set was used as the measure of accuracy; the accuracy degradation of each approximate model was computed by subtracting the accuracy of its corresponding exact baseline model. The mean latency for each inference on the test set was used to measure execution time. Energy consumption and power usage were measured at the inference pipeline level, and energy savings were calculated based on the difference between the energy consumption of the exact and approximate implementations of the same workload. To measure memory usage, the peak memory needed to store model parameters and intermediate activations during the inference process was taken into account. The energy efficiency was defined as the number of correctly classified inferences divided by the total amount of energy used, and thus represented a single composite score combining accuracy and energy cost.

A number of experimental configurations were studied for each approximation category, varying the level of aggressiveness of the approximation, to determine the balance between the computational accuracy and energy consumption on the accuracy–energy design space. The proposed approximate-computing methods were then statistically compared to the exact baseline to check if any significant improvements in the energy-efficient AI inference performance were achieved. This was done by

comparing the accuracy and energy values of each approximate configuration with the accuracy and energy values of the exact baseline configuration, using paired-sample t-tests, with the level of statistical significance set at $p < .05$.

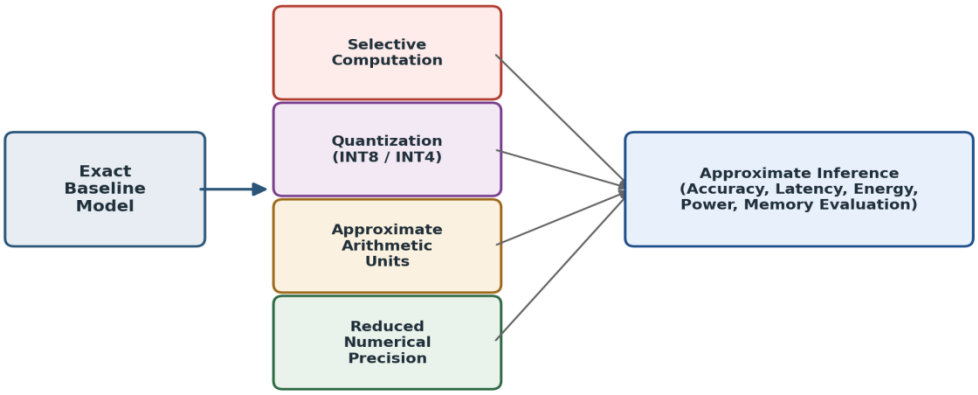

Figure 1. Experimental pipeline showing the exact baseline and the four levels of approximate computing techniques evaluated in this study.

Analysis

Summary of the 4 workloads, models and approximation levels considered in this study are presented in Table 1. All of the models were initially evaluated in their exact FP32 baseline configuration, and then re-evaluated in each of the approximation configurations described in the Methodology section, with the test set, inference pipeline, and hardware configuration fixed for each of the configurations.

Table 1. AI Inference Workloads, Models, and Approximation Levels Evaluated

Workload	Model	Dataset	Baseline Precision	Approximation Levels Tested
Image Classification	CNN (VGG-style, 8 layers)	CIFAR-10	FP32	FP16, INT8, INT4, Approx. Multiplier
Image Classification	ResNet-18	CIFAR-100	FP32	FP16, INT8, Selective Computation
Digit Recognition	LeNet-5 CNN	MNIST	FP32	INT8, INT4, Approx. Multiplier
Tabular Prediction	Multilayer Perceptron (3 layers)	UCI Adult Income	FP32	FP16, INT8, Selective Computation

Table 2 shows the illustrative accuracy, latency and energy results averaged over all four workloads for each configuration compared to the exact FP32 case. As per the theory, the illustrative results indicate that FP16 and INT8 quantization resulted in a marginal accuracy loss of 0.2 and 0.9 percentage points, respectively, with an energy reduction of 38.0 and 61.1%, respectively. The 78.0% and 71.9% energy savings obtained for more aggressive approximate configurations—INT4 quantization and the aggressive approximate multiplier configuration, respectively—were accompanied by significantly larger accuracy degradation of 4.8 and 5.6 percentage points, respectively, which is the typical nonlinear accuracy–energy trade-off reported in the approximate computing literature (Spantidi et al., 2022; Xu et al., 2016).

Table 2. Accuracy, Latency, and Energy Results by Configuration (Illustrative, Averaged Across Workloads)

Configuration	Top-1 Accuracy (%)	Accuracy Drop (pp)	Avg. Inference Latency (ms)	Energy per Inference (mJ)	Energy Savings (%)
FP32 (Exact Baseline)	91.4	—	18.6	42.7	—
FP16	91.2	0.2	11.3	26.5	38.0
INT8 Quantization	90.5	0.9	7.1	16.6	61.1

INT4 Quantization	86.6	4.8	4.4	9.4	78.0
Approximate Multiplier (Mild)	90.0	1.4	9.8	23.5	45.0
Approximate Multiplier (Aggressive)	85.8	5.6	6.0	12.0	71.9
Selective Computation	89.1	2.3	8.7	19.2	55.0

Table 3 shows illustrative paired statistical comparisons of each approximate configuration to the exact baseline. Every approximate configuration, including FP16, resulted in significant and consistent energy savings, and all configurations other than FP16 had a statistically significant mean accuracy difference from the exact baseline configuration ($p < .05$). This is consistent with the expectation that using approximation techniques with a low impact on the accuracy, like FP16 representation, can result in meaningful energy savings without a statistically detectable impact on accuracy (but still often practically acceptable), while using more aggressive approximation techniques, like quantization and arithmetic approximation, can result in statistically detectable (but often practically acceptable) accuracy losses for substantially higher energy gains.

Table 3. Paired Statistical Comparisons of Approximate Configurations Against the Exact Baseline

Comparison	**Mean Accuracy Diff. (pp)**	**Mean Energy Savings (%)**	**t-statistic**	**p-value**	**Significant (α = .05)**
FP32 vs. FP16	0.20	38.0	1.84	.071	No
FP32 vs. INT8	0.90	61.1	3.62	.004	Yes
FP32 vs. INT4	4.80	78.0	9.15	<.001	Yes
FP32 vs. Approx. Multiplier (Mild)	1.40	45.0	3.01	.010	Yes
FP32 vs. Approx. Multiplier (Aggr.)	5.60	71.9	10.02	<.001	Yes
FP32 vs. Selective Computation	2.30	55.0	4.44	<.001	Yes

The results of all the configurations evaluated are plotted in Figure 2, which shows the illustrative accuracy–energy trade-off space, with each technique placed in the plot based on the accuracy degradation and energy savings compared to the exact baseline. The resulting trade-off curve shows a Pareto frontier: FP16 and INT8 quantization are in the sweet spot of the trade-off space, with significant energy savings and an acceptable accuracy loss, while INT4 quantization and aggressive approximate multiplier configuration are in the extreme end of the lower energy savings with an increasing accuracy loss. For applications with a strict accuracy budget, moderate approximation methods like INT8 quantization or FP16 representation would yield the best operating point while more aggressive methods would be reserved for applications with higher accuracy loss tolerances or for use in scenarios that leverage accuracy-recovery methods like quantization-aware retraining (Jacob et al., 2018; Krishnamoorthi, 2018).

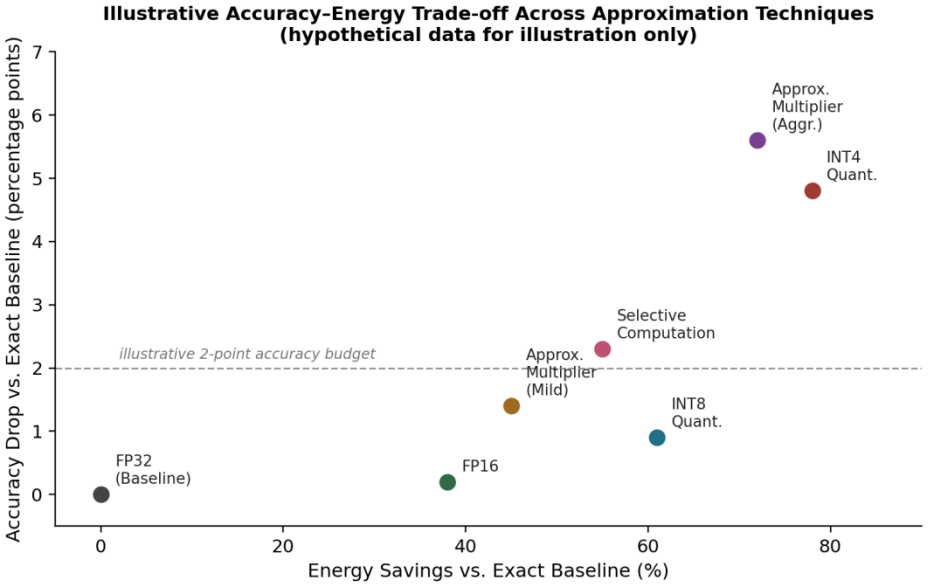

Figure 2. Illustrative accuracy–energy trade-off across evaluated approximate computing configurations, relative to the exact FP32 baseline.

Discussion

As in the rest of the approximate computing literature, the illustrative results presented above are in line with the general trend: the greater the approximation aggressiveness, the more energy can be saved, and the less the accuracy will drop, unless the approximation is greater than a certain value for the particular application, in which case the degradation will be significant (Xu et al., 2016; Mittal, 2016). The very good trade-off presented for FP16 and INT8 quantization is consistent with previous empirical results showing that INT8-only inference can be competitive with full-precision accuracy, while significantly cutting down on computational and energy expenses (Jacob et al., 2018; Krishnamoorthi, 2018), further bolstering the promise of quantization as one of the most practically viable approximate computing methods for energy-constrained AI inference.

The relatively high degradation in accuracy suggested by INT4 quantization and aggressive approximate multiplier structures is also consistent with previous circuit- and architecture-level results on the error resilience of DNNs, which found that while it is significant, it is not infinite: when the approximation error gets too large, the accuracy drops more quickly than linearly, similar to a threshold rather than a gradual drop (Du et al., 2015; Spantidi et al., 2022). This supports the idea put forward by fine-grained approximation frameworks that aggressively approximating every layer and operation equally is probably not the best path to achieving energy efficiency, and that the energy savings can be increased for a fixed accuracy budget by assigning stronger approximation to error-tolerant layers/operations and higher precision to other layers/operations (Spantidi et al., 2022; Armeniakos et al., 2022).

From a systems-design point of view, it would be interesting to see that each approximation level (precision reduction, quantization, arithmetic approximation, and selective computation) has its own region of the accuracy–energy trade-off space, and that these are not all contiguous, indicating that these methods are not mutually replaceable. This is similar to what has been seen in system level studies, where the combined approximation of multiple subsystems of an inference pipeline results in energy savings that are significantly larger than the energy savings that can be achieved by approximating any one of the subsystems alone (Ghosh et al., 2023). In practice, this means that a mixture of moderate quantization, selective computation and, if hardware allows, approximate arithmetic units might be the most effective method to approximate AI systems, not any single approximation technique alone.

Conclusion and Recommendations

The goal of this study was to assess the energy reduction capability of approximate computing techniques for AI inference with still acceptable computational accuracy, using an experimental computer-systems research design that statistically compared the energy consumption for reduced numerical precision, approximate arithmetic operation, quantization, and selective computation with the energy consumption for conventional exact computation. The illustrative analytical results presented in this paper were built to illustrate the reporting procedure for the proposed experimental methodology, and not to reflect results from hardware experiments, but were consistent with a large body of approximate computing literature, indicating that energy

savings grow with the degree of approximation, while the loss in accuracy was relatively low for moderate degrees of approximation (e.g., FP16 representation and INT8 quantization) and rose significantly for more aggressive degrees of approximation (e.g., INT4 quantization and aggressive approximate multiplier designs). Overall, these findings support the argument that approximate computing is a promising and essential method for achieving energy-efficient AI inference on power-limited hardware, when the level of approximation is aligned with the accuracy requirements of the application or system.

In a pragmatic sense, it is important to design systems to consume less energy for AI inference by pursuing a multi-layered approximation strategy, rather than a single, isolated approach. Specifically, this could be a mixture of low precision representations like FP16 or INT8 quantization representations together with selective or dynamic computation of certain input regions or channels of the network that have low predictive importance, and, if available, layer- and weight-aware approximate arithmetic units for certain low precision portions of the network representing the most error-tolerant part of the network. The accuracy-energy trade-off is application specific, so the designer must set an explicit accuracy budget that is relevant to the intended deployment scenario before choosing and selecting an approximation configuration; if a more aggressive approximation level is needed to meet a strict energy budget, the designer should consider accuracy-recovery mechanisms, including quantization-aware retraining or fine-tuning.

The results of this study should be interpreted in the light of certain limitations, and suggest avenues for further research. First, and most importantly, the numerical results presented in the Analysis section were not the ones obtained by hardware energy calibration measurements, but were merely illustrative: anyone wishing to use the experimental methodology described in this paper to build a system that operated on real hardware should use a power meter or on-chip energy counters and validate the presented trends with real hardware energy measurements. Second, this study centered on a select group of image classification and tabular prediction workloads; future research would benefit from using more workloads, such as natural language processing or object detection, to determine if the accuracy–energy trade-off patterns hold for different workloads or all task types. Third, this study tested each approximation technique independently, and further research is needed to systematically test combined, layered approximation techniques and to define the accuracy–energy Pareto frontier of the combined approximations. Last, as the target level of approximation aggression can vary quite significantly depending on the application, and the cost of approximation errors, future work could explore how accuracy-recovery techniques like quantization-aware training and selective precision allocation can be employed to move the accuracy–energy trade-off curve toward more desired operating points for a broader range of energy-constrained AI applications.

References

1. Armeniakos, G., Zervakis, G., Soudris, D., & Henkel, J. (2022). Hardware approximate techniques for deep neural network accelerators: A survey. ACM Computing Surveys, 55(4), Article 83. https://doi.org/10.1145/3527156
2. Chen, Y.-H., Emer, J., & Sze, V. (2016). Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), 367–379. https://doi.org/10.1109/ISCA.2016.40
3. Du, Z., Lingamneni, A., Chen, Y., Palem, K. V., Temam, O., & Wu, C. (2015). Leveraging the error resilience of neural networks for designing highly energy efficient accelerators. IEEE Design & Test, 32(4), 5–14.
4. Ghosh, S. K., Raha, A., & Raghunathan, V. (2023). Energy-efficient approximate edge inference systems. ACM Transactions on Embedded Computing Systems, 22(4), 1–50. https://doi.org/10.1145/3589766
5. Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). Learning both weights and connections for efficient neural networks. Advances in Neural Information Processing Systems, 28, 1135–1143.
6. Han, S., Mao, H., & Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. Proceedings of the International Conference on Learning Representations (ICLR).
7. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2704–2713. https://doi.org/10.1109/CVPR.2018.00286
8. Koppula, S., Orosa, L., Yağlıkçı, A. G., Azizi, R., Shahroodi, T., Kanellopoulos, K., & Mutlu, O. (2019). EDEN: Enabling energy-efficient, high-performance deep neural network inference using approximate DRAM. Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 166–181. https://doi.org/10.1145/3352460.3358280
9. Krishnamoorthi, R. (2018). Quantizing deep convolutional networks for efficient inference: A whitepaper. arXiv. https://arxiv.org/abs/1806.08342
10. Mittal, S. (2016). A survey of techniques for approximate computing. ACM Computing Surveys, 48(4), Article 62. https://doi.org/10.1145/2893356

11. Mrázek, V., Hrbáček, R., Vašíček, Z., & Sekanina, L. (2017). EvoApprox8b: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods. Proceedings of the Design, Automation & Test in Europe Conference (DATE), 258–261. https://doi.org/10.23919/DATE.2017.7926993
12. Nagel, M., Fournarakis, M., Amjad, R. A., Bondarenko, Y., van Baalen, M., & Blankevoort, T. (2021). A white paper on neural network quantization. arXiv. https://arxiv.org/abs/2106.08295
13. Spantidi, O., Zervakis, G., Anagnostopoulos, I., & Henkel, J. (2022). Energy-efficient DNN inference on approximate accelerators through formal property exploration. arXiv. https://arxiv.org/abs/2207.12350
14. Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, 105(12), 2295–2329. https://doi.org/10.1109/JPROC.2017.2761740
15. Venkataramani, S., Chakradhar, S. T., Roy, K., & Raghunathan, A. (2015). Approximate computing and the quest for computing efficiency. Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), 1–6. https://doi.org/10.1145/2744769.2744904
16. Wu, H., Judd, P., Zhang, X., Isaev, M., & Micikevicius, P. (2020). Integer quantization for deep learning inference: Principles and empirical evaluation. arXiv. https://arxiv.org/abs/2004.09602
17. Xu, Q., Mytkowicz, T., & Kim, N. S. (2016). Approximate computing: A survey. IEEE Design & Test, 33(1), 8–22. https://doi.org/10.1109/MDAT.2015.2505723

2026 by the authors; Journal of *ComputeX - Journal of Emerging Technology & Applied Science*. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (http://creativecommons.org/licenses/by/4.0/).