How Much Quality Loss is Acceptable? Evaluating Semantic Acceptability and Energy Gains of Approximate Memory in a RISC-V Environment

Guilherme Saides Serbai[1], **Isaías Felzmann**[2], **João Fabrício Filho**[1]

[1] Federal University of Technology – Paraná – Campo Mourão, Brazil

[2]Institute of Computing, University of Campinas, Brazil

gsserbai@gmail.com, ibf@unicamp.br, joaof@utfpr.edu.br

Abstract. *Approximate memories reduce energy consumption through non-reliable operation, but the resulting faults may compromise the usefulness of application outputs. This non-reliable operation may be tolerated up to a point that depends on application context and data types. However, uncontrolled errors may lead to execution crashes or unacceptable results, which reduce the energy gains. Defining whether a result is acceptable or not depends on the context of a complete application pipeline. This work proposes AxCept-Bench, a systematic interface for characterizing operating regions for different applications considering execution stability and output validation based on semantic acceptability. AxCept-Bench integrates the approximation source, the application pipeline and an acceptability validation of the results. Our evaluation results encompass a simulation based on memory approximation that defines acceptability datasets for the applications JPEG encoding and FFT. The JPEG application achieves energy savings of 37.01% with 90.3% more acceptable results on the pipeline than with simple error metrics, while FFT accepts 44.18% more results with 57.38% more energy savings. Therefore, energy gains must be interpreted jointly with execution validity and the semantic requirements of each application to potentialize the results with approximation.*

1. Introduction

The growing demand for data-intensive computing is increasingly constrained by energy and thermal limits. This challenge is especially relevant in image processing, signal processing, and machine learning, where large volumes of data must be moved and processed efficiently. At the same time, many applications in these domains tolerate controlled imprecision without losing the usefulness of their final outputs [Mittal 2016, Baek and Chilimbi 2010, Khudia et al. 2015]. Approximate Computing exploits this tolerance by relaxing exactness requirements in selected parts of a system. Such relaxation may reduce energy consumption or computational effort, provided that the introduced errors remain compatible with application requirements [Baek and Chilimbi 2010, Fabrício Filho et al. 2022]. The main challenge is therefore to apply approximation and determine how far it can be increased while preserving

This work was supported by Fundação Araucária, the National Council for Scientific and Technological Development (CNPq), the Coordination for the Improvement of Higher Education Personnel (CAPES) - Finance Code 001, and FAEPEX-UNICAMP grants #3770/25 and #2622/25.

useful outputs. The central challenge is not merely determining whether an application can tolerate approximation, but identifying an operating point that provides efficiency gains while preserving outputs that remain useful for their intended purpose. This operating point is application-dependent: the same amount of degradation may be acceptable in one processing pipeline and unacceptable in another. In this work, we use *acceptability* to describe whether an approximate output preserves the information required by the downstream application, independently of how far that output deviates from an exact reference according to a generic quality metric.

Low-voltage memory operation directly exposes this trade-off. Reducing the supply voltage below a reliable operating point may introduce memory errors, whose frequency increases as voltage decreases [Chang et al. 2017]. Memory faults may introduce errors into application data or control state. Depending on how these errors propagate through the application, they may lead to execution failures, malformed outputs, or degraded outputs that nevertheless remain valid and useful. Traditional metrics such as Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR), and Mean Absolute Percentage Error (MAPE) characterize structural or numerical deviation from an exact reference, but they do not directly determine whether the information required by the application remains preserved. An output may exhibit substantial degradation while still supporting its intended task, making fixed quality thresholds potentially overconservative. In this work, a *safe operating region* denotes a range of approximation levels that reduces DRAM energy while preserving execution stability, output validity, and the application-specific semantic information required by the workload. However, existing approaches control approximation through generic quality thresholds that do not capture the semantic requirements of the application [Yarmand et al. 2021, Khudia et al. 2015, Fabrício Filho et al. 2022], lacking an integrated characterization of semantic acceptability jointly with execution stability and energy.

To address this gap, this work presents **AxCept-Bench**[1], an end-to-end experimental environment that extends our previous work on proposing an acceptability metric [Felzmann et al. 2021a] with a characterization of low-voltage approximate-memory configurations in RISC-V systems. AxCept-Bench integrates AxPIKE simulator [Felzmann et al. 2021b] and AxRAM memory interface [Fabrício Filho et al. 2019] with workload-specific output validation, structural or numerical quality analysis, semantic evaluation, and trace-based DRAM energy estimation using Ramulator [Kim et al. 2016] and DRAMPower [Chandrasekar et al. nd]. Rather than imposing a generic quality threshold, the environment evaluates each workload according to the meaning and intended use of its outputs. The environment is evaluated using two complementary workloads under memory error rates ranging from `1e-5` to `1e-1`. The first evaluation performs JPEG encoding over 2,100 images from the UC Merced Land Use dataset [Yang and Newsam 2010]. JPEG outputs are evaluated through execution stability, output recovery, RGB-SSIM, and classification using ResNet-50 [He et al. 2016]. The second performs FFT-based dominant-frequency extraction over 2,526 musical-instrument recordings and evaluates output usability, numerical deviation through MAPE, and semantic acceptability through MIDI-note equivalence. For both workloads, DRAM energy is estimated from memory-command traces and compared with exact execution.

[1] `https://github.com/gserbai/AxCept-Bench`

The two workloads exhibit distinct approximation tolerances. For JPEG, at an error rate of `1e-2`, 96.52% of the outputs remain recoverable while DRAM energy is reduced by 37.01%. Even at `1e-1`, 82.95% of the JPEG outputs remain recoverable, and a ResNet-50 trained and validated using valid outputs from the same approximation regime achieves 90.3% classification accuracy, despite a mean RGB-SSIM of 0.1679. For the FFT workload, the MIDI-note match rate decreases from 44.18% at `1e-5` to 9.50% at `1e-3` and 0.04% at `1e-1`. The corresponding DRAM energy reduction increases from 14.81% at `1e-5` to 57.38% at `1e-1`. These results show that an operating point cannot be selected from energy reduction or numerical degradation alone, but must also account for execution stability, output validity, and application-specific acceptability.

The main contributions are: (i) **AxCept-Bench**, an extensible environment integrating approximate RISC-V execution, application-specific output evaluation, and trace-based DRAM energy estimation; (ii) a methodology that jointly considers execution stability, output validity, structural or numerical quality, semantic acceptability, and energy; and (iii) the evaluation of JPEG encoding and FFT-based dominant-frequency extraction using workload-specific acceptance criteria.

2. Background: Approximation Opportunities

Approximate Computing explores opportunities to reduce computational effort or energy consumption by relaxing accuracy requirements in parts of a system. Approximation can be introduced at different abstraction levels and through different mechanisms. In configurable approaches, the approximation mechanism exposes one or more control knobs whose settings define different approximation levels. For example, a memory system may use an operating parameter such as supply voltage as an approximation knob, with different settings producing different energy–error trade-offs [Mittal 2016, Fabrício Filho et al. 2022]. Therefore, configurable approximation does not define a single approximate behavior, but a set of operating points from which a configuration must be selected. Selecting an approximation level requires defining how much degradation the application can tolerate. A more aggressive configuration may provide larger efficiency gains while also increasing the probability or magnitude of errors affecting the computation. Consequently, the appropriate operating point depends on a criterion that determines whether the resulting output is still acceptable. Such a criterion is application-dependent: different applications manipulate different data types and require different information to be preserved in their outputs [Baek and Chilimbi 2010, Fabrício Filho et al. 2022].

Acceptability, however, cannot always be determined from the output of the approximated component in isolation. An approximate result may be an intermediate value consumed by subsequent processing stages, and degradation in this value is relevant only insofar as it changes the outcome required by the complete application scenario. Previous work has therefore proposed evaluating approximate outputs within their processing pipeline, considering a result acceptable when the approximation does not change the relevant outcome of the complete scenario [Felzmann et al. 2021a]. Defining this pipeline establishes the context in which acceptability is evaluated and, consequently, the criterion used to instrument an approximation experiment and identify suitable operating points.

3. Related Work

Architectural simulation and approximate-memory interfaces provide mechanisms for evaluating how hardware approximation propagates through complete applications. AxPIKE is a RISC-V-based simulation environment that allows approximation models to be injected into instructions and data accesses while collecting detailed execution statistics and memory traces for external energy evaluation [Felzmann et al. 2021b]. AxRAM provides a resilient approximate-memory interface that separates approximate and protected memory regions and reduces execution failures caused by errors affecting critical application data [Fabrício Filho et al. 2019]. Together, these works provide mechanisms for exposing applications to controlled memory approximation and observing its effects on execution, output quality, and energy. Other approaches address the problem of controlling approximation and selecting an operating point under output constraints. OPTIMA dynamically adjusts cache approximation levels according to application-specific quality feedback and a predefined minimum output accuracy [Yarmand et al. 2021]. Rumba performs online quality management by detecting large approximation errors with lightweight predictors and selectively recomputing affected outputs on exact hardware [Khudia et al. 2015]. SmartApprox determines approximate-memory configurations from application execution features and knowledge of the error tolerance of previously characterized applications, avoiding application-specific quality evaluation for the target application at runtime [Fabrício Filho et al. 2022]. These approaches demonstrate different mechanisms for controlling approximation according to error or quality requirements.

Quality requirements, however, do not necessarily express whether an approximate result remains useful in its application context. Previous work distinguishes output quality from acceptability by evaluating approximate data as part of a complete processing scenario [Felzmann et al. 2021a]. Their results show that fixed thresholds based on metrics such as SSIM, PSNR, or numerical error may reject useful outputs or accept outputs that do not preserve the information required by subsequent processing. This application-oriented view was also explored in the context of neural-network training and validation, where the acceptability of approximated images was shown to depend on the relationship between the approximation conditions used during model training and those present during validation [Serbai et al. 2025]. AxCept-Bench builds on this notion of application-level acceptability by integrating controlled approximate-memory execution, workload-specific processing pipelines, semantic output validation, execution stability, and DRAM energy estimation to characterize operating regions for different applications.

4. AxCept-Bench Overview

AxCept-Bench is an extensible experimental environment that combines AxPIKE and the AxRAM approximate-memory interface to evaluate complete RISC-V applications under controlled memory faults. Figure 1 summarizes its workflow. Application outputs are processed through workload-specific validation, structural or numerical quality analysis, and semantic evaluation, while memory traces are forwarded to Ramulator and DRAMPower for DRAM energy estimation [Chandrasekar et al. nd, Kim et al. 2016]. These measurements are jointly used to characterize safe approximation regions.

AxRAM partitions the memory space into regions that operate either exactly at the nominal voltage V_{in} or approximately at a reduced voltage

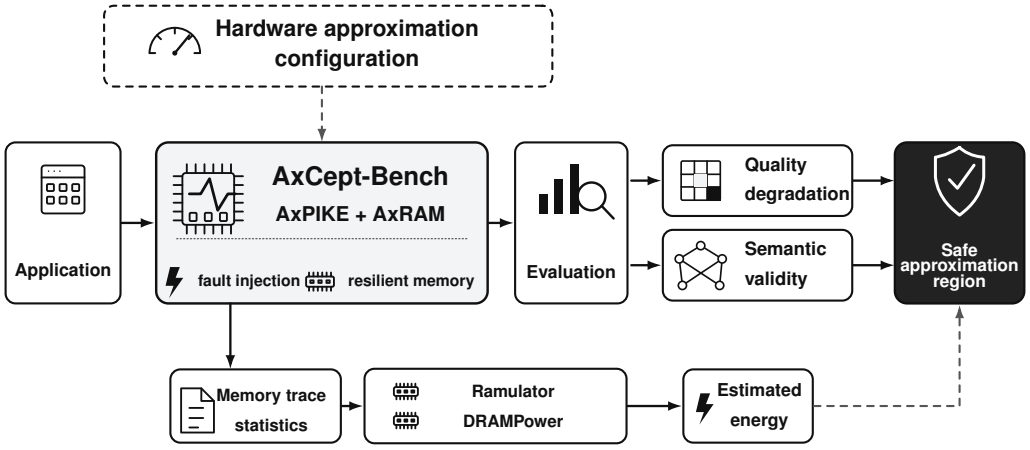


Figure 1. Overview of the AxCept-Bench evaluation pipeline.

$V_s < V_{in}$ [Fabrício Filho et al. 2019, Felzmann et al. 2021b]. The `$reg_x` register selects the regions exposed to approximation, whereas `$reg_a` controls the fault probability. In the evaluated configuration, `$reg_x` remains fixed to protect the instruction and stack regions, while the approximation intensity is varied through `$reg_a`. All cache levels remain exact, and faults affect only vulnerable memory reads that miss the L2 cache and reach DRAM. The application code remains unchanged. The current environment includes JPEG encoding and FFT-based dominant-frequency extraction, and additional workloads can be incorporated by defining their execution flow, output-validation procedure, quality metrics, and application-specific acceptability criteria.

5. Materials and Methods

Both workloads are executed on AxPIKE under configured memory error rates of `1e-5`, `1e-4`, `1e-3`, `1e-2`, and `1e-1`, controlled through the `$reg_a` register. In the simulation model, the configured error rate determines the probability of fault injection during each vulnerable DRAM read. For example, `1e-1` corresponds to a probability of 0.1 per read, or approximately one fault every ten vulnerable DRAM reads on average. The `$reg_x` configuration remains fixed throughout the experiments to keep instruction and stack regions protected. The application code is not modified to tolerate faults; approximation effects arise exclusively from probabilistic faults introduced during vulnerable memory reads.

The memory hierarchy is kept fixed throughout the experiments and is configured in AxPIKE as *sets:ways:block size*: a `128:8:32` L1 data cache (32 KB), a `256:4:32` L1 instruction cache (32 KB), and a `1024:4:32` unified L2 cache (128 KB), all with 32-byte blocks. All cache levels remain exact. In the evaluated configuration, faults affect only vulnerable memory reads that miss the L2 cache and reach DRAM. Executions are classified as normal, data crash, flow crash, or timeout. A data crash results from an invalid data-dependent memory access, whereas a flow crash indicates control-flow corruption. Outputs produced before termination are evaluated whenever they preserve the structure required by the corresponding workload.

The evaluated fault probabilities are associated with representative DDR3L supply-voltage operating points derived from the reduced-voltage characterization reported by Chang et al. [Chang et al. 2017]. The selected values follow the error-growth trend observed for Vendor B in their experimental results and are summarized in Table 1.

This mapping provides an empirically motivated association between fault-probability orders of magnitude and voltage operating points. It should not be interpreted as an exact equivalence, since Chang et al. measure the fraction of DRAM cache lines containing errors, whereas AxPIKE injects faults probabilistically into vulnerable memory reads.

Table 1. Representative mapping of fault probabilities to DDR3L supply voltages.

Fault probability	Exact	`1e-5`	`1e-4`	`1e-3`	`1e-2`	`1e-1`
V_{DD} **(V)**	1.350	1.150	1.125	1.100	1.075	1.050

5.1. JPEG Encoding Workload

The JPEG workload uses the TooJpeg encoder [Brumme 2019] with a quality factor of 100%. It processes the UC Merced Land Use dataset [Yang and Newsam 2010], which contains 2,100 TIF images distributed across 21 land-use classes. Each class contains 100 images with a resolution of 256×256 pixels. The dataset is divided into 1,680 training images and 420 validation images, corresponding to 80 and 20 images per class, respectively. During encoding, part of the JPEG stream may be written before an execution failure. When AxPIKE detects a crash, its register dump may be appended to the same output file. The recovery pipeline therefore locates the AxPIKE crash signature and discards all bytes from that position onward, preserving only the JPEG data produced before the failure.

A candidate output is retained when it contains the JPEG start marker `FFD8`, at least one JFIF, EXIF, or DQT marker, and a payload of at least 1,024 bytes. The 1,024-byte threshold is an empirically selected conservative filter used to reject fragments that are too small to contain sufficient JPEG structure for meaningful decoding; it is not a requirement of the JPEG standard. Candidate streams are then decoded using truncated-image support. Empty, missing, malformed, or undecodable outputs are classified as invalid. Only outputs that pass both binary validation and resilient decoding are included in the structural and semantic evaluations. Structural degradation is measured using RGB-SSIM between each valid JPEG output and its original TIF reference. When the decoded dimensions differ from the reference, the approximate image is resized only to enable the comparison.

Semantic acceptability is evaluated using a ResNet-50 classifier [He et al. 2016] initialized with ImageNet-1K weights and fine-tuned separately for each approximation regime. A valid JPEG output is considered semantically acceptable when its predicted land-use class matches its ground-truth class. RGB-SSIM quantifies structural degradation but is not used as an acceptance threshold. Training and validation use only the valid decoded subset available in each approximation regime. Classification accuracy is therefore reported conditionally on valid JPEG outputs, whereas the recoverability rate is computed over the complete dataset.

5.2. Dominant-Frequency Workload

The second workload estimates the dominant frequency of 2,526 musical instrument recordings from the University of Iowa Musical Instrument Samples dataset [Fritts 1997]. Audio signals are represented as raw mono, little-endian, single-precision floating-point samples at 44.1 kHz. Each signal is divided into overlapping frames of 4,096 samples with

a hop length of 512 samples. Signals shorter than one complete frame are zero-padded. A Hann window is applied to each frame, and the frequency spectrum is computed using KISS FFT [Borgerding 2003]. Spectral power is accumulated across frames, and a Harmonic Product Spectrum combining the components at k, $2k$, and $3k$ is used to emphasize the fundamental frequency. Parabolic interpolation then refines the selected spectral peak.

When the application reaches its output stage, it writes a fixed 24-byte little-endian binary record:

```
[8-byte magic] [int32 status] [int32 num_samples]
      [int32 num_frames] [float32 freq_hz].
```

A crash occurring before the record is written produces no usable application result. However, if the complete 24-byte record is written before a subsequent crash, the AxPIKE crash dump may be appended after the record. In this case, the application result remains recoverable because the parser evaluates the first 24 bytes separately from the appended crash signature. An output is considered usable when the complete record can be parsed, the magic identifier is `AXDFREQ1`, the execution status is `AX_OK`, and the estimated frequency is finite and lies within the interval $0 < f \leq 22.05$ kHz. Outputs containing a valid record followed by an AxPIKE crash dump are also considered usable.

Numerical degradation is measured using MAPE between the exact and approximate dominant frequencies. MAPE is computed only for pairs in which both executions produce usable frequencies, and it is not used as an acceptance threshold. Semantic acceptability is determined by mapping each frequency to its nearest MIDI note:

$$m(f) = \left\lfloor 69 + 12 \log_2 \left(\frac{f}{440} \right) + 0.5 \right\rfloor . \tag{1}$$

A usable approximate output is considered semantically acceptable when it maps to the same MIDI note as the exact reference. Missing, malformed, application-error, and numerically invalid approximate outputs are counted as unacceptable when computing the overall MIDI-note match rate.

5.3. DRAM Energy Estimation

For each approximation level, the representative supply voltage listed in Table 1 is used in the DRAM energy evaluation. Memory-access traces generated during execution are processed by Ramulator to obtain DRAM command traces, which are subsequently evaluated using DRAMPower. For each workload and error rate, the DRAM energy reduction relative to exact execution is computed as

$$R_E = 100 \left(1 - \frac{\overline{E}_{\text{approx}}}{\overline{E}_{\text{exact}}} \right) . \tag{2}$$

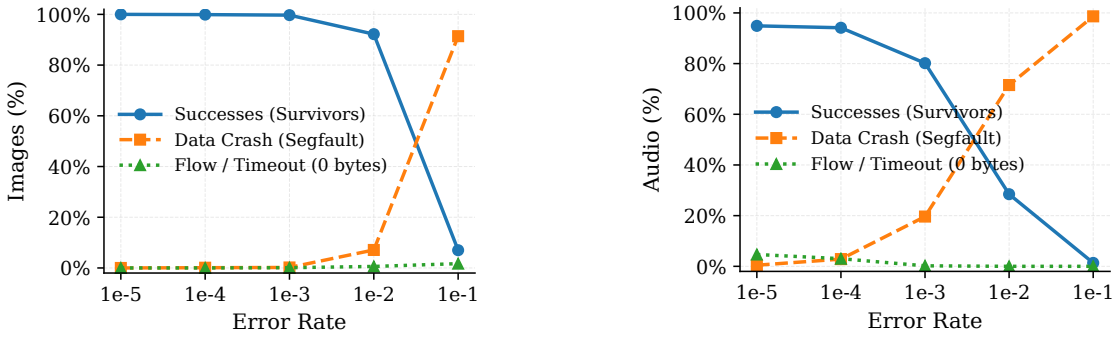
Confidence intervals are estimated using 10,000 independent bootstrap resamples and the percentile method. Independent resampling is used because exact and approximate traces cannot be reliably paired by execution identifier.

6. Experimental Results

6.1. Execution Stability and Output Validity

Figure 2 compares the execution outcomes of the JPEG and dominant-frequency workloads. For JPEG, execution remains highly stable up to `1e-3`, with more than 99% of executions completing normally. At `1e-2`, the success rate decreases to 92.2%, while 7.1% of executions fall into the data-crash category. Under the most aggressive regime, `1e-1`, only 7.0% complete normally and 91.4% are classified as data crashes. Flow-related failures and timeouts remain comparatively rare, peaking at 1.7%.

The dominant-frequency workload presents a more gradual loss of execution stability. Successful executions decrease as the fault probability increases, while data crashes become the dominant outcome from `1e-2` onward. Under `1e-1`, almost no execution produces a normal application record. Flow-related failures and timeouts remain less frequent than data-related failures.



(a) JPEG **(b) FFT**

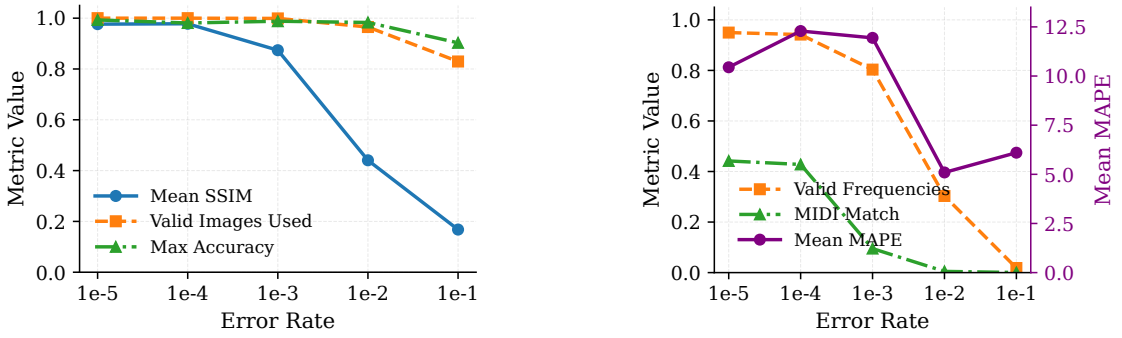
Figure 2. Execution stability under increasing memory error rates.

Execution failure does not necessarily imply that all data written before termination are unusable. Therefore, execution stability is analyzed separately from workload-specific output recoverability and usability.

6.2. Output Recoverability and Quality

For JPEG, bytes produced before an execution failure may still form a recoverable image stream. Segmentation faults interrupt future execution but do not retroactively erase bytes already written. At `1e-2`, 96.52% of the complete dataset passes the binary validation and resilient decoding pipeline. Even under `1e-1`, 82.95% of the JPEG outputs remain recoverable. These rates are substantially higher than the corresponding normal-execution rates, demonstrating that execution survival and output validity are distinct properties. Structural quality nevertheless decreases significantly as approximation becomes more aggressive. Mean RGB-SSIM reaches 0.4406 at `1e-2` and only 0.1679 at `1e-1`, indicating severe pixel-level degradation.

For the dominant-frequency workload, the proportion of usable frequencies also decreases with increasing fault probability. Mean MAPE is calculated only for pairs in which both the exact and approximate executions produce usable frequencies. Consequently, its non-monotonic behavior under aggressive approximation must not be interpreted as an improvement in numerical quality, since the metric is computed over an increasingly small subset of surviving outputs.



(a) JPEG quality and validity.

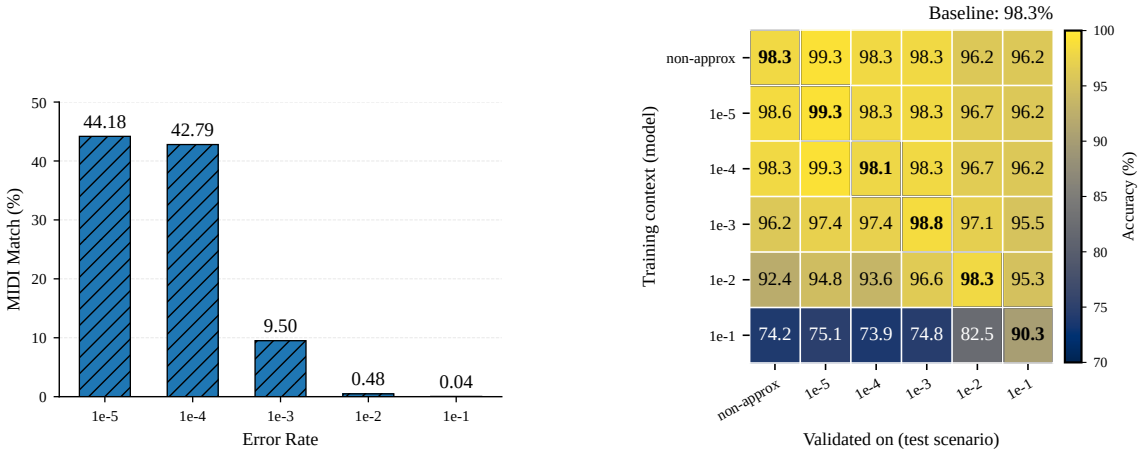
(b) FFT quality and validity.

Figure 3. Output validity and quality under approximate-memory error rates.

Figure 3 summarizes these trends, showing that structural or numerical degradation alone does not determine whether an output remains useful for its intended task.

6.3. Semantic Acceptability

Semantic acceptability is evaluated according to the meaning of each workload output. For the dominant-frequency workload, an approximate output is acceptable when it maps to the same MIDI note as the exact reference. For JPEG, a valid decoded output is acceptable when its predicted land-use class matches its ground-truth class.



(a) FFT MIDI-note match rate.

(b) JPEG train–validation accuracy.

Figure 4. Semantic acceptability under approximate-memory error rates.

Among all 2,526 audio inputs, 44.18% of the outputs match the reference MIDI note at `1e-5`, while 42.79% remain acceptable at `1e-4`. The match rate decreases to 9.50% at `1e-3`, 0.48% at `1e-2`, and 0.04% at `1e-1`. Thus, lighter approximation preserves semantic acceptability for a subset of inputs, whereas aggressive approximation almost completely eliminates MIDI-note equivalence. Unlike MAPE, the overall MIDI-note match rate uses the complete workload as its denominator. Missing, malformed, application-error, and numerically invalid outputs are therefore counted as unacceptable, avoiding a survivor-only interpretation.

For JPEG, the classification matrix in Figure 4b shows that performance depends on the relationship between the approximation regimes used during training and validation. Most importantly, when both training and validation use valid outputs from the `1e-1` regime, ResNet-50 achieves 90.3% accuracy despite the mean RGB-SSIM of only

0.1679. This dissociation between pixel-level similarity and classification accuracy supports the central argument of this work: structural quality metrics alone are insufficient proxies for application-level semantic utility. Classification accuracy is conditional on valid decoded JPEG outputs, whereas recoverability is calculated over all 2,100 input images.

6.4. JPEG Per-Class Sensitivity

Semantic degradation does not occur uniformly across the land-use classes. Figure 5 presents the accuracy of four representative classes on the fixed `1e-1` validation set as the training approximation regime changes.

Although all images have the same spatial resolution, JPEG encoding behavior depends on the visual content of each input. During encoding, each image block is transformed using the Discrete Cosine Transform and then quantized. Visually simpler regions tend to produce a larger number of zero-valued coefficients, enabling a more compact representation. In contrast, regions containing edges, textures, and stronger high-frequency content tend to retain a larger number of non-zero coefficients after quantization. These coefficients must be processed during the subsequent reordering and entropy-coding stages, potentially producing longer execution paths and a larger number of memory operations. Consequently, images containing the same number of pixels may present different levels of exposure to approximate-memory faults. This content-dependent behavior helps explain why approximation tolerance is not uniform across image classes.

When training and validation are aligned under the `1e-1` regime, *mediumresidential* and *sparseresidential* achieve 68.8% and 70.6% accuracy, respectively, whereas *baseballdiamond* and *denseresidential* both reach 87.5%. These results show that overall classification accuracy may conceal substantial differences in the sensitivity of individual classes to approximate-memory faults.

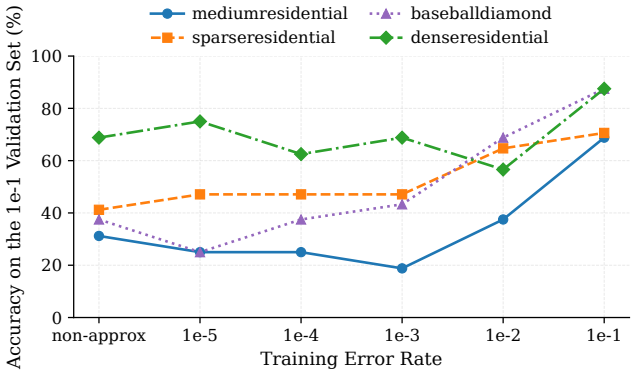


Figure 5. Per-class accuracy on the `1e-1` validation set across training regimes.

6.5. DRAM Energy Reduction

Figure 6 presents the mean DRAM energy reduction relative to the exact-memory baseline. For JPEG, the measured reductions are 31.06%, 32.48%, 35.38%, and 37.01% from `1e-5` to `1e-2`, respectively. Under `1e-1`, the reduction decreases to 25.24%. Among the evaluated points, `1e-2` provides the strongest observed trade-off, combining a 37.01% reduction in DRAM energy with 96.52% output recoverability. For the dominant-frequency workload, energy reduction increases from 14.81% at `1e-5` to 20.21% at

1e−2, reaching 57.38% under 1e−1. However, this largest reduction coincides with an almost complete loss of execution survival and MIDI-note acceptability. Therefore, this point cannot be interpreted as a safe operating configuration.

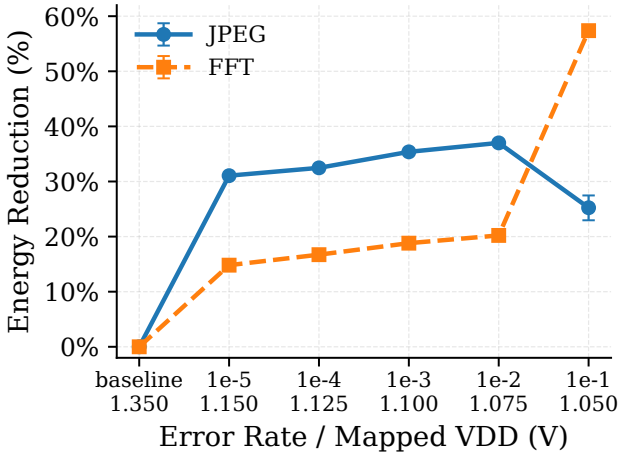


Figure 6. Mean DRAM energy reduction for JPEG and FFT. Error bars show 95% confidence intervals from 10,000 independent bootstrap resamples.

Energy reduction is not strictly monotonic because total trace energy depends not only on the representative supply voltage, but also on the execution path and on the number and distribution of DRAM commands generated during each run. Different failure modes may affect these factors in opposite ways. A data crash may terminate the application prematurely, shortening the memory-access trace and producing an apparently large energy reduction simply because less work was completed. For the execution-stability analysis, the *Flow/Timeout* category operationally groups executions that did not produce an application output or generated zero-byte files. In these cases, it is not possible to reliably distinguish control-flow corruption from an execution that remained active until the timeout limit. Therefore, the values reported for this category should not be interpreted as a complete count of all possible control-flow corruptions. Control-flow corruption may also cause the application to continue executing along unintended paths, repeat operations, or run beyond its expected termination point without immediately producing a zero-byte file. Such cases are not necessarily counted in the *Flow/Timeout* category, but their effects remain represented in the traces used for the energy analysis. Consequently, these executions may generate longer memory-access traces and a larger number of DRAM commands, potentially consuming more total energy than expected even when the memory operates at a reduced supply voltage. Thus, the measured energy reflects both voltage reduction and fault-induced changes in application behavior. A lower operating voltage does not necessarily result in lower total energy when control-flow corruption causes the application to perform additional or unintended work.

These results demonstrate that DRAM energy reduction must be interpreted jointly with execution stability, output validity, and the application-specific semantic acceptability criterion. Energy reductions resulting from premature termination or corrupted execution paths do not represent useful efficiency gains when the application fails to produce an acceptable output.

7. Discussion

The results show that safe approximation regions cannot be defined solely through structural or numerical metrics. For JPEG, the decrease in RGB-SSIM does not directly correspond to a loss of semantic utility. At `1e-1`, the mean RGB-SSIM is 0.1679, yet 82.95% of the outputs remain recoverable, and ResNet-50 reaches 90.3% accuracy when training and validation use valid outputs produced under the same approximation regime. The 90.3% accuracy must be interpreted strictly within this context. It is calculated over the subset of outputs that were successfully recovered and decoded, and it does not represent the performance of a model trained only on exact data. Nevertheless, the result demonstrates that JPEG can preserve high semantic acceptability even under aggressive approximation, provided that downstream processing is adapted to the corresponding error distribution. Among the evaluated points, `1e-2` provides the best observed trade-off for JPEG, combining 96.52% recoverability with a 37.01% reduction in DRAM energy. At `1e-1`, semantic information can still be preserved, but the high execution instability makes this configuration dependent on fault-tolerant recovery and approximation-aware training. JPEG tolerance also depends on the input content. Images containing more edges, textures, and high-frequency components may retain more non-zero coefficients after DCT and quantization, increasing the encoding workload and the exposure to memory faults. Therefore, the impact of approximation is not uniform across different images.

The FFT workload presents a more restricted acceptable region. The MIDI-note match rate decreases from 44.18% at `1e-5` to 9.50% at `1e-3`, becoming nearly zero under the more aggressive regimes. However, error rates below those evaluated, such as `1e-6` or `1e-8`, may preserve higher semantic acceptability while still providing some energy benefit. This contrast confirms that approximation tolerance is application-specific.

The energy results must also be interpreted together with execution behavior. Premature termination may reduce energy because less work is performed, whereas control-flow corruption may prolong execution and generate additional DRAM commands. Therefore, an energy reduction represents a useful gain only when the application also produces a valid and semantically acceptable output.

8. Conclusion

This work presented AxCept-Bench for evaluating low-voltage approximate memory in RISC-V systems. For JPEG, `1e-2` provided the best observed trade-off, with 37.01% DRAM energy reduction and 96.52% output recoverability. Even at `1e-1`, 82.95% of outputs remained recoverable and ResNet-50 reached 90.3% accuracy when training and validation used valid outputs from the same approximation regime. FFT showed lower semantic tolerance, with MIDI-note match decreasing from 44.18% at `1e-5` to 0.04% at `1e-1`, despite 57.38% energy reduction. These results show that safe approximation regions require joint consideration of execution stability, output validity, semantic acceptability, and energy. Future work includes additional applications and memory technologies, input-dependent approximation policies, and validation on physical low-voltage memory platforms.

References

Baek, W. and Chilimbi, T. M. (2010). Green: a framework for supporting energy-conscious programming using controlled approximation. *SIGPLAN Not.*, 45(6):198–209.

Borgerding, M. (2003). KISS FFT. `https://github.com/mborgerding/kissfft`. Software library, accessed July 2026.

Brumme, S. (2019). Toojpeg: A jpeg encoder in a single c++ file. Available at: https://github.com/stbrumme/toojpeg. Accessed: Jun. 2026.

Chandrasekar, K., Weis, C., Li, Y., Goossens, S., Jung, M., Naji, O., Akesson, B., Wehn, N., and Goossens, K. (n.d.). DRAMPower: Open-source DRAM power and energy estimation tool. `https://www.drampower.info`. Accessed: July 2026.

Chang, K. K., Yağlıkçı, A. G., Ghose, S., Agrawal, A., Chatterjee, N., Kashyap, A., Lee, D., O'Connor, M., Hassan, H., and Mutlu, O. (2017). Understanding reduced-voltage operation in modern dram devices: Experimental characterization, analysis, and mechanisms. *Proc. ACM Meas. Anal. Comput. Syst.*, 1(1).

Fabrício Filho, J., Felzmann, I., and Wanner, L. (2022). Smartapprox: Learning-based configuration of approximate memories for energy-efficient execution. *Sustainable Computing: Informatics and Systems*, 34:100701.

Fabrício Filho, J., Felzmann, I. B., Azevedo, R. J., and Wanner, L. F. (2019). A resilient interface for approximate data access. In *2019 IX Brazilian Symposium on Computing Systems Engineering (SBESC)*, pages 1–8.

Felzmann, I., Filho, J. F., de Oliveira, J. R., and Wanner, L. (2021a). Special session: How much quality is enough quality? a case for acceptability in approximate designs. In *2021 IEEE 39th International Conference on Computer Design (ICCD)*, pages 5–8.

Felzmann, I., Filho, J. F., and Wanner, L. (2021b). AxPIKE: Instruction-level injection and evaluation of approximate computing. In *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 491–494.

Fritts, L. (1997). University of iowa musical instrument samples. University of Iowa Electronic Music Studios, `https://theremin.music.uiowa.edu/mis.html`. Accessed: July 2026.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Khudia, D. S., Zamirai, B., Samadi, M., and Mahlke, S. (2015). Rumba: an online quality management system for approximate computing. *SIGARCH Comput. Archit. News*, 43(3S):554–566.

Kim, Y., Yang, W., and Mutlu, O. (2016). Ramulator: A fast and extensible dram simulator. *IEEE Computer Architecture Letters*, 15(1):45–49.

Mittal, S. (2016). A survey of techniques for approximate computing. *ACM Comput. Surv.*, 48(4).

Serbai, G., Gonçalves, R., and Filho, J. F. (2025). Um estudo sobre aceitabilidade de resultados de sistemas aproximados em redes neurais artificiais. In *Anais da XVI Escola Regional de Alto Desempenho de São Paulo*, pages 50–53, Porto Alegre, RS, Brasil. SBC.

Yang, Y. and Newsam, S. (2010). Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 270–279, New York, NY, USA. Association for Computing Machinery.

Yarmand, R., Kamal, M., Afzali-Kusha, A., Esmaeli, P., and Pedram, M. (2021). Optima: An approach for online management of cache approximation levels in approximate processing systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 29(2):434–446.