Rethinking LLM Safety on Edge Devices: Unearthing Hidden Vulnerabilities through Power Stress

Weimin Fu
Kansas State University
weiminf@ksu.edu

Zelin Lu
University of Maryland
zelinlu@umd.edu

Gang Qu
University of Maryland
gangqu@umd.edu

Xiaolong Guo
Kansas State University
guoxiaolong@ksu.edu

Abstract—**The safety of large language models (LLMs) under hardware-induced perturbations remains an underexplored dimension of model reliability. Although LLMs are often treated as stable and deterministic during inference, transient voltage fluctuations, resulting from battery degradation and computational stress on edge devices, can induce faulty behavior, including hallucinations, truncation, and inference failure, even in non-adversarial settings. This study examines the extent to which such power stress reveals model-specific vulnerabilities. Controlled undervolting experiments were conducted on a Raspberry Pi 5 using two instruction-tuned LLMs: google/gemma-3-1b-it and meta-llama/Llama-3.2-1B. Despite operating under identical hardware and load conditions, the models exhibited significantly different responses to voltage degradation. The observed discrepancies suggest that robustness to physical faults is not an inherent characteristic of LLMs, but rather a learned property influenced by training methodology and model design. These findings underscore the need to treat hardware-induced fault tolerance as a core aspect of LLM safety. Evaluating and improving such safety properties is essential for deploying LLMs in energy-constrained and long-lived edge environments.**

Index Terms—**Large Language Models, Edge AI, Robustness Evaluation, Fault Injection**

I. Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding, reasoning, and generation, enabling breakthroughs in tasks such as document summarization, question answering, and code synthesis [1], [2]. While cloud-based infrastructures have traditionally supported the intensive computational demands of LLM inference, growing concerns over data privacy and latency are now driving a shift toward deploying these models on resource-constrained edge devices. These include personal assistants [3], [4], workplace agents [5], and industrial Internet of Things (IoT) systems [6]. On-device inference offers multiple advantages: it enables faster response times without requiring network connectivity [7], improves data confidentiality by eliminating the need for transmission to external servers [8], and supports continual adaptation to user-specific contexts through local learning [9].

However, deploying LLMs in edge environments introduces new reliability challenges. Unlike data centers equipped with stable power supplies and active cooling systems, edge devices are susceptible to environmental stressors, including battery aging, thermal drift, and fluctuations in power delivery. These conditions collectively contribute to voltage instability, particularly during sustained high-load computation. The self-attention mechanism in LLMs exhibits quadratic complexity with respect to sequence length, further intensifying the computational burden on edge CPUs, GPUs, or NPUs, and exacerbating throughput bottlenecks [10], [11]. Prior work has shown that even minor voltage instability can lead to transient computational faults [12]. Although LLMs are generally considered robust under precision reduction and quantization [13], which form the basis of memory-efficient inference techniques, faults introduced by undervolting are fundamentally different from deliberate numerical approximations. The model's resilience to such faults depends on internal properties such as parameter distribution, numerical conditioning, and redundancy. Voltage fluctuations, though typically non-malicious and induced by environmental factors, effectively act as unintended fault injection mechanisms. These perturbations can cause numerical instability during forward propagation, resulting in degraded performance, erroneous outputs, or inference failure. Despite their practical significance, these issues remain underexplored in the context of LLM deployment, as most robustness evaluations emphasize adversarial inputs or synthetic noise perturbations [14]. In this work, we argue that fault resilience in LLMs is not a uniform property across models. Instead, it varies with architectural design choices, training data quality, and optimization objectives. To support this claim, we conduct an empirical study on two publicly available instruction-tuned models, gemma-3-1b-it [15] and Llama-3.2-1B [16].These models are deployed on a resource-constrained edge device, the Raspberry Pi 5, to evaluate their behavior under controlled undervolting conditions. The overall setup is illustrated in Fig. 1, where user inputs are processed locally, and power faults are introduced through voltage control. Under controlled undervolting conditions, we observe that the two models exhibit distinct degradation behaviors, suggesting that

Portions of this work were supported by the National Science Foundation (2340949 and 2419880).

979-8-3315-8924-0/25/$31.00 ©2025 IEEE

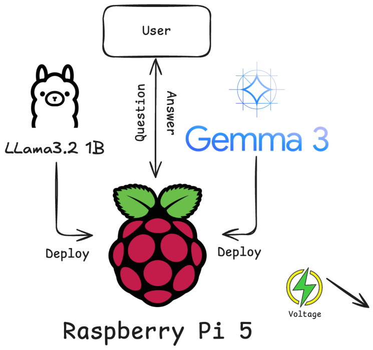

Fig. 1. Evaluation setup showing deployment of Gemma3 1B and LLaMA 3.2 1B on Raspberry Pi 5 under undervolting conditions.

fault tolerance is highly model-specific rather than a universal characteristic of LLMs. Our main contributions are as follows:

- This is the first study to systematically evaluate the reliability and safety of LLM inference under transient undervolting on edge devices, revealing a previously overlooked class of model-specific vulnerabilities caused by non-malicious voltage fluctuations.
- We empirically evaluate two 1B-parameter instruction-tuned LLMs on Raspberry Pi 5 under identical undervolting conditions, revealing distinct fault behaviors and showing that hardware fault robustness depends on model architecture and training.
- We identify two failure modes, gradual degradation and abrupt collapse, and show that complex tasks such as code generation are particularly vulnerable, underscoring the need for hardware-aware LLM robustness evaluation.

II. Background

While LLMs demonstrate strong capabilities in language understanding, reasoning, and generation, their high memory and computational requirements pose major challenges for typical edge devices. A practical trade-off has emerged through compact models with fewer than 3 billion parameters, which sacrifice some performance for improved deployability. Although these models cannot match the capabilities of hundred-billion-parameter LLMs, they can fit entirely in memory on low-power hardware and run locally, slow but enabling offline inference and better data privacy [8].

Recent progress in efficient transformer architectures, mixed-precision inference, and hardware-aware optimization has further facilitated this transition. Toolchains such as GGUF, llama.cpp, and Ollama have demonstrated the feasibility of running quantized LLM variants on consumer-grade CPUs or lightweight GPUs with minimal external dependencies. Despite these advances, most prior research on secure or fault-tolerant inference in edge environments has focused on traditional deep learning models, such as convolutional neural networks and recurrent neural networks [11], which have smaller model sizes and simpler computation graphs. In contrast, LLMs introduce distinct challenges due to their deeper transformer stacks, attention mechanisms with quadratic complexity, and heightened sensitivity to minor numerical perturbations.

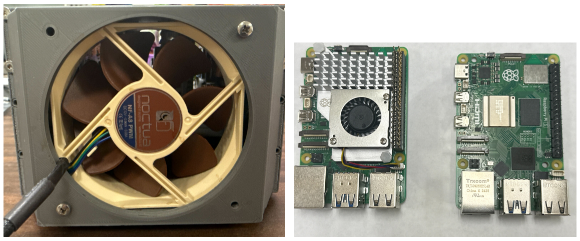

Fig. 2. Hardware setup for voltage fault experiments. Left: custom Noctua active cooler; right: Pi 5 boards with and without active cooling.

Moreover, edge devices often operate under non-ideal hardware conditions, including battery degradation, thermal fluctuation, and unstable power delivery. These factors can cause dynamic voltage variation, particularly during sustained high-load computation, which may in turn introduce silent numerical faults [17] in critical operations such as matrix multiplications, normalization layers, or attention mechanisms. Such faults stem from physical hardware constraints and differ fundamentally from adversarial attacks or quantization-induced approximations; they cannot be effectively modeled as Gaussian noise. Although prior studies have explored adversarial robustness and quantization-aware inference for LLMs [13], little attention has been paid to the vulnerability of instruction-tuned LLMs to transient, hardware-induced faults [12], [18].

This work aims to address this gap by investigating how voltage instability affects the inference robustness of LLMs deployed on edge platforms. Understanding this underexplored dimension of reliability is essential as LLMs become increasingly integrated into safe and long-running edge applications.

III. Methodology

A. System and Deployment Configuration

All experiments were conducted on a Raspberry Pi 5 (model SC1113), powered by a Broadcom BCM2712 system-on-chip featuring a 64-bit quad-core Arm Cortex-A76 CPU clocked at 2.4GHz, with 512KB L2 cache per core and a shared 2MB L3 cache. The system was equipped with 16GB of Micron LPDDR4X-4267 SDRAM. Power was provided by the official Raspberry Pi 27W USB-C power adapter (model SC1153), capable of delivering up to 5.1V and 5A.

To maintain thermal stability during sustained computation, we employed an active cooling unit (model SC1148). Additionally, we designed a custom 3D-printed enclosure housing a Noctua NF-A8 fan for enhanced heat dissipation. Without active cooling, the Raspberry Pi 5 frequently throttled its CPU frequency to approximately 1.5GHz and reached the thermal limit of 85°C under continuous load. The complete experimental hardware setup is shown in Fig. 2.

All models were deployed using PyTorch with full precision on Raspberry Pi OS. While the device was connected to a local network, no external cloud inference was involved; all computations were performed locally.

B. LLM Selection and Token-Level Workload Design

Two instruction-tuned LLMs were selected for evaluation: gemma-3-1b-it and Llama-3.2-1B. These models represent contrasting architectural and training philosophies.

The Gemma model features a 26-layer transformer with 1152 hidden dimensions, narrower MLP channels (6912), and a normalized attention mechanism based on QK-normalization and grouped query attention (GQA). It applies multiple normalization layers, including pre- and post-FFN LayerNorms and separate q_norm and k_norm components. These characteristics make Gemma computationally efficient, with reduced memory requirements and improved regularization, suggesting better suitability for constrained environments.

In contrast, the LLaMA model employs a 16-layer transformer with wider channels (2048 hidden dimensions and 8192 in the MLP), using standard multi-head attention with RoPE positional encoding and softmax normalization. This configuration introduces greater redundancy and wider numerical margins, potentially offering higher tolerance to local faults or perturbations. The model also uses fewer normalization points and a simpler computation graph, which may reduce the propagation of numerical drift across layers.

To ensure fair evaluation, both models were prompted with identical inputs and assessed on the same set of benchmarks. The evaluation suite comprises seven representative tasks, spanning code generation, factual reasoning, commonsense QA, toxicity detection, and multilingual understanding:

- Code Generation:
  - HumanEval-X [19] (164 samples, max generation: 512 tokens): multilingual code synthesis; test-only.
  - MBPP [20] (974 samples, max generation: 256 tokens): Python function generation from natural language; test-only.
- Knowledge & Reasoning:
  - ARC-Challenge [21] (1172 samples, max generation: 64 tokens): multiple-choice questions requiring complex reasoning.
  - TruthfulQA [22] (817 samples, max generation: 64 tokens): tests robustness against factual misinformation.
  - OpenBookQA [23] (500 samples, max generation: 16 tokens): science-based elementary-level multiple-choice QA.
- Toxicity Detection:
  - Toxigen [24] (940 samples, max generation: 32 tokens): evaluates whether the model produces toxic responses from ambiguous prompts.
- Multilingual Understanding:
  - Belebele [25] (900 samples, max generation: 32 tokens): English subset (eng_Latn) of a multilingual reading comprehension benchmark.

Each task was formatted to enforce consistent maximum generation lengths (the upper limit on newly generated tokens), independent of input prompt length, to ensure comparability across models with different output behaviors.

C. Voltage Undervolting and Fault Injection Protocol

To simulate hardware-level instability, voltage perturbations were introduced through controlled undervolting on the Raspberry Pi 5. The `over_voltage` configuration parameter was set to negative values ranging from 0 to `-2`. Voltage levels below `-3` were found to destabilize the operating system, making such conditions infeasible for controlled experimentation.

Due to hardware constraints, undervolting settings required a full system reboot and could not be modified dynamically during inference. The system voltage was monitored using the `vcgencmd` utility; however, it remained nominally constant across runs, due to hardware-level voltage locking.

Unlike traditional fault injection approaches based on bit-flips or synthetic weight perturbation, voltage-induced faults originate from actual physical conditions. These faults cannot be easily modeled as additive Gaussian noise and tend to affect multiple computation components simultaneously, including matrix multiplications and normalization layers.

D. Drift Tracing and Output Consistency Metrics

To quantify the impact of undervolting on inference stability, both behavioral and internal consistency metrics were employed. For each input prompt, the full sequence of generated tokens was recorded.

Model output under fault conditions was compared against a baseline run at nominal voltage. The primary metric used for internal deviation was the layer-wise L2 distance between hidden state tensors, computed across the token sequence. This measure captures the extent of accumulated numerical drift across the transformer stack.

In addition to hidden-state drift, qualitative output consistency was assessed. A generation was considered to have failed if it produced empty outputs, crashed during decoding, or yielded semantically incoherent content. These failure modes reflect practical degradation in usability under real-world deployment conditions.

IV. Experiment

A. Experimental Design and Objectives

This section investigates the effects of voltage instability on the internal computations and functional behavior of large language models. Both google/gemma-3-1b-it and meta-llama/Llama-3.2-1B were evaluated under three voltage settings, corresponding to over_voltage levels of 0, -1, and -2. The test platform, Raspberry Pi 5, required a full system reboot to apply new voltage configurations. During undervolting at level -2, the system occasionally exhibited segmentation faults, and in severe cases, full system hangs that required physical power cycling.

All experiments used the same set of benchmark prompts and a fixed random seed (42) to ensure consistency. Two forms of degradation were monitored: token-level activation drift and observable output failure. Activation drift was measured

TABLE I
PASS RATE COMPARISON FOR GEMMA3 1B ACROSS VOLTAGE SETTINGS AND BENCHMARK TASKS

Voltage (V)	Voltage Level	ARC-Challenge			Belebele (eng Latn)			HumanEval-X			MBPP		
		Pass	**Fail**	**Pass Rate**	**Pass**	**Fail**	**Pass Rate**	**Pass**	**Fail**	**Pass Rate**	**Pass**	**Fail**	**Pass Rate**
0.8674	0	465	707	39.68%	46	854	5.11%	134	30	81.71%	488	486	50.10%
0.8433	-1	461	711	39.33%	38	862	4.22%	124	40	75.61%	471	503	48.36%
0.8126	-2*	447	725	38.14%	33	867	3.67%	92	72	56.10%	289	685	29.67%
		OpenBookQA			**Toxigen**			**TruthfulQA**					
		Pass	**Fail**	**Pass Rate**	**Pass**	**Fail**	**Pass Rate**	**Pass**	**Fail**	**Pass Rate**			
0.8674	0	10	490	2.00%	4	936	0.43%	147	670	17.99%			
0.8433	-1	11	489	2.20%	5	935	0.53%	156	661	19.09%			
0 .8126	-2*	15	485	3.00%	9	931	0.96%	169	648	20.69%			

* For over_voltage = -2, results were selected from multiple trials due to frequent segmentation faults and output instability.

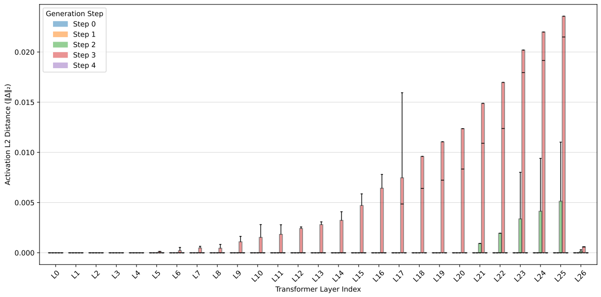

Fig. 3. Layer-wise L2 activation drift in google/gemma-3-1b-it under undervolting across generation steps. Deviation accumulates noticeably in deeper layers.

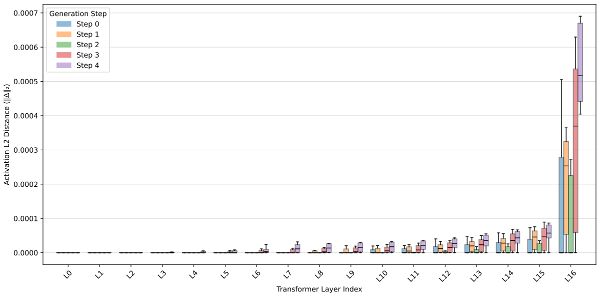

Fig. 4. Layer-wise L2 activation drift in meta-llama/Llama-3.2-1B under undervolting across generation steps. Drift remains uniformly low across layers.

by computing the layer-wise L2 distance between baseline (normal voltage level) and undervolted hidden states. Output degradation was defined to include both semantic errors and decoding collapse. Notably, repeated runs under identical conditions sometimes produced different results due to nondeterministic hardware behavior. Therefore, multiple trials were conducted per setting, and the most representative runs were selected for analysis and visualization.

B. Drift Patterns under Voltage Stress

To investigate internal numerical stability, the layer-wise L2 distance of activation tensors was computed between baseline runs and undervolted runs at different generation steps. Fig.3 and Fig.4 illustrate the distribution of drift magnitudes across transformer layers for `google/gemma-3-1b-it` and `meta-llama/Llama-3.2-1B`, respectively. Each box plot summarizes results from 10 independent inference runs conducted under controlled undervolting.

In the case of Gemma3 1B (Fig. 3), the L2 distance grows progressively with depth, particularly after layer 10. Later layers exhibit significantly larger deviations, indicating a drift accumulation effect along the transformer stack. This trend is especially pronounced at generation steps beyond the second token, suggesting that error propagation intensifies with sequence length. The presence of multiple normalization points throughout the architecture did not prevent the amplification of drift, implying that internal regularization alone may be insufficient for fault attenuation under voltage-induced perturbations.

By contrast, the LLaMA 3.2 1B model (Fig. 4) demonstrates markedly lower L2 distances overall, with most values remaining below 0.0005 across all layers and generation steps. The drift distribution is more uniform, and no specific region exhibits sharp instability. Despite having fewer normalization layers, the wider hidden dimension and more redundant attention structure appear to suppress local perturbations more effectively. Notably, the absence of high-frequency spikes across layers suggests that LLaMA's architecture is less sensitive to voltage-related noise in matrix multiplications.

These results indicate that numerical drift under undervolting is not only depth-dependent but also highly model-specific. Structural properties such as hidden width, normalization placement, and attention scaling significantly affect the model's resilience to low-level physical disturbances.

C. Functional Degradation on Benchmarks

To evaluate the real-world impact of undervolting on end-task performance, the Gemma3 1B model was tested across seven benchmarks: ARC, Belebele, HumanEval-X, MBPP, OpenBookQA, Toxigen, and TruthfulQA. Table I summarizes the results under three voltage levels. Pass rates were computed based on whether the generated output matched reference answers, following benchmark-specific criteria.

Even under mild undervolting (over_voltage = -1), degradation was observed across most tasks. ARC and MBPP showed slight decreases in pass rate, while TruthfulQA exhibited improved robustness, possibly due to the discrete nature of its question types. At more aggressive undervolting (over_voltage = -2), performance dropped considerably in all tasks. For instance, HumanEval-X pass rate fell from 81.7% to 56.1%, and MBPP dropped from 50.1% to 29.7%. Output collapse was also observed, including responses composed solely of repeated tokens (e.g., “F”), indicating a loss of decoding stability. Despite multiple runs, these artifacts persisted, and results were assembled from the most coherent outputs across trials.

To further analyze task-specific sensitivity, the degradation in functional accuracy across benchmarks was examined. Among all evaluated tasks, MBPP and HumanEval-X exhibited the most pronounced decline under undervolting. MBPP’s pass rate dropped by over 20 percentage points when the over_voltage level was reduced from 0 to -2, while HumanEval-X declined by more than 25 percentage points. These results suggest that code generation tasks, which rely on long-form consistency and multi-token reasoning, are particularly vulnerable to voltage-induced errors.

In contrast, tasks such as ARC and Belebele showed relatively minor reductions, under 2 percentage points. Surprisingly, tasks including TruthfulQA, OpenBookQA, and Toxigen displayed slight improvements under undervolting, though these changes likely fall within statistical noise margins. The inverted trend in TruthfulQA may be attributed to the binary-choice nature of the task, where random variation occasionally aligns with the correct answer.

These findings indicate that robustness to hardware-level faults is not uniform across benchmarks. Tasks involving structured generation and token coherence appear significantly more susceptible to drift accumulation, while short-response or classification-style tasks show higher tolerance.

D. Summary of Observations

The experimental results demonstrate that voltage instability leads to both internal numerical drift and measurable degradation in model performance. Drift accumulates progressively across transformer layers and manifests differently depending on architectural properties. Output collapse and response inconsistency were frequently observed, especially under moderate to severe undervolting. These effects did not always result in total failure but frequently degraded semantic correctness.

These findings underscore the need for fault-aware evaluation protocols in LLM deployment on edge platforms. Voltage-induced failures, while nondeterministic and non-adversarial, have the potential to silently compromise reliability. Robustness in LLM inference should therefore be assessed not only under synthetic perturbations but also in the presence of realistic hardware constraints.

V. Related Work

A. Voltage-Induced Faults and Hardware-Level Resilience

Prior research has extensively explored the impact of hardware-level perturbations on the reliability of deep neural networks. Studies on bit-flip errors, memory faults, and undervolting have demonstrated that even low-level physical disturbances can significantly degrade model accuracy, especially in convolutional and recurrent architectures. For example, hardware-level fault attacks such as Rowhammer–based bit flips have been shown to impair DNNs dramatically, sometimes reducing accuracy by over 90% with only a few corrupted bits [26], [27]. In addition, undervolting experiments on FPGA-based CNN accelerators have revealed a clear reliability-power trade-off, where reduced supply voltage leads to timing faults and increased error rates despite energy savings [18]. These works often propose algorithmic fault detection or masking mechanisms to mitigate degradation. However, their focus remains on traditional CNN/RNN architectures and synthetic fault injection methods, leaving transformer-based LLMs under realistic undervolting conditions largely unexamined.

B. LLM Robustness to Perturbations

The robustness of LLMs has been widely studied in the context of adversarial attacks [28], prompt injections [29], and data poisoning [30]. These perturbations are typically introduced at the input or training data level to manipulate model behavior without modifying the underlying architecture. For example, adversarial prompts can elicit harmful or incorrect outputs, while training-time poisoning can implant persistent backdoors into LLMs. Although such attacks differ in mechanism from hardware-induced faults, they share a common goal: disrupting inference consistency and semantic correctness. Despite the growing literature on adversarial robustness in LLMs, little is known about how physical computation faults, such as those induced by unstable power supply, interact with model architectures to cause output instability. This study highlights a complementary and underexamined dimension of LLM vulnerability.

VI. Conclusion

This work investigates the fault resilience of instruction-tuned LLMs on edge platforms via controlled undervolting on Raspberry Pi 5. We identify two key failure modes: numerical drift in hidden activations and degraded benchmark performance. Results show that robustness varies by model—Gemma3 1B suffers severe drift and collapse, while LLaMA 3.2 1B remains more stable. Structured generation tasks like MBPP and HumanEval-X are especially vulnerable. These findings underscore the need for hardware-aware reliability evaluation in LLM edge deployments.

References

[1] C. Dong, Y. Li *et al.*, “A survey of natural language generation,” *ACM Comput. Surv.*, vol. 55, no. 8, pp. 173:1–173:38, 2023. [Online]. Available: https://doi.org/10.1145/3554727

[2] L. Dong, N. Yang *et al.*, "Unified language model pre-training for natural language understanding and generation," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle *et al.*, Eds., 2019, pp. 13 042–13 054. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/c20bb2d9a50d5ac1f713f8b34d9aac5a-Abstract.html

[3] R. Rawassizadeh and Y. Rong, "Odsearch: Fast and resource efficient on-device natural language search for fitness trackers' data," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 4, pp. 178:1–178:25, 2022. [Online]. Available: https://doi.org/10.1145/3569488

[4] H. Wen, Y. Li *et al.*, "Autodroid: Llm-powered task automation in android," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking, ACM MobiCom 2024, Washington D.C., DC, USA, November 18-22, 2024*, W. Shi, D. Ganesan, and N. D. Lane, Eds. ACM, 2024, pp. 543–557. [Online]. Available: https://doi.org/10.1145/3636534.3649379

[5] I. Gur, H. Furuta *et al.*, "A real-world webagent with planning, long context understanding, and program synthesis," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=9JQtrumvg8

[6] J. Jeong, Y. Zou *et al.*, "Winclip: Zero-/few-shot anomaly classification and segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 19 606–19 616. [Online]. Available: https://doi.org/10.1109/CVPR52729.2023.01878

[7] Q. Cao, P. Khanna *et al.*, "Mobivqa: Efficient on-device visual question answering," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 2, pp. 44:1–44:23, 2022. [Online]. Available: https://doi.org/10.1145/3534619

[8] Z. Fan, Q. Zhang *et al.*, "Taskfusion: An efficient transfer learning architecture with dual delta sparsity for multi-task natural language processing," in *Proceedings of the 50th Annual International Symposium on Computer Architecture, ISCA 2023, Orlando, FL, USA, June 17-21, 2023*, Y. Solihin and M. A. Heinrich, Eds. ACM, 2023, pp. 5:1–5:14. [Online]. Available: https://doi.org/10.1145/3579371.3589040

[9] R. Bhardwaj, Z. Xia *et al.*, "Ekya: Continuous learning of video analytics models on edge compute servers," in *19th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2022, Renton, WA, USA, April 4-6, 2022*, A. Phanishayee and V. Sekar, Eds. USENIX Association, 2022, pp. 119–135. [Online]. Available: https://www.usenix.org/conference/nsdi22/presentation/bhardwaj

[10] A. Tschand, A. T. R. Rajan *et al.*, "Mlperf power: Benchmarking the energy efficiency of machine learning systems from μwatts to mwatts for sustainable AI," in *IEEE International Symposium on High Performance Computer Architecture, HPCA 2025, Las Vegas, NV, USA, March 1-5, 2025*. IEEE, 2025, pp. 1201–1216. [Online]. Available: https://doi.org/10.1109/HPCA61900.2025.00092

[11] M. M. H. Shuvo, S. K. Islam *et al.*, "Efficient acceleration of deep learning inference on resource-constrained edge devices: A review," *Proc. IEEE*, vol. 111, no. 1, pp. 42–91, 2023. [Online]. Available: https://doi.org/10.1109/JPROC.2022.3226481

[12] J. Lin, X. Jiao *et al.*, "Vulnerability of hardware neural networks to dynamic operation point variations," *IEEE Des. Test*, vol. 37, no. 5, pp. 75–84, 2020. [Online]. Available: https://doi.org/10.1109/MDAT.2020.2986742

[13] R. Gong, Y. Ding *et al.*, "A survey of low-bit large language models: Basics, systems, and algorithms," *CoRR*, vol. abs/2409.16694, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2409.16694

[14] E. Ozen, "Algorithm-centric design of reliable and efficient deep learning processing systems," Ph.D. dissertation, University of California, San Diego, USA, 2023. [Online]. Available: https://www.escholarship.org/uc/item/515341v3

[15] G. Team, "Gemma 3," 2025. [Online]. Available: https://goo.gle/Gemma3Report

[16] A. Dubey, A. Jauhri *et al.*, "The llama 3 herd of models," *CoRR*, vol. abs/2407.21783, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2407.21783

[17] B. Salami, E. B. Onural *et al.*, "An experimental study of reduced-voltage operation in modern fpgas for neural network acceleration," in *50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2020, Valencia, Spain, June 29 - July 2, 2020*. IEEE, 2020, pp. 138–149. [Online]. Available: https://doi.org/10.1109/DSN48063.2020.00032

[18] M. Rinkinen, L. Koskinen *et al.*, "Shavette: Low power neural network acceleration via algorithm-level error detection and undervolting," *CoRR*, vol. abs/2410.13415, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2410.13415

[19] Q. Zheng, X. Xia *et al.*, "Codegeex: A pre-trained model for code generation with multilingual benchmarking on humaneval-x," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, A. K. Singh, Y. Sun *et al.*, Eds. ACM, 2023, pp. 5673–5684. [Online]. Available: https://doi.org/10.1145/3580305.3599790

[20] J. Austin, A. Odena *et al.*, "Program synthesis with large language models," *CoRR*, vol. abs/2108.07732, 2021. [Online]. Available: https://arxiv.org/abs/2108.07732

[21] S. Bhakthavatsalam, D. Khashabi *et al.*, "Think you have solved direct-answer question answering? try arc-da, the direct-answer AI2 reasoning challenge," *CoRR*, vol. abs/2102.03315, 2021. [Online]. Available: https://arxiv.org/abs/2102.03315

[22] S. Lin, J. Hilton, and O. Evans, "Truthfulqa: Measuring how models mimic human falsehoods," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Association for Computational Linguistics, 2022, pp. 3214–3252. [Online]. Available: https://doi.org/10.18653/v1/2022.acl-long.229

[23] T. Mihaylov, P. Clark *et al.*, "Can a suit of armor conduct electricity? A new dataset for open book question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, E. Riloff, D. Chiang *et al.*, Eds. Association for Computational Linguistics, 2018, pp. 2381–2391. [Online]. Available: https://doi.org/10.18653/v1/d18-1260

[24] T. Hartvigsen, S. Gabriel *et al.*, "Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Association for Computational Linguistics, 2022, pp. 3309–3326. [Online]. Available: https://doi.org/10.18653/v1/2022.acl-long.234

[25] L. Bandarkar, D. Liang *et al.*, "The belebele benchmark: a parallel reading comprehension dataset in 122 language variants," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 749–775. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-long.44

[26] M. Beyer, A. Morozov *et al.*, "Fault injectors for tensorflow: Evaluation of the impact of random hardware faults on deep cnns," *CoRR*, vol. abs/2012.07037, 2020. [Online]. Available: https://arxiv.org/abs/2012.07037

[27] S. Hong, P. Frigo *et al.*, "Terminal brain damage: Exposing the graceless degradation in deep neural networks under hardware fault attacks," in *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, N. Heninger and P. Traynor, Eds. USENIX Association, 2019, pp. 497–514. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/hong

[28] K. Zhu, J. Wang *et al.*, "Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts," in *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis, LAMPS 2024, Salt Lake City, UT, USA, October 14-18, 2024*, B. Li, W. Xu *et al.*, Eds. ACM, 2024, pp. 57–68. [Online]. Available: https://doi.org/10.1145/3689217.3690621

[29] Y. Liu, Y. Jia *et al.*, "Formalizing and benchmarking prompt injection attacks and defenses," in *33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024*, D. Balzarotti and W. Xu, Eds. USENIX Association, 2024. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/liu-yupei

[30] D. A. Alber, Z. Yang *et al.*, "Medical large language models are vulnerable to data-poisoning attacks," *Nature Medicine*, vol. 31, no. 2, pp. 618–626, 2025.