

Reliability Profiling of LLMs and VLMs for Edge AI under Activation Corruption

Georgios Mermigkis^{*}, Argiris Sofotasios, Panagiotis Hadjidoukas, and
George Papadimitriou

Computer Engineering and Informatics Department, University of Patras, Patras, Greece
{g.mermigkis, a.sofotasios, phadjido}@ceid.upatras.gr, and gpapad@upatras.gr

Abstract

Generative Artificial Intelligence (AI) is increasingly considered for edge platforms, where model selection must account not only for accuracy and execution cost, but also for resilience to hardware-induced corruption. However, conventional fault injection studies primarily characterize whether a perturbation affects the final output, providing limited insight into how corruption reshapes the autoregressive decoding process itself. In this paper, we use Entropy Production Rate (EPR) to characterize the response of Large Language Models (LLMs) and Vision-Language Models (VLMs) to activation corruption. We evaluate two LLMs and two VLMs across four datasets under bit-flip (BF) and probabilistic-NaN (pNaN) perturbations. Fault-free executions remain well below the EPR ceiling, whereas pNaN drives every evaluated model-dataset pair close to saturation. BF produces a more model-dependent response, with Llama-3.2-3B increasing by approximately 1.6x to 2x over its baseline while Qwen3-32B and both VLMs approach the entropy ceiling, and this profiling adds less than 2% to total generation time. Overall, our results show that corruption produces distinct decoding-level responses across models and fault types, positioning EPR as a practical, low-cost resilience signal for guiding model and hardware choices in generative Edge AI.

1 Introduction

Edge Artificial Intelligence (AI) is expanding beyond conventional discriminative models toward generative Large Language Model (LLM) and Vision Language Model (VLM) workloads that combine language reasoning with increasingly rich sensor and visual inputs [1], [2]. This transition introduces a different reliability challenge, since a hardware-induced fault during autoregressive decoding can modify the probability distribution of the next token and consequently influence the trajectory of all subsequent generation steps [3]. Therefore, resilience characterization of generative Edge AI should capture not only whether the final output changes, but also how corruption propagates through the decoding process.

^{*}Corresponding Author

Hardware reliability is particularly relevant to edge deployments because model, accelerator, and operating-point choices jointly determine the behavior of the deployed AI system. Existing fault-injection frameworks, such as PyTorchFI [4] and Ares [5], primarily quantify vulnerability by comparing corrupted executions against a sample-level golden reference, using output deviations, application-level correctness, or the identification of sensitive network components [6]. These metrics determine whether corruption eventually affects the computation but compress the propagation process into an end-point outcome, and their reliance on a golden reference means they cannot assess corruption when no such reference exists, as is the case during real-time deployment. Additionally, corruption that does not immediately alter the output can propagate through generation undetected, since no existing mechanism currently reveals that corruption has reached the model in the absence of a golden reference. For autoregressive models, this leaves an important characterization gap, since different models may react differently to the same low-level corruption even when they execute the same type of generative workload.

To characterize this behavior without relying on a golden reference, we turn to Entropy Production Rate (EPR), which provides a compact representation of decoding dynamics by aggregating the entropy of the top-ranked token probabilities across the complete generated sequence. It was originally proposed in the context of hallucination assessment [7] and subsequently employed to characterize explanation stability [8]. In this paper, we employ EPR for a different purpose, to quantify how activation corruption changes the decoding regime of a generative model. Specifically, we characterize (a) the fault-free entropy regime of each model-workload pair, (b) the displacement induced by different corruption mechanisms, and (c) the extent to which this response varies across models and modalities. This provides a reliability profile that can complement accuracy, model size, and execution cost when reasoning about candidate models for Edge AI.

To this end, this paper makes the following contributions:

- We present an Edge-AI-oriented reliability characterization methodology for autoregressive LLMs and VLMs, combining controlled activation corruption with sequence-level EPR measurements.
- We characterize two corruption mechanisms across eight model-dataset pairs and quantify how fault-induced entropy behavior depends on the model, workload, and corruption type.
- We demonstrate that severe non-finite corruption consistently drives generation toward entropy saturation, whereas bit-flip (BF) produces a more model-dependent response, with EPR extraction adding less than 2% to total generation time.

2 EPR-Based Reliability Profiling

Autoregressive generation exposes a probability distribution over candidate tokens at every decoding step. EPR [7] summarizes how concentrated or diffused these distributions remain over the complete generated sequence and therefore provides a compact representation of decoding behavior. Let q denote the input query and $T = \{t_1, \dots, t_L\}$ the generated sequence of length L produced in response to q . At each step j , generation is conditioned on q and the preceding tokens $t_{<j} = \{t_1, \dots, t_{j-1}\}$. EPR is defined as follows:

$$\text{EPR}_K(q, T) = \frac{1}{L} \sum_{j=1}^L \tilde{H}_K(q, t_{<j}) \quad \tilde{H}_K(q, t_{<j}) = - \sum_{k=1}^K p_{r(k),j} \log_2(p_{r(k),j})$$

where $\tilde{H}_K$ is the entropy estimated from the top K token probabilities at generation step j , and $p_{r(k),j}$ denotes the probability of the k -th ranked candidate token at step j , given q and $t_{<j}$. Higher

EPR indicates that the probability mass is distributed more uniformly across the top-K candidates, whereas lower EPR indicates stronger concentration on fewer candidates. Since the maximum entropy for $K = 10$ is $\log_2 10 = 3.32$ bits, this ceiling also provides a common reference for comparing models whose fault-free EPR values differ. We use this property to quantify how far activation corruption displaces each model from its normal decoding regime toward entropy saturation.



Figure 1: Edge-AI reliability-profiling workflow. Fault-free and corrupted executions produce token-probability traces from which EPR distributions are obtained to characterize baseline behavior, corruption-induced displacement, variability, and measurement cost for each model-workload pair.

3 Fault Injection and Reliability Profiling

We generate fault-free and corrupted executions using an activation-level fault-injection framework adapted to autoregressive LLM and VLM decoding. Each injection targets one randomly selected activation in one randomly selected layer, sampled once per generation, and is injected once per forward pass. We employ two corruption mechanisms with different perturbation characteristics. BF [6] inverts one randomly selected bit of the targeted activation, so its impact depends on the significance of the selected bit. Probabilistic-NaN (pNaN) [3] replaces the targeted activation with a non-finite value according to probability p , checked at each forward pass, which lets p be set to a hardware-grounded bit error rate rather than a fixed corruption event, producing a more consistently disruptive perturbation regardless of location.

For every sampled prompt, we record the fault-free EPR and repeat the generation under BF and pNaN using the same decoding configuration. The resulting distributions form the reliability profile of each model-dataset pair. We report their median and interquartile range to capture both the typical entropy response and its variability across prompts. Since different models exhibit different fault-free entropy levels, comparisons focus on both absolute EPR and the displacement from the corresponding fault-free regime. Because EPR is computed from the top-10 token log-probabilities already produced during decoding, it requires no additional model forward pass, keeping profiling overhead minimal.

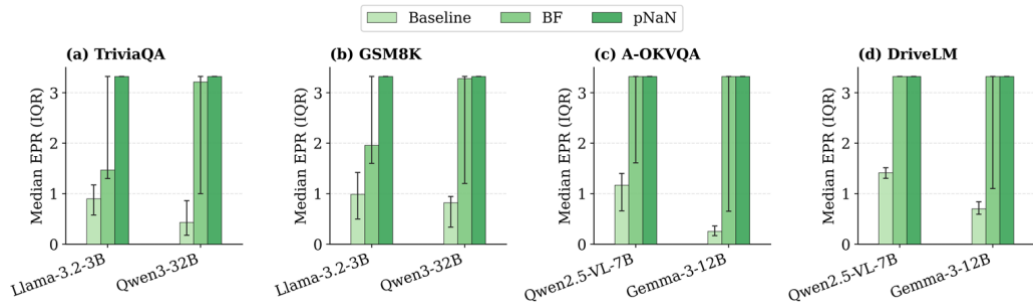


Figure 2: Median EPR with interquartile ranges (25th-75th percentile) for the fault-free baseline and both injection strategies, on (a) TriviaQA, (b) GSM8K, (c) A-OKVQA, and (d) DriveLM, for Llama-3.2-3B, Qwen3-32B, Qwen2.5-VL-7B, and Gemma-3-12B.

4 Evaluation

4.1 Experimental Setup

We evaluate reliability profiles across four generative models spanning text-only and multimodal inference, namely Llama-3.2-3B and Qwen3-32B [9], [10], and Qwen2.5-VL-7B and Gemma-3-12B [11], [12]. The LLMs are evaluated on TriviaQA and GSM8K [13], [14], and the VLMs on A-OKVQA and DriveLM [15], [16], with DriveLM providing a smart-mobility workload representative of an important Edge-AI application domain. For each model-dataset pair, we sample 500 prompts. Generation uses non-greedy decoding with a temperature of 0.6, a fixed seed of 42, up to 2,048 new tokens, and a repetition penalty. We discard one warm-up generation per prompt and retain the top-10 token log-probabilities ($K = 10$) for EPR computation. Offline profiling records fault-free EPR for every model-dataset pair. For the corruption experiments, each prompt is rerun with either BF or pNaN ($p = 0.05$, consistent with prior work [3]), targeting a single randomly selected activation site and keeping the comparison prompt-matched against the clean run. All experiments run on one NVIDIA H100 GPU (80GB HBM3), using Python 3.12.3, with all models loaded and executed in bfloat16 precision.

4.2 Results

Having established the profiling methodology, we now characterize how activation corruption reshapes the decoding-level entropy response across models, quantify this response independently of each model's fault-free regime, and assess the measurement cost of EPR extraction itself.

Fig. 2 characterizes the entropy response of the evaluated model-dataset pairs under BF and pNaN corruption. Fault-free EPR remains well below the theoretical ceiling of $\log_2 10 = 3.32$ bits, ranging from approximately 0.25 for Gemma-3-12B on A-OKVQA to approximately 1.4 for Qwen2.5-VL-7B on DriveLM, with Llama-3.2-3B on GSM8K reaching approximately 1.0. pNaN produces a common response across the evaluated models, driving EPR close to the entropy ceiling with a tight spread. BF behaves differently [17], [18], [19]. Llama-3.2-3B reaches median values of approximately 1.45 to 1.95, i.e., 1.6x to 2x its fault-free baseline, whereas Qwen3-32B and both VLMs approach 3.2 to 3.3. Moreover, Qwen2.5-VL-7B on DriveLM exhibits a wider range extending toward its fault-free regime. Overall, these results show that the corruption mechanism and the model jointly determine the decoding-level reliability profile, in that severe non-finite corruption produces a largely common saturation regime, while bit-level corruption exposes considerably greater model dependence.

To compare models independently of their different fault-free entropy regimes, we define the normalized EPR shift as

$$S_{\text{EPR}} = \frac{\text{EPR}_{\text{corr}} - \text{EPR}_{\text{ref}}}{\log_2(K) - \text{EPR}_{\text{ref}}}$$

where EPR_{ref} is the reference, corruption-free EPR and EPR_{corr} is the EPR under a given corruption strategy. Fig. 3 reports this normalized EPR shift per model-dataset pair under BF and pNaN corruption. A value close to one indicates that corruption consumes most of the entropy headroom available between the fault-free execution and the theoretical ceiling. This normalization exposes the cross-model contrast under BF more clearly than absolute EPR alone, while pNaN remains close to saturation across the evaluated pairs. Therefore, fault susceptibility cannot be inferred solely from the nominal model or workload class and should be characterized for the candidate model-workload combination.

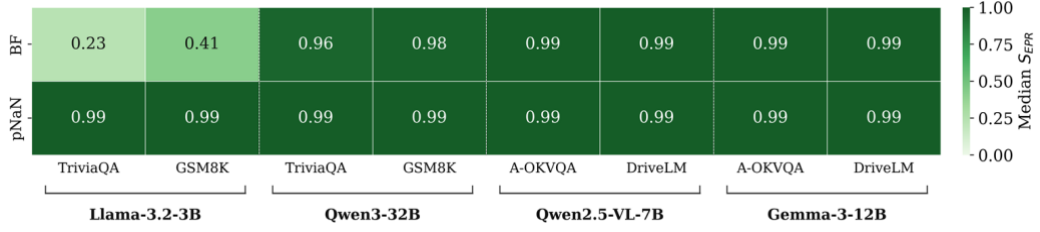


Figure 3: Median normalized EPR shift (SEPR) per model-dataset pair under BF and pNaN corruption, with values near one indicating near-complete consumption of the available entropy headroom.

Characterizing every combination is only practical if the profiling itself is inexpensive. To measure the extra cost imposed by EPR computation, we record, for each prompt, the wall-clock generation time t_{gen} and the additional wall-clock time t_{EPR} needed to compute EPR. We define the EPR overhead as:

$$\eta_{EPR} = \frac{t_{EPR}}{t_{gen}} \times 100\%$$

and report its mean across all prompts for each model-dataset pair (see Fig. 4). The overhead decreases from approximately 1.73% for Llama-3.2-3B to 0.55% for Qwen3-32B among the text models, and from approximately 1.35% for Qwen2.5-VL-7B to 0.7% for Gemma-3-12B among the VLMs. Therefore, EPR characterization remains below 2% across all evaluated models, allowing reliability profiling without materially altering the execution cost of the underlying generative workload.

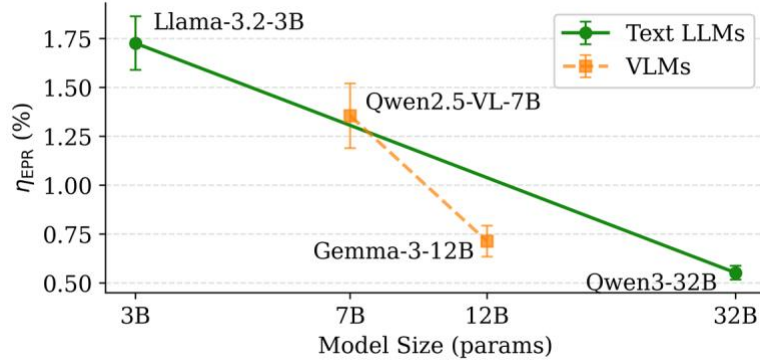


Figure 4: EPR characterization cost as a percentage of total generation time across the evaluated LLMs and VLMs.

5 Implications for Reliable Edge AI

The results reveal two properties relevant to the deployment of generative models in Edge-AI systems. First, activation corruption does not produce a model-independent decoding response. While severe pNaN perturbations consistently drive generation toward entropy saturation, BF exposes considerably stronger differences among the evaluated models. Therefore, hardware resilience should be treated as a model-dependent deployment characteristic rather than assumed from model functionality alone. Second, EPR makes this behavior observable with less than 2% computation overhead. It is, to our knowledge, the first signal that reveals corruption has reached a model's decoding process without requiring a golden reference against which to compare the output. EPR is thus an unveiling signal rather than a correctness signal, exposing how

strongly a candidate model's decoding process reacts to low-level corruption, not whether the resulting output is semantically acceptable. Future work will combine this signal with semantic degradation, model footprint, latency, and energy measurements to support reliability-aware model selection for heterogeneous Edge-AI platforms. It will also extend evaluation to actual edge hardware platforms.

6 Acknowledgement

This paper has been financed by the funding program “MEDICUS” of the University of Patras.

References

- [1] Y. Zheng, Y. Chen, B. Qian, X. Shi, Y. Shu, and J. Chen, “A Review on Edge Large Language Models: Design, Execution, and Applications,” *ACM Comput. Surv.*, vol. 57, no. 8, pp. 1–35, Aug. 2025, doi: 10.1145/3719664.
- [2] A. Sharshar, L. U. Khan, W. Ullah, and M. Guizani, “Vision-Language Models for Edge Networks: A Comprehensive Survey,” *IEEE Internet Things J.*, vol. 12, no. 16, pp. 32701–32724, Aug. 2025, doi: 10.1109/JIOT.2025.3579032.
- [3] O. Chatzopoulos, G. Papadimitriou, D. Gizopoulos, H. D. Dixit, and S. Sankar, “From Gates to SDCs: Understanding Fault Propagation Through the Compute Stack,” in *2025 Design, Automation & Test in Europe Conference (DATE)*, Mar. 2025, pp. 1–7. doi: 10.23919/DATe64628.2025.10992727.
- [4] A. Mahmoud *et al.*, “PyTorchFI: A Runtime Perturbation Tool for DNNs,” in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, Jun. 2020, pp. 25–31. doi: 10.1109/DSN-W50199.2020.00014.
- [5] B. Reagen *et al.*, “Ares: A framework for quantifying the resilience of deep neural networks,” in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, Jun. 2018, pp. 1–6. doi: 10.1109/DAC.2018.8465834.
- [6] D. Ma *et al.*, “Dr. DNA: Combating Silent Data Corruptions in Deep Learning using Distribution of Neuron Activations,” in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, La Jolla CA USA: ACM, Apr. 2024, pp. 239–252. doi: 10.1145/3620666.3651349.
- [7] C. Moslonka, H. Randrianarivo, A. Garnier, and E. Malherbe, “Learned Hallucination Detection in Black-Box LLMs using Token-level Entropy Production Rate,” Jan. 20, 2026, *arXiv*: arXiv:2509.04492. doi: 10.48550/arXiv.2509.04492.
- [8] G. Mermigkis *et al.*, “A Two-Stage LLM Framework for Accessible and Verified XAI Explanations,” Apr. 14, 2026, *arXiv*: arXiv:2604.12543. doi: 10.48550/arXiv.2604.12543.
- [9] A. Grattafiori *et al.*, “The Llama 3 Herd of Models,” Nov. 23, 2024, *arXiv*: arXiv:2407.21783. doi: 10.48550/arXiv.2407.21783.
- [10] A. Yang *et al.*, “Qwen3 Technical Report,” May 14, 2025, *arXiv*: arXiv:2505.09388. doi: 10.48550/arXiv.2505.09388.
- [11] S. Bai *et al.*, “Qwen2.5-VL Technical Report,” Feb. 19, 2025, *arXiv*: arXiv:2502.13923. doi: 10.48550/arXiv.2502.13923.
- [12] G. Team *et al.*, “Gemma 3 Technical Report,” Mar. 25, 2025, *arXiv*: arXiv:2503.19786. doi: 10.48550/arXiv.2503.19786.
- [13] M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer, “TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver,

- Canada: Association for Computational Linguistics, 2017, pp. 1601–1611. doi: 10.18653/v1/P17-1147.
- [14] K. Cobbe *et al.*, “Training Verifiers to Solve Math Word Problems,” Nov. 18, 2021, *arXiv*: arXiv:2110.14168. doi: 10.48550/arXiv.2110.14168.
 - [15] D. Schwenk, A. Khandelwal, C. Clark, K. Marino, and R. Mottaghi, “A-OKVQA: A Benchmark for Visual Question Answering Using World Knowledge,” in *Computer Vision – ECCV 2022*, vol. 13668, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., in Lecture Notes in Computer Science, vol. 13668, Cham: Springer Nature Switzerland, 2022, pp. 146–162. doi: 10.1007/978-3-031-20074-8_9.
 - [16] C. Sima *et al.*, “DriveLM: Driving with Graph Visual Question Answering,” in *Computer Vision – ECCV 2024*, vol. 15110, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds., in Lecture Notes in Computer Science, vol. 15110, Cham: Springer Nature Switzerland, 2025, pp. 256–274. doi: 10.1007/978-3-031-72943-0_15.
 - [17] G. Papadimitriou, M. Kaliorakis, A. Chatzidimitriou, D. Gizopoulos, G. Favor, K. Sankaran, and S. Das, “A System-Level Voltage/Frequency Scaling Characterization Framework for Multicore CPUs,” In 13th IEEE Workshop on Silicon Errors in Logic - System Effects (SELSE ’17), Boston, MA, USA, 2017.
 - [18] D. Agiakatsikas, G. Papadimitriou, V. Karakostas, D. Gizopoulos, M. Psarakis, and C. Bélanger-Champagne, “Impact of Voltage Scaling on Soft Errors Susceptibility of Multicore Server CPUs,” in *2023 56th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Oct. 2023, pp. 957–971.
 - [19] O. Chatzopoulos, N. Karystinos, G. Papadimitriou, D. Gizopoulos, H. D. Dixit, and S. Sankar, “Veritas – Demystifying Silent Data Corruptions: μ Arch-Level Modeling and Fleet Data of Modern x86 CPUs,” in *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Mar. 2025, pp. 1–14. doi: 10.1109/HPCA61900.2025.00012.