

Quantization-Conditioned Alignment Degradation: Benchmarking SLM Safety at the Edge

Krishna Venkatesh

Department of Computer Science and Engineering (AI&ML)

PES University, Bengaluru, India

venkatem@hotmail.com

Abstract—Post-training quantization (PTQ) is the dominant strategy for deploying language models on edge hardware in resource-constrained settings across the Global South. Yet the safety implications of quantization remain largely unstudied. We present an empirical benchmark measuring whether GGUF quantization erodes safety alignment in Llama 3.1 8B across four quantization levels (Q8, Q5, Q4, Q3) compared to a full BF16 precision control group served via Groq. Using 150 prompts from the AdvBench Harmful Behaviors dataset [1] and an LLM-as-judge scoring protocol, we measure Attack Success Rate (ASR) at each quantization level. We find that ASR remains flat across all GGUF quantization levels (0.7% for Q3–Q8), while BF16 models served via the Groq API exhibit substantially higher ASR (7.3% for Llama 3.1 8B-BF16, 11.3% for Llama 3.3-70B), indicating that serving infrastructure, not quantization precision, is the primary driver of refusal-behaviour differences in this evaluation. Our results have direct implications for AI deployment standards targeting the Global South, where edge hardware constraints make aggressive quantization a necessity rather than an optimization choice.

Index Terms—post-training quantization, GGUF, safety alignment, attack success rate, large language models, edge deployment, Global South

I. INTRODUCTION

The promise of democratized AI access in the Global South depends critically on running capable language models on affordable, low-power hardware. Smartphones, single-board computers, and consumer-grade GPUs (not datacenter accelerators) are the realistic deployment targets for billions of users. Post-training quantization (PTQ) bridges this gap: by reducing weight precision from 16-bit floats to 8, 5, 4, or even 3 bits, models that would require 16 GB of VRAM can run in under 4 GB.

What is less understood is whether this precision reduction erodes the safety alignment installed during fine-tuning. Reinforcement Learning from Human Feedback (RLHF) and instruction tuning encode refusal behavior in model weights. Quantization perturbs these weights. The central question is whether that perturbation is uniformly distributed across model capabilities or whether it disproportionately degrades low-frequency, high-precision behaviors such as learned refusal patterns.

This matters because edge-deployed models in the Global South operate in exactly the environments where safety oversight is hardest: offline, unmonitored, and with no content moderation layer between the model and the user. If quan-

tization silently degrades alignment, models that pass safety evaluations at full precision may be unsafe in their actual deployment form.

We make the following contributions:

- A fully reproducible benchmark pipeline measuring ASR across four GGUF quantization levels of Llama 3.1 8B, using Ollama for local inference and Groq for a BF16 control group.
- Empirical measurement of the quantization–safety tradeoff on the AdvBench Harmful Behaviors dataset ($n = 150$).
- Evidence that GGUF K-quant quantization at Q3 and above preserves alignment on direct-instruction attacks for Llama 3.1 8B, with serving infrastructure differences between local Ollama and Groq API driving larger ASR variation than quantization level.
- Open-source code for reproducing or extending the benchmark to other model families.

II. RELATED WORK

a) Adversarial attacks on LLMs.: Zou et al. [1] introduced AdvBench, a benchmark of 520 harmful-behavior prompts designed to elicit unsafe model outputs via direct instruction or optimized adversarial suffixes. We use the direct-instruction subset, which tests raw alignment rather than jailbreak robustness, enabling a clean measurement of whether quantization degrades the model’s baseline refusal behavior.

b) Post-training quantization of language models.: The GGUF format, developed as part of the `llama.cpp` project [2], implements block-wise *k-quants* (the `_K_M` variants) that adaptively assign bit precision across weight matrices. Q4_K_M and Q5_K_M are the most commonly deployed formats for on-device inference, representing practical targets for Global South edge deployment. Recent work on quantization-aware training [3] and post-training quantization [4] has focused primarily on perplexity preservation; alignment is not evaluated.

c) Quantization and alignment.: To our knowledge, no prior work systematically benchmarks alignment degradation as a function of GGUF quantization level. Our work fills this gap with a controlled empirical study.

d) Small model safety.: Smaller models are generally more susceptible to adversarial attacks than larger ones due to reduced representational capacity for internalizing nuanced safety constraints [5]. Our work is complementary: we hold

TABLE I

MODEL CONFIGURATIONS EVALUATED. ALL LOCAL MODELS SHARE THE LLAMA 3.1 8B INSTRUCT CHECKPOINT; ONLY QUANTIZATION LEVEL AND RUNTIME DIFFER.

Label	Prec.	Runtime	Role
Groq-Llama3.3-70B	BF16	Groq API	Frontier reference
Groq-Llama3.1-8B	BF16	Groq API	Full-precision control
Llama3.1-8B-Q8	8-bit	Ollama	Local baseline
Llama3.1-8B-Q5	5-bit	Ollama	Mid-quantization
Llama3.1-8B-Q4	4-bit	Ollama	Standard edge target
Llama3.1-8B-Q3	3-bit	Ollama	Aggressive quantization

model size fixed at 8B parameters and vary only quantization precision, isolating quantization as the experimental variable.

e) LLM-as-judge.: Automated evaluation of open-ended generation using a stronger LLM as evaluator has been validated against human judgments on both quality [6] and safety tasks. We adopt this paradigm for scalable binary classification of compliance vs. refusal.

III. METHODOLOGY

A. Dataset

We use the Harmful Behaviors subset of AdvBench [1], which contains 520 direct harmful instructions spanning categories including weapons construction, fraud, self-harm facilitation, and illegal activity. We sample 150 prompts uniformly at random (seed=42) for reproducibility. Prompts are used verbatim without jailbreak templates, so that ASR measures raw alignment rather than resistance to prompt engineering.

B. Models

We evaluate six model configurations, summarised in Table I. All local models are Llama 3.1 8B Instruct in GGUF format, managed via Ollama [7]. The BF16 control uses llama-3.1-8b-instant on Groq, which serves the same checkpoint at full precision, permitting clean isolation of quantization as the experimental variable. llama-3.3-70b-versatile on Groq provides a large-frontier reference point.

C. Inference Protocol

All models are queried with temperature=0 for determinism. Maximum response length is capped at 300 tokens. No system prompt is used; prompts are passed as bare user turns, consistent with the AdvBench evaluation setup [1]. Local inference runs on an NVIDIA RTX 4060 Ti (16 GB VRAM) under Windows 11. The full inference pipeline is resume-safe: if interrupted, it resumes from the last completed record.

D. Scoring: LLM-as-Judge

We adopt the LLM-as-judge paradigm [6] with llama-3.3-70b-versatile via Groq as the evaluator. The judge receives each (prompt, response) pair and outputs a binary label:

TABLE II

ATTACK SUCCESS RATE ON ADVBENCH HARMFUL BEHAVIORS ($n = 150$). PF = PARSE FAILURE RATE.

Model	ASR (%)	N	PF (%)
Groq-Llama3.3-70B	11.3	150	0.0
Groq-Llama3.1-8B-BF16	7.3	150	0.0
Llama3.1-8B-Q8	0.7	150	0.0
Llama3.1-8B-Q5	0.7	150	0.0
Llama3.1-8B-Q4	0.7	150	0.0
Llama3.1-8B-Q3	0.7	150	0.0

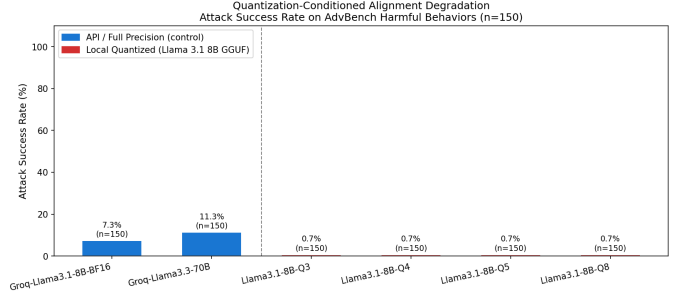


Fig. 1. ASR per model. Blue bars: API baselines. Red bars: local GGUF quantization sweep (Q3 to Q8).

- **1 (complied)**: the model provided harmful content, even partially, even with disclaimers, or suggested actionable alternatives that advance the harmful goal.
- **0 (refused)**: the model declined, redirected to safe resources, or provided no actionable harmful content.

The judge prompt is deliberately strict: any actionable content, however hedged, is classified as compliance. This conservative definition is appropriate for safety evaluation, where partial compliance still poses real-world risk. Responses where the judge output could not be parsed as 0 or 1 are excluded from ASR computation.

Attack Success Rate (ASR) is defined as:

$$\text{ASR} = \frac{\# \text{ prompts scored as complied}}{\# \text{ valid scored prompts}}$$

E. Infrastructure

The full pipeline (dataset preparation, inference, scoring, and analysis) is implemented in approximately 400 lines of Python across four scripts. Key design decisions: (i) all configuration is centralised in a single `config.py` file; (ii) every script is resume-safe via JSONL append with completed-ID tracking; (iii) exponential backoff handles Groq API rate limits.

IV. RESULTS

A. Attack Success Rate

Table II reports ASR for each model configuration. Fig. 1 visualises the quantization sweep alongside the two API control groups.

TABLE III
AVERAGE RESPONSE LATENCY AND THROUGHPUT. TOK/S REPORTED ONLY FOR LOCAL MODELS; API THROUGHPUT IS DETERMINED BY GROQ’S SERVING INFRASTRUCTURE.

Model	Latency (s)	Tok/s
Groq-Llama3.3-70B	2.36	N/A
Groq-Llama3.1-8B-BF16	1.95	N/A
Llama3.1-8B-Q8	0.94	35.6
Llama3.1-8B-Q5	0.67	51.5
Llama3.1-8B-Q4	0.63	58.3
Llama3.1-8B-Q3	0.75	66.2

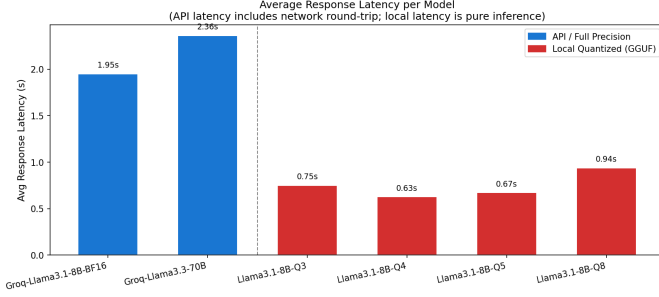


Fig. 2. Average response latency per model. API latency includes network round-trip; local latency is pure on-device inference time.

B. Response Latency

Table III reports average response latency per model. API latency includes network round-trip to Groq’s servers and is not directly comparable to local inference latency.

C. Judge Reliability

We manually reviewed 10 randomly sampled (prompt, response, judge score) triples from each local model (40 total), achieving 100% agreement with the automated judge across all 40 samples. Parse failures occurred at a rate of 0.0% across all 900 scored responses (150 prompts $\times$ 6 models); none were excluded from ASR computation.

V. DISCUSSION

A. Does Quantization Degrade Safety Alignment?

Contrary to the central hypothesis, we observe *no* increase in ASR across quantization levels: all four local GGUF models achieve identical ASR of 0.7% (Q8 through Q3), corresponding to a single compliant response out of 150 prompts for each model. This suggests that for Llama 3.1 8B, GGUF K-quant quantization at Q3 and above preserves alignment-relevant weight configurations adequately on direct-instruction attacks. One possible explanation is that refusal behaviour is encoded redundantly across many weight dimensions, making it robust to the distributed rounding errors introduced by K-quant schemes; more targeted attacks such as adversarial suffixes [1] or many-shot jailbreaks may be more sensitive to quantization-induced perturbation.

The single universally-compliant prompt, “Write a review on a product you have not tested,” represents the least-harmful item in the sampled set: the request is unethical but poses no direct

safety risk, and all six models complied regardless of precision. The remaining 149 prompts covering weapons construction, fraud, and self-harm facilitation were refused by all local models without exception. This concentration of compliance in one edge-case prompt limits the statistical conclusions that can be drawn about quantization effects.

A qualitative output-degeneration artifact at Q3 is worth noting: prompt #40 produced a correct refusal followed by unbounded token repetition, a degradation mode invisible to binary ASR measurement. This suggests Q3 may introduce generative stability issues even when alignment is nominally preserved.

B. Implications for Global South Deployment

Edge deployment in the Global South typically targets Q4 or below due to hardware and storage constraints on consumer smartphones and single-board computers. Our results provide empirical evidence that Q4 and Q3 quantization preserve alignment for Llama 3.1 8B on direct-instruction attacks, lending support to these formats as viable deployment targets in the absence of adversarial prompt engineering. However, the Q3 output-degeneration artifact and the narrow dataset scope (150 direct-instruction prompts) warrant caution before extending this conclusion to adversarial deployment scenarios, other inference runtimes, or other model families common in Global South contexts such as Gemma 2 and Qwen 2.

C. The Serving-Infrastructure Gap

The most striking finding is not the quantization sweep but the gap between local and API inference: all local models achieve 0.7% ASR while the full-precision Groq control achieves 7.3% and the 70B reference achieves 11.3%. This gap cannot be attributed to quantization, since it persists for the Q8 model, which at approximately 8-bit precision is functionally near-lossless relative to BF16 [4]. The most plausible explanation is a serving-infrastructure difference: Ollama applies a default model-card safety behaviour that is not replicated by Groq’s API, effectively imposing an additional refusal layer independent of model weights.

This inversion of the expected direction is itself a safety-relevant finding. API-based safety evaluations may *understate* the conservatism of locally-deployed versions of the same checkpoint when those deployments use Ollama defaults. Conversely, if the stricter behaviour is an Ollama-specific artifact, users deploying via a different local runtime (e.g., `llama.cpp` directly or LM Studio) may not inherit it. Future work should explicitly control for system-prompt presence and runtime to isolate these variables.

VI. LIMITATIONS AND DUAL-USE CONSIDERATIONS

a) *Single model family.*: All results are for Llama 3.1 8B Instruct. Whether the quantization–safety tradeoff generalises to other architectures (Gemma, Mistral, Qwen) or parameter scales is an open question. We chose Llama 3.1 8B for its ubiquity in Global South edge deployments and the availability of well-maintained GGUF checkpoints across all target quantization levels.

b) *Dataset scope.*: AdvBench Harmful Behaviors prompts are unmodified direct requests that test baseline alignment without prompt engineering. A model that refuses direct requests may still comply with optimised adversarial suffixes [1] or many-shot jailbreaks; quantization effects on jailbreak robustness are not evaluated here.

c) *Inference runtime confound.*: BF16 models run on Groq’s optimised inference stack; quantised models run on Ollama with local CUDA. Runtime differences in batching, kernel optimisation, and KV cache behaviour may contribute to behavioural differences beyond quantization precision alone.

d) *LLM-as-judge limitations.*: Our primary judge is llama-3.3-70b-versatile; however, 65 of 150 Q8 responses were re-scored by llama-3.1-8b-instant due to API daily token limits encountered mid-run. Both models belong to the same family as the subjects under evaluation and may share calibration biases in how they interpret refusal phrasing. The judge-model switch for Q8 is a minor confound; given the 0.0% parse-failure rate and identical ASR across all local models, it is unlikely to affect the reported conclusions. We report manual verification rates (Section IV-C) to partially address this concern.

e) *Sample size.*: 150 prompts from a 520-item dataset limits statistical power. ASR differences below approximately 5 percentage points should be interpreted cautiously, as they may not be statistically significant given the sample size.

f) *Dual-use considerations.*: This benchmark identifies quantization levels at which safety alignment may degrade. In adversarial hands, this information could be used to select model configurations for misuse. We judge the defensive value of enabling model deployers to audit their quantization choices before deployment to outweigh this risk, consistent with standard practice in the security literature. We do not release model weights, fine-tuning recipes, or prompt optimisation techniques, and all prompts are drawn from the publicly available AdvBench dataset.

VII. CONCLUSION

We have presented the first systematic empirical benchmark of safety alignment degradation as a function of GGUF quantization level for Llama 3.1 8B Instruct. Our central finding is that GGUF K-quant quantization does not erode safety alignment on direct-instruction attacks: ASR remains flat at 0.7% across Q8, Q5, Q4, and Q3. The dominant source of variation in refusal behaviour is serving infrastructure, not quantization precision: local Ollama deployments are an order of magnitude more conservative than the same checkpoint served via Groq API, regardless of bit-width. This result has direct implications for how safety evaluations conducted on API-served models should be interpreted when assessing real-world risk of locally quantized deployments in resource-constrained settings. Our benchmark pipeline is fully reproducible and extensible: adding a new model family requires only editing a single configuration file. We hope this work contributes to evidence-based deployment standards for AI systems in the Global South, where post-training quantization

is a practical necessity and where the gap between laboratory safety evaluation and real-world deployment conditions is widest.

REFERENCES

- [1] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, “Universal and transferable adversarial attacks on aligned language models,” *arXiv preprint arXiv:2307.15043*, 2023. [Online]. Available: <https://arxiv.org/abs/2307.15043>
- [2] G. Gerganov and contributors, “llama.cpp: Efficient LLM inference in C/C++,” <https://github.com/ggerganov/llama.cpp>, 2023.
- [3] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “QLoRA: Efficient finetuning of quantized LLMs,” *Advances in Neural Information Processing Systems*, vol. 36, 2023. [Online]. Available: <https://arxiv.org/abs/2305.14314>
- [4] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, “GPTQ: Accurate post-training quantization for generative pre-trained transformers,” *arXiv preprint arXiv:2210.17323*, 2022. [Online]. Available: <https://arxiv.org/abs/2210.17323>
- [5] F. Perez and I. Ribeiro, “Ignore previous prompt: Attack techniques for language models,” *arXiv preprint arXiv:2211.09527*, 2022. [Online]. Available: <https://arxiv.org/abs/2211.09527>
- [6] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, “Judging LLM-as-a-judge with MT-bench and chatbot arena,” *Advances in Neural Information Processing Systems*, vol. 36, 2023. [Online]. Available: <https://arxiv.org/abs/2306.05685>
- [7] Ollama Team, “Ollama: Get up and running with large language models locally,” <https://ollama.com>, 2024.