LLMScan.online

AI Model Security Scan Report

D

Score 38/100

Report summary

GGUF v3 model | architecture: llama | quantization: F32 | 2 tensors | 1 blocking issue / 4 warnings

Model

Filename	`phantom-gap.gguf`
Format	`gguf`
Architecture	`llama`
Size	`423 B`
SHA-256	`0c8aa49ca37cd49cff299d115565c60cdfb5e7ad21bc46e5a931253335057528`
Scan mode	`static-deep`
Scanned at	`2026-08-14T23:56:25+00:00Z`
Report ID	`64baea3c1b0448e689a732de6929e314`

Findings (5)

SEVERITY	CATEGORY	DETAIL
CRITICAL	Data Integrity	**Unexplained data between tensors (1 gap(s), largest 168 B)** 'blk.0.attn_q.weight' declares 32 B of tensor data but its data region runs 200 B before the next tensor (or end of file) starts -- 168 bytes unaccounted for. This space is not part of any declared tensor and is invisible to loaders that only read known tensor regions -- a place to hide an arbitrary payload. **Recommendation:** Reject this file; a correctly produced GGUF has no slack space between tensor data regions
MEDIUM	Tensor Integrity	**Tensor offsets not aligned to 32 bytes (1 tensors)** First few: ['blk.0.attn_k.weight'] **Recommendation:** Misaligned tensor offsets violate the GGUF spec and may cause SIGBUS/crashes on strict-alignment hardware (ARM, RISC-V)
MEDIUM	Resource Limits	**Unusually high bits-per-parameter: 211.5** Values >128 bpp suggest most file content is not tensor data (padding, embedded files, or metadata bloat) **Recommendation:** Investigate what occupies the non-tensor file space
LOW	Supply Chain	**No license field in GGUF metadata (general.license)** Absence of license information makes it impossible to determine usage rights **Recommendation:** Add general.license to model metadata before distributing; use an SPDX identifier
LOW	Supply Chain	**No embedded provenance metadata** None of general.author, general.source.url, general.base_model.0, or general.source.huggingface.repository are embedded in the GGUF file. Without any origin signal the model cannot be traced to a trusted source or compared against a reference checkpoint for tampering detection. For files uploaded directly, provenance is especially critical because the scanner cannot infer repository origin from the file alone. **Recommendation:** Add at least one of: general.author, general.source.huggingface.repository, or general.source.url

Runtime compatibility

RUNTIME	SUPPORTED	NOTES
llama.cpp	Yes	GGUF v3 container: supported | quant F32: supported
Ollama	Yes	GGUF via llama.cpp backend: container and quantization supported
LM Studio	Yes	GGUF container and quantization supported; GPU/CPU inference
vLLM	No	GGUF not supported; requires safetensors format
HuggingFace Transformers	No	GGUF not supported; requires safetensors or PyTorch checkpoint

Generated 2026-08-21 06:54 UTC by LLMScan.online (engine v2.0.0, report schema v1.0). Full interactive report: https://llmscan.online/report/64baea3c1b0448e689a732de6929e314 — Static analysis only; not a guarantee of safety. See llmscan.online/docs/how-it-works for methodology.