GGUF File Won't Load: Format and Compatibility Fixes

February 18, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** Most common: your llama.cpp or Ollama is too old for the GGUF version. Update to latest. Second most common: corrupted or partial download — check the file size against HuggingFace and re-download if it doesn't match. If you accidentally grabbed a GPTQ or safetensors file instead of GGUF, you need a different file. If it loads partway then crashes, the model is too big for your RAM+VRAM — try a smaller quantization.

📚 **More on this topic:** Model Formats Explained: GGUF, GPTQ, AWQ, EXL2 · Quantization Explained · Run Your First Local LLM · Planning Tool

You downloaded a GGUF file. It should just load. It doesn't. Here's every reason why and how to fix each one.

Quick Diagnostic

Error Message	Jump To
`invalid magic number` or `not a GGUF file`	Wrong File Format
`unsupported GGUF version`	Version Mismatch
`failed to load model` with no details	Corrupted Download
`unexpected EOF` or truncated data	Corrupted Download or Split Files
`imatrix` in the error	imatrix Quants
Loads partway, then crashes or OOM	Too Big for Memory
Ollama says `invalid model` on import	Ollama Import

---

1. GGUF Version Mismatch

Error:

```
error: unsupported GGUF version: 3
```

Cause: GGUF is a versioned format. Newer models use GGUF v3, but older builds of llama.cpp or Ollama only understand v2 or earlier. Quantization tools on HuggingFace always use the latest version.

Fix: Update your tools to latest:

```
# llama.cpp
cd llama.cpp && git pull
cmake -B build -DGGML_CUDA=ON && cmake --build build --config Release -j $(nproc)

# Ollama
curl -fsSL https://ollama.com/install.sh | sh
```

This is the most common GGUF loading failure. If the file loaded fine a month ago but a newly downloaded model won't — your tools are behind.

---

2. Corrupted Download

Error: `failed to load model`, segfault on load, or gibberish output after loading.

Cause: GGUF files are 2-50GB. Large downloads fail silently — your browser shows "complete" but the file is truncated. Network interruptions, disk space running out mid-download, and CDN issues all cause this.

Fix:

1. **Check the file size.** Compare your local file against what HuggingFace lists:

```
ls -lh model-Q4_K_M.gguf
# Compare with the size shown on the HuggingFace repo page
```

If your file is smaller than listed, it's truncated. Re-download.

1. **Verify the sha256 hash** if the repo provides one:

```
sha256sum model-Q4_K_M.gguf
# Compare with the hash in the repo's README or .sha256 file
```

1. **Use a download manager** for large files. `wget -c` supports resumable downloads:

```
wget -c https://huggingface.co/user/repo/resolve/main/model-Q4_K_M.gguf
```

In Ollama, a corrupted pull is rare but possible. Fix: `ollama rm model:tag` then `ollama pull model:tag` again.

---

3. Split Files — Missing Parts

Error: `unexpected EOF`, file seems too small, or model loads with missing layers.

Cause: Large GGUF files are sometimes split into multiple parts for hosting. You downloaded one part but not all of them.

Split files look like:

```
model-Q4_K_M.gguf-split-a-of-b
model-Q4_K_M.gguf-split-b-of-b
```

Or sometimes:

```
model-Q4_K_M-00001-of-00003.gguf
model-Q4_K_M-00002-of-00003.gguf
model-Q4_K_M-00003-of-00003.gguf
```

Fix: Download every split file into the same directory. llama.cpp reads them automatically when you point it at the first file:

```
llama-cli -m model-Q4_K_M-00001-of-00003.gguf -p "Hello"
```

Some repos also have a non-split version. Check if there's a single file option before downloading 3+ parts.

---

4. Wrong File Format

Error:

```
error: invalid magic number: 0x46545347 (expected 0x46475547)
```

Or: `not a GGUF file`.

Cause: You downloaded a file that isn't GGUF. Common mistakes:

What You Downloaded	How to Tell
GPTQ model	File ends in `.safetensors`, repo says "GPTQ"
AWQ model	Repo says "AWQ" in the name
Raw safetensors	Multiple `.safetensors` files, no `.gguf`
EXL2 model	Repo says "EXL2", needs ExLlamaV2
Older GGML format	File ends in `.bin` (pre-GGUF legacy format)

Fix: Find the GGUF version. Look for repos by these quantizers:

- **bartowski** — high-quality GGUF quants for most popular models
- **mradermacher** — wide coverage, imatrix quants
- **TheBloke** — older models (mostly Llama 2 era, still valid)
- **Mungert** — Unsloth GGUF quants

Search HuggingFace for `[model name] GGUF` and filter by the quantization level you want (Q4_K_M is the sweet spot for most users).

---

5. Model Too Big for Memory

Error: Loading starts, progress bar moves, then crash — OOM, segfault, or the process gets killed.

Cause: The model needs more RAM + VRAM than you have. A 70B Q4_K_M file is ~40GB — you need that much memory available just to load it, plus overhead for the KV cache during inference.

Fix:

- **Use a smaller quantization.** Q4_K_M → Q3_K_M saves ~25% memory. Q2_K saves more but quality drops noticeably. See the quantization guide for the full tradeoff table.
- **Use a smaller model.** If 70B won't fit, try 32B. If 32B won't fit, try 14B. The quality gap between sizes is real, but a model that loads beats one that doesn't.
- **Partial GPU offloading.** Load some layers on GPU, the rest on CPU:

```
# llama.cpp — offload 20 layers to GPU, rest stays in RAM
llama-cli -m model.gguf -ngl 20

# Ollama — set in Modelfile
PARAMETER num_gpu 20
```

Check our VRAM requirements guide for exact memory needs by model and quant level.

---

6. Ollama Can't Import Custom GGUF

Error: `invalid model format` or Ollama ignores your GGUF file.

Cause: Ollama doesn't load raw GGUF files directly. You need a Modelfile that tells Ollama where the file is and how to format prompts.

Fix: Create a Modelfile:

```
FROM ./model-Q4_K_M.gguf

TEMPLATE """<|im_start|>system
{{ .System }}<|im_end|>
```

```
<|im_start|>user
{{ .Prompt }}<|im_end|>
<|im_start|>assistant
"""

PARAMETER stop "<|im_end|>"
```

Then import:

```
ollama create mymodel -f Modelfile
ollama run mymodel
```

The TEMPLATE must match the model's chat format. The example above is ChatML (works for Qwen, many others). Llama 3 models need a different template. If the model loads but gives incoherent output, the template is wrong.

---

7. imatrix Quants

Error: Mentions `imatrix`, `importance matrix`, or fails on a quant level that should be standard (like Q4_K_M).

Cause: imatrix (importance matrix) quantization is a newer method that produces better quality at low bit rates. It requires a recent version of llama.cpp to load. Older builds don't recognize the format.

Fix: Same as section 1 — update llama.cpp or Ollama to the latest version. imatrix quants have been supported since mid-2025, so anything reasonably current handles them.

imatrix quants are identified by `imatrix` or `im` in the filename (e.g., `model-Q4_K_M-imatrix.gguf`). They load and run identically to standard quants once your tools are current.

Bottom Line

GGUF loading failures come down to three things: your tools are outdated, your file is damaged, or you grabbed the wrong format. Update first, check file size second, verify you actually have a GGUF third. That catches 90% of issues.

For the remaining 10%: the model is too big (use a smaller quant), you're missing split file parts, or your Ollama Modelfile needs the right chat template.

Get notified when we publish new guides.

Subscribe — free, no spam

Source: https://insiderllm.com/guides/gguf-file-wont-load-fix/