Quantization Models (GGML Vs GGUF)

**What is Quantization?**

**Quantization** is the process of reducing the precision of a neural network’s weights **(e.g., from 32-bit floating point to 8-bit integer)** to: ✅ Reduce **model size**  
✅ Increase **inference speed**  
✅ Lower **RAM & VRAM usage**  
✅ Make **LLMs run on CPUs instead of GPUs**

**Types of Quantization**

1. **Post-training Quantization (PTQ)**
   * Applied **after training** a model.
   * Example: GGML, GGUF.
2. **Quantization-aware Training (QAT)**
   * Applied **during training** for better accuracy.

**What is GGML?**

**GGML (Georgi Gerganov Machine Learning)** is a format optimized for **running LLMs on CPUs efficiently**.

✅ **Supports 4-bit, 5-bit, 8-bit quantization**  
✅ Uses **matrix multiplication optimized for CPUs**  
✅ Used in **Llama.cpp**, ctransformers, llama-cpp-python  
✅ Works with **Llama-2, Mistral, GPT-4-All** models

Example:

llama-2-7b-chat.ggmlv3.q4\_K\_M.bin

**GGUF (GGML Unified Format)** is an improved version of GGML.

✅ **Standardized format** for quantized models  
✅ More **efficient metadata storage**  
✅ Backward-compatible with GGML  
✅ **Smaller file size & faster inference**

Example:

llama-2-7b-chat.Q4\_K.gguf

**GGML vs. GGUF**

| **Feature** | **GGML** | **GGUF** |
| --- | --- | --- |
| **Efficiency** | Moderate | High |
| **File Size** | Larger | Smaller |
| **Model Support** | Llama-2, Mistral, Falcon | All GGML models + more |
| **Backward Compatibility** | No | Yes |
| **Recommended for New Models?** | ❌ No | ✅ Yes |

📌 **Takeaway:** GGUF is the future! Most **new models are released in GGUF format**.

Practical:

pip install ctransformers

Download **Llama-2 GGUF model** from Hugging Face:

wget <https://huggingface.co/TheBloke/Llama-2-7B-GGUF/resolve/main/llama-2-7b-chat.Q4_K.gguf>

from ctransformers import AutoModelForCausalLM

# Load the GGUF model

model = AutoModelForCausalLM.from\_pretrained(

"llama-2-7b-chat.Q4\_K.gguf",

model\_type="llama",

gpu\_layers=50 # Adjust for GPU acceleration

)

# Run inference

response = model("Tell me a joke about AI.")

print(response)