

NAME

cmgenerate - run, inspect, score and convert language models

SYNOPSIS

cmgenerate [*OPTION*]... *MODEL* [*PROMPT*]

DESCRIPTION

cmgenerate loads a transformer language model and generates text from *PROMPT*, or with a mode option inspects, scores, traces or re-quantises it. Without *MODEL*, a demonstration runs on stand-in weights.

MODEL is a GGUF file, a safetensors file, a llama2.c **.bin** file, or a Hugging Face directory containing **config.json**. For a Hugging Face cache, give the snapshot directory.

If the vocabulary carries a known chat template (ChatML, Llama 3, Apertus, Muse Glimmer, Gemma 4), the prompt is wrapped as one user turn and generation stops at the end of the model's turn. **--raw** disables the template.

OPTIONS**Generation**

-n, --steps *N*

Tokens to generate. Default: until the end of the turn, bounded by the context; 64 without a chat template.

-c, --context *N*

Context length to allocate, clamped to the model's maximum. Default: 4096. The KV cache is sized for this length.

--window *N*

Sliding-window attention: layers without a trained window attend to the last *N* tokens and the first **--sinks** tokens, stored in a ring buffer of that size. Tokens outside the window are not recalled.

--sinks *N*

Initial tokens kept visible with **--window**. Default: 4.

-t, --temperature *F*

Sampling temperature; 0 is greedy. Default: 1.0.

-k, --top-k *N*

Top-k sampling; 0 disables. Default: 40.

-p, --top-p *F*

Nucleus sampling; 1 disables. Default: 0.9.

-s, --seed *N*

Random seed. Default: derived from the time.

--raw

No chat template; continue *PROMPT* as text.

--effort *LEVEL*

How hard a reasoning model is asked to think: **none**, **low**, **medium** or **high**. Kolibri 1 gets the matching "# Reasoning effort" sentence of its chat template in the system turn (default high); with **none** the answer also opens with an empty thought, as it does for models whose template opens thinking (Qwen3.5, Qwen3.8). Muse Glimmer gets its "Reasoning strength", lowest for **none**. Other models are unchanged.

--speculate *N*

Speculative decoding with up to *N* draft tokens per step; 0 disables. Drafts come from the model's multi-token-prediction (MTP) layer if it has one (at most 3 per step, each kept only while its probability is at least 0.5), otherwise from prompt lookup (trigram matches in the prompt and output). The drafts are verified in one batched forward pass. Default: enabled when a batch of tokens costs little more than one token on the chosen device (GPU with batched kernels, or a dense model on the CPU).

--no-mtp

Draft by prompt lookup even if the model has an MTP layer. The MTP tensors of a GGUF are then not loaded.

--mtp-from *PATH*

Load the MTP layer (**mtp.*** tensors) from a safetensors file or Hugging Face directory, for a model file that lacks it. For Qwen3.8-27B these tensors are in the last shard.

Modes**--repl**

Interactive session; the model stays loaded between questions.

--serve [*ADDR*]

HTTP server. Default address: 127.0.0.1:8080.

--tokens

Print the tokenisation of the prompt and exit.

--score

Print the model's perplexity on the prompt and exit.

--inspect

Print the model's metadata and exit. With *PROMPT*, list the tensors whose names contain it.

--trace

Run the prompt, with the vocabulary's BOS token, one token at a time and print the first values and the sum of the residual stream after each layer, in the format of llama.cpp's **llama-eval-callback**.

--experts

Report how often each expert of a mixture-of-experts model was routed to.

--quantise *FMT*

Write the model as *FMT* (Q4_K, Q5_K, Q6_K, Q8_0; BITCOS for ternary models) and exit.

-o, --out *PATH*

Output file of **--quantise**. Default: beside the model.

Placement**--gpu**

Place as many weight tensors on the GPU as fit, dense tensors first; the rest stays in host memory and runs on the CPU. On Vulkan, expert tensors that do not fit are kept in pinned host memory and served through an expert cache in VRAM.

--gpu-backend *API*

auto, cuda, metal or vulkan. Implies **--gpu**.

--gpu-device *N*

Vulkan physical device index.

--gpu-as *FMT*

Quantise tensors to *FMT* while uploading them, if the model does not fit as stored. Formats the GPU kernels cannot read (FP8, NVFP4) are always converted.

--cpu-as *FMT*

Quantise weight matrices to *FMT* (Q4_K, Q5_K, Q6_K, Q8_0) at load, one tensor at a time. Matrices already as narrow as *FMT* or narrower are kept; norms and vectors stay in f32. CPU decode is memory-bandwidth-bound, so this raises its throughput roughly in proportion to the size reduction. Q4_K encodes about 10 times slower than Q8_0.

--kv-format *FMT*

KV cache format: q8 (Q8_0 blocks on the host, int8 with per-32 scales on Vulkan) or f32. Default: q8.

--lora *PATH*

Merge an adapter written by **cmfinetune**(1) into the weights at load.

--lora-as *FMT*

Format of the matrices the adapter changes. Default: Q8_0.

ENVIRONMENT**CMGENERATE_THREADS**

Worker threads. Default: chosen from the CPU topology.

CMGENERATE_KERNELS

Limit the SIMD kernels, for comparison: **scalar** (none), **float** (no integer dot products), **avx2** (AVX2 on a CPU with AVX-512).

CMGENERATE_BATCH

Prompt tokens read in one pass. Default: 512 on the CPU, 64 or the backend's own choice (128 for Vulkan's tensor cores) with a GPU, 64 when **CMGENERATE_PANEL_TOKENS** is 0.

CMGENERATE_PANEL_TOKENS

A CPU batch of at least this many tokens multiplies each matrix as 16-row panels against tiles of tokens, decoded once a call: as 8-bit codes for Q8_0, Q4_0, Q4_1, Q5_0, Q4_K, Q5_K and Q6_K with AVX-512 VNNI or ARM dot-product instructions, and for all but Q8_0 with AVX2 (activations rounded to 8 bits per block, as token-by-token decoding does); otherwise as f32 (activations kept f32). About 2 to 4 times faster than a dot product per row and token at 512 tokens. 0 turns it off. Default: 24.

CMGENERATE_QOS

macOS: how many threads request performance cores (**0** or **all**).

CMGENERATE_SPIN

Spin iterations before an idle worker sleeps.

CMGENERATE_GPU_MEMORY

Cap on GPU memory used, in GiB.

CMGENERATE_EXPERT_CACHE

Fraction (0..1) of the VRAM left for expert tensors that is used as expert cache.

CMGENERATE_EXPERT_FETCH

Fixed fraction (0..1) of expert-cache misses copied to the GPU instead of computed on the CPU.

CMGENERATE_TRANSFER_QUEUE=0

Do not use a separate Vulkan transfer queue; disables the expert cache.

CMGENERATE_ANCHORS=0

Do not keep recurrent-state snapshots for rewinds.

CMGENERATE_METAL_PROFILE

Metal: submit each dispatch in its own command buffer and report GPU time per kernel. Slows execution.

CMGENERATE_VULKAN_PROFILE=1

Vulkan: per-operation GPU timestamps as JSON lines on standard error.

FILES

~/.cache/cmgenerate/

Vulkan pipeline cache, one file per device, driver version and shader build.

EXAMPLES

```
cmgenerate model.gguf "Explain Rust ownership." -n 128 -t 0
```

```
cmgenerate model.gguf --gpu --gpu-as Q4_K "Hello"
```

```
cmgenerate model.gguf --cpu-as Q8_0 --repl
```

```
cmgenerate model.gguf --inspect attn_q
```

```
cmgenerate model.safetensors --quantise Q4_K -o model.q4_k.gguf
```

SEE ALSO

cmagent(1), **cmfinetune(1)**