The Open-Weights

LLM Landscape

February 2026

Table of Contents

1. Introduction

The open-weights LLM ecosystem has grown explosively since the release of Meta's LLaMA in early 2023. Today, practitioners can choose from dozens of model families spanning 1B to 405B+ parameters, released under licenses ranging from fully permissive (Apache 2.0) to restricted commercial use. These models power everything from local chatbots on consumer hardware to production inference systems serving millions of users. This report surveys the major open-weights model families, their architectural characteristics, licensing terms, and practical deployment considerations with an emphasis on running them on small-GPU systems.

2. What "Open Weights" Means

An important distinction exists between open-source and open-weights models. A truly open-source model would release not only the trained weights but also the complete training data, data processing pipeline, training code, and hyperparameters—everything needed to reproduce the model from scratch. Most "open" LLMs release only the weights and sometimes the inference code, making them open-weights rather than open-source. A few projects (notably OLMo from AI2 and LLM360) do publish full training artifacts and qualify as genuinely open-source.

The licensing landscape adds further nuance. Apache 2.0 and MIT licenses are fully permissive for commercial use. Meta's Llama license permits commercial use below a 700-million monthly active user threshold. Some models (like certain DeepSeek releases) use licenses that restrict specific use cases. Practitioners must evaluate licensing terms against their deployment scenario.

3. Major Model Families

3.1 Llama (Meta)

Meta's Llama family is arguably the most influential open-weights release. Llama 3.1 introduced 8B, 70B, and 405B parameter variants with a 128K context window and multilingual support across eight languages. Llama 3.2 extended the family to include 1B and 3B lightweight models for edge deployment, plus multimodal 11B and 90B vision-language models. Llama 4 introduced a mixture-of-experts architecture. The dense models use grouped-query attention (GQA), RoPE positional embeddings, and SwiGLU activations—architectural choices that have become standard across the ecosystem. Licensed under Meta's community license (commercial use permitted with MAU limits).

3.2 Mistral / Mixtral (Mistral AI)

Mistral AI has released several notable models. Mistral 7B introduced sliding window attention and demonstrated that a well-trained 7B model could match or exceed Llama 2 13B. Mixtral 8x7B was a landmark sparse mixture-of-experts model: 47B total parameters but only 13B active per forward pass, achieving near-70B-class quality at a fraction of the compute cost. Mistral also released Mistral Large, Codestral (for code), and Mistral Small under various

licenses. The smaller models (7B, Mixtral) are particularly well-suited to small-GPU deployments.

3.3 Qwen (Alibaba Cloud)

Alibaba's Qwen family has emerged as one of the strongest multilingual model lines. Qwen 2.5 spans 0.5B to 72B parameters (and a 32B "sweet spot" variant), with specialized variants for code (Qwen-Coder) and mathematics (Qwen-Math). The models feature strong performance on both English and Chinese benchmarks and are released under Apache 2.0. The smaller Qwen models (0.5B–7B) are excellent candidates for draft models in speculative decoding setups.

3.4 Gemma (Google DeepMind)

Google's Gemma family provides high-quality small models. Gemma 2 includes 2B, 9B, and 27B variants, where the 27B model uses an interleaved local-global attention pattern that reduces KV-cache size. Gemma 3 introduced a 1B model and multimodal capabilities. Released under a permissive license with minimal restrictions. The 2B and 9B models are particularly efficient on single small GPUs and make excellent local inference models.

3.5 DeepSeek

DeepSeek has produced models notable for their efficiency and reasoning capability. DeepSeek-V2 introduced Multi-head Latent Attention (MLA), which compresses the KV-cache by projecting keys and values into a lower-dimensional latent space—a significant memory optimization for inference. DeepSeek-V3 and DeepSeek-R1 pushed further with mixture-of-experts architectures and reinforcement-learning-based reasoning training. DeepSeek's MLA innovation is particularly relevant for small-GPU deployments where KV-cache memory is a binding constraint.

3.6 Other Notable Families

- **Phi (Microsoft):** Small but capable models (1.5B–14B) trained on synthetic data and curated sources. Phi-4 achieves remarkable quality for its size, making it ideal for single-GPU or edge deployment.
- **Yi (01.AI):** 34B and 9B models with strong multilingual performance and 200K context support.
- **Falcon (TII):** Early open-weights leader (40B, 180B) using multi-query attention. The 180B model was one of the first very large open-weights releases.
- **OLMo (AI2):** Truly open-source with full training data and code. OLMo 2 7B and 13B models with competitive performance.
- **Command R (Cohere):** 35B and 104B models optimized for RAG and tool use, with strong long-context performance.

4. Choosing a Model for Small-GPU Inference

4.1 Parameter Count and Quantization

Model Size	FP16	INT8	INT4	Min GPU (INT4)	Quality Tier
1–3B	2–6 GB	1–3 GB	0.5–1.5 GB	1x 4 GB	Basic tasks
7–9B	14–18 GB	7–9 GB	4–5 GB	1x 8 GB	Good general use
27–35B	54–70 GB	27–35 GB	14–18 GB	2x 12 GB	Strong
70B	140 GB	70 GB	35 GB	2x 24 GB	Near frontier

4.2 Architectural Features That Matter

- **Grouped-Query Attention (GQA):** Reduces KV-cache size by sharing key-value heads across query heads. Most modern models use this. Llama 3, Mistral, Qwen 2.5, and Gemma 2 all employ GQA.
- **Multi-head Latent Attention (MLA):** DeepSeek's innovation compresses KV-cache further through learned projections. Extremely beneficial for memory-constrained systems.
- **Mixture of Experts (MoE):** Allows larger effective model size with less active compute, but requires memory for all experts. Best when total memory across chips is ample but per-chip compute is limited.
- **Sliding Window Attention:** Reduces memory for long sequences by limiting attention to a local window. Mistral's approach trades off some long-range capability for efficiency.

4.3 Practical Selection Criteria

Beyond raw benchmark scores, practitioners should consider quantization compatibility (some models quantize more gracefully than others; models trained with quantization awareness generally perform better at INT4), community tooling support (availability of GGUF, GPTQ, AWQ, and EXL2 quantized versions on repositories like Hugging Face), the model's context length requirements versus the available KV-cache budget, and whether fine-tuned variants exist for the target domain (instruction-tuned, code-tuned, or domain-specific fine-tunes can outperform a larger general-purpose model).

5. Distribution Formats and Serving

5.1 Quantization Formats

- **GGUF:** The format used by llama.cpp. Supports 2–8 bit quantization with fine-grained per-block scaling. Widely available on Hugging Face. Ideal for CPU+GPU hybrid inference on consumer hardware.

- **GPTQ:** GPU-focused format using per-group quantization with calibration. Well-supported by vLLM, TGI, and AutoGPTQ. Best for pure GPU serving.
- **AWQ:** Activation-aware weight quantization that preserves salient weights at higher precision. Often achieves better quality than GPTQ at the same bit width.
- **EXL2:** ExLlamaV2's format supporting mixed bit widths within a single model. Allows fine-grained control over the quality-size trade-off.

5.2 Serving Frameworks

vLLM is the de facto standard for serving open-weights models, supporting most major model architectures with PagedAttention, continuous batching, and tensor/pipeline parallelism. For consumer hardware, llama.cpp offers excellent flexibility with CPU/GPU splitting and network-distributed inference via its RPC backend. TensorRT-LLM provides the lowest latency on NVIDIA hardware but requires model-specific compilation. For peer-to-peer setups, exo enables distributed inference across heterogeneous consumer devices.

6. Conclusion

The open-weights LLM landscape offers a rich set of options for every hardware budget and use case. For small-GPU deployments, the 7–9B class (Llama 3.1 8B, Qwen 2.5 7B, Gemma 2 9B, Mistral 7B) provides the best quality-to-resource ratio when quantized to INT4 or INT8. For higher quality requirements, the 27–35B class (Gemma 2 27B, Qwen 2.5 32B) is accessible on two consumer GPUs. The 70B class remains the sweet spot for near-frontier quality and is practical on clusters of 4–8 small GPUs with 4-bit quantization. The ecosystem continues to improve rapidly, with each new release pushing the quality-per-parameter frontier forward.