AI Reference Guide

Common Terms & Concepts

- **Model**: A mathematical system trained on data to perform tasks like generating text, translating languages, or answering questions. In AI, a language model learns patterns in language so it can predict what comes next in a sentence.
- **Training / Fine-Tuning**:
  - **Training** is the initial process of teaching an AI model by feeding it massive amounts of data so it can learn patterns.
  - **Fine-tuning** happens after initial training, it's when a model is further adjusted using specialised or smaller datasets to improve performance on specific tasks (e.g., medical chatbots, legal assistants).
- **Inference**: The process of using a trained model to generate predictions or responses. For example, when you ask ChatGPT a question, it performs inference to produce the answer.
- **Parameter Count**: Number of trainable weights in a model (e.g., 7 B = 7 billion). These parameters determine how well the model can learn and generalise from data.
- **Token**: A text piece (word or subword) the model processes sequentially. For example, "cat" might be one token, while "predictable" could be broken into several subword tokens.
- **Context Window (Token Limit)**: Max length of input + output the model can handle at once (e.g., 128 k tokens). Longer windows allow models to handle bigger documents or longer conversations.
- **Accuracy / Benchmarks**: Performance on standardised tasks like MMLU (Massive Multitask Language Understanding), HellaSwag (commonsense reasoning), HumanEval (code generation), etc. Higher scores indicate stronger real-world understanding.
- **Quantization**: A technique to reduce model size and speed up performance by using lower precision numbers (e.g., INT8 instead of FP16), making it easier to run large models on smaller hardware.
- **RAG**: RAG (Retrieval-Augmented Generation) enhances an LLM by retrieving relevant documents or chunks from your knowledge base and injecting them into the prompt before generating a response. RAG automates the retrieval of relevant context.
- **MCP**: MCP (Model Context Protocol) is an open protocol that standardises how applications provide context to LLMs. MCP gives you a repeatable and structured format to inject context into the model.

Open-Source LLM Comparison

Model	Params	Benchmarks	Possible Hardware
LLaMA 3.1 (Meta)	8B / 70B / 405B	MMLU: 87.3% (405B), 82.0% (70B), 69.4% (8B); HumanEval: 89.0% (405B)	**8B**: Mac Mini M4 (24GB), RTX 4070 **70B**: Mac Studio M3 Ultra (128GB), 2x RTX 4090 **405B**: Enterprise GPU clusters
Qwen 3 (Alibaba)	1B – 235B (22B active MoE)	MMLU: 64.3% (8B), strong multilingual, coding excellence	**8B**: Mac Mini M4 (24GB), RTX 4060 Ti **22B**: Mac Studio M3 Max (64GB), RTX 4080 **235B**: Mac Studio M3 Ultra (512GB), Multi-GPU workstation
DeepSeek-V3 (DeepSeek)	671B total / 37B active	MMLU: 88.5%, MATH: 90.2%, HumanEval: 82.6%, state-of-the-art performance	**37B active**: Mac Studio M3 Ultra (128GB), 2x RTX 4090Full model: Enterprise clusters
Mistral Large 2	123B (dense)	MMLU: ~84.0%, strong instruction following, 128K context	Mac Studio M3 Ultra (256GB), 4x RTX 4090, H100
DeepSeek-R1 (DeepSeek)	671B total / 37B active	MMLU: 90.8%, MATH: 97.3%, reasoning specialist competitive with OpenAI o1	**37B active**: Mac Studio M3 Ultra (128GB), 2x RTX 4090Full model: Enterprise clusters
Kimi K2 (Moonshot AI)	1T total / 32B active (MoE)	LiveCodeBench: 53.7%, SWE-bench: 65.8%, GPT-4-class performance	**32B active**: Mac Studio M3 Ultra (128GB), RTX 4090 Full model: Multi-GPU clusters

GPU Size & Hardware Requirements

Model Size	Precision	Memory Needed	Mac Options	PC/GPU Options	Performance Notes
3–7B	INT4	~3.5–4 GB	Mac Mini M4 (16GB) MacBook Pro M4 (16GB)	RTX 3060, RTX 4060	Budget-friendly, excellent Mac performance
3–7B	FP16	~14–16 GB	Mac Mini M4 (24GB) Mac Studio M3 Max (32GB)	RTX 4090, RTX 5090	High-end consumer setup
8–13B	INT4	~6.5–7 GB	Mac Mini M4 (32GB) Mac Studio M3 Max (64GB)	RTX 4070, RTX 5070	Good balance of cost/performance
8–13B	FP16	~26–28 GB	Mac Studio M3 Max (64GB) Mac Studio M3 Ultra (128GB)	2x RTX 4090, RTX 5090	Professional workstation level
20–30B	INT4	~15–20 GB	Mac Studio M3 Max (64GB) Mac Studio M3 Ultra (128GB)	RTX 4090, A6000	High-end workstation
20–30B	FP16	~60–65 GB	Mac Studio M3 Ultra (128GB) Mac Studio M3 Ultra (256GB)	4x RTX 4090, A6000, H100	High-memory workstation/server
65–70B	INT4	~35–42 GB	Mac Studio M3 Ultra (128GB) Mac Studio M3 Ultra (256GB)	A6000 (48GB), H100	Great Mac performance at this size
65–70B	FP16	~140–150 GB	Mac Studio M3 Ultra (256GB) Mac Studio M3 Ultra (512GB)	4x A100, 2x H100	Mac now viable for 70B FP16!
120–200B	INT4	~60–100 GB	Mac Studio M3 Ultra (256GB) Mac Studio M3 Ultra (512GB)	2x H100, 4x A6000	Mac competitive for large models
405B+	FP16	~200+ GB	Mac Studio M3 Ultra (512GB) *for smaller 405B variants*	4x H100 (80GB)	Enterprise clusters preferred
405B+	FP16	~800+ GB	*N/A*	8x H100 (80GB)	Enterprise clusters / Cloud GPUs

Note: Quantization (e.g., INT4) can reduce memory needs dramatically (e.g., 70B INT4 can fit on a 24 GB GPU).

FAQs

1. What's the difference between an LLM and general AI?

A **Large Language Model (LLM)** is a type of AI trained to understand and generate human language. It excels at tasks like writing, summarising, and answering questions.

Artificial Intelligence (AI) is a broader field that includes LLMs but also covers vision, robotics, decision-making systems, etc.

2. Do LLMs think or understand like humans?

No. LLMs generate text based on statistical patterns learned from massive datasets. They don't have **intentions**, **self-awareness**, or **true understanding**—but they often **appear** intelligent due to the quality of their training data.

3. How do I connect to my organisations documents and knowledge?

You can connect to your organisation's documents by combining **Retrieval-Augmented Generation (RAG)**—which retrieves relevant internal content at runtime—with the **Model–Context–Prompt (MCP)** protocol, which cleanly defines the model used, the context retrieved, and the prompt given. This setup enables grounded, auditable answers from local AI systems without sending data to the cloud.

4. Where do you find datasets for training?

Training datasets are often sourced from public internet data such as **webpages**, **books**, **scientific articles**, **GitHub code**, and **forums**. Common repositories include **Hugging Face**, **The Pile**, **Common Crawl**, and **OpenWebText**. For fine-tuning, organisations may use curated internal data or domain-specific corpora.

5. Why are some LLMs good at some tasks and not others?

Performance varies based on a model's **training data**, **architecture**, and **number of parameters**. LLMs trained on diverse, high-quality datasets tend to generalise well. Others may specialise—for example, coding models are often fine-tuned on code. Larger models typically perform better but can be less efficient or harder to deploy.