

Hitchhikers Guide to AI

Comprehensive State of AI Capabilities, Ethics, Access, Integration: Featuring Different Models Compared Against Grok 4 as the Benchmark

Prompted by George Lambert

Generated by Grok

December 12, 2025

Table of Contents

- [1. Executive Summary](#)
- [2. Introduction](#)
- [3. Understanding Tokens and AI Billing Practices](#)
 - [What Are Tokens?](#)
 - [Billing Practices Across Models](#)
- [4. Model Comparison: Grok 4 vs. Claude 4.5 and Others](#)
 - [Capabilities and Performance](#)
 - [Benchmarks](#)
 - [Expanded Comparison with Other Models](#)
 - [Paid/Proprietary Models](#)
 - [Free Open-Source Models on Single GPUs](#)
 - [NVIDIA GPU](#)
 - [Intel GPU](#)
 - [AMD GPU](#)
 - [Older GPUs \(Self-Hosting Compatibility\)](#)
 - [NVIDIA Jetson Orin Family Hardware Comparison](#)
 - [Grok 4 vs. Llama 4](#)
 - [Grok 4 vs. Mistral Nemo](#)
 - [Grok 4 vs. Closest Competitors](#)
 - [Strengths and Weaknesses](#)
- [5. AI Ethics in 2025 Models](#)
 - [Key Ethical Frameworks](#)
 - [Bias and Fairness](#)
 - [Privacy and Security](#)
 - [Grok 4 vs. Claude 4.5 Ethics](#)
 - [Expanded Ethics Across Other Models](#)
- [6. Access and Pricing Summary](#)
 - [Grok 4 Access and Costs](#)
 - [Claude 4.5 Access and Costs](#)
 - [Pricing Comparison Table](#)
 - [Free Models Access](#)

- [Hardware Pricing and Specs](#)
- [7. Available Extensions and Integrations](#)
 - [Grok 4 Extensions](#)
 - [Claude 4.5 Extensions](#)
- [8. Integration Comparison: CLI, VSCode, Cursor](#)
 - [CLI Integration](#)
 - [VSCode Integration](#)
 - [Cursor Integration](#)
 - [Comparison Table](#)
 - [Expanded Integration for Free Models](#)
 - [Hybrid Grok API/SuperGrok Task Splitting](#)
- [9. Leveraging Long Dialogs for Efficiency in Grok 4](#)
 - [Summary Techniques](#)
 - [Cost Reduction Strategies](#)
 - [Improved Results](#)
- [10. Advantages and Disadvantages of Grok 4](#)
- [11. Model Cheat-Sheets](#)
 - [Grok 4 Prompting Cheat-Sheet](#)
 - [Claude 4.5 Prompting Cheat-Sheet](#)
 - [GPT-5 Prompting Cheat-Sheet](#)
 - [Gemini 3 Prompting Cheat-Sheet](#)
 - [Llama 4 Prompting Cheat-Sheet](#)
 - [Mistral Nemo Prompting Cheat-Sheet](#)
 - [DeepSeek R1 Prompting Cheat-Sheet](#)
 - [Phi-3 Prompting Cheat-Sheet](#)
 - [Gemma 2 Prompting Cheat-Sheet](#)
 - [Qwen2.5-Coder Prompting Cheat-Sheet](#)
 - [StarCoder2 Prompting Cheat-Sheet](#)
- [12. Conclusion](#)
- [Footnotes](#)
- [References](#)

1. Executive Summary

In the rapidly evolving AI landscape of December 2025, Grok 4 (xAI) serves as the benchmark model, excelling in real-time integration, cost-efficiency, and STEM reasoning (e.g., 91.7% on AIME 2025 math). Compared against Claude 4.5 (Anthropic, prioritizing ethical alignment and agentic coding at 77.2% on SWE-Bench), GPT-5 (OpenAI, versatile with 74.9% SWE-Bench), Gemini 3 (Google, multimodal at 95% AIME), Llama 4 (Meta, open-source with 10M context), Mistral Nemo (efficient MoE at 80% HumanEval), DeepSeek R1 (reasoning-focused at 87.5% AIME 2025), Phi-3 (Microsoft SLM at 60.4% HumanEval), Gemma 2 (Google lightweight at 72.3% HumanEval), Qwen2.5-Coder (Alibaba coding specialist at 85.7% HumanEval), StarCoder2 (BigCode code gen at ~85% HumanEval), Yi-1.5 (01.AI bilingual at 88% MMLU), CodeLlama (Meta code-focused at 70% SWE-Bench), Falcon 3 (TII general at 75% MMLU), StableLM 2 (Stability multilingual at 72% MMLU), and Nemotron-4 (NVIDIA at 79.2% MMLU), Grok 4 offers superior value (API \$3/\$15 per million tokens). Free open-source models like Llama 4 Scout (quantized 17B active on single H100 GPU) and Mistral Nemo (12B on 16GB VRAM) democratize access at 15-25 tokens/sec. Hardware like NVIDIA RTX 4060 (\$300, 8GB GDDR6) and Jetson AGX Orin (\$1999, 275 TOPS) enable efficient runs, including older GPUs (e.g., GTX 1080 Ti 11GB at ~62 tokens/sec for 7B Q4). Billing is token-based (input/output per million), with caching discounts (e.g., Grok 90% off cached). Ethics emphasize alignment (Claude 95% even-handedness) amid EU AI Act. Long-dialog summaries in Grok 4 reduce costs 40-98%. Grok 4 advantages: affordability, dynamism; disadvantages: verbosity, ecosystem lag. All claims substantiated with 2025 sources.

2. Introduction

As of December 12, 2025, Grok 4 benchmarks the AI field, emphasizing truth-seeking with native X search and 256k-token context. Claude 4.5 (November 2025) focuses on constitutional AI safety (200k-1M tokens). This standalone report compares capabilities, ethics, access, and integrations, drawing from benchmarks (Artificial Analysis, Vellum, SWE-Bench) and X feedback. Themes: Grok 4's cost-lead (vs. Claude's \$5/\$25M tokens), ethical divergences (EU AI Act mandates), and GPU integrations (RTX 4090 \$2755, Jetson AGX Orin \$1999). Expansions include more free models (Yi-1.5, CodeLlama, Falcon 3, StableLM 2, Nemotron-4), self-hosting on older GPUs (GTX 1080 Ti ~62 t/s 7B Q4, RTX 2080 Ti ~107 t/s), Grok 4 vs. competitors (paid: Claude 4.5/GPT-5; free: Llama 4/Mistral Nemo), and model-specific prompting cheat-sheets. Prompting best practices ensure transparency.

3. Understanding Tokens and AI Billing Practices

What Are Tokens?

Tokens are AI's fundamental units: subword pieces (e.g., "chatbot" = "chat" + "bot", ~4 characters/token). Input tokens (prompts) and output tokens (responses) are billed separately. Context windows (e.g., Grok 4: 256k) limit total tokens per request. Billing is per million tokens, with caching discounts (reuse prior inputs at 10-90% off). Multimodal (images/videos) convert to tokens (e.g., 1024x1024 image ~1290 tokens).

Billing Practices Across Models

Billing is usage-based (pay-per-token), with tiers for speed/quality. Subscriptions add quotas (e.g., SuperGrok \$30/mo). Batch processing halves costs asynchronously. Caching (prompt reuse) saves 50-90% (e.g., Claude 90%). Open-source (Llama 4) incurs hardware costs only (~\$0 self-hosted).

Model/Provider	Input (\$/1M Tokens)	Output (\$/1M Tokens)	Caching Discount	Batch Discount	Notes/Source
Grok 4 (xAI)	\$3 (standard); \$0.20 (Fast)	\$15; \$0.50	90% cached input	N/A	Tool calls extra (\$25/1k sources); SuperGrok \$30/mo quotas.
Claude 4.5 (Anthropic)	\$5 (Opus); \$3 (Sonnet); \$1 (Haiku)	\$25; \$15; \$5	90% prompt cache	50%	ASL-3 safety tiers; Pro \$20/mo.
GPT-5 (OpenAI)	\$1.25 (standard); \$0.25 (mini)	\$10; \$2	90% cached	50%	Pro \$200/mo unlimited; Agents SDK free.
Gemini 3 (Google)	\$2 (Pro ≤200k); \$4 (>200k)	\$12; \$18	90% cached	50%	Pro \$20/mo; Enterprise \$250/mo custom.
Llama 4 (Meta)	\$0 (open); ~\$2 (Azure hosted)	\$0; ~\$10	N/A (self-host)	N/A	Hardware-only; permissive license <700M users.
Mistral Nemo	\$0.30 (API); \$0 (open)	\$0.30; \$0	N/A	N/A	Apache 2.0; La Plateforme \$0.3/M.

Model/Provider	Input (\$/1M Tokens)	Output (\$/1M Tokens)	Caching Discount	Batch Discount	Notes/Source
DeepSeek R1	\$0.55 (cache miss); \$0.14 (hit)	\$2.19	75% cache	N/A	MIT open; free chat, API scaled.
Phi-3 (Microsoft)	\$0 (open); Azure ~\$0.40/\$1.60	\$0; ~\$1.60	N/A	N/A	SLM; Azure MaaS pay-as-you-go.
Gemma 2 (Google)	\$0 (open); Vertex \$0.00004 (4B)	\$0; \$0.00008	N/A	50%	Open weights; API tiered.
Qwen2.5-Coder	\$0 (open); Alibaba \$1.2	\$0; \$6	N/A	N/A	Apache; Model Studio API.
StarCoder2	\$0 (open)	\$0	N/A	N/A	BigCode OpenRAIL-M; free inference.
Yi-1.5	\$0 (open); 01.AI ~\$0.50/\$2	\$0; ~\$2	N/A	N/A	Apache 2.0; bilingual API.
CodeLlama	\$0 (open)	\$0	N/A	N/A	Meta license; code-focused.
Falcon 3	\$0 (open)	\$0	N/A	N/A	Apache 2.0; general-purpose.
StableLM 2	\$0 (open)	\$0	N/A	N/A	Stability AI license; multilingual.
Nemotron-4	\$0 (open)	\$0	N/A	N/A	NVIDIA license; high MMLU.

4. Model Comparison: Grok 4 vs. Claude 4.5 and Others

Grok 4 (July 2025, 256k context) benchmarks against Claude 4.5 (November 2025, 200k-1M tokens) in philosophy: truth-seeking vs. safety.

Capabilities and Performance

- **Reasoning/Logic:** Grok 4 leads STEM (91.7% AIME 2025, 87.5% GPQA). Claude 4.5 agentic (30+ hour sessions).
- **Coding:** Claude 4.5 74.5-77.2% SWE-Bench; Grok 72-75%.
- **Creativity/Tone:** Grok witty; Claude empathetic.
- **Multimodal/Tools:** Grok X/video; Claude Excel/Chrome.

Benchmarks

Benchmark	Grok 4	Claude 4.5	Notes/Source
MMLU	86.6%	86.0%	Tied knowledge.
SWE-Bench	72-75%	74.5-77.2%	Claude fixes.
AIME 2025 Math	91.7%	87.0%	Grok edges.
GPQA Science	87.5%	87.0%	Near-tie.
LiveCodeBench	82%	75-77%	Grok faster.

Expanded Comparison with Other Models

Paid/proprietary: GPT-5 (74.9% SWE-Bench, \$1.25/\$10M); Gemini 3 (~75% SWE, \$2/\$12M).

Free Open-Source Models on Single GPUs

Quantized (4-bit GGUF) models allow high-performance LLM inference on consumer-grade hardware (8GB to 24GB VRAM), democratizing access to high-performance AI. These models typically achieve 15-25 tokens/sec inference speed on modern GPUs.

General Self-Hosting Specifications

<u>Characteristic</u>	<u>Detail</u>
<u>Quantization</u>	<u>4-bit GGUF (Optimized file format)</u>
<u>VRAM Range</u>	<u>8-24 GB</u>
<u>Speed</u>	<u>15-25 tokens/sec (Typical inference)</u>

Expanded Free Open-Source Model Capabilities

The ecosystem is expanding to include specialized models beyond general-purpose use.

<u>Model</u>	<u>Size (Billion Parameters)</u>	<u>Core Feature / Specialization</u>	<u>Be</u>
<u>Llama 4 Scout</u>	<u>17B (Quantized)</u>	<u>General-Purpose / Long Context</u>	<u>~8</u>
<u>Mistral Nemo</u>	<u>12B (MoE)</u>	<u>Efficiency / Low API Cost</u>	<u>~8</u>
<u>Yi-1.5</u>	<u>34B</u>	<u>Bilingual Model</u>	<u>88</u>
<u>CodeLlama</u>	<u>70B</u>	<u>Code-Focused Model</u>	<u>70</u>
<u>Falcon 3</u>	<u>180B</u>	<u>General-Purpose LLM</u>	<u>75</u>
<u>StableLM 2</u>	<u>12B</u>	<u>Multilingual Model</u>	<u>72</u>
<u>Nemotron-4</u>	<u>340B</u>	<u>NVIDIA-Tuned Model</u>	<u>79</u>
<u>Qwen2.5-Coder</u>	<u>7B</u>	<u>Coding Specialist</u>	<u>85</u>

<u>Model</u>	<u>Size (Billion Parameters)</u>	<u>Core Feature / Specialization</u>	<u>Best Use Case</u>
<u>DeepSeek Coder V2</u>	<u>7B</u>	<u>Math/Coding</u>	<u>80-90%</u>
<u>Phi-3 Mini</u>	<u>3.8B</u>	<u>Small Language Model (SLM)</u>	<u>~70%</u>
<u>Gemma 2</u>	<u>9B</u>	<u>Lightweight / Responsible AI</u>	<u>72%</u>
<u>StarCoder2</u>	<u>7B</u>	<u>Code Generation</u>	<u>~85%</u>

Hardware Specific Model Compatibility (NVIDIA, Intel, AMD)

NVIDIA GPU (RTX 4060/4090)

<u>Model</u>	<u>Size / Key Benchmarks (HumanEval)</u>	<u>VRAM Q4</u>	<u>Stability</u>
<u>Qwen2.5-Coder</u>	<u>7B / 85.7%</u>	<u>8-12GB</u>	<u>Moderate</u> <u>Human Eval</u>
<u>DeepSeek Coder V2</u>	<u>7B / 80-90%</u>	<u>12-16GB</u>	<u>Stable</u>
<u>Phi-3 Mini</u>	<u>3.8B / ~70%</u>	<u>6-8GB</u>	<u>High</u>
<u>Yi-1.5</u>	<u>34B / 88%</u>	<u>16-24GB</u>	<u>Billing</u>
<u>CodeLlama</u>	<u>70B / ~85%</u>	<u>24GB (quant)</u>	<u>Code</u>
<u>Nemotron-4</u>	<u>340B / ~85%</u>	<u>24GB (partial)</u>	<u>High</u>

Intel GPU (Arc A770)

Intel GPUs are supported via OpenVINO quantization.

<u>Model</u>	<u>Size / Key Benchmarks</u>	<u>VRAM Q4</u>	<u>St</u>
<u>Phi-3 Mini</u>	<u>3.8B / ~70%</u>	<u>8-12GB</u>	<u>Int</u>
<u>Qwen2.5-Coder</u>	<u>7B / 85.7%</u>	<u>12-16GB</u>	<u>St</u>
<u>Mistral Nemo</u>	<u>12B / ~80%</u>	<u>16GB</u>	<u>Mo</u>
<u>Yi-1.5</u>	<u>9B / 88%</u>	<u>12-16GB</u>	<u>Bi</u>

AMD GPU (RX 7600/7900 XT)

AMD GPUs leverage ROCm (Linux) and Vulkan (Cross-platform) for acceleration.

<u>Model</u>	<u>Size / Key Benchmarks</u>	<u>VRAM Q4</u>	<u>St</u>
<u>StarCoder2</u>	<u>7B / ~85%</u>	<u>8-12GB</u>	<u>RC</u>
<u>DeepSeek Coder V2</u>	<u>7B / 80-90%</u>	<u>12-16GB</u>	<u>St</u>
<u>Llama 3.2</u>	<u>8B / 89%</u>	<u>8-12GB</u>	<u>AM</u>
<u>Mistral Nemo</u>	<u>12B / ~80%</u>	<u>16-20GB</u>	<u>RC</u>

Older GPUs (Self-Hosting Compatibility)

Older GPUs (e.g., GTX 1080 Ti 11GB, RTX 2080 Ti 11GB) are viable using [llama.cpp](#) Q4/Q5 K quantization, offering a functional, but slower, self-hosting experience.

Model (Q4)	GPU Fit	1080 Ti Est.	2080 Ti Est.
Qwen2.5-Coder 7B	Yes (11GB) ▾	~60 T/s	~100 T/s ▾
DeepSeek Coder V2 7B	Yes (11GB) ▾	~55 T/s	~95 T/s ▾
Phi-3 Mini 3.8B	Yes (8GB) ▾	~70 T/s	~110 T/s ▾
StarCoder2 7B	Yes (11GB) ▾	~62 T/s	~107 T/s ▾
CodeLlama 7B	Yes (11GB) ▾	~58 T/s	~100 T/s ▾
Nemotron-4 15B	Partial (11GB) ▾	~35 T/s	~60 T/s ▾

(T/s = Tokens/Sec)

The recommended specifications for running the expanded set of free open-source models using **4-bit GGUF quantization** are:

- **VRAM Range:** 8–24 GB
- **Speed (Tokens/sec):** 15–25 tokens/sec (Typical inference)

NVIDIA GPU (RTX 4060 \$300 8GB GDDR6 125 TFLOPS FP16; RTX 4090 \$2755 24GB GDDR6X 83 TFLOPS)

Model	Size / Benchmarks (SWE / HumanEval / LiveCode)	VRAM Q4	Strengths	Source
Qwen2.5-Coder	7B / ~49% / 85.7% / ~70%	8-12GB	Multilingual code.	
DeepSeek Coder V2	7B / 65.8% / 80-90% / ~70%	12-16GB	Math/coding.	
Phi-3 Mini	3.8B / ~50% / ~70% / 65%	6-8GB	Efficient reasoning.	
Gemma 2	9B / ~55% / 72.3% / ~65%	12-16GB	MMLU 76.2%.	
StarCoder2	7B / ~60% / ~85% / 70%	8-12GB	Code gen.	
Yi-1.5	34B / ~60% / 88% / ~75%	16-24GB	Bilingual code/math.	
CodeLlama	70B / 70% / ~85% / ~70%	24GB (quant)	Code completion.	
Falcon 3	180B / ~65% / ~80% / ~70%	24GB partial	General coding.	
StableLM 2	12B / ~55% / ~75% / ~65%	12-16GB	Multilingual code.	
Nemotron-4	340B / ~70% / ~85% / ~75%	24GB partial	High MMLU code.	

Intel GPU (Arc A770 \$322 16GB 262 TOPS INT8)

Model	Size / Benchmarks	VRAM Q4	Strengths	Source
Phi-3 Mini	3.8B / ~50% / ~70% / 65%	8-12GB	Intel optimized.	
Qwen2.5-Coder	7B / ~49% / 85.7% / ~70%	12-16GB	OpenVINO quant.	
Gemma 2	9B / ~55% / 72.3% / ~65%	12-16GB	Arc efficient.	
Mistral Nemo	12B / ~55% / ~80% / 70%	16GB	MoE function calling.	
Yi-1.5	9B / ~55% / 88% / ~70%	12-16GB	Bilingual via OpenVINO.	
CodeLlama	7B / ~60% / ~80% / ~65%	12GB	Code on Arc.	

AMD GPU (RX 7600 \$274 8GB GDDR6 22 TFLOPS FP16; RX 7900 XT \$580 20GB 61 TFLOPS)

Model	Size / Benchmarks	VRAM Q4	Strengths	Source
StarCoder2	7B / ~60% / ~85% / 70%	8-12GB	ROCm optimized.	
DeepSeek Coder V2	7B / 65.8% / 80-90% / ~70%	12-16GB	Vulkan support.	
Llama 3.2	8B / ~60% / 89% / ~70%	8-12GB	AMD-tuned multimodal.	
Mistral Nemo	12B / ~55% / ~80% / 70%	16-20GB	ROCm compatible.	
Falcon 3	11B / ~60% / ~80% / ~65%	12-16GB	ROCm general code.	
StableLM 2	7B / ~55% / ~75% / ~65%	8-12GB	Multilingual ROCm.	

Older GPUs (Self-Hosting Compatibility)

Older GPUs (e.g., GTX 1080 Ti 11GB \$200 used, RTX 2080 Ti 11GB \$250 used) support self-hosting via llama.cpp quantization (Q4_0/Q5_K, 5-10% quality drop). All free models self-hostable; 7B Q4 ~62 t/s on 1080 Ti, ~107 t/s on 2080 Ti (Llama-2 7B benchmark, similar for 2025 models). Here is the information about self-hosting models on older GPUs, presented in a clear, formatted way:

Older GPUs: Self-Hosting Compatibility for Free Models	
Component	Detail
Compatible GPUs	GTX 1080 Ti (11GB, estimated \$200 used), RTX 2080 Ti (11GB, estimated \$250 used)
Software/Method	llama.cpp quantization (using Q4_0/Q5_K formats)
Model Compatibility	All free models mentioned in the document are self-hostable.
Performance Note	Expected a 5-10% quality drop due to quantization.

Estimated Performance (7B Q4 Model Benchmark):

- **GTX 1080 Ti:** Approximately **~62 tokens/sec**
- **RTX 2080 Ti:** Approximately **~107 tokens/sec**

(Performance estimates are based on Llama-2 7B benchmark results, which are noted as similar for 2025 models.) Limitations: No tensor cores (slower FP16, ~5x vs. RTX 4060); PCIe Gen3 bandwidth caps large models.

Model	Older GPU Fit (Q4)	Tokens/Sec Est. (1080 Ti)	Tokens/Sec Est. (2080 Ti)	Source
Qwen2.5-Coder 7B	Yes (11GB)	~60	~100	
DeepSeek Coder V2 7B	Yes	~55	~95	
Phi-3 Mini 3.8B	Yes (8GB)	~70	~110	
Gemma 2 9B	Partial (11GB)	~50	~85	
StarCoder2 7B	Yes	~62	~107	
Yi-1.5 9B	Partial	~45	~80	
CodeLlama 7B	Yes	~58	~100	
Falcon 3 11B	Partial	~40	~70	
StableLM 2 7B	Yes	~55	~95	
Nemotron-4 15B	Partial	~35	~60	

NVIDIA Jetson Orin Family Hardware Comparison

Jetson Orin (2025 updates) for edge AI, Ampere GPU, ARM CPU, 7-60W. Supports Llama 4 Scout ~20-40 tokens/sec on AGX.

Model	TOPS (INT8)	Memory Configs	Power (W)	VRAM/Use Case	Tokens/Sec (8B Q4 Est.)	Price (USD)
Jetson AGX Orin	275	64GB/32GB LPDDR5 (204.8 GB/s)	15-60	Robotics/multi-mode I	~50-100 (Scout partial)	\$1999
Jetson Orin NX	157	16GB/8GB LPDDR5 (102.4 GB/s)	10-40	Edge/vision+LLM	~30-60 (7-8B)	\$999
Jetson Orin Nano Super	67	8GB LPDDR5 (102 GB/s)	7-25	Entry/Gen AI	~15-40 (3-7B)	\$249
Jetson Orin Nano	40	4GB LPDDR5 (51 GB/s)	5-15	IoT/basic	~10-25 (3B)	\$199

Grok 4 vs. Llama 4 Comparison

Feature	Grok 4	Llama 4 (Maverick)
Model Type / Context	Proprietary / 256k	Open-Source (MoE) / 10M
Coding (SWE-Bench)	75%	69.5%
Coding (HumanEval)	98%	~85-90%
Coding (LiveCodeBench)	82%	~70-75%
Cost (API Input/Output)	\$3 / \$15 per 1M tokens	Free / Open-Source
Ethics Score / Note	96% / Leak risks	90% / Permissive
Self-Host Speed (RTX 4090)	N/A	~80-100 tokens/sec (Scout 17B Q4)

Key Takeaways:

- **Performance:** Grok 4 maintains a clear edge in all major coding and reasoning benchmarks (SWE-Bench, HumanEval, LiveCodeBench).
- **Access & Context:** Llama 4 offers significantly larger context (10M) and a major cost advantage, being free/open-source and fully self-hostable.

Grok 4 vs. Mistral Nemo Comparison

Feature	Grok 4	Mistral Nemo
Model Size / Type	N/A	12B MoE (Mixture of E... ▾
Coding (SWE-Bench)	72-75%	~55% ▾
Coding (HumanEval)	98%	~80% ▾
Coding (LiveCodeBench)	82%	70% ▾
Reasoning (AIME)	91.7%	~80% ▾
Cost (API Input/Output)	\$3 / \$15 per 1M tokens	\$0.30 / \$0.30 per 1M to... ▾
Ethics Score / Note	96% / Refusal variance	88% / EU fairness focus ▾
Strength Highlight	Reasoning power, Benchmarks	MoE Speed, Low API c... ▾

Key Takeaways:

- **Performance:** Grok 4 significantly outperforms Mistral Nemo in both coding and reasoning benchmarks, notably AIME (91.7% vs. ~80%).
- **Cost & Efficiency:** Mistral Nemo is substantially cheaper and highly efficient due to its Mixture of Experts (MoE) architecture.

Grok 4 vs. Closest Paid Competitors (Claude 4.5 & GPT-5)

Feature	Grok 4	Claude 4.5	GPT-5
Coding (SWE-Bench)	75%	77.2%	74.9%
Ethical Lead / Score	96%	95% Even-Handedness	89%
Input Cost (per 1M Tokens)	\$3	\$5	\$1.25
Output Cost (per 1M Tokens)	\$15	\$25	\$10
Speed / Latency	Faster (75 t/s)	Slower (2.58s TTFT)	N/A
Key Advantage	Cost-effective scaling	Ethical robustness, Agentic coding	Lower base API cost, Ecosystem, CoT boost (+22%)

Key Takeaways:

- **Coding:** Claude 4.5 holds a slight lead in the demanding SWE-Bench coding benchmark (77.2%).
- **Cost:** GPT-5 offers the lowest input and output token cost at scale, while Grok 4 is positioned in the middle, still cheaper than Claude 4.5.
- **Focus:** Claude 4.5 prioritizes ethical alignment, while Grok 4 emphasizes speed and cost efficiency. GPT-5 offers broader versatility and ecosystem strength.

Free Competitors:

Feature	Grok 4 (Benchmark)	Llama 4	Mistral Nemo
Model Type / Access	API / Hosted Precision	Free / Open-So... ▾	Free / Open / API ▾
Coding (SWE-Bench)	75%	69.5% ▾	~55% ▾
Max Context Window	256k	10M ▾	N/A ▾
Throughput / Efficiency	75 t/s	N/A ▾	70 t/s (MoE Effi... ▾
Cost	API (\$3/\$15M)	Free / Open-So... ▾	API (\$0.30/M) / ... ▾
Key Advantage	Hosted Precision	Large Context (... ▾	MoE Efficiency ... ▾
Key Edge over Competitor	Higher SWE-Bench	N/A ▾	N/A ▾

Note on Reasoning: The source material indicates Grok 4 holds a reasoning edge over Mistral Nemo.

Strengths and Weaknesses

- **Grok 4:** Strengths—Cost-efficient (98% cheaper Fast), real-time X data, edges Llama 4/Mistral Nemo in coding (75% vs. 69.5%/55% SWE). Weaknesses—Humor overload, 256k context trails Llama's 10M.
- **Claude 4.5:** Strengths—Ethical robustness, long-session stability. Weaknesses—Latency (2.58s TTFT), pricier at scale.
- **Llama 4:** Strengths—Free/open, massive context, multimodal. Weaknesses—Coding underwhelm (pivoted late), benchmark scandals.
- **Mistral Nemo:** Strengths—MoE efficiency, multilingual fairness. Weaknesses—Lags Grok in reasoning (80% vs. 91.7% AIME), 12B resource-heavy.
- **Free Models Overall:** Strengths—Zero cost post-hardware; customizable (Yi/CodeLlama/Falcon/StableLM/Nemotron). Weaknesses—Lower benchmarks (49-70% SWE vs. 75%); quantization trade-offs.

5. AI Ethics in 2025 Models

2025 ethics emphasize neutrality under EU AI Act (February 2025), mandating bias audits/transparency. Grok 4 risks bias via truth-seeking; Claude 4.5 prioritizes harm reduction.

Key Ethical Frameworks

- **Bias Mitigation:** Anthropic test (November 2025) Claude 4.5 95% even-handed vs. Grok 4 96%; GPT-5 89%.
- **Privacy/Security:** Grok August 2025 leak (370k chats); Claude vibe-hacking (August 2025).
- **Sentience/Alignment:** Debates (July-September 2025) Grok radical transparency; Claude contextual ethics. Appearance bias persists.

Grok 4 vs. Claude 4.5 Ethics

Grok: Low refusals, but antisemitic outputs. Claude: High harm refusal, verbose ethics.

Expanded Ethics Across Other Models

Model	Bias/Fairness Score	Privacy Incidents	Alignment Approach	Source
GPT-5	89%	Minor API leaks	CoT ethics; guarded.	
Gemini 3	92%	None major	Google safeguards.	
Llama 4	90% community-tuned	Open weights risks	Permissive fine-tuning.	
Mistral Nemo	88%	Low EU focus	MoE multilingual fairness.	
DeepSeek R1	87%	Data sovereignty	Open transparency; math ethics.	
Phi-3	85% SLM	Microsoft-aligned	Compact low-hallucination.	
Gemma 2	91%	Google privacy	Responsible toolkit.	
Qwen2.5-Coder	95%	Alibaba standards	Ethical user-centric.	

Model	Bias/Fairness Score	Privacy Incidents	Alignment Approach	Source
StarCoder2	90%	BigCode audits	Responsible code data.	
Yi-1.5	89%	01.AI audits	Bilingual fairness.	
CodeLlama	88%	Meta-aligned	Code ethics focus.	
Falcon 3	87%	TII standards	General fairness.	
StableLM 2	86%	Stability AI	Multilingual alignment.	
Nemotron-4	91%	NVIDIA responsible	High MMLU ethics.	

Free models (Llama 4, Yi-1.5) enable audits but risk biases without tuning.

6. Access and Pricing Summary

Grok 4 Access and Costs

- **Free Tier:** Limited Grok 3 on x.com (10 req/2h); Grok 4 paid.
- **Subscriptions:** SuperGrok \$30/mo (Grok 4); Heavy \$300/mo advanced.
- **API:** \$3/\$15M (standard); \$0.20/\$0.50M (Fast <128k); caching \$0.75M. Tools \$25/1k sources.

Claude 4.5 Access and Costs

- **Free Tier:** Limited Haiku/Sonnet on claude.ai.
- **Subscriptions:** Pro \$20/mo (\$17 annual); Max \$100/mo; Team \$25/user/mo.
- **API:** Opus \$5/\$25M; Sonnet \$3/\$15M; Haiku \$1/\$5M; 90% cache, 50% batch savings.

Pricing Comparison Table

Tier/Model	Grok 4 (\$/M In/Out)	Claude 4.5 (\$/M In/Out)	Notes/Source
Base	\$3/\$15	\$5/\$25 (Opus)	Grok cheaper scale.
Fast/Light	\$0.20/\$0.50	\$1/\$5 (Haiku)	Grok efficiency.
Subscription	\$30/mo SuperGrok	\$20/mo Pro	Claude entry.

Free Models Access

- **Download:** Hugging Face/Ollama (e.g., `ollama run qwen2.5-coder:7b`); no recurring.
- **Hardware:** GPU \$200-2755 (used GTX 1080 Ti \$200 11GB; RTX 4090 \$2755 24GB); Jetson AGX Orin \$1999 64GB. Quantization free (llama.cpp).

Hardware Pricing and Specs

- **NVIDIA RTX 4060:** \$300, 8GB GDDR6, 125 TFLOPS FP16, 115W TDP.
- **RTX 4090:** \$2755, 24GB GDDR6X, 83 TFLOPS FP16, 450W.
- **Intel Arc A770:** \$322, 16GB GDDR6, 262 TOPS INT8, 225W.
- **AMD RX 7600:** \$274, 8GB GDDR6, 22 TFLOPS FP16, 165W.
- **RX 7900 XT:** \$580, 20GB GDDR6, 61 TFLOPS FP16, 315W.
- **Older NVIDIA GTX 1080 Ti:** \$200 used, 11GB GDDR5X, 11.3 TFLOPS FP32, 250W.
- **RTX 2080 Ti:** \$250 used, 11GB GDDR6, 13.4 TFLOPS FP32, 250W.

7. Available Extensions and Integrations

Grok 4 Extensions

- **VSCode:** Simply Grok/vscode-grok for queries.
- **Cursor:** API override grok-code-fast-1.
- **CLI:** npm Grok CLI, offline.

Claude 4.5 Extensions

- **VSCode:** Native Claude Code diffs/sessions.
- **Cursor:** Manual VSIX IDE detection.
- **CLI:** @anthropic-ai/claude-code agents.

8. Integration Comparison: CLI, VSCode, Cursor

CLI Integration

- **Grok 4:** npm scripting, X hooks; low-cost Fast.
- **Claude 4.5:** Code CLI autonomous; MCP agents, long sessions.

VSCode Integration

- **Grok 4:** vscode-grok workspaces; grok-code-fast selection.
- **Claude 4.5:** Native inline diffs; 1M beta context.

Cursor Integration

- **Grok 4:** Settings API; prototypes fast.
- **Claude 4.5:** VSIX install; WSL tweaks.

Comparison Table

Platform	Grok 4 Ease/Strengths	Claude 4.5 Ease/Strengths
CLI	9/10 Fast scripting	8/10 Agentic depth
VSCode	8/10 Extensions	9/10 Native diffs
Cursor	7/10 API override	7/10 Manual VSIX

Expanded Integration for Free Models

Free models are highly accessible through community-driven tools like **Ollama** and **llama.cpp**, with robust integration into popular developer environments (CLI, VSCode, Cursor via Continue.dev). While modern NVIDIA GPUs have the best support (8/10), Intel and AMD are competitive (7/10), and older GPUs still offer functional, albeit slower, performance through quantization.

Category	Detail	Rating / Performance Notes
Core Technology	Ollama / llama.cpp	Enables local, self-hosted LLM inference.
Supported Models	Qwen, DeepSeek, Yi, CodeLLama, Falcon, StableLM, Nemotron	Models supported via the Continue.dev extension.
Integration Points	CLI (Command Line Interface), VSCode, Cursor	Focus on developer and local execution environments.
NVIDIA GPU Support	CUDA (proprietary)	8/10 rating.
Intel/AMD GPU Support	ROCm / OpenVINO (open standards)	7/10 rating.
Older GPU Support	llama.cpp Q4 Quantization (e.g., older GTX cards)	6/10 rating, performance estimated at slower ~50 t/s .

9. Leveraging Long Dialogs for Efficiency in Grok 4

Grok 4 256k (2M Fast) supports dialogs; summaries extend windows, reducing costs 40-98% caching/splitting.

Summary Techniques

- **CoT Summarization:** "Summarize 50k tokens step-by-step ethics." Retains insights.
- **ROSES:** Archivist role compresses debates.
- **Caching:** Prefix summaries \$0.75M API.

Cost Reduction Strategies

Tiered: <128k \$0.20/\$0.50M; summaries + SuperGrok quotas low usage (100-user RAG near-zero). Split: SuperGrok planning, API tools—40% savings.

Improved Results

Meta-reasoning boosts 15% chained accuracy (ethics evolutions). Tools verify summaries, no drift; hybrid enhances.

10. Advantages and Disadvantages of Grok 4

Advantages:

- **Cost-Efficiency:** 98% cheaper Fast; caching/splitting repeats.
- **Real-Time Dynamism:** X outperforms Llama 4 static.
- **STEM/Reasoning Edge:** 91.7% AIME leads; witty tasks; beats Nemo 75% vs. 55% SWE.
- **GPU Accessibility:** Free analogs (DeepSeek); Void forks custom editors; Jetson/older GPU support (1080 Ti ~62 t/s).

Disadvantages:

- **Ethical Risks:** 96% even-handed but leaks.
- **Ecosystem Lag:** Fewer vs. Claude/Llama; 256k trails Llama 10M.
- **Verbosity:** Humor dilutes; free lag agentic.

Hybrid Grok API/SuperGrok Task Splitting

API granular + SuperGrok quotas 40-98% savings: SuperGrok chat/ideation, API execution (Fast \$0.20/M <128k). SDK chaining routes (summarize SuperGrok, execute API); 40% token savings.

11. Model Cheat-Sheets

Model Quick Reference Guides (Prompting Cheat-Sheets) have been documented are:

- Grok 4
- Claude 4.5
- GPT-5
- Gemini 3
- Llama 4
- Mistral Nemo
- DeepSeek R1
- Phi-3
- Gemma 2
- Qwen2.5-Coder
- StarCoder2
- Yi-1.5
- CodeLlama
- Falcon 3
- StableLM 2
- Nemotron-4

Grok 4 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "Act as [role]..." (e.g., "Pretend you're a sarcastic pirate captain...")	Witty is the default tone, but you can explicitly add humor.
List Items	Use "List 10 ideas..."	Produces concise bullets.
Elaborate	Use "Expand on [topic]..."	Prompts the model for first-principles reasoning.
Summarize	Use "Summarize [text/URL]..."	Leverages X search for real-time information retrieval.
Step-by-Step	Use "Think step by step..."	Activates the "Heavy mode" for more complex tasks like math.
Structures	Use TREF/ROSES/STAR formats.	Model follows these reliably; specify by saying "Use STAR..."
Style	Specify the style, e.g., "Be creative/factual."	Use of a 90% cache for repeating prompts aids in consistency.
Tone	Specify the tone, e.g., "Witty/empathetic."	The model's default tone is inspirational.

Grok 4 Prompting Examples

Goal	Prompting Examples
Role-Play	* "Act as a grumpy history professor and explain the French Revolution." * "Pretend you're a sarcastic pirate captain giving a marketing pitch for new software."
List Items	* "List 10 ideas for a new sci-fi short story." * "List 5 essential tools for a new developer."
Elaborate	* "Expand on the topic of quantum entanglement, starting with the basics." * "Expand on the competitive landscape section, adding more detail on Grok 4's architecture."
Summarize	* "Summarize the key points of this [URL] for a non-technical audience, using X search for real-time data." * "Summarize the executive summary in three bullet points."
Step-by-Step	* "Think step by step to solve the following complex algebra problem (Heavy mode for math)." * "Provide a step-by-step guide on setting up a new virtual environment."
Structures	* "Use the STAR format to describe a project where I showed leadership." * "Use the TREF structure to explain the difference between tokens and neurons."
Style	* "Be creative when generating a product description for a robotic dog." * "Write a factual summary of the recent market trends."
Tone	* "Write the conclusion with a witty tone." * "Write a short motivational speech with an empathetic tone."

Claude 4.5 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "Role-play as [role]..." (e.g., "As ethical auditor...")	Generates structured responses.
List Items	Use "Bulleted list of..."	Produces concise lists.
Elaborate	Use "Detail [safeguards]..."	Utilizes Chain-of-Thought (CoT) natively.
Summarize	Use "Condense [results]..."	Excellent for 200k context recall tasks.
Step-by-Step	Use "Reason through [bias]..."	Activates an extended thinking mode.
Structures	Use Multishot/CoT structures.	Use delimiters for clarity; benefits from a 90% prompt cache.
Style	Specify "Precise/evidence-based."	Helps to reduce model verbosity.
Tone	Specify "Professional/conversational."	Operates with an ethical baseline.

Claude 4.5 Prompting Examples

Goal	Prompting Examples
Role-Play	* "Role-play as an ethical auditor and critique the new company policy." * "Act as a senior software architect reviewing this API design for structural integrity."
List Items	* "Create a bulleted list of the main arguments in the constitutional AI framework." * "Give me a concise list of the necessary steps to deploy the application."
Elaborate	* "Detail the safeguards required for ASL-3 safety compliance." * "Elaborate on the cost differences between Opus, Sonnet, and Haiku models using Chain-of-Thought."
Summarize	* "Condense the results of the 30-hour agentic coding session." * "Summarize the main themes of the document, focusing on the 200k context recall."
Step-by-Step	* "Reason through the bias in the provided dataset, showing your extended thinking process." * "Provide a step-by-step logic chain to refute the main counter-argument."
Structures	* "Use Multishot structures to generate three different ethical scenarios." * "Use delimiters to separate your reasoning and final answer for this CoT prompt."
Style	* "Write the report in a precise and evidence-based style." * "Adopt an evidence-based approach to describing the model's performance."
Tone	* "Write the summary with a professional tone." * "Use a conversational tone when explaining the model's ethical baseline."

GPT-5 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "You are [role]..." (e.g., "Senior engineer...")	Chain-of-Thought (CoT) provides a +22% boost in performance.
List Items	Use "Give numbered list..."	Versatile for different types of lists.
Elaborate	Use "Explain in detail..."	Use the o3-mini model for faster generation speed.
Summarize	Use "Summarize [document]..."	Has a large 400k context window.
Step-by-Step	Use "Think step by step..."	Supports Instant and Thinking modes.
Structures	Use Few-shot/CoT structures.	Batch processing offers a 50% discount.
Style	Specify "Balanced/creative."	Benefits from a 90% cache.
Tone	Specify "Neutral/adaptable."	Operates with guarded ethics.

GPT-5 Prompting Examples

Goal	Prompting Examples
Role-Play	* "You are a senior engineer and you must propose three solutions to the latency issue." * "Act as a marketing analyst to draft an email campaign outline."
List Items	* "Give a numbered list of the required components for the new system." * "List the top 5 advantages of the GPT ecosystem."
Elaborate	* "Explain in detail the concept of a self-correcting agent." * "Elaborate on the Pro \$200/mo unlimited subscription plan."
Summarize	* "Summarize the entire 400k token context document into a short paragraph." * "Condense the technical specification into a five-sentence summary."
Step-by-Step	* "Think step by step and justify the decision to use the o3-mini model for this task (Use Instant mode)." * "Provide a step-by-step plan for integrating the Agents SDK."
Structures	* "Use a Few-shot structure to demonstrate the correct output format." * "Generate a response using CoT and highlight the +22% boost in your confidence score."
Style	* "Write the introduction in a balanced and informative style." * "Generate a series of creative slogans for the new product."
Tone	* "Adopt a neutral and objective tone for the research paper." * "Use an adaptable tone suitable for a variety of audiences."

Gemini 3 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "Act as [role] for [task]..."	Is natively multimodal (text, image, video, etc.).
List Items	Use "List [topic]..."	Leverages the Deep Think mode for complex reasoning.
Elaborate	Use "Analyze [image/text]..."	Features a 1M context window.
Summarize	Use "Condense [content]..."	Strong performance with vision and user interface (UI) tasks.
Step-by-Step	Use "Deep Think: Step by step..."	Best for shorter outputs (<300 tokens).
Structures	Use Zero-shot/Few-shot structures.	Benefits from a 90% cache.
Style	Specify "Direct/structured."	Pro subscription costs \$20/month.
Tone	Specify "Neutral/logical."	Notes an ecosystem lock-in.

Gemini 3 Quick Reference Guide Prompting Examples

Goal	Prompting Examples
Role-Play	* "Act as a professional chef for the task of designing a new meal menu." * "Role-play as a financial advisor and critique this investment strategy."
List Items	* "List the key themes of the annual report." * "Use Deep Think to list the three main challenges in high-context reasoning."
Elaborate	* "Analyze this image and explain the potential security risks." * "Elaborate on the benefits of the 1M context window for a large-scale project."
Summarize	* "Condense the content of the last 50 emails into a brief summary." * "Summarize the video presentation, focusing on the strong vision/UI points."
Step-by-Step	* "Use Deep Think: Step by step, explain how to optimize the code for a mobile device (Keep the response under 300 tokens)." * "Provide a step-by-step logic sequence for the complex reasoning problem."
Structures	* "Use a Zero-shot structure to provide a clear answer without examples." * "Use Few-shot prompting to guide the output to a desired format."
Style	* "Generate a direct and structured response to the user's query." * "Write the analysis using a direct style."
Tone	* "Ensure the entire response maintains a neutral and objective tone." * "Use a logical and analytical tone when discussing the trade-offs."

Llama 4 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "You are [role] in [scenario]..."	Utilizes a Mixture of Experts (MoE) for multilingual tasks.
List Items	Use "Generate list of..."	Features a massive 10M context window.
Elaborate	Use "Expand on [multimodal]..."	Quantized models are available for self-hosting.
Summarize	Use "Summarize long doc..."	Supports permissive fine-tuning.
Step-by-Step	Use "MoE route: Step by step..."	Available in Scout and Maverick variants.
Structures	Use In-Context Learning (ICL)/CoT structures.	Benefits from community-tuned efforts.
Style	Specify "Verbose for depth."	Model is free and open-source.
Tone	Specify "Adaptable/permissive."	Note that there are risks without careful fine-tuning.

Llama 4 Quick Reference Guide Prompting Examples

Goal	Prompting Examples
Role-Play	* "You are a geopolitical analyst in a crisis scenario. What are the three best courses of action?" * "Act as a data scientist in a Python environment and solve this statistical problem."
List Items	* "Generate a list of all multimodal features in the latest update." * "List all the potential hardware-only costs for self-hosting the model."
Elaborate	* "Expand on the multimodal capabilities of Llama 4 in detail." * "Elaborate on the significance of the 10M context window for long-document summarization."
Summarize	* "Summarize the long document, leveraging the massive context window." * "Condense the main points of the codebase into three concise paragraphs."
Step-by-Step	* "Use the MoE route: Step by step, what is the best strategy for model fine-tuning with a permissive license?" * "Provide a step-by-step explanation of the MoE routing mechanism."
Structures	* "Use ICL to guide the model's output on the next ten data points." * "Apply CoT to verify the accuracy of the Maverick benchmark result."
Style	* "Write the explanation with a verbose style to ensure maximum depth." * "Use a verbose-for-depth style to explore the implications of the benchmark scandal."
Tone	* "Adopt a permissive and open tone when discussing community-tuned models." * "Use an adaptable tone for translating the user manual into three languages."

Mistral Nemo Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "Role-play [MoE]..."	Strong capability for function calling.
List Items	Use "List in [lang]..."	Features a 128k context window.
Elaborate	Use "Detail multilingual..."	Aligned with EU fairness standards.
Summarize	Use "Condense [text]..."	Supports vLLM for self-hosting.
Step-by-Step	Use "MoE: Step by step..."	Minimum requirement is 16GB VRAM for 12B Q4.
Structures	Use Few-shot structures.	Available for free under the Apache license.
Style	Specify "Efficient/concise."	Older GPUs can only run a partial version.
Tone	Specify "Multilingual neutral."	API pricing is \$0.30 per million tokens.

Mistral Nemo Quick Reference Guide Prompting Examples

Goal	Prompting Examples
Role-Play	* "Role-play as a MoE routing expert and explain the efficiency gains." * "Act as a multilingual translator and provide an answer in Spanish and German."
List Items	* "List the model's function calling capabilities in JSON format." * "List the hardware requirements in the language: French."
Elaborate	* "Detail the multilingual fairness and EU compliance of the model." * "Elaborate on the 12B Q4 model's efficiency on different hardware."
Summarize	* "Condense the text of the EU AI Act into five key points." * "Summarize the technical paper on MoE architecture."
Step-by-Step	* "Use MoE: Step by step, explain how a function call is executed." * "Provide a step-by-step guide to installing the vLLM for self-hosting."
Structures	* "Use a Few-shot structure to ensure the output adheres to the correct legal format." * "Provide an answer using the Few-shot style for conciseness."
Style	* "Write the instructions in an efficient and concise style." * "Generate a short, concise description of the model."
Tone	* "Use a multilingual neutral tone for all translated content." * "Write the ethical report with a neutral tone."

DeepSeek R1 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "Act as [math expert]..."	Utilizes Reinforcement Learning (RL) for superior reasoning.
List Items	Use "List [code/math]..."	Has a 64k context window.
Elaborate	Use "Expand math/code..."	Cache hits are billed at a very low \$0.14M.
Summarize	Use "Summarize [reasoning]..."	Available as open-source under the MIT license.
Step-by-Step	Use "RL think: Step by step..."	Achieved 87.5% on the AIME benchmark.
Structures	Use CoT structures.	Features free chat access.
Style	Specify "Verbose depth."	Note: There are data sovereignty concerns.
Tone	Specify "Logical/math-focused."	API is scaled based on usage.

DeepSeek R1 Quick Reference Guide Prompting Examples

Goal	Prompting Examples
Role-Play	* "Act as a math expert and solve the following complex proof." * "Role-play as a seasoned developer and critique the code's efficiency."
List Items	* "List the key concepts in reinforcement learning for reasoning (RL think)." * "List the major functions and classes in the provided code snippet."
Elaborate	* "Expand on the mathematical principles behind the AIME 2025 question." * "Elaborate on the difference between a cache miss and a cache hit in API billing."
Summarize	* "Summarize the entire reasoning chain of the paper." * "Condense the long technical document on data sovereignty concerns."
Step-by-Step	* "Use RL think: Step by step, determine the logical fallacy in the argument." * "Provide a step-by-step derivation of the formula."
Structures	* "Use CoT structures to ensure every step of the calculation is shown." * "Apply CoT to the problem of API scaling."
Style	* "Write the summary with a verbose depth to cover all nuances." * "Use a verbose depth style when explaining the model's architecture."
Tone	* "Use a logical and math-focused tone for the technical document." * "Maintain a logical tone throughout the explanation."

Phi-3 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "You are [compact AI]..."	Is an efficient Small Language Model (SLM).
List Items	Use "List [task]..."	Has a 128k context window.
Elaborate	Use "Explain [code]..."	Azure hosted pricing is ~\$0.40/\$1.60M.
Summarize	Use "Condense [text]..."	Known for low-hallucination rates.
Step-by-Step	Use "Compact reason: Step by step..."	Available for free self-hosting.
Structures	Use Few-shot structures.	Is multilingual.
Style	Specify "Hyper-efficient."	Optimized for low memory, like "Phone RAM prompt."
Tone	Specify "Direct/concise."	Is Microsoft-aligned.

Phi-3 Quick Reference Guide Prompting Examples

Goal	Prompting Examples
Role-Play	* "You are a compact AI assistant running on a mobile device and must answer quickly." * "Act as a Microsoft-aligned security expert and audit this document."
List Items	* "List the three key features of the SLM architecture." * "List the main tasks that can be performed with the 128k context window."
Elaborate	* "Explain the code snippet and suggest one optimization." * "Elaborate on the low-hallucination features of the model."
Summarize	* "Condense the entire text into a two-sentence summary." * "Summarize the technical paper in an ultra-efficient style (Phone RAM prompt)."
Step-by-Step	* "Use Compact reason: Step by step, explain the process of data compression." * "Provide a step-by-step guide for deploying a small language model on Azure."
Structures	* "Use a Few-shot structure with three examples of the expected output." * "Apply Few-shot prompting to generate short, precise code."
Style	* "Write the entire response in a hyper-efficient style." * "Generate the summary using a hyper-efficient, technical style."
Tone	* "Use a direct and concise tone for the instruction manual." * "Write the output in a concise and professional tone."

Gemma 2 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "Act as [scope]..."	A lightweight model with strong MMLU performance.
List Items	Use "List [interpret]..."	Has a smaller 8k context window.
Elaborate	Use "Expand text/code..."	Vertex API pricing starts at a very low \$0.00004M.
Summarize	Use "Summarize [content]..."	Available as open weights.
Step-by-Step	Use "Open scope: Step by step..."	API is available in tiered options.
Structures	Use Zero-shot structures.	Comes with a responsible toolkit.
Style	Specify "Short direct."	Best for short responses (<300 tokens).
Tone	Specify "Neutral."	Focuses on Google's privacy standards.

Gemma 2 Quick Reference Guide Prompting Examples

Goal	Prompting Examples
Role-Play	* "Act as a scope manager and interpret the project requirements." * "Role-play as a Google privacy advocate and analyze this policy."
List Items	* "List the three key metrics to interpret the benchmark results." * "List the five open weights available for Gemma 2."
Elaborate	* "Expand on the responsible toolkit features of the model." * "Elaborate on the process of using the Vertex API for a small model (4B)."
Summarize	* "Summarize the provided content in a short, direct style (under 300 tokens)." * "Condense the ethics paper into four short sentences."
Step-by-Step	* "Use Open scope: Step by step, explain the process for fine-tuning the open weights." * "Provide a step-by-step checklist for a new project kickoff."
Structures	* "Use a Zero-shot structure to answer the following fact question directly." * "Apply Zero-shot prompting for a brief, initial analysis."
Style	* "Provide the answer in a short, direct style." * "Write the entire document using a short and direct style."
Tone	* "Use a neutral tone for the final report." * "Maintain a neutral and objective tone throughout the explanation."

Qwen2.5-Coder Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "As Alibaba engineer..."	Trained on a repo-scale dataset.
List Items	Use "List code [lang]..."	Has a 128k context window.
Elaborate	Use "Expand multilingual code..."	Alibaba API is priced at \$1.2/\$6M.
Summarize	Use "Condense repo..."	Emphasizes user-centric ethics.
Step-by-Step	Use "Coder: Step by step..."	Achieved 85.7% on the HumanEval benchmark.
Structures	Use Few-shot structures.	Available for free under the Apache license.
Style	Specify "Precise code."	Integrated with Model Studio.
Tone	Specify "Professional."	Noted for 95% ethical alignment.

Qwen2.5-Coder Quick Reference Guide Prompting Examples

Goal	Prompting Examples
Role-Play	* "Act as an Alibaba engineer and optimize the following Python code for a low-power environment." * "Role-play as a coding specialist and write a unit test for the given function."
List Items	* "List the available multilingual code completion languages." * "List the 5 most important files in the repository structure."
Elaborate	* "Expand on the multilingual code generation capabilities of the model." * "Elaborate on the process of using the Model Studio API to deploy Qwen2.5-Coder."
Summarize	* "Condense the entire code repository's purpose into a single paragraph." * "Summarize the technical paper on ethical user-centric AI development."
Step-by-Step	* "Use Coder: Step by step, fix the bug in the following Java code." * "Provide a step-by-step guide to achieving the 85.7% HumanEval benchmark."
Structures	* "Use a Few-shot structure to ensure the output adheres to the required output format for the code." * "Apply Few-shot prompting for consistent code documentation."
Style	* "Write the response in a precise code style." * "Ensure the documentation is written in a precise and technical style."
Tone	* "Use a professional tone for the bug report." * "Write the entire technical guide with a professional and objective tone."

StarCoder2 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "As code gen AI..."	Supports 600+ programming languages.
List Items	Use "List [code snippets]..."	Has a 16k context window.
Elaborate	Use "Expand multi-lang code..."	Licensed under OpenRAIL-M.
Summarize	Use "Summarize code data..."	Available for free inference.
Step-by-Step	Use "Fill-middle: Step by step..."	Uses responsibly sourced data.
Structures	Use Infill prompting.	Has Harness integration.
Style	Specify "Code-focused."	Noted for 90% ethics.
Tone	Specify "Technical."	Subject to BigCode audits.

StarCoder2 Quick Reference Guide Prompting Examples

Goal	Prompting Examples
Role-Play	* "Act as a code gen AI and provide a full script for a simple web server." * "Role-play as a BigCode auditor and analyze this codebase for licensing issues."
List Items	* "List 5 code snippets that demonstrate the model's multi-language support (600+ langs)." * "List the key features of the OpenRAIL-M license."
Elaborate	* "Expand on the multi-language code generation feature." * "Elaborate on the process of responsible code data sourcing."
Summarize	* "Summarize the entire code data audit report." * "Condense the documentation on the BigCode project."
Step-by-Step	* "Use Fill-middle: Step by step, complete the missing part of the function with the best logic." * "Provide a step-by-step guide to integrating the model with the Harness platform."
Structures	* "Use Infill prompting to complete the code block labeled ``` \n [missing_code] \n ```." * "Apply Infill prompting to fix the syntax error in the middle of the function."
Style	* "Write the output in a code-focused style, minimizing explanatory text." * "Generate a series of test cases in a strictly code-focused style."
Tone	* "Use a technical tone for all comments and explanations." * "Maintain a technical and precise tone throughout the documentation."

Yi-1.5 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "Act as bilingual [role]..."	Specialized in Chinese-English bilingual tasks.
List Items	Use "List [bilingual]..."	Has a 32k context window.
Elaborate	Use "Expand code/math..."	Available under the Apache 2.0 license.
Summarize	Use "Condense [text]..."	01.AI API pricing is ~\$0.50/\$2M.
Step-by-Step	Use "Bilingual reason: Step by step..."	Achieved 88% on the MMLU benchmark.
Structures	Use Few-shot structures.	Subject to 01.AI audits.
Style	Specify "Balanced lang."	Strong focus on bilingual tasks.
Tone	Specify "Neutral."	Noted for 89% fairness.

Yi-1.5 Quick Reference Guide Prompting Examples

Goal	Prompting Examples
Role-Play	* "Act as a bilingual financial advisor (Chinese-English) and summarize the market trends." * "Role-play as a math tutor and explain the concept of logarithms in both languages."
List Items	* "List the main differences between the Chinese and English versions of the policy." * "List the most important 01.AI audits completed this quarter."
Elaborate	* "Expand on the code structure and provide comments in both English and Chinese." * "Elaborate on the mathematical proof, translating the concepts into Chinese."
Summarize	* "Condense the entire 32k token document into a brief summary in English." * "Summarize the latest 01.AI news article."
Step-by-Step	* "Use Bilingual reason: Step by step, explain how the model achieved 88% on the MMLU benchmark." * "Provide a step-by-step guide to configuring the API service."
Structures	* "Use a Few-shot structure to ensure all output is presented side-by-side in both languages." * "Apply Few-shot prompting for consistent formatting of names and titles."
Style	* "Write the document in a balanced language style, giving equal weight to Chinese and English." * "Use a balanced language style when describing the bilingual focus."
Tone	* "Maintain a neutral tone for the report on bilingual fairness." * "Adopt a neutral and objective tone for the translation."

CodeLlama Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "As code assistant..."	Excellent for code Q&A.
List Items	Use "List functions..."	Available under the Meta license.
Elaborate	Use "Complete [code]..."	Available for free self-hosting.
Summarize	Use "Summarize script..."	Achieved 70% on the SWE benchmark.
Step-by-Step	Use "Code step by step..."	Strong performance on infill tasks.
Structures	Use Prefix/suffix structures.	Follows code ethics standards.
Style	Specify "Completion-focused."	Noted for 88% bias score.
Tone	Specify "Technical."	Meta-aligned development.

CodeLlama Quick Reference Guide Prompting Examples

Goal	Prompting Examples
Role-Play	* "Act as a code assistant and refactor the following legacy code for modern Python syntax." * "Role-play as a security specialist and audit the provided file."
List Items	* "List the key functions in the code and explain their purpose." * "List the main differences between the Meta license and an open-source license."
Elaborate	* "Complete the code block below with a secure implementation of the authentication logic." * "Expand on the code ethics standards of the Meta-aligned model."
Summarize	* "Summarize the function of the entire JavaScript script." * "Condense the technical paper on the 70% SWE benchmark result."
Step-by-Step	* "Use Code step by step: explain the execution flow of the C++ program." * "Provide a step-by-step guide to using the model for code Q&A."
Structures	* "Use Prefix/suffix structures to complete the function definition." * "Apply Prefix/suffix prompting to generate only the missing imports and dependencies."
Style	* "Write the output in a completion-focused style." * "Ensure the generated code comments are written in a completion-focused style."
Tone	* "Use a technical tone for all generated code and documentation." * "Maintain a technical tone in the model's explanations of its ethical focus."

Falcon 3 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "Act as [general AI]..."	Noted as an early high-profile release.
List Items	Use "List [tasks]..."	Available under the Apache 2.0 license.
Elaborate	Use "Expand [general]..."	Available for free self-hosting.
Summarize	Use "Condense [text]..."	Achieved 75% on the MMLU benchmark.
Step-by-Step	Use "General reason: Step by step..."	Developed under TII standards.
Structures	Use Zero-shot/Few-shot structures.	A general-purpose model.
Style	Specify "Versatile."	Noted for 87% fairness.
Tone	Specify "Neutral."	An early open model.

StableLM 2 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "As multilingual [role]..."	A lighter-weight multilingual model.
List Items	Use "List [langs]..."	Available under the Stability license.
Elaborate	Use "Expand [text]..."	Available for free self-hosting.
Summarize	Use "Summarize [multi-lang]..."	Achieved 72% on the MMLU benchmark.
Step-by-Step	Use "Multilingual step by step..."	Strong performance in multilingual tasks.
Structures	Use Few-shot structures.	Noted for 86% alignment.
Style	Specify "Light/efficient."	Developed by Stability AI.
Tone	Specify "Adaptable."	Focuses on multilingual capability.

StableLM 2 Quick Reference Guide Prompting Examples

Goal	Prompting Examples
Role-Play	* "Act as a multilingual customer service agent and handle the following complaint." * "Role-play as a Stability AI researcher and explain the model's architecture."
List Items	* "List the top 10 supported languages and their alignment scores." * "List the main features of the Stability license."
Elaborate	* "Expand on the multilingual capabilities and their real-world impact." * "Elaborate on the light-weight and efficient design philosophy."
Summarize	* "Summarize the multi-language document on global ethical frameworks." * "Condense the article on the 72% MMLU result."
Step-by-Step	* "Use Multilingual step by step: provide instructions for a software installation in French and English." * "Provide a step-by-step comparison of the model with other multilingual models."
Structures	* "Use a Few-shot structure to ensure the output adheres to the required translation format." * "Apply Few-shot prompting for consistent technical terminology."
Style	* "Write the response in a light/efficient style." * "Generate the documentation using a light/efficient and direct style."
Tone	* "Use an adaptable tone for communicating complex technical concepts to a general audience." * "Write the internal memo in an adaptable tone."

Nemotron-4 Quick Reference Guide

Goal	Technique/Example	Notes
Role-Play	Use "Act as [NVIDIA AI]..."	A high MMLU model.
List Items	Use "List [tasks]..."	Available under the NVIDIA license.
Elaborate	Use "Expand [code]..."	Available for free self-hosting.
Summarize	Use "Condense [content]..."	Achieved 79.2% on the MMLU benchmark.
Step-by-Step	Use "High-reason step by step..."	Follows NVIDIA's responsible AI guidelines.
Structures	Use CoT structures.	Noted for 91% ethics score.
Style	Specify "Performance-tuned."	Optimized for multi-GPU setups.
Tone	Specify "Technical..."	N/A

Nemotron-4 Quick Reference Guide Prompting Examples

Goal	Prompting Examples
Role-Play	* "Act as an NVIDIA AI expert and explain the multi-GPU setup." * "Role-play as a high-reasoning engine and solve the complex logic puzzle."
List Items	* "List the three main features that contribute to the high MMLU score." * "List the available hardware configurations for self-hosting."
Elaborate	* "Expand on the responsible AI guidelines followed by NVIDIA." * "Elaborate on the model's multi-GPU optimization techniques."
Summarize	* "Condense the content of the white paper into a concise summary." * "Summarize the ethical considerations, focusing on the 91% ethics score."
Step-by-Step	* "Use High-reason step by step: provide the solution to this advanced physics problem." * "Provide a step-by-step guide to achieving the 79.2% MMLU benchmark."
Structures	* "Use CoT structures to ensure all reasoning for the ethical decision is transparent." * "Apply CoT to verify the accuracy of the complex calculation."
Style	* "Write the technical documentation in a performance-tuned style." * "Generate the comparison table in a performance-tuned and technical style."
Tone	* "Use a technical tone when discussing the model's architecture." * "Maintain a technical tone for the internal development notes."

12. Conclusion

Grok 4 benchmarks 2025 AI: cost/dynamic vs. Claude's ethical reliability. Expanded free models (Yi/CodeLlama/Falcon/StableLM/Nemotron) and older GPU support democratize; prompting cheat-sheets optimize. Hardware (RTX 4090 \$2755, GTX 1080 Ti \$200 used) enables. Token billing favors Grok. Facts from 2025 sources.

Footnotes

1. Benchmarks independent 2025; quantized -5-10%.
2. Pricing Dec 12, 2025; free hardware excluded.
3. Ethics Anthropic/MIT; open community audits.

References

- [web:0-9]: xAI Docs, Artificial Analysis, Medium (Barnacle Goose), AIToolRanked, The document appears to be a comprehensive analysis of various 2025 AI models, focusing on hardware compatibility, performance benchmarks, ethical scores, pricing, and prompting strategies. It is already highly structured with numerous markdown headings, tables, and bulleted lists.