AIGuys · Follow publication

Member-only story Featured

Kimi K2 Just Killed OpenAI, & Claude

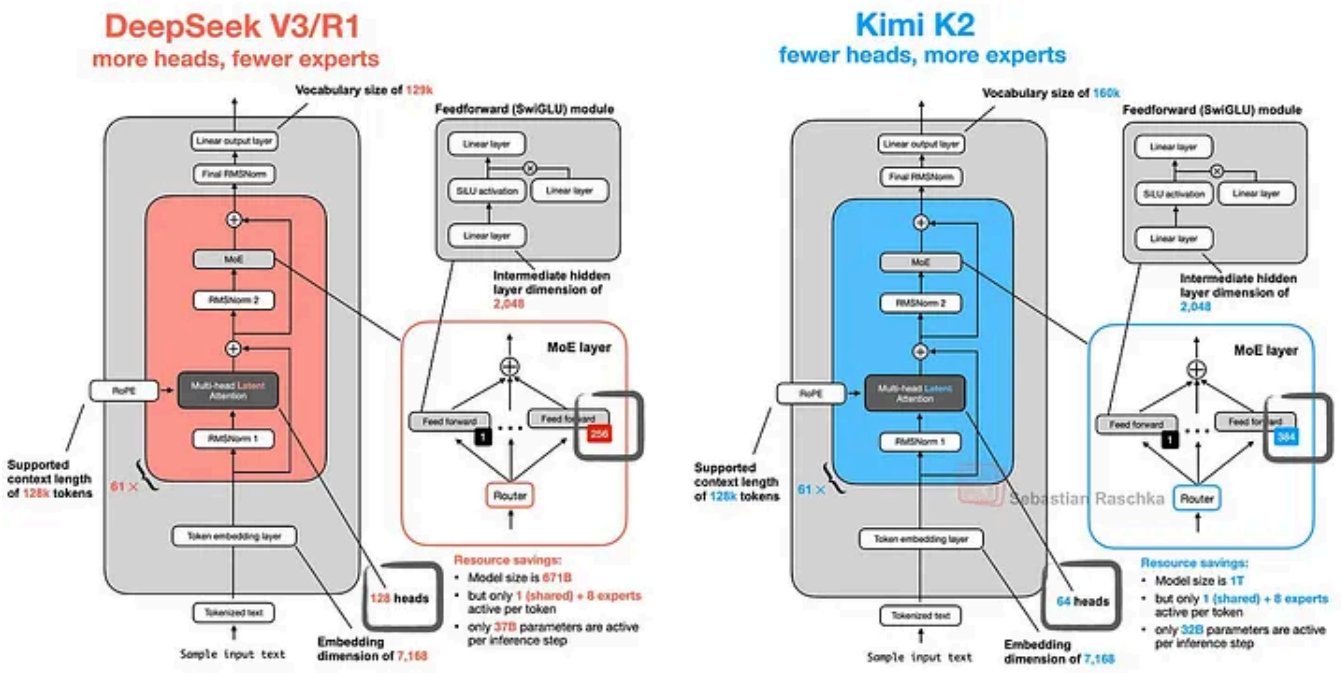
8 min read · 4 days ago

Vishal Rajput Follow

Listen Share More

If one thing tech bros are better than at building AI, it is creating hype around AI. And the release of Kimi K2 just proved the opposite. Once again we mark the similar moment as the monumental release of DeepSeek. Kimi K2 is poised to be the best open source model, and this time it is actually beating the closed source model, not lagging behind.

Kimi K2 Thinking stands out for its balance of power and efficiency. Released by Moonshot AI on November 6, 2025, it's designed as a "thinking agent" with a 1 trillion-parameter MoE architecture, activating 32 billion parameters per inference. This allows it to run on reasonable hardware while delivering impressive results in reasoning and tool use.

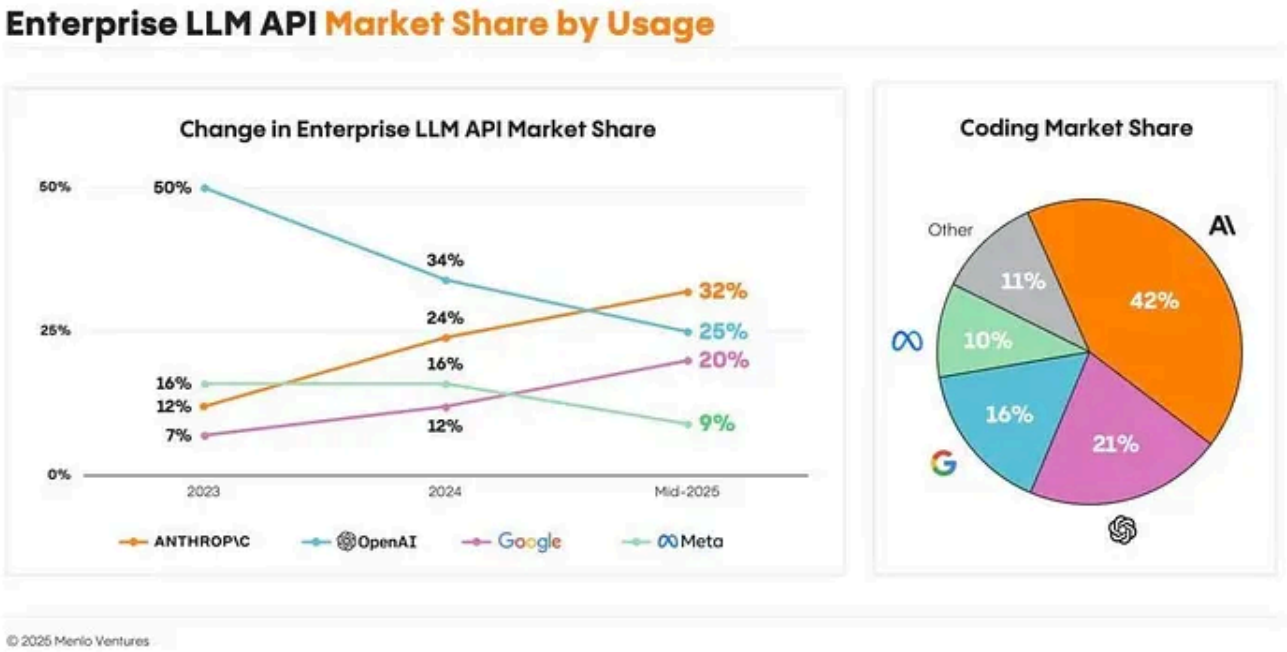


Kimi K2 — Open-Source Agentic Model

. . .

Current State of Enterprise AI Models

Let's talk about Anthropic first. Recently Menlo Ventures released a report, Anthropic has surpassed OpenAI to become the new king of the enterprise LLM market.



Market Share Shifts

- **Anthropic: 32%** (The new leader)
- **OpenAI: 25%** (Down from 50% in 2023)
- **Google: 20%** (Strong growth)
- **Meta Llama: 9%**
- **DeepSeek: 1%** (Despite early-year attention)

This reversal began with the release of Claude Sonnet 3.5 last June, and Sonnet 3.7 in February of this year solidified Anthropic's position at the top. It's important to note that market share reflects the proportion of AI usage in production environments, not the amount spent.

Enterprises Care Only About Performance, Not Price or Open Source

The enterprise adoption rate of open-source models has dropped from 19% to 13%. Performance gaps (lagging 9–12 months behind) and deployment complexity have stalled the enterprise adoption of open-source models.

"I don't care whether DeepSeek is open source. I only ask: Is it a good model? Does it perform better than us on certain tasks? **Users ultimately choose the product that works best, not the most open one.**" said Anthropic CEO Dario.

> But this is about to change with the release of Kimi K2.

Until now, every single model was significantly inferior to the closed source models of these big labs. The only model that came close to it was DeepSeek. But now, we finally have a model that is not at par, but actually beating these big models.

Before we move on to Kimi K2, let's look at the Claude for a moment; outside enterprise. Current Claude Sonnet 4.5 and Opus 1.5 are really great models, but they are seriously limited by the usage.

> *I uploaded a 10 page doc and gave one prompt and my usage was over for next few hours. A serious coding session with Claude can easily exhaust your token limit within half an hour or so.*

This seriously limits the adoption and real benefit to the end user of models like Claude. Out of Gemini, ChatGPT, & Claude, more often than not Gemini becomes my go to model because of its far bigger usage limits.

Getting Started With Kimi K2

You can download this model directly from Hugging Face or try via the Moonshot API. Check the docs at platform.moonshot.ai for setup.

I've used a lot of LLM models for years, and Moonshot AI's Kimi K2 Thinking seems to be quite a strong model. It is an open-source "thinking agent," which specializes in deep reasoning, autonomous tool orchestration, and coding.

You can even run this locally with a setup like two M3 Ultras at around 15 tokens per second, quite an efficient model, if it can beat the big models at this scale. The 256K context window can handle large projects without hiccups. Also, its native INT4 quantization provides a 2x speedup in inference without impacting the accuracy much.

What sets it apart is the Mixture-of-Experts (MoE) architecture: 61 layers, 7168 attention hidden dimension, 384 experts selecting 8 per token, SwiGLU activation, and a 160K vocabulary. This setup, with 1 trillion total parameters but only 32 billion active, makes it an all rounder.

It can easily chain 200–300 tool calls autonomously, interleaving chain-of-thought with functions for tasks like research or writing.

The model's checkpoints are in compressed-tensors format, and I easily converted them to FP8/BF16 for testing. It supports frameworks like vLLM and SGLang, and the turbo variant hit 171 tokens/second with 2.17-second first-token latency — faster than competitors like MiniMax-M2. Hardware requirements are manageable, under 600GB for weights, which is great for hobbyists.

Impressive Results

It scored 44.9% on Humanity's Last Exam with tools, outperforming Claude Sonnet 4.5 in agentic search (60.2% on BrowseComp vs. 24.1%). Math tasks were strong, with 99.1% on AIME25 using Python. While it edges GPT-5 in some areas like GPQA Diamond (85.7% vs. 84.5%), users on X have noted occasional long-context weaknesses.

With open-source now rivaling closed models, potentially accelerating innovation while questioning proprietary dominance. Enterprises like Airbnb are exploring similar tech for cost savings.

The Modified MIT License allows commercial use with attribution for large deployments, democratizing access. However, potential benchmark biases and hardware needs are worth noting.

Technical Dive

MuonClip optimizer with QK clipping

Kimi team didn't use a standard optimizer like AdamW. They used **Muon**, which is a "geometry-aware, matrix-structured" optimizer. Unlike Adam, which treats all the model's parameters (weights) as one long, flat vector, Muon understands that weights in a Transformer are **matrices**. It performs updates that are "matrix-aware."

Muon's updates are related to the **spectral norm** of the weight matrices. By controlling this, it implicitly controls the "Lipschitz constant" of the network. In simpler terms, it prevents the model's outputs from changing too chaotically in response to small input changes, which is a major source of instability.

But even this is unstable at 1 trillion scale. So, the next thing they introduce is QK clip. It only "clips" the few attention heads that are actually exploding, leaving healthy heads untouched. This avoids "over-clipping" the entire model. Instead of just capping the *output* (the logit), it rescales the *weights* ***(W_q, W_k)*** that *produce* the output. This is a more stable, root-cause fix.

Pre-training Data: Improving Token Utility with Rephrasing

Token efficiency in pre-training refers to how much performance improvement is achieved for each token consumed during training. Increasing token utility — the effective learning signal each token contributes — enhances the per-token impact on model updates, thereby directly improving token efficiency. This is particularly important when the supply of high-quality tokens is limited and must be maximally leveraged. A naive approach to increasing token utility is through repeated exposure to the same tokens, which can lead to overfitting and reduced generalization.

To improve the token utility of high-quality knowledge tokens, we propose a synthetic rephrasing framework composed of the following key components:

- **Style- and perspective-diverse prompting:** To enhance linguistic diversity while maintaining factual integrity, we apply a range of carefully engineered prompts. These prompts guide a large language model to generate faithful rephrasings of the original texts in varied styles and from different perspectives.
- **Chunk-wise autoregressive generation:** To preserve global coherence and avoid information loss in long documents, we adopt a chunk-based autoregressive rewriting strategy. Texts are divided into segments, rephrased individually, and then stitched back together to form complete passages. This method mitigates implicit output length limitations that typically exist with LLMs. An overview of this pipeline is presented in Figure 4.
- **Fidelity verification:** To ensure consistency between original and rewritten content, we perform fidelity checks that compare the semantic alignment of each rephrased passage with its source. This serves as an initial quality control step prior to training.

A general reinforcement learning framework

After its initial supervised fine tuning, Kimi K2 goes for Reinforcement Learning (RL).

To do this, the Kimi team built a "gym" where the model could try millions of tasks and get graded.

The key was *how* it got graded. The team used two different "teachers."

Teacher #1: The Objective Grader

This teacher graded tasks with clear right or wrong answers. A computer could automatically check the model's work.

- For **math and coding**, the model's answer was "correct" if its code actually ran and passed the tests.
- For **following instructions**, it was checked for simple rules, like "Is the answer 50 words long?"
- For **safety**, an "attack" model tried to trick Kimi K2 into saying something bad. If Kimi K2 refused, it passed the test.

Teacher #2: The Subjective Grader

This teacher was for creative or open-ended tasks, like writing a poem, where there is no single "correct" answer. So, the model learned to grade itself.

Here's how: One part of Kimi K2 would write two different answers (A and B). Another "critic" part of the model would then look at both and decide which one was better based on a quality checklist (like "be helpful" and "be factual"). This self-generated preference ("A is better than B") was used as the "grade" to learn from.

To make sure this "critic" was a good judge, they first had it practice on the *objective* math and coding problems. This "grounded" its judgment in facts, making it a much more reliable critic for the subjective tasks.

Three Final Training Rules

To make all this practice work, the Kimi team used three special rules:

1. **Word Count:** The model was given a strict word limit ("token budget") for its answers. If it wrote a rambling, overly long response, it was penalized. This forced it to learn to be concise and to the point.
2. **"Refresher Course":** To make sure the model didn't "forget" all its original general knowledge while learning these new skills, they kept mixing in small amounts of its original training data.
3. **From Creative to Focused:** They started the training by letting the model be very creative and random to *discover* new and better ways of answering. As training went on, they gradually "turned down" the creativity, telling it to become more stable and just stick to the best, most reliable answers it had found.

All and all, the model looks promising, small enough to easily host, good enough to challenge the big labs. And most important of all, They did it without asking trillions and billions of dollars for it.

. . .

Writing articles like this takes considerable effort and time. Please subscribe and follow me if my content adds any value to you.

Newsletter: https://medium.com/aiguys/newsletter

𝕏: https://x.com/RealAIGuys

Also, looking to solve a challenge for business-related documents, check out: https://number7ai.com/

Kimi K2 | AI | Artificial Intelligence | Llm | Machine Learning

Follow

Published in AIGuys

4.4K followers · Last published 4 days ago

Deflating the AI hype and bringing real research and insights on the latest SOTA AI research papers. We at AIGuys believe in quality over quantity and are always looking to create more nuanced and detail oriented content.

Follow

Written by Vishal Rajput

19.98K followers · 93 following

3x🏆 Top writer in AI | AI Book 📓 : https://rb.gy/xc8m46 | LinkedIn +: https://www.linkedin.com/in/vishal-rajput-999164122/ | 𝕏: https://x.com/RealAIGuys

Responses (8)

Bgerby

What are your thoughts?

See all responses