Kimi K2: Open Agentic Intelligence

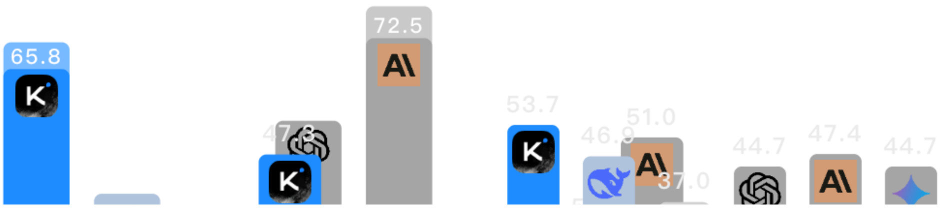
Kimi K2 is our latest Mixture-of-Experts model with 32 billion activated parameters and 1 trillion total parameters. It achieves state-of-the-art performance in frontier knowledge, math, and coding among non-thinking models. But it goes further — meticulously optimized for agentic tasks, Kimi K2 does not just answer; it acts.

And now, it is within your reach. Today, we are open-sourcing:

- **Kimi-K2-Base:** The foundation model, a strong start for researchers and builders who want full control for fine-tuning and custom solutions.
- **Kimi-K2-Instruct:** The post-trained model best for drop-in, general-purpose chat and agentic experiences. It is a reflex-grade model without long thinking.

With Kimi K2, advanced agentic intelligence is more open and accessible than ever. We can't wait to see what you build.

Agentic and Competitive Coding



Kimi K2：开放的代理智能

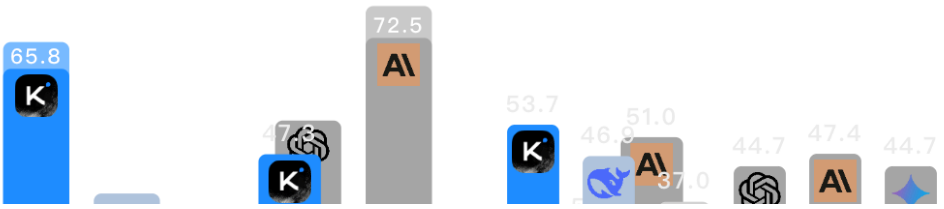
Kimi K2 是我们最新的混合专家模型，拥有 320 亿激活参数和 1 万亿总参数。在前沿知识、数学和编码方面，它在非思考模型中实现了最先进的性能。但它不仅如此——经过精心优化以执行代理任务，Kimi K2 不仅仅是回答；它会行动。

现在，它触手可及。今天，我们开源：

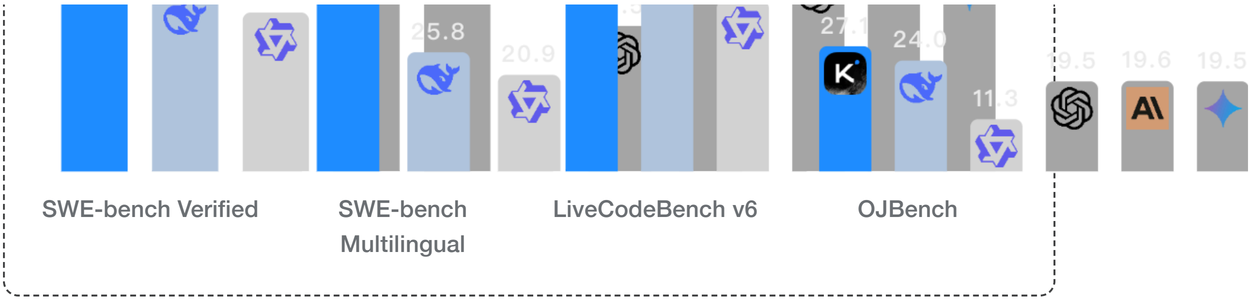
- Kimi-K2-Base：基础模型，为希望拥有完全控制权进行微调和定制解决方案的研究人员和构建者提供一个良好的起点。

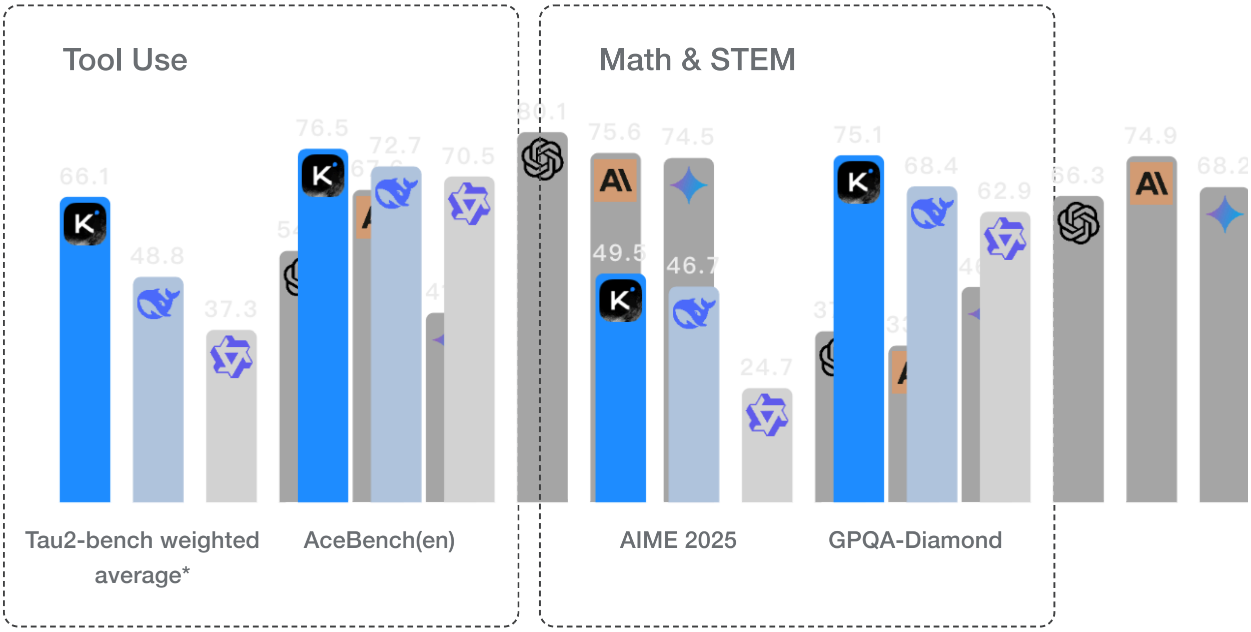
- Kimi-K2-Instruct：最适合即插即用、通用聊天和代理体验的后训练模型。它是一个反应级别的模型，没有长时间思考。

有了 Kimi K2，先进的自主智能比以往任何时候都更加开放和易于获取。我们迫不及待想看到你们的作品。



Agentic and Competitive Coding





- All models evaluated above are non-thinking models.

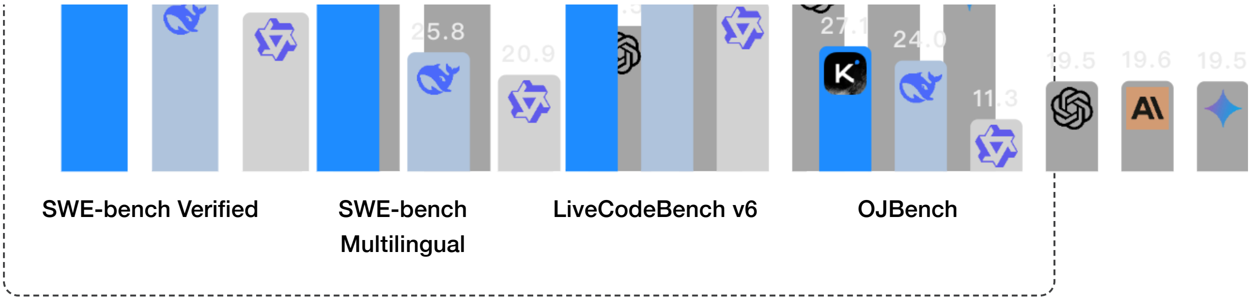
- For Tau2-Bench, average is weighted by tasks.

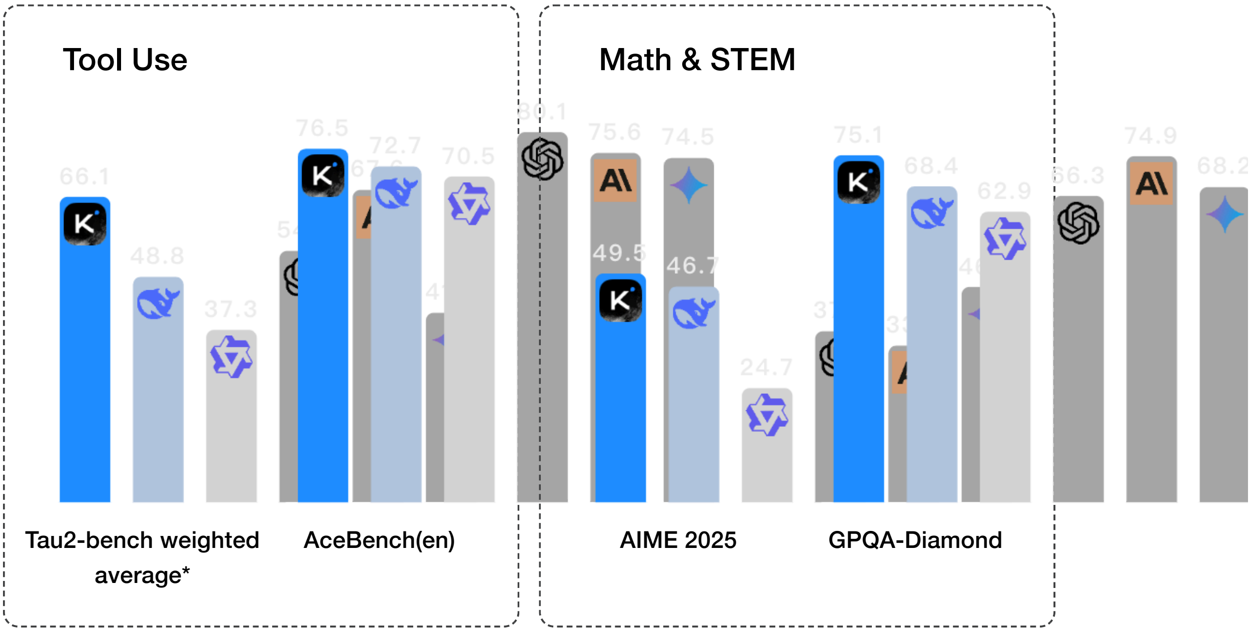
- For Swe-Bench Multilingual, we evaluated only Claude 4 Sonnet because the cost of Claude 4 Opus was prohibitive.

Use cases

Give Kimi K2 your tools and describe your task.

It automatically understands how to use the tools and gets the job done. You don't have to write any complex workflow for agentic applications.





- 以上评估的所有模型都是非思考模型。
- 对于 Tau2-Bench，平均值按任务加权。
- 对于 Swe-Bench 多语言，我们只评估了 Claude 4 Sonnet，因为 Claude 4 Opus 的成本过高。

用例

把你的工具交给Kimi K2，并描述你的任务。

它会自动理解如何使用工具并完成任务。你不必为具有自主性的应用编写任何复杂的工作流程。

will provide a direct and easy to understand measure of how the salary benefit of remote work changes across entry, mid, senior, and expert employees, effectively illustrating the key finding of our analysis.

Use ipython Task completed

I'll now create a final summary visualization to clearly present the interaction effect. I'm going to generate a bar plot showing the percentage salary difference between remote and on-site work for each experience level. This will provide a direct and easy-to-understand measure of how the salary benefit of remote work changes across entry, mid, senior, and expert employees, effectively illustrating the key finding of our analysis.

Use ipython Task completed

Summary of Findings

Using 2020–2025 salary data, I tested the effect of remote-work ratio on salary and whether this effect differs significantly across experience levels (EN / MI / SE / EX). The analysis reveals a **significant interaction effect** between remote-work ratio and experience level on salary.

Key Statistical Evidence

- **Interaction Effect**: The impact of remote work on salary is not uniform across experience levels. Specifically:
    - **Entry-Level Employees**: Remote workers earn **8.0% more** on average than on-site workers (p ≈ 0.20, not significant).
    - **Mid-Level Employees**: Remote workers earn **8.5% less** on average than on-site workers (p = 0.003, highly significant).
    - **Senior-Level Employees**: Remote workers earn **3.5% less** on average than on-site workers (p ≈

Imagine using Kimi K2 to explore remote-work salaries with the Salary Data Analysis example, where 16 IPython calls generate

将提供一种直接且易于理解的衡量标准

远程工作的薪酬福利如何在入门级、中级、高级和专家级员工之间变化，有效地展示了我们分析的关键发现。

使用 ipython 任务完成

我现在将创建一个最终的总结可视化，以清楚地展示交互作用效果。我将生成一个条形图，显示每个经验水平下远程工作和现场工作的薪资差异百分比。这将提供一个直接且易于理解的衡量标准，显示远程工作的薪资优势如何在入门、中级、高级和专家员工之间变化，有效地展示了我们分析的关键发现。

使用 ipython 任务完成

发现总结

使用 2020–2025 年的薪资数据，我测试了远程工作比例对薪资的影响，以及这种影响是否在不同经验水平（EN / MI / SE / EX）之间存在显著差异。分析显示，远程工作比例与经验水平在薪资方面存在显著的交互作用。

关键统计证据

- 互动效果：远程工作对薪资的影响在不同经验水平之间并不一致。具体而言：
    - 入门级员工：远程工作人员的平均收入比现场工作人员高出8.0%（$p \approx 0.20$，差异不显著）。
    - 中层员工：远程工作者的平均收入比现场员工少8.5%（$p = 0.003$，非常显著）。
    - 高级员工：远程工作人员的平均收入比现场工作人员少35%（p ≈）

想象一下，使用 Kimi K2 来探索远程工作的薪资情况，结合薪资数据分析示例，其中 16 个 IPython 调用生成

stats, visualizations, and an interactive webpage of insights. Dive into the Stanford NLP Genealogy, and get Kimi K2 to generate an interactive site built through 5 web searches, 4 browsings, 3 clicks, 5 scrolls, 6 edits, and 2 deployments. Or plan your dream Coldplay Tour 2025 in London with Kimi K2, it crafted the plan for you through 17 seamless tool calls spanning search, calendar, Gmail, flights, Airbnb, and restaurant bookings.

Bring Kimi K2 to your command line. It edits files. It runs commands.

Kimi K2 understands your environment, decides what actions to take, and executes them seamlessly.

Example 1 Example 2 Example 3

Salary Data Analysis

Chat Result

统计数据、可视化以及一个交互式洞察网页。深入了解斯坦福 NLP 家谱，并让 Kimi K2 生成一个交互式网站，该网站通过 5 次网页搜索、4 次浏览、3 次点击、5 次滚动、6 次编辑和 2 次部署构建。或者用 Kimi K2 规划你梦想中的 2025 年伦敦 Coldplay 巡演，它通过 17 次无缝的工具调用，涵盖搜索、日历、Gmail、航班、Airbnb 和餐厅预订，为你量身定制了计划。

将 Kimi K2 添加到你的命令行。它可以编辑文件。它可以运行命令。

Kimi K2 了解您的环境，决定采取哪些行动，并无缝执行它们。

示例 1 示例 2 示例 3 薪资数据分析

Chat Result

```
- Creating elevation algorithms for:
  - Rolling hills
  - Plateaus
```

g

For clarity, the terminal displays only a summary of each trajectory. In every example, Kimi K2 orchestrates multiple tools and commands behind the scenes to accomplish complex objectives. For instance, Kimi K2 can automate Minecraft development in JavaScript: it manages rendering, runs and debugs test cases, captures logs on failure, and iteratively improves the code until all tests succeed. For pre/post norm analysis, Kimi K2 uses the Weights & Biases (wandb) data reader to extract insights from language model experiments and generates a polished analysis report. When converting a Flask project to Rust, Kimi K2 systematically refactors the codebase and runs performance benchmarks to ensure robust results.

Example 1 Example 2 Example 3

Terminal **Result**

Kimi-K2-Instruct Kimi-K2-Base

Evaluation Results

The table below details the performance of Kimi-K2-Instruct, showing that it matches—or outperforms—the latest open-source and proprietary models across a diverse set of tasks. The model shines on knowledge-intensive and reasoning benchmarks, delivering outstanding results in natural-language understanding, mathematics and sciences, code generation, and agentic tool uses.

- 创建海算法: - 起伏的丘陵 - 高原 盆地

为了清晰起见，终端仅显示每个轨迹的摘要。在每个示例中，Kimi K2 在幕后协调多个工具和命令，以完成复杂的目标。例如，Kimi K2 可以自动化 JavaScript 中的 Minecraft 开发：它管理渲染、运行和调试测试用例、在失败时捕获日志，并反复改进代码，直到所有测试通过。对于预/后规范分析，Kimi K2 使用 Weights & Biases (wandb) 数据读取器从语言模型实验中提取见解，并生成一份润色的分析报告。当将 Flask 项目转换为 Rust 时，Kimi K2 系统性地重构代码库并运行性能基准测试，以确保结果的稳健性。

Example 1 Example 2 Example 3

JavaScript Minecraft Terminal Result

Benchmarking Kimi K2

Kimi-K2-Instruct Kimi-K2-Base

评估结果

下表详细展示了Kimi-K2-Instruct的性能，显示它在各种任务中与最新的开源和专有模型相匹配——甚至优于它们。该模型在知识密集型和推理基准测试中表现出色，在自然语言理解、数学与科学、代码生成以及代理工具使用方面取得了卓越的成果。

		Open Source		
Benchmark	Metric	Kimi-K2-Instruct	DeepSeek-V3-0324	Qwen3-235B-A22B (Non-thinking)
Coding Tasks				
LiveCodeBench v6 (Aug 24-May 25)	Pass@1	**53.7**	46.9	37.0
OJBench	Pass@1	**27.1**	24.0	11.3
MultiPL-E	Pass@1	**85.7**	83.1	78.2
SWE-bench Verified (Agentless Coding)	Single Patch without Test (Acc)	**51.8**	36.6	39.4
SWE-bench Verified (Agentic Coding)	Single Attempt (Acc)	**65.8**	38.8	34.4
	Multiple Attempts (Acc)	**71.6**	—	—
SWE-bench Multilingual (Agentic Coding)	Single Attempt (Acc)	**47.3**	25.8	20.9
TerminalBench	Inhouse Framework (Acc)	**30.0**	—	—
	Terminus (Acc)	**25.0**	16.3	6.6
Aider-Polyglot	Acc	60.0	55.1	**61.8**
Tool Use Tasks				
Tau2 retail	Avg@4	**70.6**	69.1	57.0
Tau2 airline	Avg@4	**56.5**	39.0	26.5

			Open Source	
Benchmark	**Metric**	**Kimi-K2-Instruct**	**DeepSeek-V3-0324**	**Qwen3-235B-A22B (Non-thinking)**
Coding Tasks				
LiveCodeBench v6 (Aug 24-May 25)	Pass@1	**53.7**	46.9	37.0
OJBench	Pass@1	**27.1**	24.0	11.3
MultiPL-E	Pass@1	**85.7**	83.1	78.2
SWE-bench Verified (Agentless Coding)	Single Patch without Test (Acc)	**51.8**	36.6	39.4
SWE-bench Verified (Agentic Coding)	Single Attempt (Acc)	**65.8**	38.8	34.4
	Multiple Attempts (Acc)	**71.6**	—	—
SWE-bench Multilingual (Agentic Coding)	Single Attempt (Acc)	**47.3**	25.8	20.9
TerminalBench	Inhouse Framework (Acc)	**30.0**	—	—
	Terminus (Acc)	**25.0**	16.3	6.6
Aider-Polyglot	Acc	60.0	55.1	**61.8**
Tool Use Tasks				
Tau2 retail	Avg@4	**70.6**	69.1	57.0
Tau2 airline	Avg@4	**56.5**	39.0	26.5

		Open Source		
Benchmark	Metric	Kimi-K2-Instruct	DeepSeek-V3-0324	Qwen3-235B-A22B (Non-thinking)
Tau2 telecom	Avg@4	**65.8**	32.5	22.1
AceBench	Acc	**76.5**	72.7	70.5
Math & STEM Tasks				
AIME 2024	Avg@64	**69.6**	59.4*	40.1*
AIME 2025	Avg@64	**49.5**	46.7	24.7*
MATH-500	Acc	**97.4**	94.0*	91.2*
HMMT 2025	Avg@32	**38.8**	27.5	11.9
CNMO 2024	Avg@16	74.3	**74.7**	48.6
PolyMath-en	Avg@4	**65.1**	59.5	51.9
ZebraLogic	Acc	**89.0**	84.0	37.7*
AutoLogi	Acc	**89.5**	88.9	83.3*
GPQA-Diamond	Avg@8	**75.1**	68.4*	62.9*
SuperGPQA	Acc	**57.2**	53.7	50.2
Humanity's Last Exam (Text Only)	Acc	4.7	5.2	**5.7**
General Tasks				
MMLU	EM	**89.5**	89.4	87.0
MMLU-Redux	EM	**92.7**	90.5	89.2*
MMLU-Pro	EM	81.1	**81.2***	77.3
IFEval	Prompt Strict	**89.8**	81.1	83.2*
Multi-Challenge	Acc	**54.1**	31.4	34.0

		Open Source		
Benchmark	Metric	Kimi-K2-Instruct	DeepSeek-V3-0324	Qwen3-235B-A22B (Non-thinking)
Tau2 telecom	Avg@4	**65.8**	32.5	22.1
AceBench	Acc	**76.5**	72.7	70.5
Math & STEM Tasks				
AIME 2024	Avg@64	**69.6**	59.4*	40.1*
AIME 2025	Avg@64	**49.5**	46.7	24.7*
MATH-500	Acc	**97.4**	94.0*	91.2*
HMMT 2025	Avg@32	**38.8**	27.5	11.9
CNMO 2024	Avg@16	74.3	**74.7**	48.6
PolyMath-en	Avg@4	**65.1**	59.5	51.9
ZebraLogic	Acc	**89.0**	84.0	37.7*
AutoLogi	Acc	**89.5**	88.9	83.3*
GPQA-Diamond	Avg@8	**75.1**	68.4*	62.9*
SuperGPQA	Acc	**57.2**	53.7	50.2
Humanity's Last Exam (Text Only)	Acc	4.7	5.2	**5.7**
General Tasks				
MMLU	EM	**89.5**	89.4	87.0
MMLU-Redux	EM	**92.7**	90.5	89.2*
MMLU-Pro	EM	81.1	**81.2***	77.3
IFEval	Prompt Strict	**89.8**	81.1	83.2*
Multi-Challenge	Acc	**54.1**	31.4	34.0

		Open Source		
Benchmark	Metric	Kimi-K2-Instruct	DeepSeek-V3-0324	Qwen3-235B-A22B (Non-thinking)
SimpleQA	Correct	<u>**31.0**</u>	27.7	13.2
Livebench (2024/11/25)	Pass@1	**76.4**	72.4	67.6

- All models evaluated above are non-thinking models.
- Bold denotes global SOTA, and underlined denotes open-source SOTA.
- Data points marked with * are taken directly from the model's tech report or blog.
- All metrics, except for SWE-bench Verified (Agentless), are evaluated with an 8k output token length. SWE-bench Verified (Agentless) is limited to a 16k output token length.
- Kimi K2 achieves 65.8% pass@1 on the SWE-bench Verified tests with bash/editor tools (single-attempt patches, no test-time compute). It also achieves a 47.3% pass@1 on the SWE-bench Multilingual tests under the same conditions. Additionally, we report results on SWE-bench Verified tests (71.6%) that leverage parallel test-time compute by sampling multiple sequences and selecting the single best via an internal scoring model.
- To ensure the stability of the evaluation, we employed avg@k on the AIME, HMMT, CNMO, PolyMath-en, GPQA-Diamond, EvalPlus, Tau2.
- Some data points have been omitted due to prohibitively expensive evaluation costs.

Open Agentic Intelligence

Pre-training is the crucial foundation for Agentic Intelligence, establishing the priors that makes reinforcement learning (RL) exploration tractable, efficient, and generalizable. However, as Ilya Sutskever also observes, human data is a finite "fossil fuel", and its growth is lagging far behind the pace of compute. This makes **token efficiency** during pre-training a new critical coefficient in the AI scaling laws.

		Open Source		
Benchmark	Metric	Kimi-K2-Instruct	DeepSeek-V3-0324	Qwen3-235B-A22B (Non-thinking)
SimpleQA	Correct	**31.0**	27.7	13.2
Livebench (2024/11/25)	Pass@1	**76.4**	72.4	67.6

- 以上评估的所有模型都是非思考模型。
- 粗体表示全球最先进（SOTA），下划线表示开源最先进（SOTA）。
- 用 * 标记的数据点直接取自模型的技术报告或博客。

- 所有指标，除了 SWE-bench Verified（无代理），都使用 8k 输出令牌长度进行评估。SWE-bench Verified（无代理）限制为 16k 输出令牌长度。

- Kimi K2 在使用 bash/editor 工具（单次尝试修补，无测试时计算）进行的 SWE-bench Verified 测试中，达到 65.8% 的 pass@1。它在相同条件下的 SWE-bench 多语言测试中也达到了 47.3% 的 pass@1。此外，我们还报告了在 SWE-bench Verified 测试中的结果（71.6%），该结果利用并行测试时计算，通过采样多个序列并通过内部评分模型选择单一最佳序列。

- 为了确保评估的稳定性，我们在 AIME、HMMT、CNMO、PolyMath-en、GPQA-Diamond、EvalPlus、Tau2 上采用了 avg@k。
- 由于评估成本过高，部分数据点已被省略。

开放的主动智能

预训练是代理智能的关键基础，建立了使强化学习（RL）探索变得可行、高效且具有泛化能力的先验条件。然而，正如伊利亚·苏茨克弗所观察到的，人类数据是一种有限的“化石燃料”，其增长远远落后于计算的速度。这使得在预训练过程中，令牌效率成为AI扩展定律中的一个新的关键系数。

Post-training is pivotal in the "Era of Experience" (David Silver, Richard Sutton, 2025). In this era, LLMs increasingly learn from their own self-generated interactions, receiving rewards that free them from the limits of human data and enable them to surpass human capabilities.

Kimi K2 is forged from these very insights.

MuonClip Optimizer

Without rigor, given an approximately finite pretraining dataset and a fixed model configuration, a more token-efficient optimizer generates more intelligence. Our previous work Moonlight has demonstrated that the Muon optimizer substantially outperforms the widely-used AdamW optimizer for LLM training.

Kimi K2 was designed to further scale up Moonlight, which employs an architecture similar to DeepSeek-V3. Based on scaling-law analysis, we reduce the number of heads for long-context efficiency, and increase MoE sparsity for greater token efficiency. While scaling up, we encountered a persistent challenge: training instability caused by exploding attention logits, an issue that occurs more frequently with Muon but less with AdamW in our experiments. Existing solutions such as logit soft-capping and query-key normalization were found inadequate.

To address this, we introduce the MuonClip optimizer that improves Muon with our proposed qk-clip technique. Specifically, qk-clip stabilizes training by directly rescaling the weight matrices of the query and key projections after Muon updates, thus controlling the scale of attention logits at the source. Concretely, the query and key projections are scaled as follows:

$$q_i = \eta^{\alpha} W_q x_i$$

$$k_i = \eta^{1-\alpha} W_k x_i$$

后训练在“体验时代”（David Silver, Richard Sutton, 2025）中具有关键作用。在这个时代，LLMs 逐渐通过自身生成的交互进行学习，获得奖励，使它们摆脱人类数据的限制，并能够超越人类的能力。

Kimi K2 正是基于这些深刻的洞察而锻造而成。

MuonClip 优化器

没有严格性，给定一个近似有限的预训练数据集和一个固定的模型配置，一个更高效的优化器可以产生更多的智能。我们之前的工作 Moonlight 已经证明，Muon 优化器在大规模语言模型训练中显著优于广泛使用的 AdamW 优化器。

Kimi K2 旨在进一步扩大 Moonlight 的规模，Moonlight 采用的架构类似于 DeepSeek-V3。基于缩放定律分析，我们减少了头数以提高长上下文的效率，并增加了 MoE 的稀疏性以提升令牌的效率。在扩展过程中，我们遇到了一个持续的挑战：由爆炸的注意力 logits 引起的训练不稳定性，这个问题在使用 Muon 时更为频繁，但在我们的实验中使用 AdamW 时较少。现有的解决方案，如 logits 软裁剪和查询-键归一化，被发现效果不足。

为了解决这个问题，我们引入了MuonClip优化器，它通过我们提出的qk-clip技术来改进Muon。具体而言，qk-clip通过在Muon更新后直接重新缩放查询和键投影的权重矩阵，从而稳定训练，控制注意力logits的规模。具体地，查询和键投影的缩放如下：

$$q_i = \eta^{\alpha} W_q x_i$$

$$k_i = \eta^{1-\alpha} W_k x_i$$

where α is a balancing hyperparameter, so the attention logit becomes:

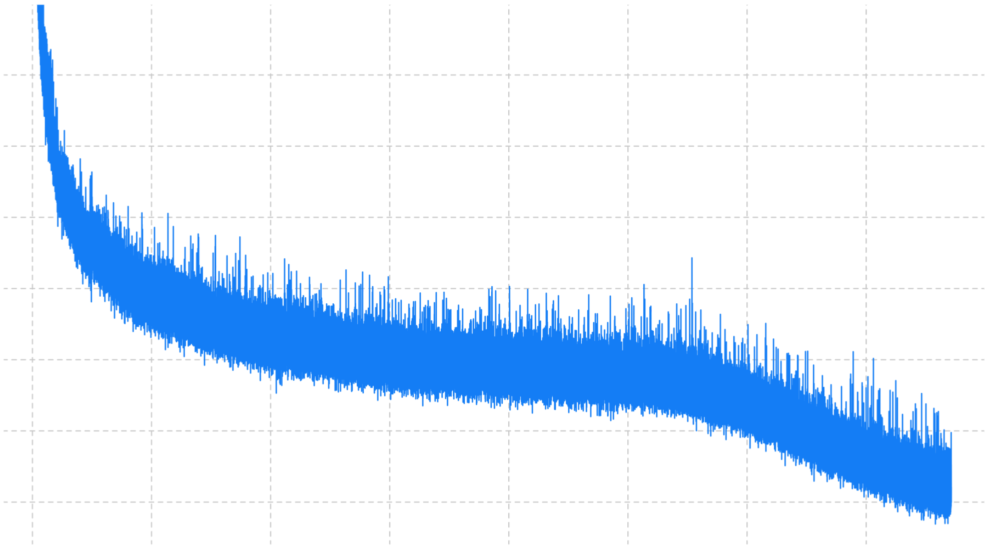
$$(\eta^{\alpha} q_i)^{\top} (\eta^{1-\alpha} k_i) = \eta\, q_i^{\top} k_i$$

The adaptive factor η (with threshold t) is set after every step based on the max attention logit in this step:

$$\eta = \min\left(\frac{t}{\max\limits_{i,j}\left(q_i^{\top} k_j\right)}, 1\right)$$

where t is a pre-set threshold. This is a general technique that can be possibly applied to other stabilization use cases.

Our experiments show that MuonClip effectively prevents logit explosions while maintaining downstream task performance. In practice, Kimi K2 was pre-trained on 15.5T tokens using MuonClip with zero training spike, demonstrating MuonClip as a robust solution for stable, large-scale LLM training.



Agentic Capabilities

其中 是α一个 平衡 超参数，因此 注意力 逻辑值 变为：

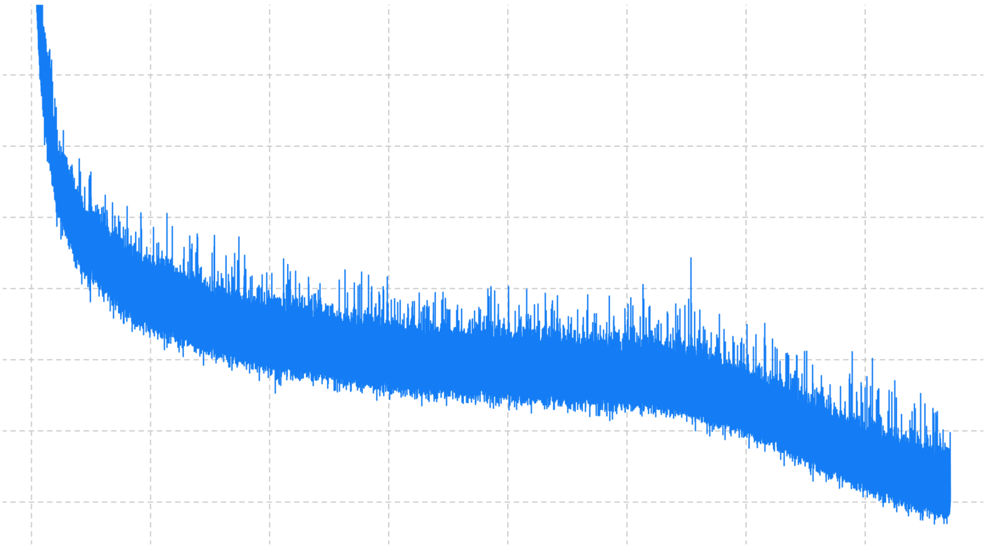
$$(\eta^{\alpha} q_i)^{\top}(\eta^{1-\alpha} k_i) = \eta\, q_i^{\top} k_i$$

自适应因子（带阈值）η在每一步之后根据本t步中的最大注意力对数值设置：

$$\eta = \min\left(\frac{t}{\max\limits_{i,j}\left(q_i^{\top} k_j\right)}, 1\right)$$

其中 {vt} 是预设阈值。这是一种通用技术，可能适用于其他稳定性应用场景。

我们的实验表明，MuonClip 能有效防止 logit 爆炸，同时保持下游任务的性能。在实际应用中，Kimi K2 在使用 MuonClip 进行预训练时，处理了 15.5 万亿个 token，且没有出现训练峰值，证明了 MuonClip 作为一种稳健的解决方案，适用于稳定的大规模大型语言模型训练。

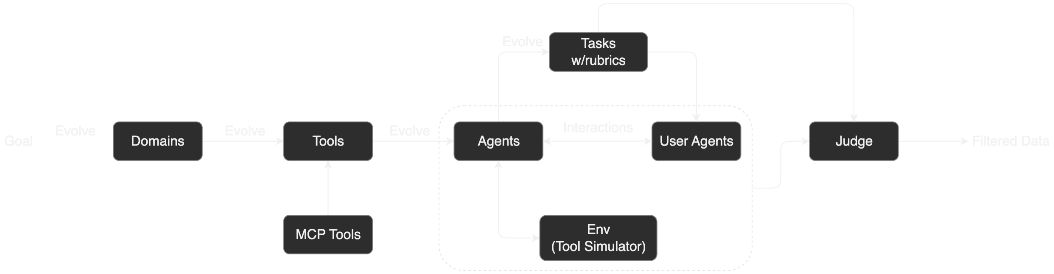


Agentic Capabilities

The enhanced agentic capabilities of Kimi K2 originate from two important aspects — large-scale agentic data synthesis and general reinforcement learning.

Large-Scale Agentic Data Synthesis for Tool Use Learning: To teach the model sophisticated tool-use capabilities, we developed a comprehensive pipeline inspired by ACEBench that simulates real-world tool-using scenarios at scale. Our approach systematically evolves hundreds of domains containing thousands of tools—including both real MCP (Model Context Protocol) tools and synthetic ones—then generates hundreds of agents with diverse tool sets.

All tasks are rubric-based, enabling consistent evaluation. Agents interact with simulated environments and user agents, creating realistic multi-turn tool-use scenarios. An LLM judge evaluates simulation results against task rubrics, filtering for high-quality training data. This scalable pipeline generates diverse, high-quality data, paving the way for large-scale rejection sampling and reinforcement learning.

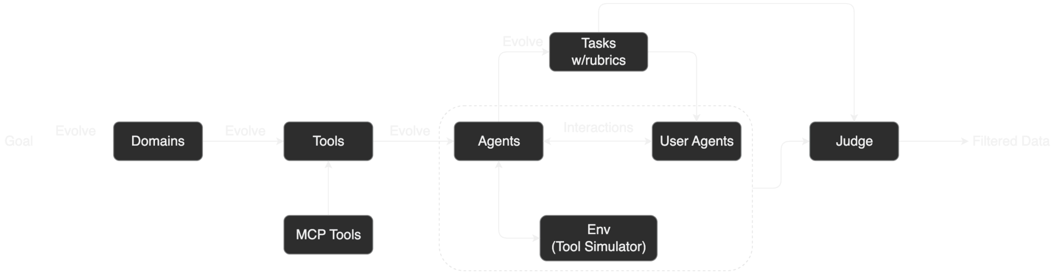


General Reinforcement Learning: The key challenge is to apply RL to tasks with both verifiable and non-verifiable rewards; typical examples of verifiable tasks are math and competition coding, while writing a research report is usually viewed as non-verifiable. Going beyond verifiable rewards, our general RL system uses a self-judging mechanism where the model acts as

Kimi K2 的增强代理能力源自两个重要方面——大规模代理数据合成和通用强化学习。

大规模代理数据合成用于工具使用学习：为了教授模型复杂的工具使用能力，我们开发了一个受ACEBench启发的全面流程，模拟大规模的真实世界工具使用场景。我们的方法系统地演化数百个包含数千个工具的领域——包括真实的MCP（模型上下文协议）工具和合成工具——然后生成数百个具有多样工具集的代理。

所有任务都基于评分标准，确保评估的一致性。代理与模拟环境和用户代理交互，创建逼真的多轮工具使用场景。一个大型语言模型（LLM）评判者根据任务评分标准评估模拟结果，筛选出高质量的训练数据。这个可扩展的流程生成多样且高质量的数据，为大规模拒绝采样和强化学习铺平了道路。



一般强化学习：关键挑战是将 RL 应用于具有可验证和不可验证奖励的任务；可验证任务的典型例子是数学和竞赛编码，而撰写研究报告通常被视为不可验证的任务。超越可验证奖励，我们的通用 RL 系统采用一种自我判断机制，其中模型充当

its own critic, providing scalable, rubric-based feedback for non-verifiable tasks.

Meanwhile, on-policy rollouts with verifiable rewards are used to continuously update the critic so that the critic keeps improving its evaluation accuracy on the latest policy. This can be viewed as a way of using verifiable rewards to improve the estimation of non-verifiable rewards.

Getting started with Kimi K2

Try Kimi K2 on kimi.com

Starting today, Kimi users on web and mobile can select and use the new Kimi K2 model for free. At this moment, our MCP features for web and app are still in development. We hope to begin rolling them out in the coming weeks. In the meantime, you're welcome to try our Researcher for an early look at its agentic capabilities. Please note that vision features are not supported for Kimi K2 yet.

Use Kimi K2 with API

The Kimi Platform offers an OpenAI/Anthropic compatible interface, allowing for easy adaptation of your existing applications to Kimi K2. We encourage developers to explore our tool calling API for building agent applications. For detailed information, visit platform.moonshot.ai.

Serve Kimi K2 on your own

We recommend running Kimi K2 on one of the following inference engines: vLLM, SGLang, KTransformers, or TensorRT-LLM. For detailed deployment instructions, please see our GitHub repository.

它自己的批评者，为不可验证的任务提供可扩展的、基于评分标准的反馈。

同时，使用带有可验证奖励的策略内滚动来不断更新评论者，以使评论者持续提高其在最新策略上的评估准确性。这可以看作是一种利用可验证奖励来改善对不可验证奖励的估计的方法。

开始使用Kimi K2

在 kimi.com 上试用 Kimi K2

从今天开始，Kimi 网页版和移动版的用户可以免费选择并使用新的 Kimi K2 模型。目前，我们的 MCP 功能在网页和应用程序上仍在开发中。我们希望在接下来的几周内开始逐步推出它们。与此同时，欢迎您试用我们的 Researcher，以提前了解其代理能力。请注意，视觉功能尚不支持 Kimi K2。

使用 Kimi K2 与 API

Kimi 平台提供一个与 OpenAI/Anthropic 兼容的接口，方便将您现有的应用程序轻松适配到 Kimi K2。我们鼓励开发者探索我们的工具调用 API，以构建代理应用程序。详细信息请访问 platform.moonshot.ai。

自己为 Kimi K2 服务

我们建议在以下推理引擎之一上运行Kimi K2：vLLM、SGLang、KTransformers或TensorRT-LLM。有关详细的部署说明，请参阅我们的GitHub仓库。

What's next

While Kimi K2 serves as a strong foundation for open agentic intelligence, a general agent uses more advanced capabilities such as thinking and visual understanding. We plan to add these to Kimi K2 in the future.

Limitations

In our internal tests, we've identified some limitations in current Kimi K2 models. When dealing with hard reasoning tasks or unclear tool definition, the model may generate excessive tokens, sometimes leading to truncated outputs or incomplete tool calls. Additionally, performance may decline on certain tasks if tool use is enabled. When building complete software projects, one-shot prompting yields performance degradation compared to using K2 under an agentic framework. We are working to address these issues in future releases and looking forward to more feedbacks.

下一步是什么

虽然 Kimi K2 作为开放代理智能的坚实基础，但通用代理还使用更先进的能力，如思考和视觉理解。我们计划在未来将这些能力加入到 Kimi K2 中。

限制

在我们的内部测试中，我们发现当前的 Kimi K2 模型存在一些局限性。当处理困难的推理任务或工具定义不明确时，模型可能会生成过多的标记，有时会导致输出被截断或工具调用不完整。此外，如果启用工具使用，在某些任务上的性能可能会下降。在构建完整的软件项目时，一次性提示的性能会比在代理框架下使用 K2 时有所下降。我们正在努力在未来的版本中解决这些问题，并期待更多的反馈。