Kimi-K2 简介和有意思的用法

2025年7月19日　5 分钟阅读

#AI #Kimi #Kimi-K2 #代码 #技术 #MoonshotAI

> *本文介绍了MoonshotAI公司Kimi-K2模型简介和相关有意思的用法。*

目录

目录

文章信息

字数

阅读时间

发布时间

更新时间

标签

#AI #Kimi #Kimi-K2 #MoonshotAI

Kimi-K2 简介

Kimi K2 模型由月之暗面（Moonshot AI）于 2025 年 7 月 11 日发布并开源，凭借其在 Agentic（任务代理）能力、编程效率与成本优势上的突破性表现，迅速引发全球 AI 业界高度关注。 Kimi-K2是一种大规模混合专家（MoE）语言模型，拥有1万亿参数及32亿激活参数，专为**工具使用、推理和自主问题解决**而优化，适用于知识、推理和编码任务等领域。

Homepage: https://www.moonshot.ai/

Huggingface 模型地址：https://huggingface.co/moonshotai/Kimi-K2-Instruct

Github Kimi-K2: https://github.com/moonshotai/Kimi-K2

Paper Link (coming soon)

Kimi Chat: https://www.kimi.com/

Kimi API: https://platform.moonshot.ai

每百万输入令牌 0.15 美元（缓存命中）

每百万输入令牌 0.60 美元（缓存未命中）

每百万输出令牌 2.50 美元

技术特点

主要特点：

通过Muon优化器在1万亿参数和15.5万亿Token上进行大规模训练，解决训练不稳定性问题。

提供基础模型（Kimi-K2-Base）和指令模型（Kimi-K2-Instruct），分别适用于细调及通用聊天。

强大的自主工具调用能力。

评估性能：

在多项指标如代码生成、工具使用、数学和通用任务中表现优秀，甚至在部分开源基准任务中达到全球SOTA。

部署：

提供开放兼容API，并支持多种推理引擎如vLLM和TensorRT-LLM。

部署示例和完整使用指南可在项目页获取。

许可：

模型权重和代码基于修改版MIT许可协议开放。

资源及联系信息：

提供技术博客、API接入、以及未来的论文链接。如有疑问，可邮件至 support@moonshot.cn。

这是一个强大的开源项目，旨在推进大语言模型的创新与应用。

行业评价

综合各大技术媒体、开发者社区及行业领袖的评价，业界对 Kimi K2 的认可主要集中在以下方面：

🏆 一、技术突破与性能表现

开源模型的性能新标杆

在多项权威测试中，Kimi K2 刷新开源模型 SOTA（State-of-the-Art），包括：

SWE Bench Verified（代码错误修复）：65.8% 准确率，超越 GPT-4.1（44.7%）[1][2]；

LiveCodeBench（交互式编程）：53.7% 准确率，显著领先 Claude 4（48.5%）[2]；

AceBench（语言理解）：80.1% 准确率，媲美顶级闭源模型[3]。

在 LMArena 大模型竞技场中位列开源第一、总榜第五，编程能力与 GPT-4.5 持平[4]。

万亿参数的训练稳定性突破

通过自研 MuonClip 优化器，成功控制 Attention logits 爆炸问题，完成 15.5T token 的无损训练（全程无 loss spike），被 AI 研究员 Cedric Chee 称为"机器学习史上最优美的损失曲线之一"[3][5]。

⚡ 二、成本效率与实用价值

"Claude 4 平替，成本仅 20%"

开发者实测显示，Kimi K2 在编程任务中性能接近 Claude 4，但 API 成本仅为后者的 1/5（输入 0.6/百万 token，输出 2.5/百万 token）[6][7]。

例如：生成完整前端组件库、处理 13 万行数据分析报告等复杂任务，“成本仅需几分钱”[8]。

硬件适配性优化

通过 MoE 稀疏架构（1T 总参数，仅激活 32B）和路由策略优化，Kim K2 可在非英伟达硬件流畅运行，被评价为 “压力下的创新”，可能动摇英伟达硬件垄断地位[6][8]。

🤖 三、Agentic 能力：重新定义“生产力工具”

首个完全开源的 Agentic 模型

支持 多轮工具调用、任务拆解与并行执行，例如：

自动分析数据生成统计报告（含图表解读）；

规划演唱会行程并发送邮件；

生成可直接执行的 ToolCall 结构[1][3]。

前 Anthropic 工程师 Pietro Schirano 称其为 “Claude 3.5 后唯一可在生产中放心使用的非 Anthropic 模型”[5][2]。

大规模合成数据驱动泛化能力

通过覆盖 数百领域、数千工具 的合成 pipeline 生成高质量训练数据，解决真实数据稀缺问题，大幅提升模型在复杂场景的泛化能力[4][8]。

🌍 四、开源生态与行业影响

全球开发者社区热捧

上线 20 分钟，Hugging Face 下载量破 1.2 万次，登顶热搜榜[7]；

OpenRouter 平台 token 使用量 发布两天超越 xAI（Grok 4），跻身全球前十[5][2]。

领军企业与学术界的认可

英伟达黄仁勋：公开称赞 Kimi K2 为 “全球最优秀推理模型之一”[1]；

Hugging Face 联合创始人 Thomas Wolf：称其“令人难以置信，开源模型正挑战闭源极限”[7]；

《自然》杂志：评价为 “又一个 DeepSeek 时刻”，推动全球开源生态进步[4][1]。

⚠️ 五、当前局限性与未来期待

短板：暂不支持多模态输入，数学推理能力弱于顶级闭源模型（如 o3 在 AIME 测试近满分，K2 为 49.5%）[6][2]；

未来方向：月之暗面计划加入思维链推理（CoT）与视觉理解能力，进一步扩展 Agentic 场景[3]。

💎 总结

Kimi K2 被业界视为 “行动派 AI” 的开源典范——不追求榜单刷分，而是聚焦 “执行力下沉”，以超高性价比推动 AI 从对话工具转向生产力引擎。其技术突破与生态贡献，不仅验证了中国 AI 团队的创新能力，更在全球范围内点燃了开源模型实用化的新浪潮。

参考文献

黄仁勋评论

社区反响

华尔街见闻

自然杂志报道

开发者实测报告

成本分析文章

Hugging Face动态

CSDN技术博客

有意思的功能

Partial

在使用大型语言模型时，有时我们希望通过预先填充部分响应来引导模型的输出。在 Kimi 大型语言模型中，我们提供了“部分模式”来实现这一功能。它有助于我们控制输出格式、引导内容，并在角色扮演场景中保持更好的一致性。你只需在最后一个以助手角色发送的消息条目中添加 “partial”： True 即可启用“部分模式”。

```python
from openai import OpenAI

client = OpenAI(
    api_key="$MOONSHOT_API_KEY",
    base_url="https://api.moonshot.ai/v1",
)

completion = client.chat.completions.create(
    model="moonshot-v1-32k",
    messages=[
        {
            "role": "system",
            "content": "Extract the name, size, price, and color from the product description and output them in a JSON object.",
        },
        {
            "role": "user",
            "content": "The DaMi SmartHome Mini is a compact smart home assistant available in black and silver. It costs 998 yuan and measures 256 x 128 x 128mm. It allows you to control lights, thermostats, and other connected devices via voice or app, no matter where you place it in your home.",
        },
        {
            "role": "assistant",
            "content": "{",
            "partial": True    # <<<<< 关键点
        },
    ],
    temperature=0.3,
)

print('{'+completion.choices[0].message.content)
```

1. Partial Mode 的本质作用

当你在最后一个 assistant 消息中设置 `"partial": true`，**你是在告诉模型：这一条回复只是“部分内容”，后续还会继续补充。**这会引发模型在生成时的“引导”行为，具体表现为：

> **模型会以你提供的内容为“开头”，继续补全剩余内容**；
>
> 你可以精确控制输出的开头格式、风格，甚至是结构（比如 JSON 的左大括号、角色扮演的前缀等）；
>
> 这对于需要严格格式化输出、或需要模型“在某种状态下”继续生成的场景非常有用。

2. Partial Mode 的典型效果和用法

场景一：结构化内容补全（如 JSON）

假设你希望模型输出标准 JSON，但又怕它乱加前缀或格式出错。你可以这样做：

```
{
  "role": "assistant",
  "content": "{",
  "partial": true
}
```

此时模型会以 `{` 为起点，**只补全后续 JSON 字段**，而不会再多加内容或乱改结构。你拿到的结果会更容易拼接和解析。

场景二：角色扮演一致性

比如你要让模型扮演某个角色，可以这样：

```
{
  "role": "assistant",
  "name": "Dr. Kelsier",
  "content": "",
  "partial": true
}
```

这样模型生成的回复就会以 Dr. Kelsier 的身份、风格进行，**不会自动加多余的开头**，角色一致性更好。

场景三：多轮对话中的格式引导

你可以动态地在每轮对话中根据实际需要，**预填充一部分内容**，让模型在这个基础上继续生成，确保风格/格式/内容不跑偏。

3. Partial Mode 的底层原理

Prompt 拼接控制：本质上，相当于你把一段"前导文本"强制塞给模型，模型只负责"续写"。

防止模型自作主张：在没有 partial 时，模型有时会自动添加解释、前缀或格式，导致输出不可控；有了 partial，模型会更"听话"地只做补全。

流式输出友好：对于需要流式生成（如实时聊天、代码补全等），partial 可以让你逐步引导模型输出，提升交互体验。

4. 注意事项

不要和 response_format=json_object 混用，否则模型可能混淆结构，导致输出异常。

API 返回内容不包含你预填的 leading_text，你需要自己拼接完整输出。

5. 脑洞大开的建议

你可以结合 partial mode 实现以下高级玩法：

多 Agent 协同写作：每个 Agent 先写一段开头，后续用 partial 让模型自动续写，形成风格统一的长文或剧本。

AI 自动代码补全 IDE：你先给出部分函数头，用 partial 让模型只补全函数体，保证代码格式和风格一致。

多模态输出引导：在多模态场景里，先让模型生成图片描述的开头，再 partial 续写详细内容，实现分段式内容生成。

AI 安全对齐：通过 partial 预填"安全声明"或"免责声明"，让模型每次输出都带有合规性前缀，降低风险。

总结一句话：
`partial: true` 让你像"牵着模型的手"一样，精准引导它的输出起点和风格，特别适合高格式化、角色扮演和多轮交互等场景，是高级工程师玩转大模型的利器！

自动选择合适上下文窗口（context window）模型

参　考　：　https://platform.moonshot.ai/docs/guide/choose-an-appropriate-kimi-model

1. 背景与需求分析

问题本质：多轮对话中，历史消息不断增长，Token数可能超过当前模型的最大Token限制（如8k、32k、128k）。

手工方案：提前计算Token数，选择合适的模型，但不灵活且易出错。

自动化目标：根据当前对话上下文的Token数，自动选择合适的Kimi大模型，既节省成本又避免超限报错。

2. 官方推荐方案

Moonshot官方已提供**moonshot-v1-auto**模型，作为"模型路由器"：

工作机制：自动检测当前上下文Token数，选择8k、32k或128k模型。

调用方式：与普通模型一致，仅需将model参数改为`moonshot-v1-auto`即可。

计费规则：按最终实际选用的模型计费。

代码示例：

```
from openai import OpenAI

client = OpenAI(
    api_key = "MOONSHOT_API_KEY", # Replace MOONSHOT_API_KEY with the API Key you obtained from the Kimi Open Platform
    base_url = "https://api.moonshot.ai/v1",
)
messages.append({
  "role": "user",
  "content": input,
})

# We engage in a conversation with the Kimi large language model, carrying the messages along
completion = client.chat.completions.create(
        model="moonshot-v1-auto",  # <-- Note the change here, from moonshot-v1-8k to moonshot-v1-auto
        messages=messages,
        temperature=0.3,
    )

# Through the API, we obtain the response message (role=assistant) from the Kimi large language model
   assistant_message = completion.choices[0].message

   # To ensure the Kimi large language model has a complete memory, we must also add the message it returns to us to the messages list
   messages.append(assistant_message)

   return assistant_message.content
# Through the API, we obtain the response message (role=assistant) from the Kimi large language model
   assistant_message = completion.choices[0].message

   # To ensure the Kimi large language model has a complete memory, we must also add the message it returns to us to the messages list
   messages.append(assistant_message)

   return assistant_message.content
```

Kimi-K2 API

它同时支持与 OpenAI 兼容和与 Anthropic 兼容的 API。 这是使用它替代 OpenAI 模型所需的全部 Python 代码。

API key

国内API主页：https://platform.moonshot.cn/console/account #我用的是这个。 国外API主页：https://platform.moonshot.ai/console/account

代码示例

```python
from openai import OpenAI

client = OpenAI(
    api_key="$MOONSHOT_API_KEY",
    base_url="https://api.moonshot.ai/v1",
)

completion = client.chat.completions.create(
    model="kimi-k2-0711-preview",
    messages=[
        {"role": "system", "content": "You are Kimi, an AI assistant provided by Moonshot AI. You are proficient in Chinese and English conve"},
        {"role": "user", "content": "Hello, my name is Li Lei. What is 1+1?"}
    ],
    temperature = 0.3,
)

print(completion.choices[0].message.content)
```

使用 Tools

```
from openai import OpenAI

client = OpenAI(
    api_key = "$MOONSHOT_API_KEY",
    base_url = "https://api.moonshot.ai/v1",
)

completion = client.chat.completions.create(
    model = "moonshot-v1-8k",
    messages = [
        {"role": "system", "content": "You are
Kimi, an AI assistant provided by Moonshot AI,
who is more proficient in Chinese and English
conversations. You will provide users with safe,
helpful, and accurate answers. At the same time,
you will reject any questions involving
terrorism, racism, pornography, and violence.
Moonshot AI is a proper noun and should not be
translated into other languages."},
        {"role": "user", "content": "Determine
whether 3214567 is a prime number through
programming."}
    ],
    tools = [{
        "type": "function",
        "function": {
            "name": "CodeRunner",
            "description": "A code executor that
supports running Python and JavaScript code",
            "parameters": {
                "properties": {
                    "language": {
                        "type": "string",
                        "enum": ["python",
"javascript"]
                    },
                    "code": {
                        "type": "string",
                        "description": "The code
is written here"
                    }
                },
            "type": "object"
            }
        }
    }],
    temperature = 0.3,
)

print(completion.choices[0].message)
```

集成到AI 辅助编程助手

集成到Cline： https://platform.moonshot.ai/docs/guide/agent-support#using-kimi-k2-model-in-cline

集成到Roocode： https://platform.moonshot.ai/docs/guide/agent-support#using-kimi-k2-model-in-roocode

集成到Cursor： https://platform.moonshot.ai/docs/guide/agent-support#using-kimi-k2-model-in-cursor

集成到Claude Code CLI

分享这篇文章

相关文章推荐

> **DeepSeek FlashMLA 代码..**

本文介绍了深度求索（ DeepSeek ） 公 ...

> **Google I/O 2025 大会亮点**

本文介绍了Google I/O 2025 大会亮点。

> **多智能体强化学习（MARL）在..**

本文介绍了多智能体强化学习（MARL）在...