



南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY



DeepSeek本地化部署

南方科技大学
唐 博
tangb3@sustech.edu.cn



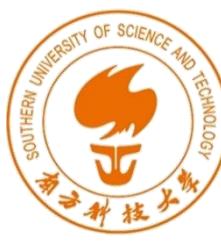


内容提纲



- 大模型DeepSeek历史简介
 - 从ChatGPT到DeepSeek
- 本地化部署DeepSeek模型
 - 理解DeepSeek-R1/V3模型参数
 - 本地化部署DeepSeek蒸馏模型
- 调用DeepSeek的API服务
 - 开发专有大模型APP: LatexChecker

此课件中部分内容由李润中同学和DBGroup部分成员帮忙准备，特此致谢！



内容提纲



✓ 大模型DeepSeek历史简介

- 从ChatGPT到DeepSeek

□ 本地化部署DeepSeek模型

- 理解DeepSeek-R1/V3模型参数

- 本地化部署DeepSeek蒸馏模型

□ 调用DeepSeek的API服务

- 开发专有大模型APP: LatexChecker



从 ChatGPT 说起



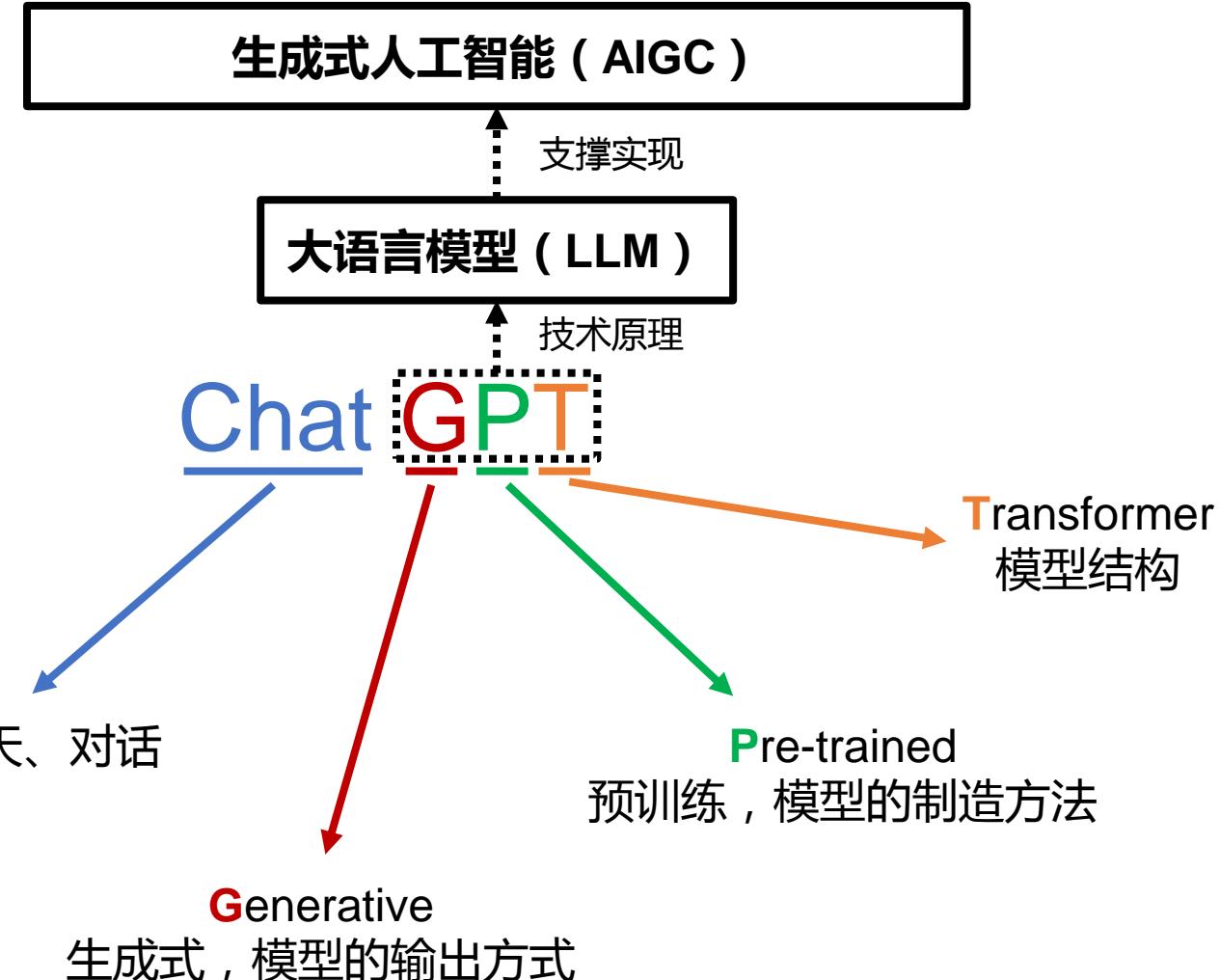
2022 年 11 月 30 日

OpenAI 正式发布了基于 GPT-3.5 模型的 ChatGPT

里程碑事件：把 AI 从“天上”带到了“人间”

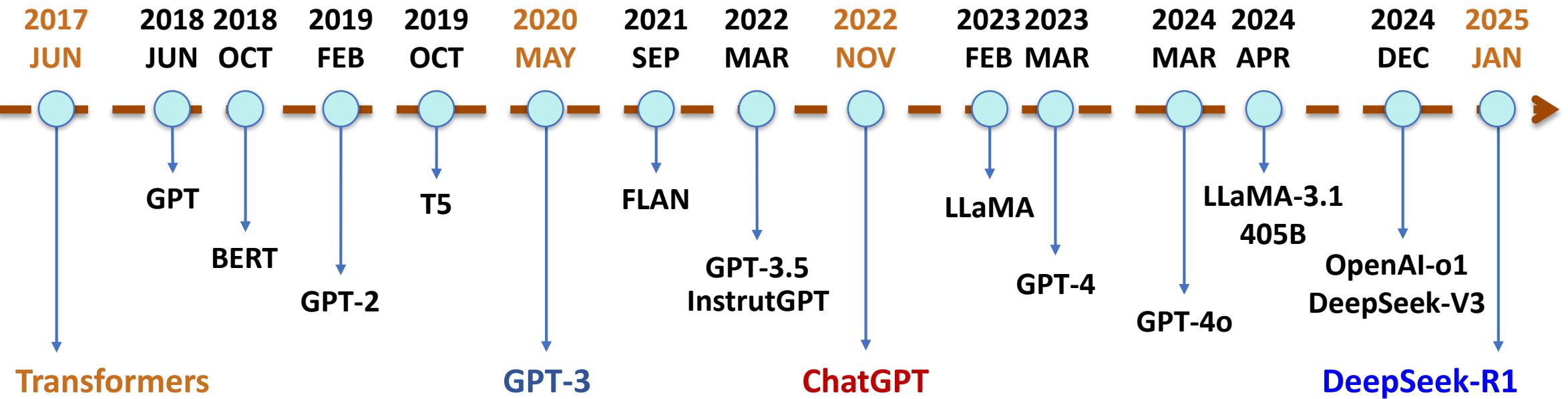


大语言模型的基础概念





大语言模型(LLM)简史





大模型工作原理



OpenAI



LLaMA



deepseek



Qwen



智谱·AI



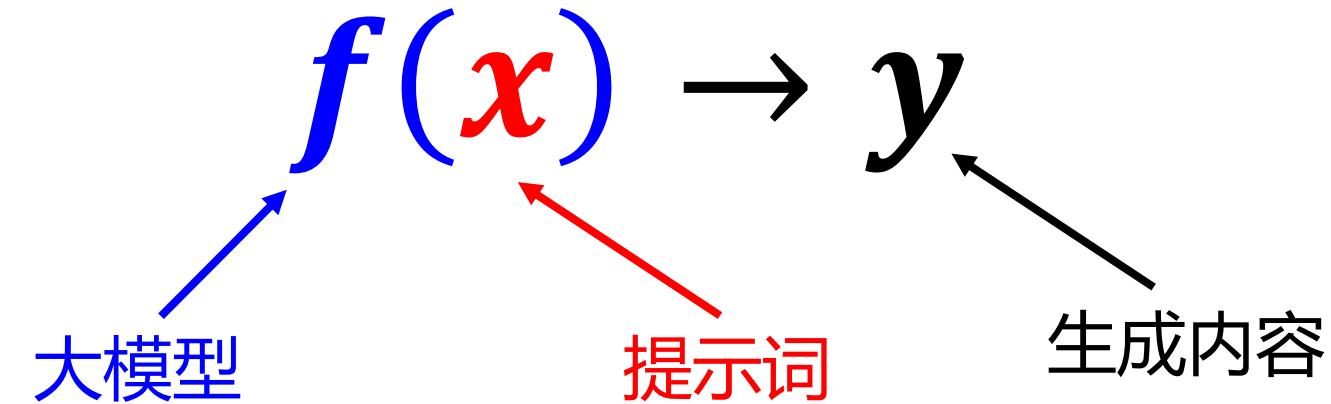
面壁智能



Midjourney



Colossal-AI





大模型工作原理示例



大模型 $f()$



返回结果 y

什么样的学生适合报考南科大？

Thought for 20 seconds

南方科技大学（简称“南科大”）作为一所创新型、研究型大学，自2011年建校以来，凭借其独特的办学理念和优质资源吸引了众多学生。以下从多个维度分析适合报考南科大的学生类型，帮助判断是否与学校特点匹配：

一、学术兴趣与学科方向

- 理工科优势领域**
南科大以理工医商为主，尤其在材料科学、计算机、电子工程、生物医学等前沿学科领域具有较强实力。例如，化学、材料科学在ESI排名中进入全球前1%。若学生对人工智能、新能源、生物技术等新兴领域有浓厚兴趣，南科大的科研平台（如粤港澳大湾区量子科学中心）能提供优质资源。
- 交叉学科探索**
学校推行“通识+专业教育”，鼓励跨学科学习。例如“金融数学”“生物医学工程”等交叉专业，适合希望突破传统学科边界的学生。

二、学习模式适应性

- 研究导向型学习**
本科生即可进入实验室参与课题（如大二开始的“URP计划”），2022年本科生发表SCI论文超百篇。适合热衷科研、希望早进实验室的学生。
- 国际化培养**
全英文专业课程占比超60%，与麻省理工、剑桥等高校联合培养。需具备较强英语能力（高考英语120+/150分制较有竞争力），并有志于海外深造者（近年境外升学率约40%）。

提示词 x



大语言模型工作原理



提示词
 x

什么样的学生适合报考南科大？

大模型
 $f(x)$



生成内容
 y

南方科技大学（简称“南科大”）作为一所创新型、研究型大学，自2011年[...]



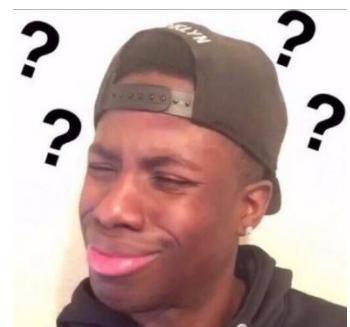
大语言模型原理



文字接龙

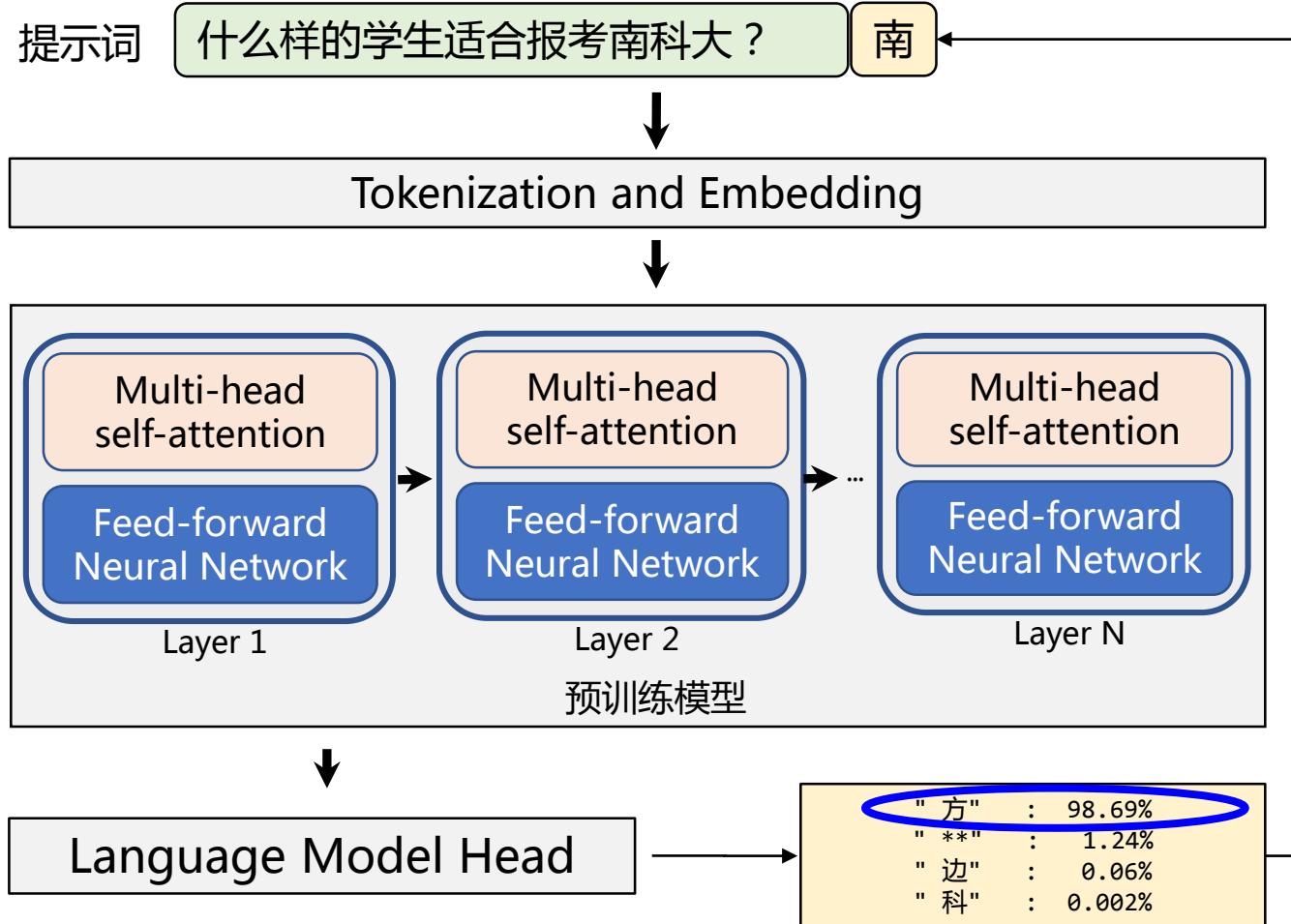


鸡饭团体操心上进步...





大语言模型：靠谱的文字接龙





大语言模型两阶段:训练和推理



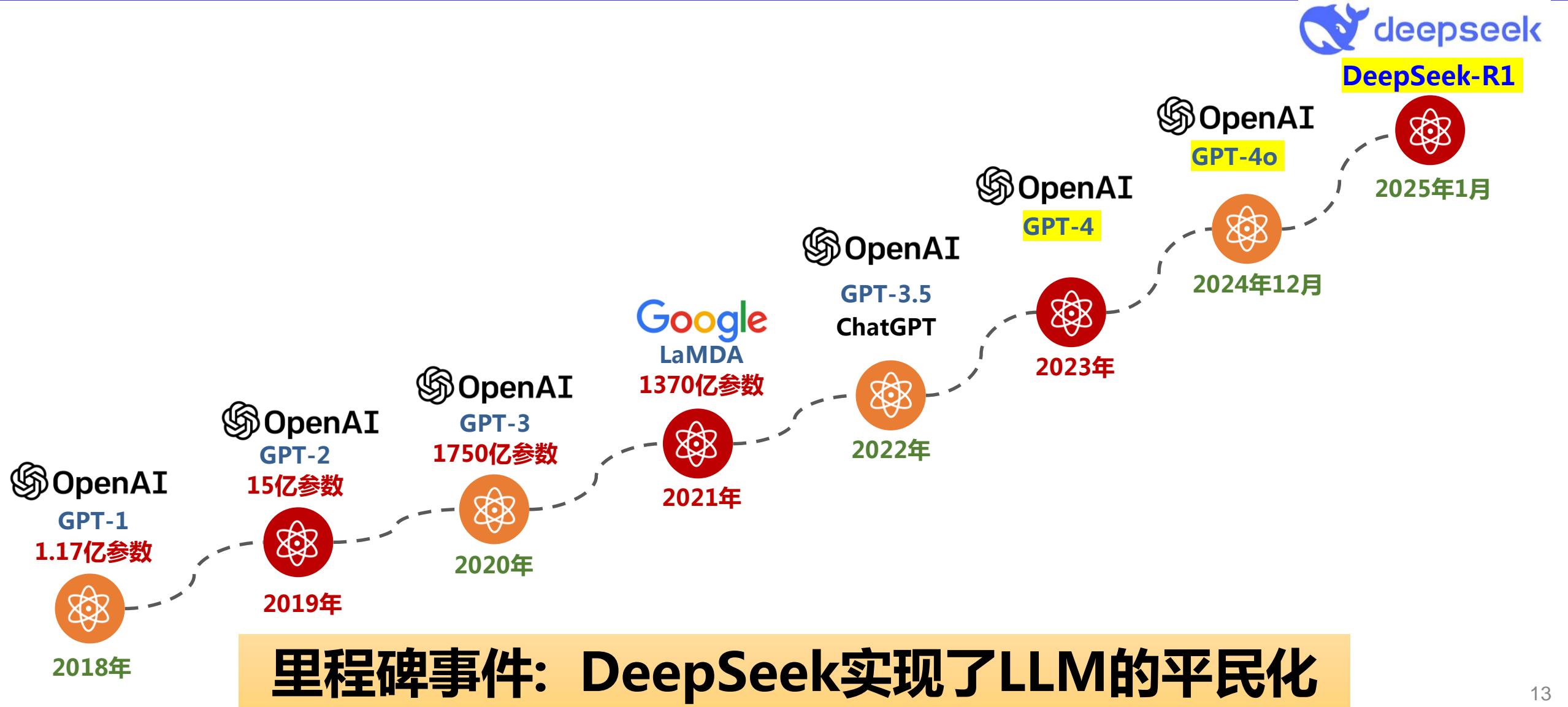
模型训练

模型推理

行业共识: 训练数据越多,训练开销更昂贵,大模型能力更强



LLM历史中的东方神秘力量





DeepSeek模型发展历程



- 2019年,幻方量化跻身**百亿私募**,幻方AI成立 (500卡): **1万块A100是准入门槛**
- 2021年,投入增加到10亿,搭载约**10000块**英伟达A100: **国内不超过5家有1万枚**
- 2023年7月,DeepSeek公司成立,致力于AGI: **量化挣钱, AI追梦**
- 2023年11月,**开源**DeepSeek LLM 7B和67B模型: **开源基因,严谨求证**
- 2024年5月,**开源**DeepSeek-V2系列模型: **重大创新,效果显著**
- 2024年12月,**开源**DeepSeek-V3系列模型: **基座模型SOTA**
- 2025年1月,**开源推理模型**DeepSeek-R1: **勇于原始创新,敢入无人之境**



DeepSeek的历史意义



南科大排名从国内第十到国内前三需要做哪些努力



Thinking... ^

好的，用户问南方科技大学要从国内第十提升到前三需要做哪些努力。首先，我需要了解南科大目前的状况。根据之前的对话，南科大是一所创新型、研究型大学，成立于2011年，以理工医商为主，采用综合评价录取，国际化程度高，科研平台不错，本科生有参与科研的机会，毕业生深造率也高。但可能还存在一些不足，比如文科规模小，传统名校光环不足。

用户现在关心的是排名提升，国内前三通常是清北复交浙这样的顶尖学府。南科大作为年轻大学，要进入前三挑战很大。需要分析排名指标，通常大学排名考虑学术声誉、科研产出、师资力量、国际化、学生质量、资金投入等。

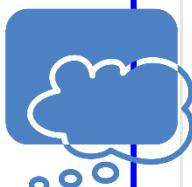
首先，学术声誉和科研产出方面，南科大已经有一定基础，比如材料科学进入ESI前1%，但可能需要在更多学科领域取得突破，尤其是国际顶尖期刊论文的数量和影响力。可能需要引进更多顶尖学者，增加科研经费，建立国家级重点实验室。

其次，师资力量是关键。虽然南科大可能已经有一些优秀教师，但要达到前三，可能需要更多国际知名的教授，提高师生比，吸引诺奖级学者或者院士级别的教授加入。

国际化程度方面，南科大已经和麻省理工、剑桥有合作，但可能需要进一步扩大国际合作，增加国际学生和教师的比例，提升全英文课程的数量和质量，参与更多国际联合研究项目。

学生质量方面，虽然现在采用631模式，但可能需要提高录取标准，吸引更多高分考生，同时加强本科生的科研和实践机会，提升毕业生就业和深造的质量，尤其是进入世界顶尖学府的比例。

资金投入也是重要因素。南科大位于深圳，有地域优势，可能需要争取更多政府拨款和企业合作，增加校友捐赠，提升资金实力以支持科研和教学。



南方科技大学（简称“南科大”）若要从国内第十名左右跃升至前三名（对标清北复交浙等顶尖高校），需在学科建设、科研实力、师资队伍、人才培养、国际化、社会声誉等多维度实现跨越式突破。以下从具体策略角度展开分析：

一、学科布局：聚焦前沿领域，打造“不可替代性”

1. 集中资源突破“卡脖子”领域

- 聚焦方向：在量子计算、人工智能、合成生物学、新能源材料等国家战略领域建立“学科高地”，例如建立国家级实验室（对标中国科大“量子信息国家实验室”）。
- 案例参考：西湖大学以生物医学、人工智能为突破口，5年内建成10个世界级研究中心，快速提升国际影响力。

2. 交叉学科创新机制

- 设立跨学科研究院（如“智能科学与医学工程研究院”），打破院系壁垒，推动“AI+生物”“材料+能源”等融合方向，产出颠覆性成果。

二、科研实力：从“量”到“质”的跃升

1. 提升原创性研究比重

- 鼓励教授在《Nature》《Science》《Cell》等顶刊发表论文，设立专项奖励（如港科大对顶刊论文奖励50万/篇）。
- 数据对标：2023年清华在NS主刊发文58篇，南科大需从当前年均10篇左右向30篇以上突破。

2. 技术转化与产业赋能

- 依托深圳产业优势，与大疆、华为、比亚迪共建联合实验室，推动专利商业化（参考斯坦福-硅谷模式）。
- 目标：技术转化率从目前的15%提升至30%，孵化独角兽企业3-5家。



历史意义一：首个展示思维链过程的推理模型



DeepSeek的历史意义

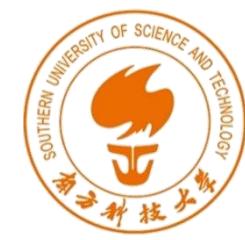


模型 ⁽¹⁾	deepseek-chat	deepseek-reasoner
上下文长度	64K	64K
最大思维链长度 ⁽²⁾	-	32K
最大输出长度 ⁽³⁾	8K	8K
标准时段价格 (北京时间 08:30-00:30)	百万tokens输入 (缓存命中) ⁽⁴⁾	0.5元
	百万tokens输入 (缓存未命中)	2元
	百万tokens输出 ⁽⁵⁾	8元
优惠时段价格 ⁽⁶⁾ (北京时间 00:30-08:30)	百万tokens输入 (缓存命中)	0.25元 (5折)
	百万tokens输入 (缓存未命中)	1元 (5折)
	百万tokens输出	4元 (5折)

DeepSeek-R1模型API价格为每百万tokens输入约为**4元**，每百万tokens输出价格为**16元**，网页聊天**完全免费**！

ChatGPT-o1模型API价格为每百万tokens输入约为**55元**，每百万tokens输出价格为**438元**，网页聊天需要**240美元/年的会员**才能使用

历史意义二：打破OpenAI的垄断,价格亲民



DeepSeek的历史意义



DeepSeek-R1 Public

Watch 583

main · 1 Branch · 0 Tags

Go to file Add file · Code

DeepSeekDDM	Delete CITATION.cff	95aaec7 · 4 days ago	30 Commits
.github/workflows	Apply suggestions from code review	3 weeks ago	
figures	Release DeepSeek-R1	last month	
DeepSeek_R1.pdf	Update README.md & DeepSeek_R1.pdf	last month	
LICENSE	Release DeepSeek-R1	last month	
README.md	Update bib info	4 days ago	

README MIT license

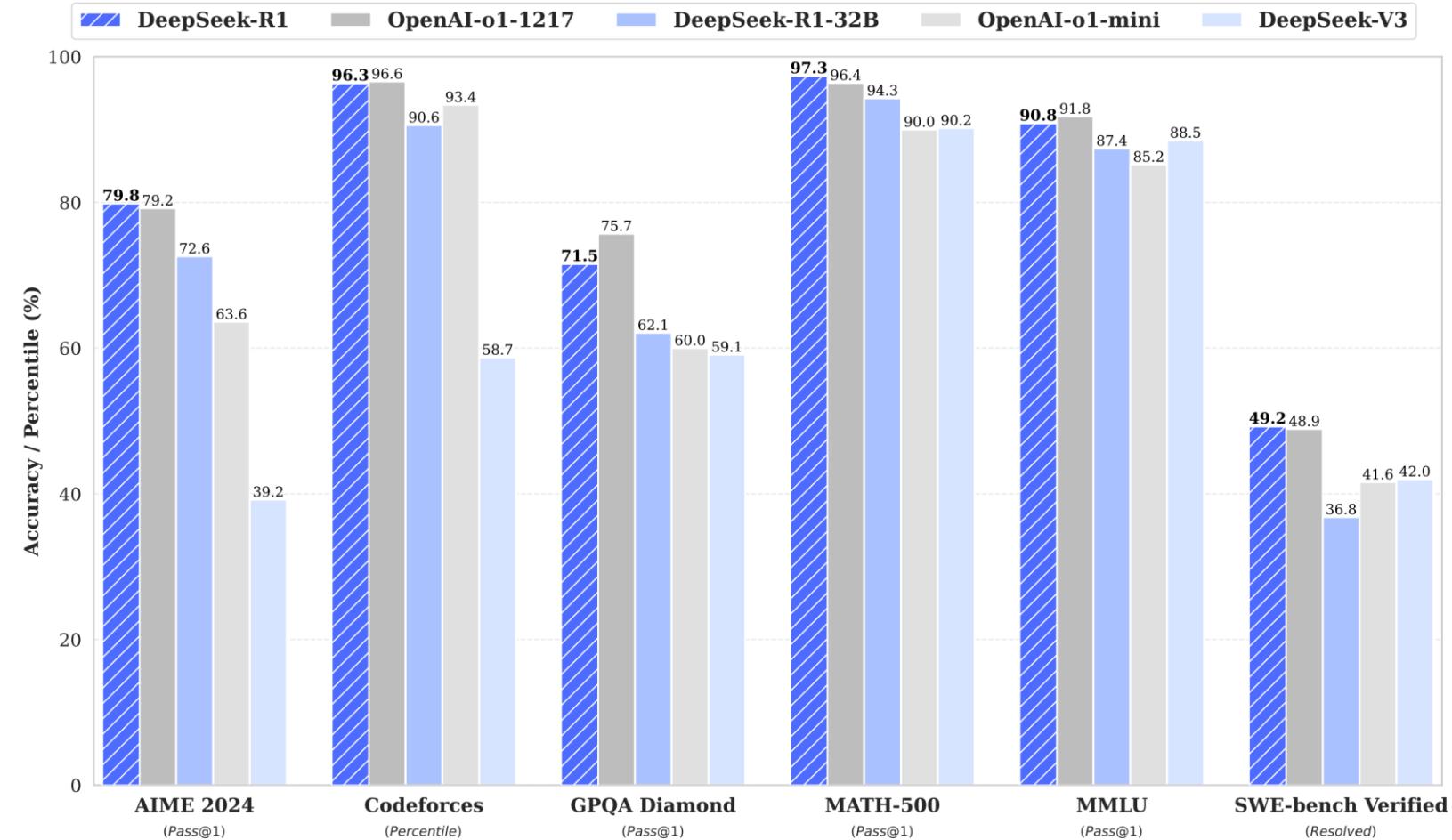
DeepSeek-R1

 deepseek

历史意义三：首个开源的推理大模型



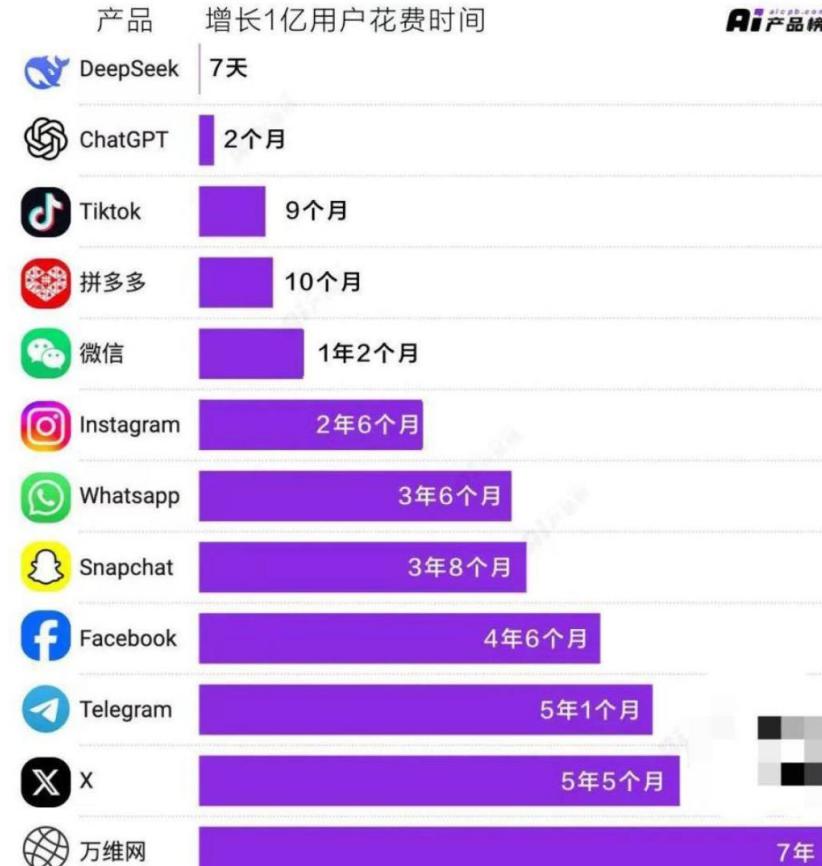
DeepSeek的历史意义



历史意义四: 源于中国，自主创新，领先世界



DeepSeek的历史意义



注：DeepSeek 包含网站Web/应用App累加不去重，Tiktok 不包含国内版抖音

数据来源：AI产品榜 aicpb.com 感谢邓瑞恒提供作图思路

历史意义五：AI平民化,显著提升生产效率



内容提纲



- ✓ 大模型DeepSeek历史简介
 - 从ChatGPT到DeepSeek
- ✓ 本地化部署DeepSeek模型
 - 理解DeepSeek-R1/V3模型参数
 - 本地化部署DeepSeek蒸馏模型
- 调用DeepSeek的API服务
 - 开发专有大模型APP: LatexChecker



DeepSeek-R1 模型参数



DeepSeek-R1 Models

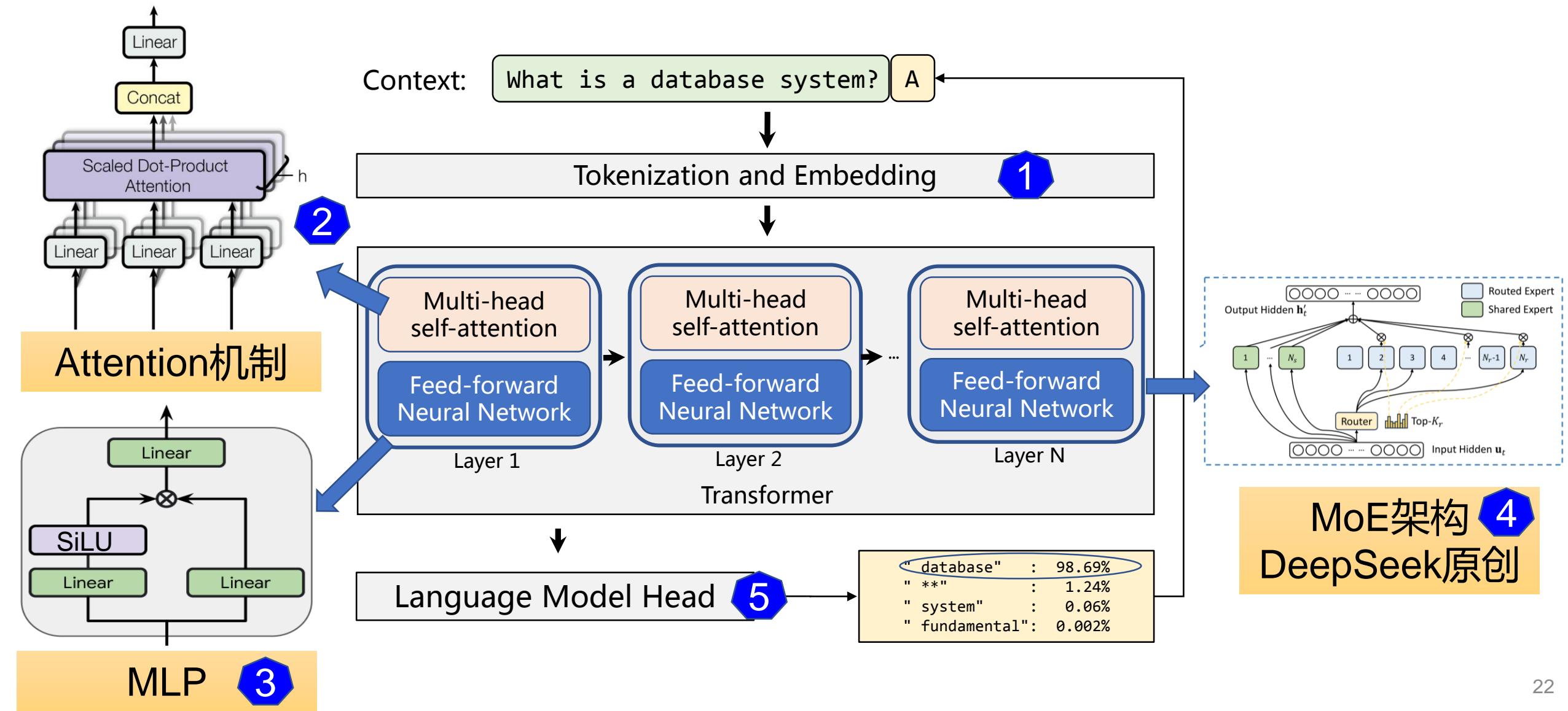
Model	#Total Params	#Activated Params	Context Length	Download
DeepSeek-R1-Zero	671B	37B	128K	HuggingFace
DeepSeek-R1	671B	37B	128K	HuggingFace

DeepSeek-R1-Zero & DeepSeek-R1 are trained based on DeepSeek-V3-Base. For more details regarding the model architecture, please refer to [DeepSeek-V3](#) repository.

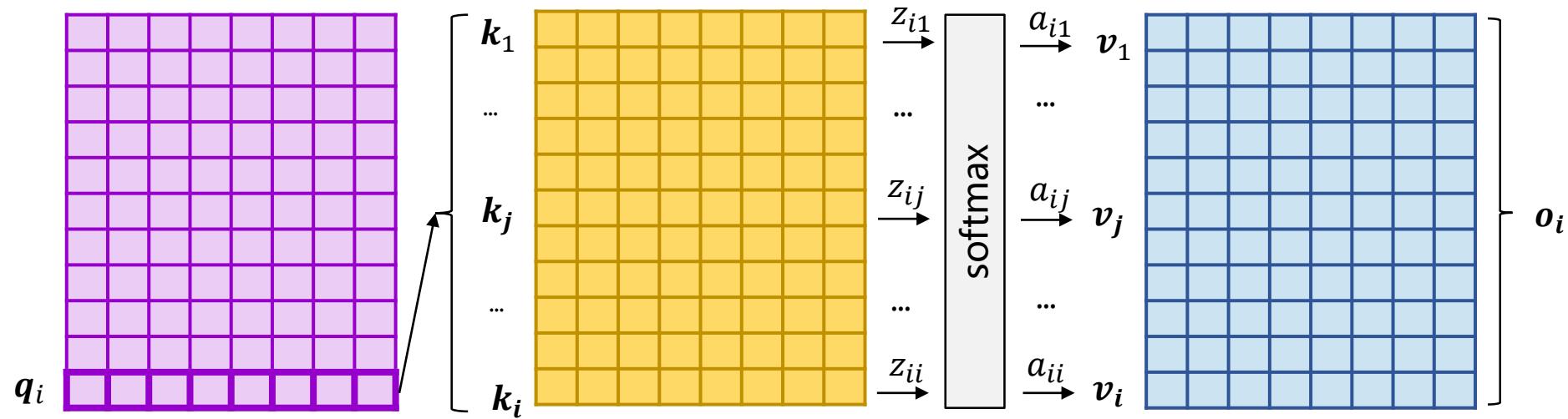
- 什么是总参数量？什么是激活参数量？
- R1为什么总参数量是671B，激活参数量是37B？



LLM基本工作流程及核心组件



Self-attention



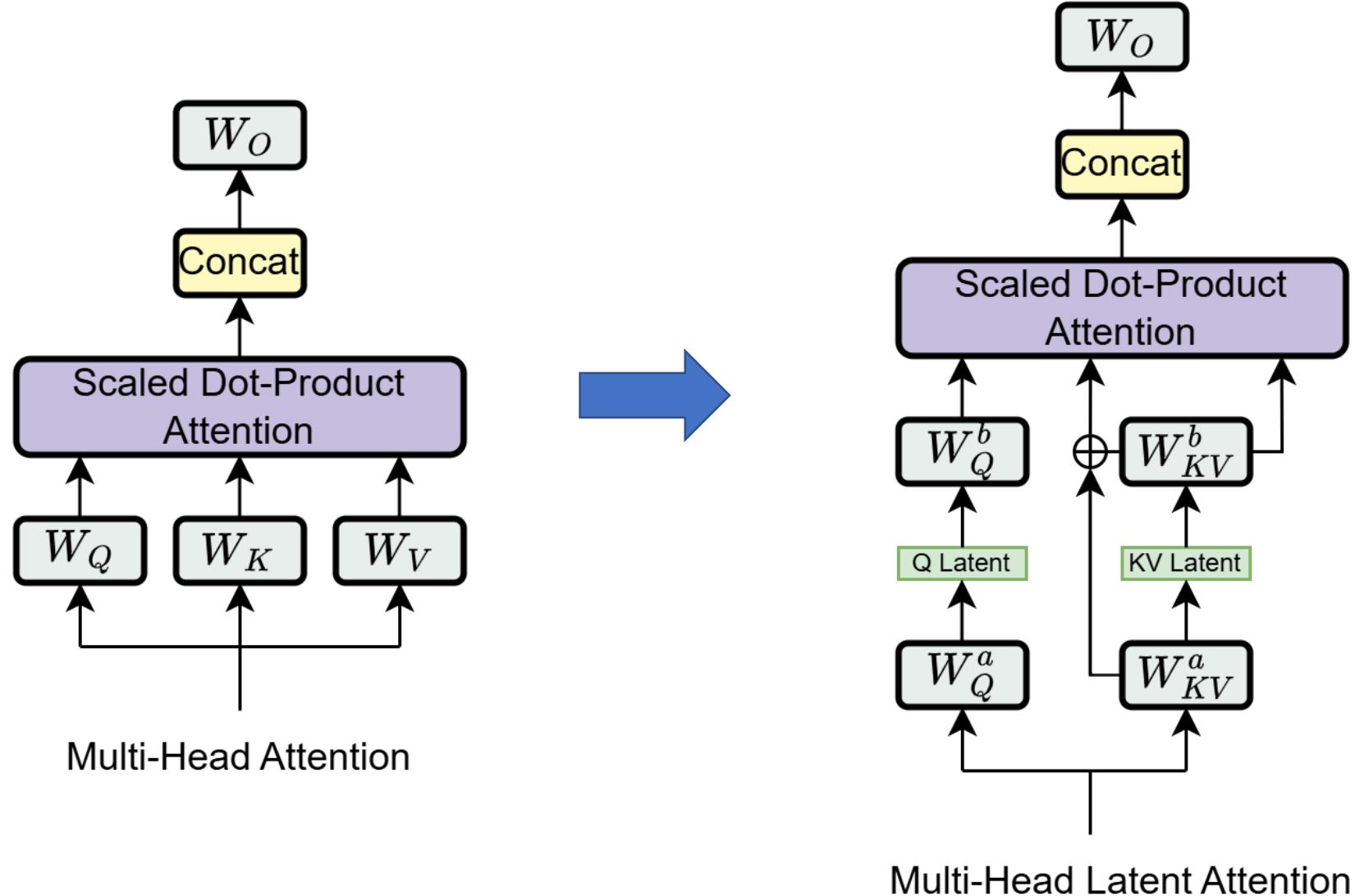
Query matrix
 $Q = X W_Q$

Key matrix
 $K = X W_K$

Value matrix
 $V = X W_V$

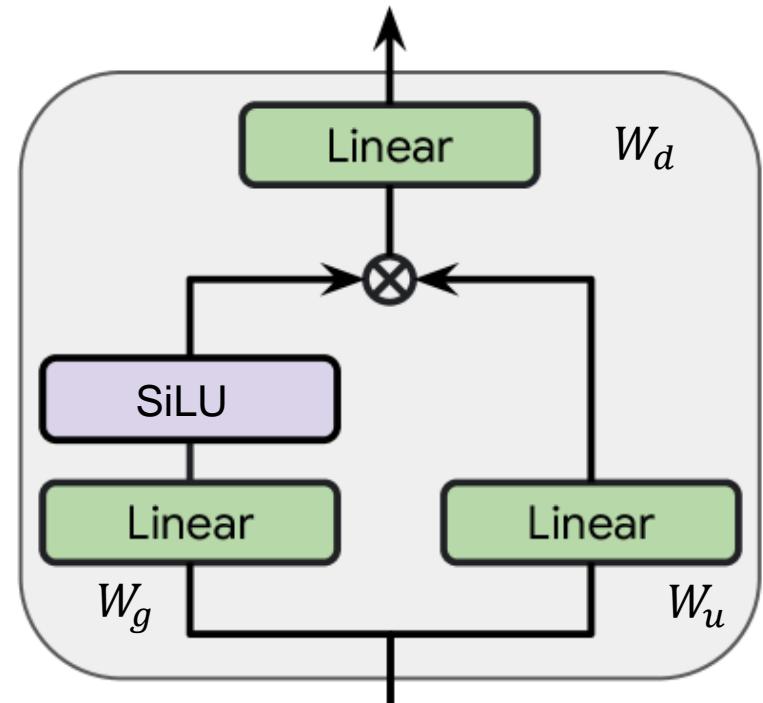
$$o_i = \text{softmax}\left(\frac{\mathbf{q}_i \mathbf{K}^T}{\sqrt{d}}\right) \mathbf{V}$$

DeepSeek 原创：MLA



FFN: MLP 非线性变换

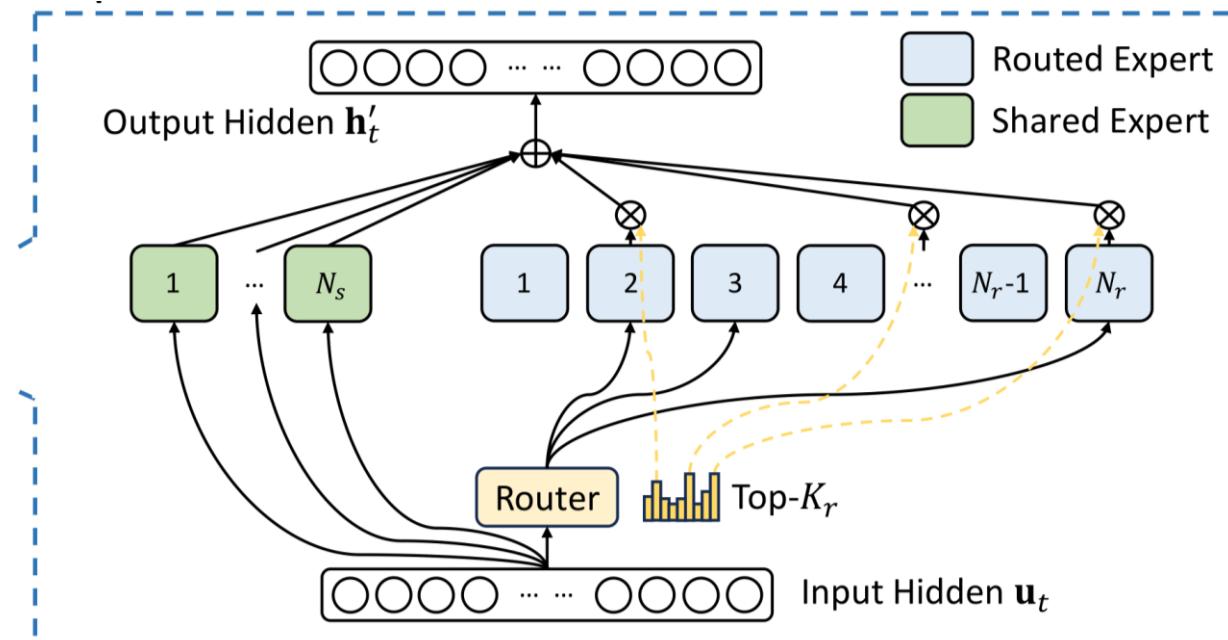
- d 维向量通过非线性变换转成新的 d 维向量
- DeekSeek前三层FFN使用MLP
- MLP: $y = (\sigma(xW_g) \otimes xW_u)W_d$
- W_g , W_u 和 W_d 是三个可学习参数。
- 一个MLP的参数数量为 $3 \times d \times d_i$



FFN: MoE (DeepSeek后 58层用)

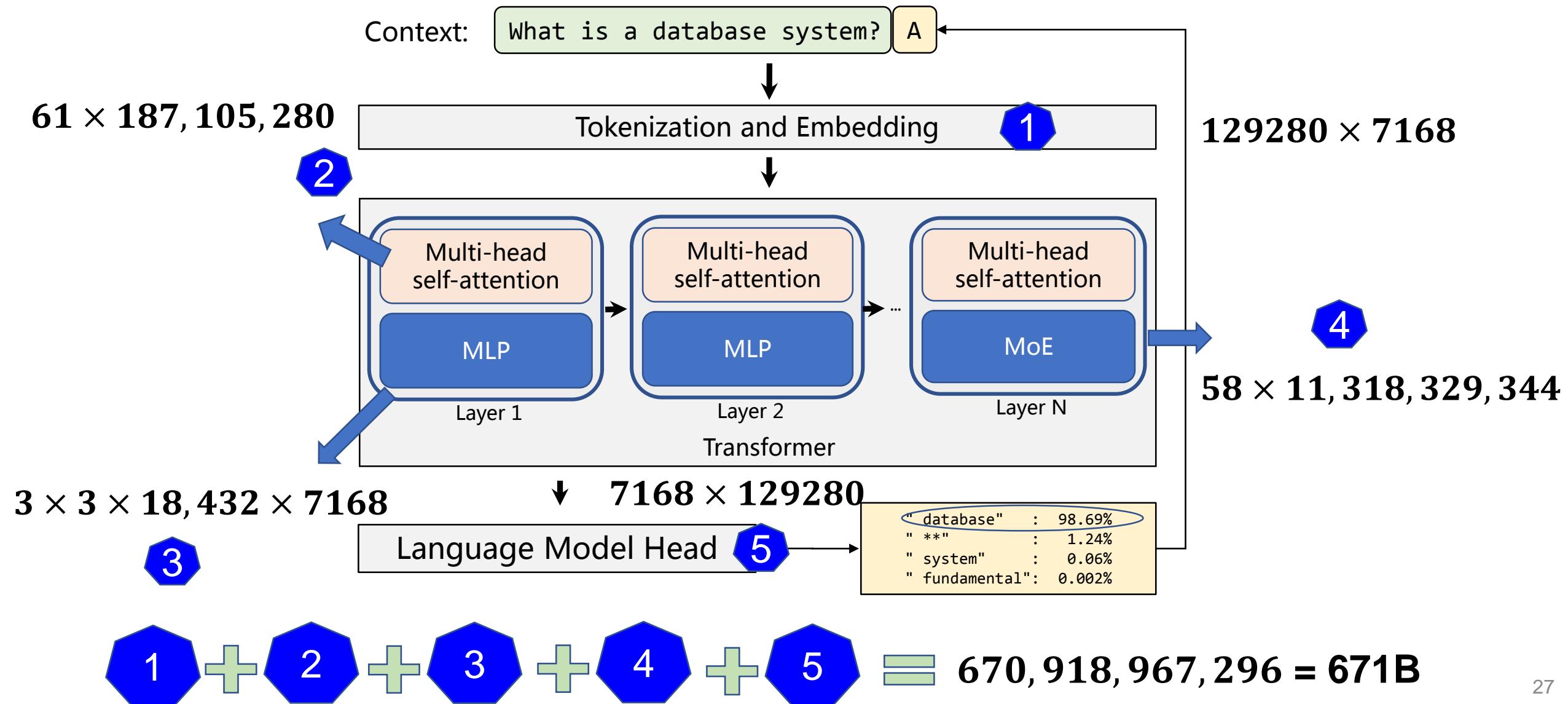


- Mixture-of-Experts: 一个MoE由多个MLP组成, 据说GPT-4用了MoE技术
- 每一个MoE共有257个MLP, 总参数量 : $257 \times 3 \times d \times d_i$
- 生成每个token时每一个MoE会用到9个, 激活参数量 : $9 \times 3 \times d \times d_i$





DeepSeek 总参数量





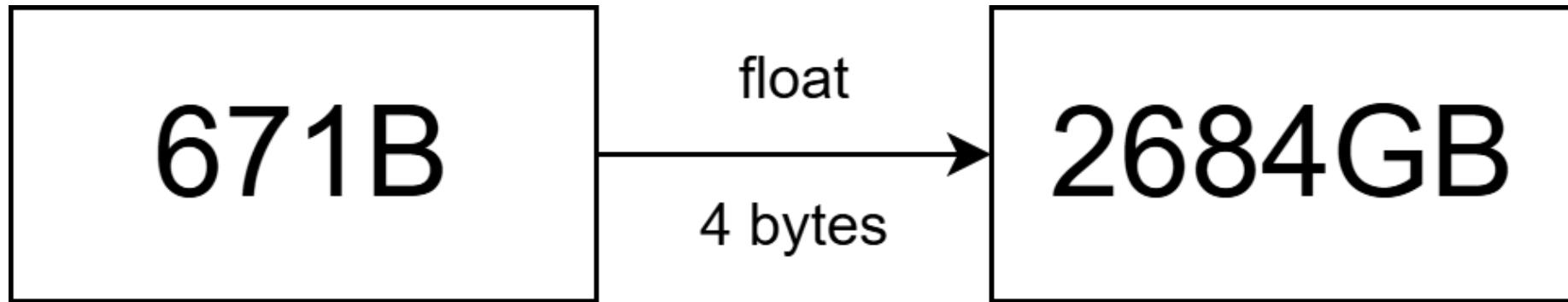
满血版DeepSeek有671B参数



需要多大的内存才能存下671B参数？



671B参数需要多大内存？



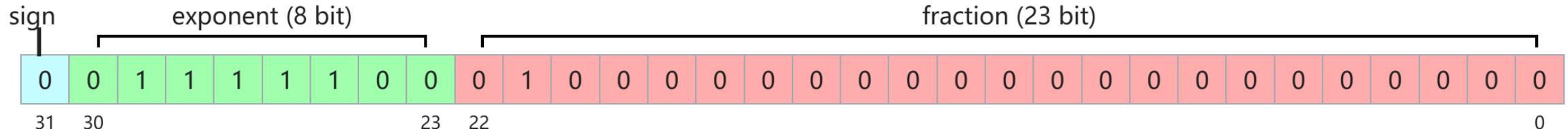
计算机导论告诉我们一个float需要4个字节？



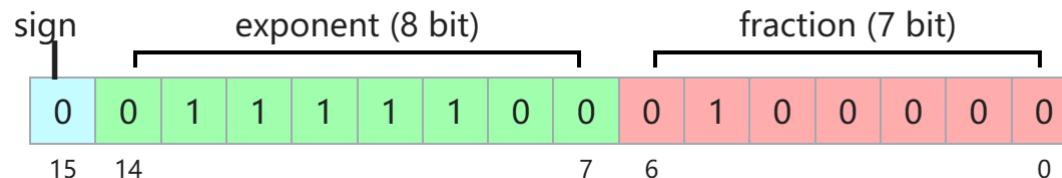
671B参数需要多大内存？



IEEE 754 single-precision 32-bit float



bfloat16

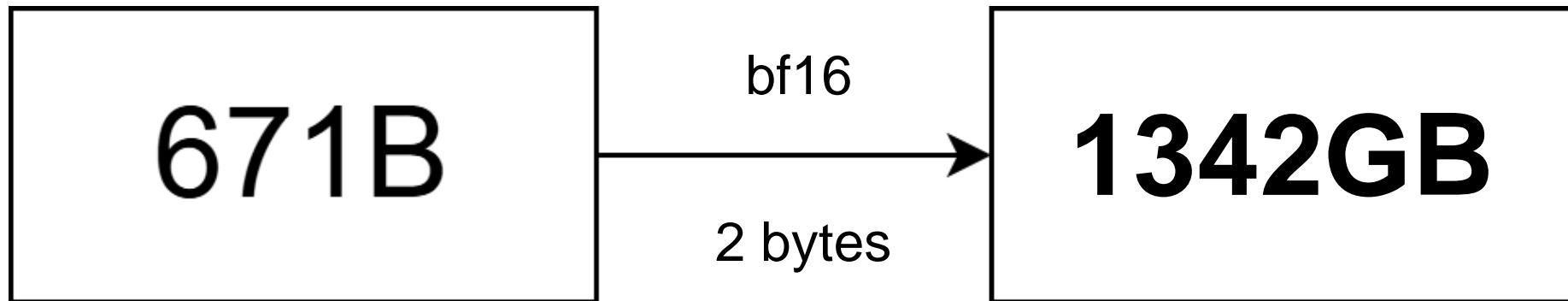


英伟达标准：brain float 16, 缩写bf16

英伟达告诉我们float类型的小数点后精度对模型并不关键！



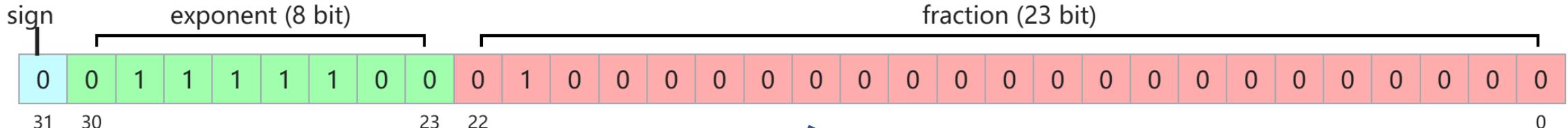
671B参数需要多大内存？



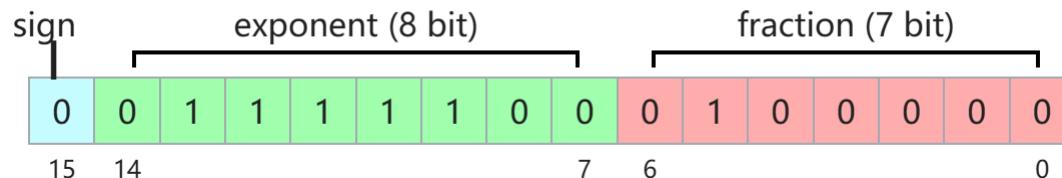
但是HuggingFace上满血版DeepSeek小于700G !

671B参数需要多大内存？

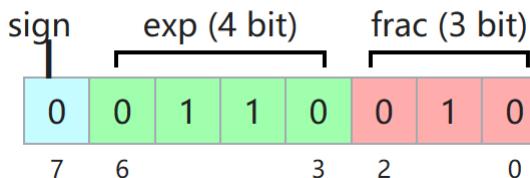
IEEE 754 single-precision 32-bit float



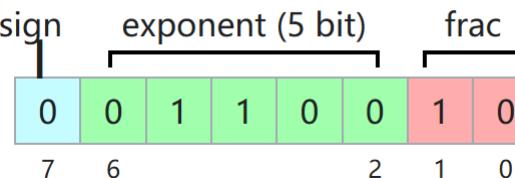
bfloat16



8-bit float e4m3



8-bit float e5m2



英伟达标准：brain float 16, 缩写bf16

$[-3.39 \times 10^{38}, 3.39 \times 10^{38}]$

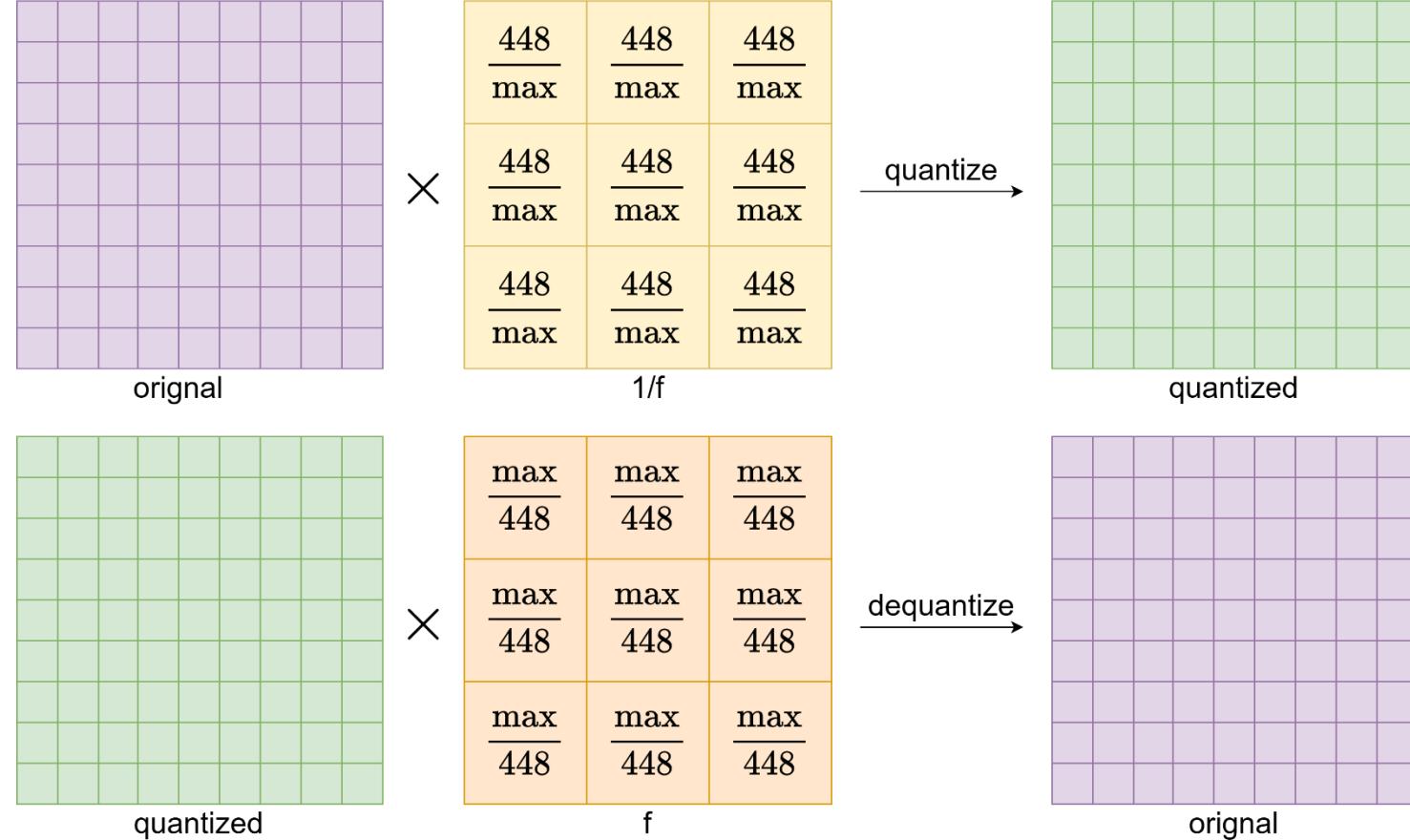
fp8: 尝试着用8个bits来表示浮点数

fp8的e4m3: 值域范围 $[-448, 448]$

fp8能降低存储需求但是值域范围极其有限



第一个商用原生fp8训练的模型



量化缩小值域范围，反量化得到原始值（有精度损失）



核心思想示例



```
>>> a  
tensor([[ 126.5718, -1515.1697,  -681.2960],  
       [ 1291.2880,    348.8044,     23.0466],  
       [  660.5675,   650.4839, -1164.3694]])  
>>> m = a.abs().amax()  
>>> m  
tensor(1515.1697)  
  
>>> q = (a / m * 448).to(dtype=torch.float8_e4m3fn)  
>>> q.to(torch.float32) / 448 * m  
tensor([[ 121.7547, -1515.1697,  -703.4716],  
       [ 1298.7169,    351.7358,     23.6745],  
       [  649.3585,   649.3585, -1190.4905]])
```

quantize

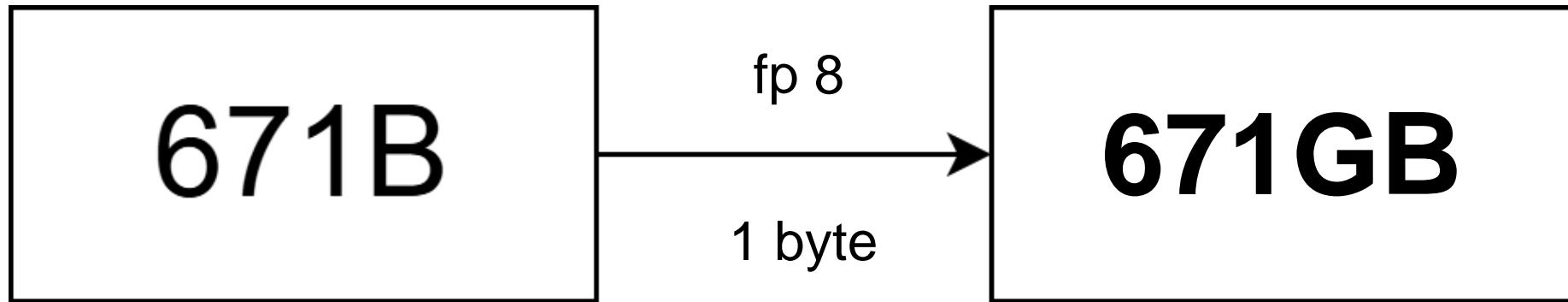
```
>>> a / m * 448  
tensor([[ 37.4243, -448.0000, -201.4432],  
       [ 381.8035,  103.1333,    6.8143],  
       [ 195.3143,  192.3328, -344.2766]])  
>>> (a / m * 448).to(dtype=torch.float8_e4m3fn)  
tensor([[ 36., -448., -208.],  
       [ 384.,  104.,    7.],  
       [ 192.,  192., -352.]], dtype=torch.float8_e4m3fn)
```

dequantize

量化缩小值域范围，反量化得到原始值（有精度损失）



671B参数需要多大内存？



这就是Hugging Face上满血版DeepSeek的模型大小！



满血版DeepSeek部署需要什么



推荐配置

硬件型号: [REDACTED]

CPU:Intel Xeon 8480+ * 2

内存 : 64G*32

系统盘: 960GB SSD * 2

数据盘 : 3.84T SSD * 8

GPU: **Nvidia HXX * 8**

网卡: 双口10G网卡 * 1 (单机)
/8 or 4 *400G CX7 (多机)

软件平台: [REDACTED] V1.2

单机支持满血版本部署

原生FP8无精度损失

单机支持20-30个用户并发

满血版 Deepseek V3/R1 671B 参数

- 模型消耗 671GB 显存
- 8 张 H20 (96GB) 768GB , 剩余 97GB (KV-Cache)



满血版DeepSeek部署需要什么



推荐配置

硬件型号: [REDACTED]

CPU: Intel Xeon 8480+ * 2

内存: 64G*32

系统盘: 960GB SSD * 2

数据盘: 3.84T SSD * 8

GPU: Nvidia HXX * 8

网卡: 双口10G网卡 * 1 (单机)

/8 or 4 *400G CX7 (多机)

软件平台: [REDACTED] V1.2

单机支持满血版本部署

原生FP8无精度损失

单机支持20-30个用户并发

满血版 Deepseek V3/R1 671B 参数

- 模型消耗 671GB 显存
- 8 张 H20 (96GB) 768GB , 剩余 97GB (KV-Cache)

AlayaDB 使用1%的KV-Cache

用户并发数提升数十倍



Deepseek



Deepseek + AlayaDB



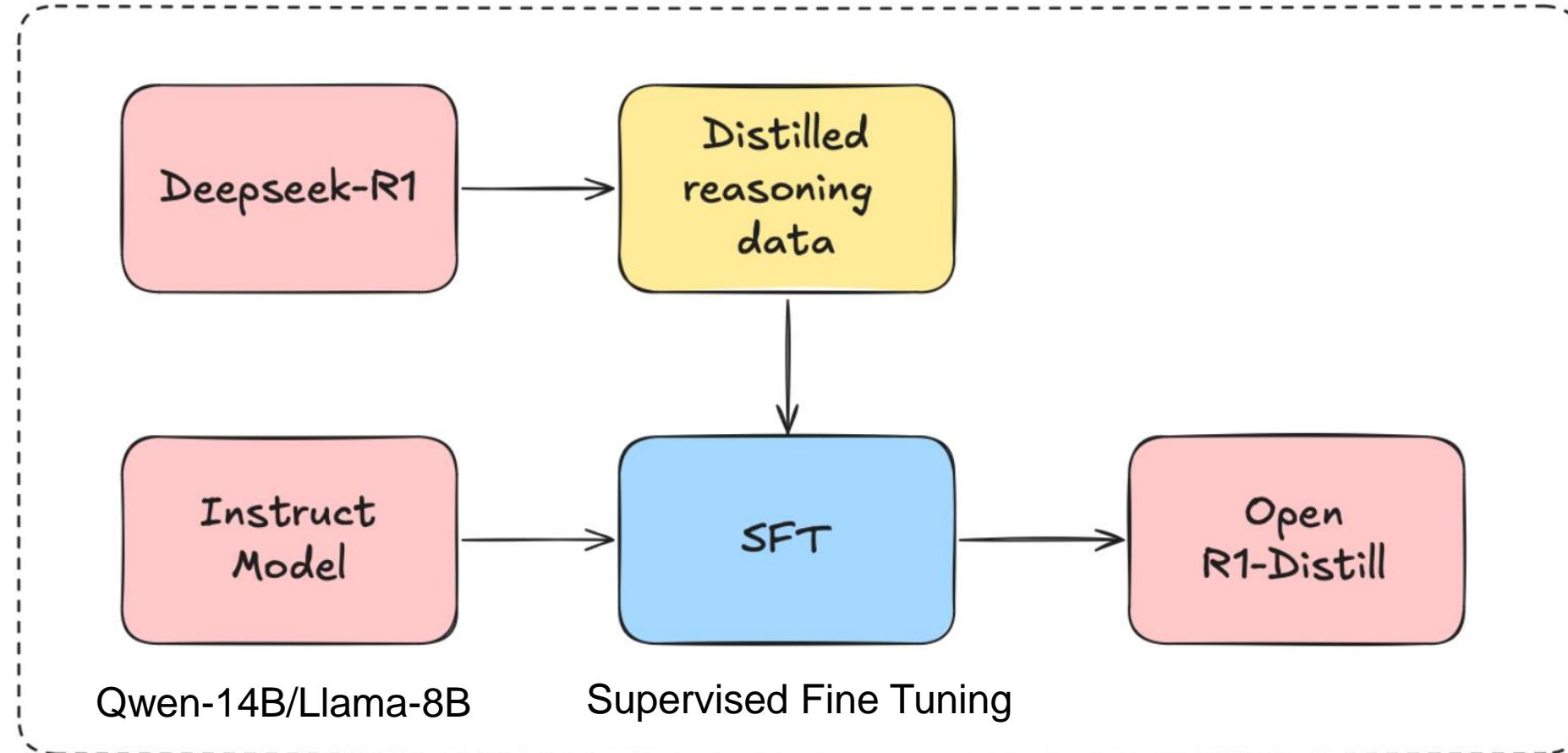
本地化部署DeepSeek



满血版DeepSeek太大，我们如何本地化部署？

答：部署DeepSeek蒸馏模型

什么是DeepSeek蒸馏模型？



DeepSeek通过蒸馏可以给其他模型赋能推理能力



DeepSeek 蒸馏模型



DeepSeek-R1-Distill Models

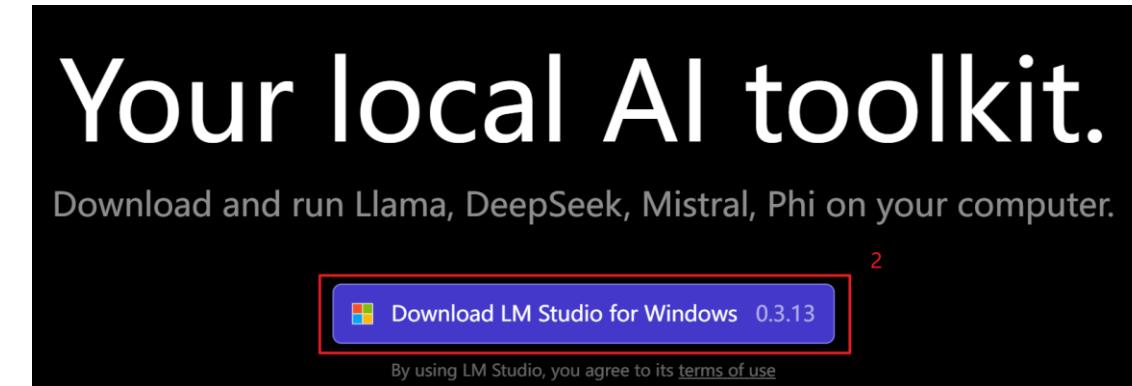
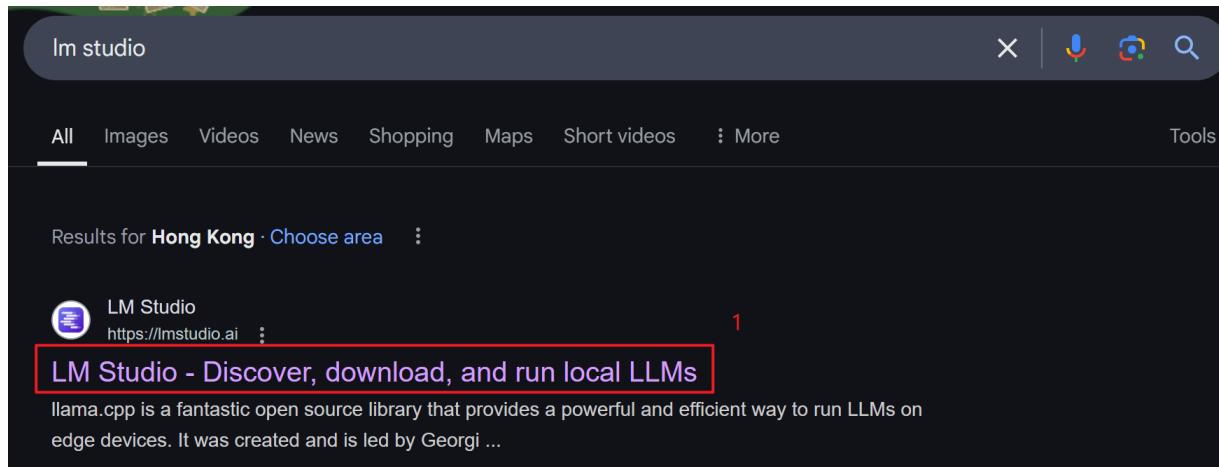
Model	Base Model	Download
DeepSeek-R1-Distill-Qwen-1.5B	Qwen2.5-Math-1.5B	HuggingFace
DeepSeek-R1-Distill-Qwen-7B	Qwen2.5-Math-7B	HuggingFace
DeepSeek-R1-Distill-Llama-8B	Llama-3.1-8B	HuggingFace
DeepSeek-R1-Distill-Qwen-14B	Qwen2.5-14B	HuggingFace
DeepSeek-R1-Distill-Qwen-32B	Qwen2.5-32B	HuggingFace
DeepSeek-R1-Distill-Llama-70B	Llama-3.3-70B-Instruct	HuggingFace

DeepSeek-R1-Distill models are fine-tuned based on open-source models, using samples generated by DeepSeek-R1. We slightly change their configs and tokenizers. Please use our setting to run these models.

DeepSeek-R1-Distill-Qwen-14B的参数是多少？
需要多大的存储空间？



本地化部署DeepSeek蒸馏模型



(0) 通过LM Studio来下载和运行本地化模型



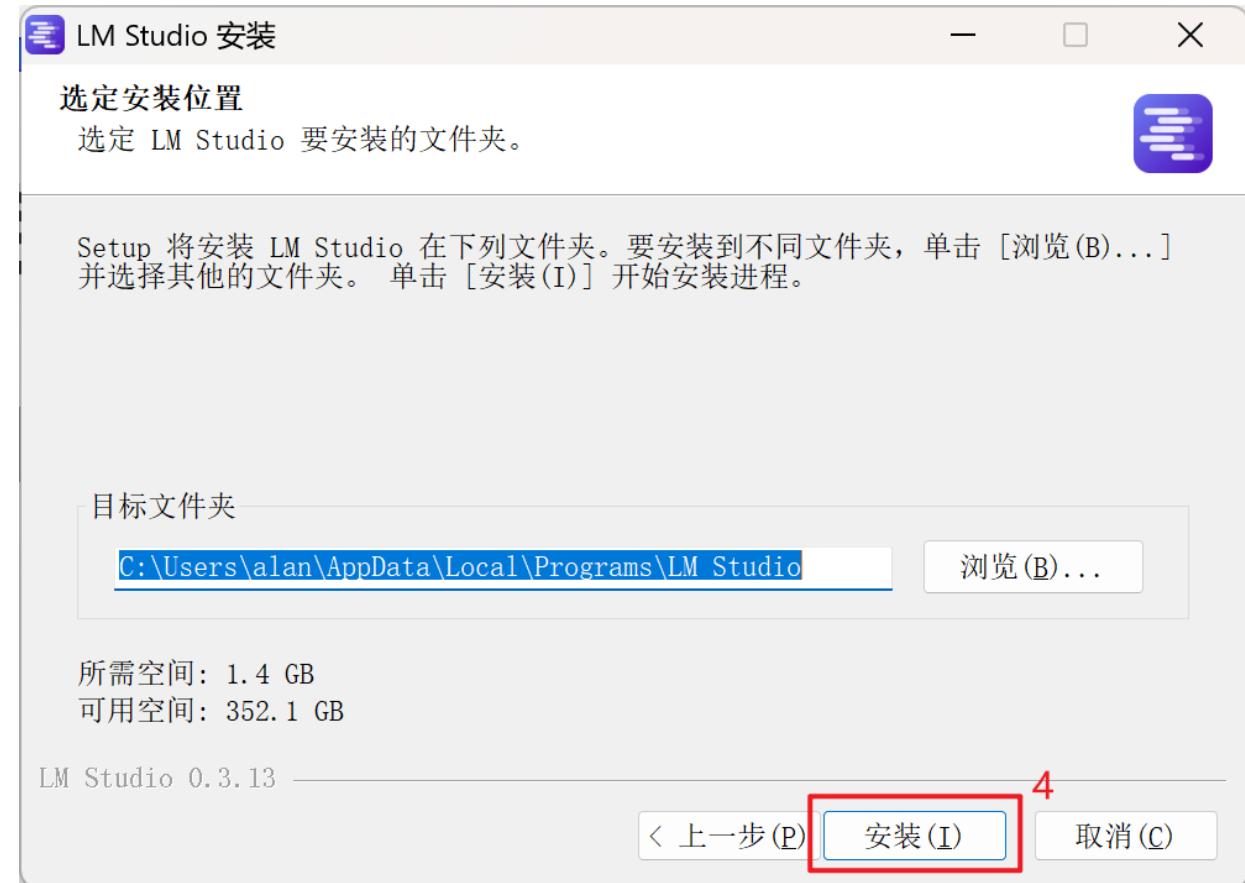
本地化部署DeepSeek蒸馏模型



(1) 安装LM Studio软件



本地化部署DeepSeek蒸馏模型



(1) 安装LM Studio软件



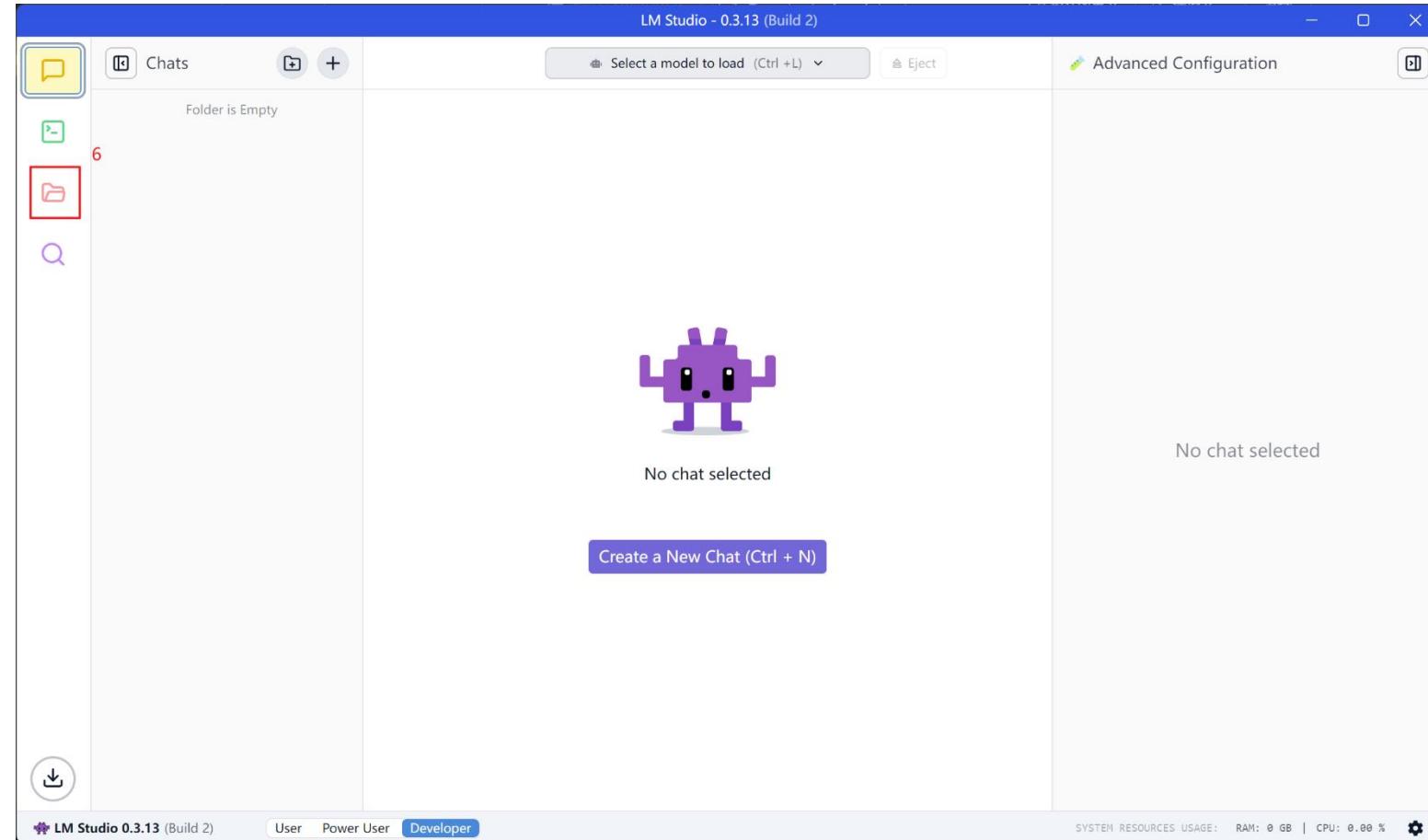
本地化部署DeepSeek蒸馏模型



(1) 安装LM Studio软件



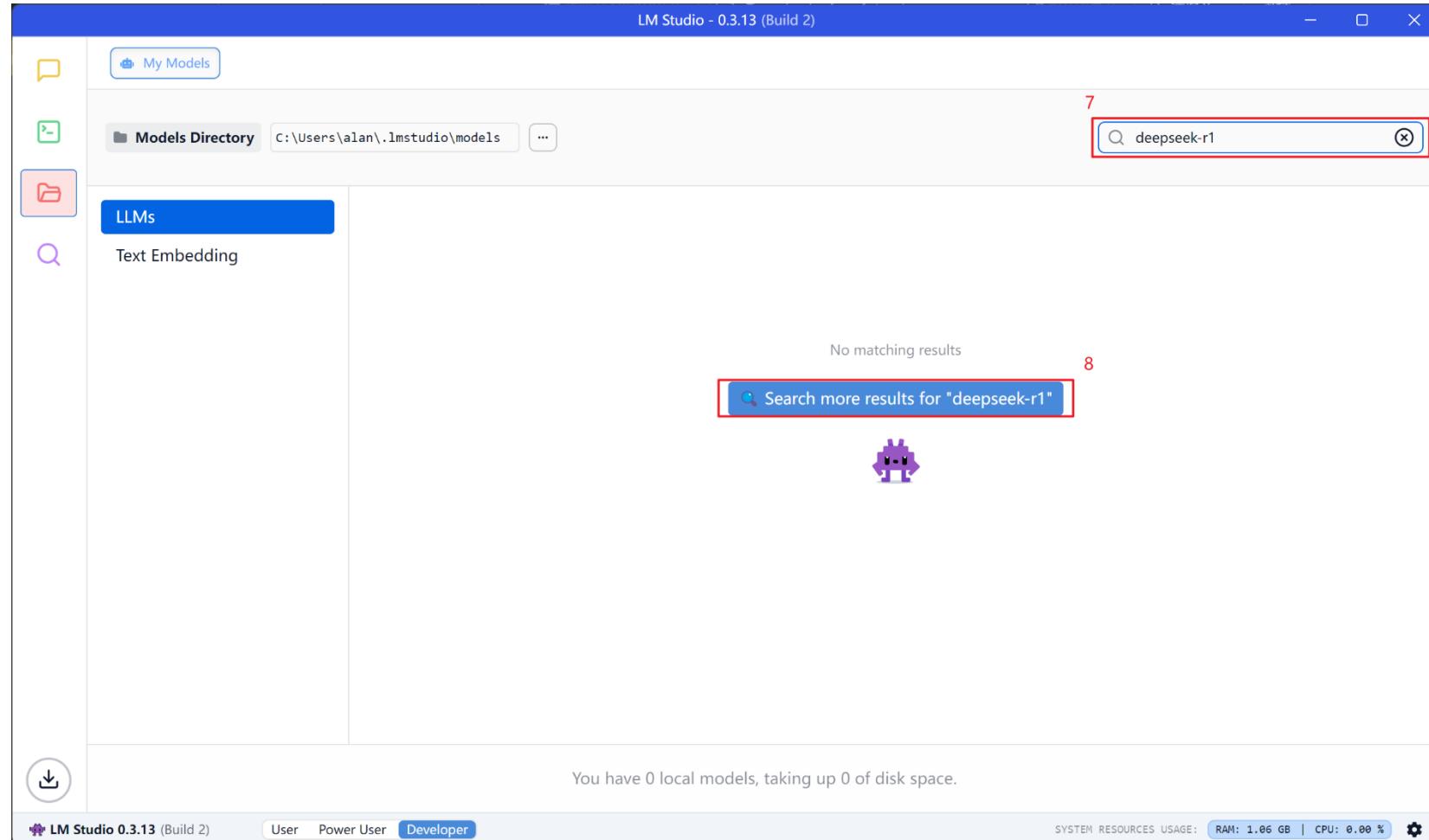
本地化部署DeepSeek蒸馏模型



(2) 运行LM Studio下载蒸馏模型



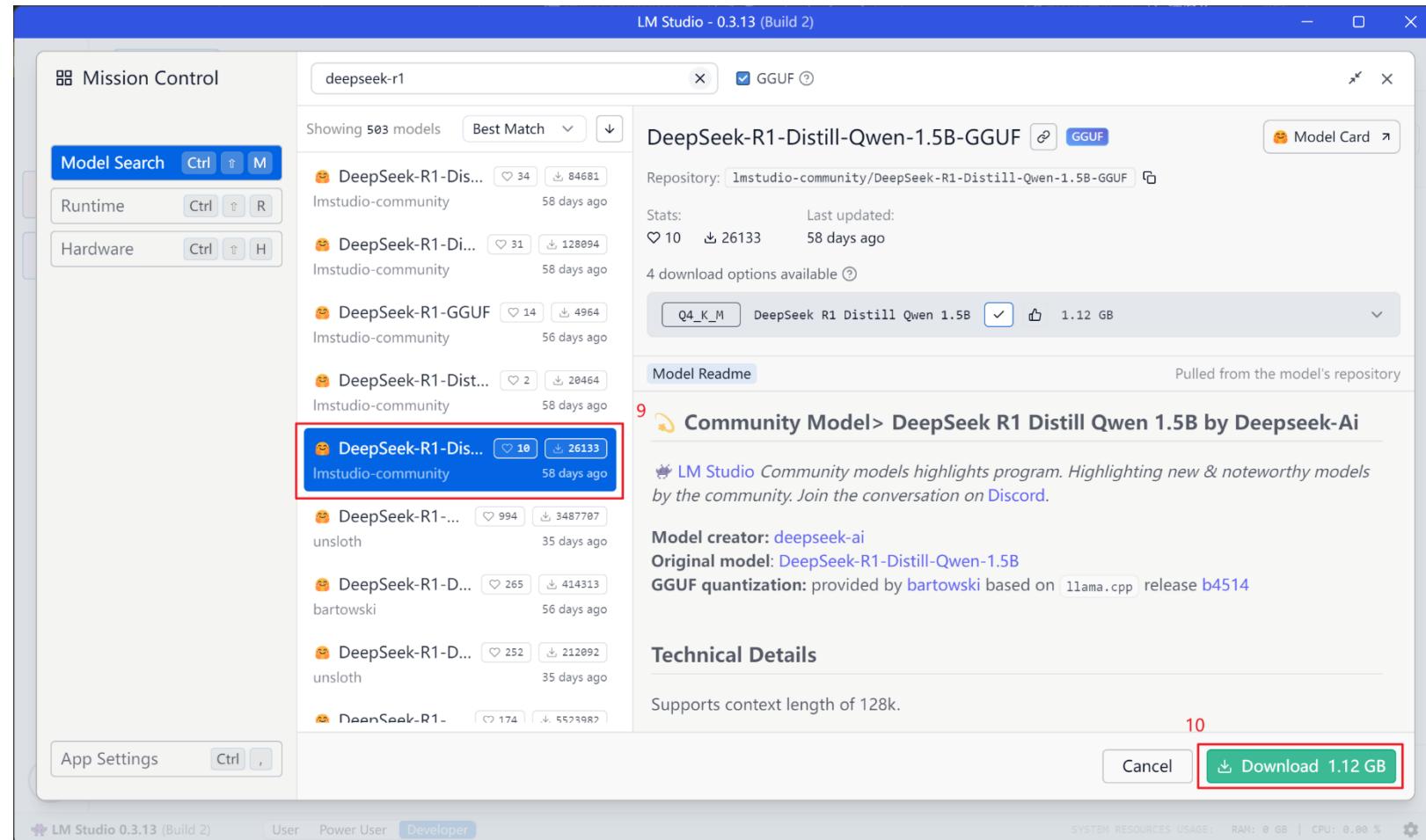
本地化部署DeepSeek蒸馏模型



(3) 搜索要下载的相应模型



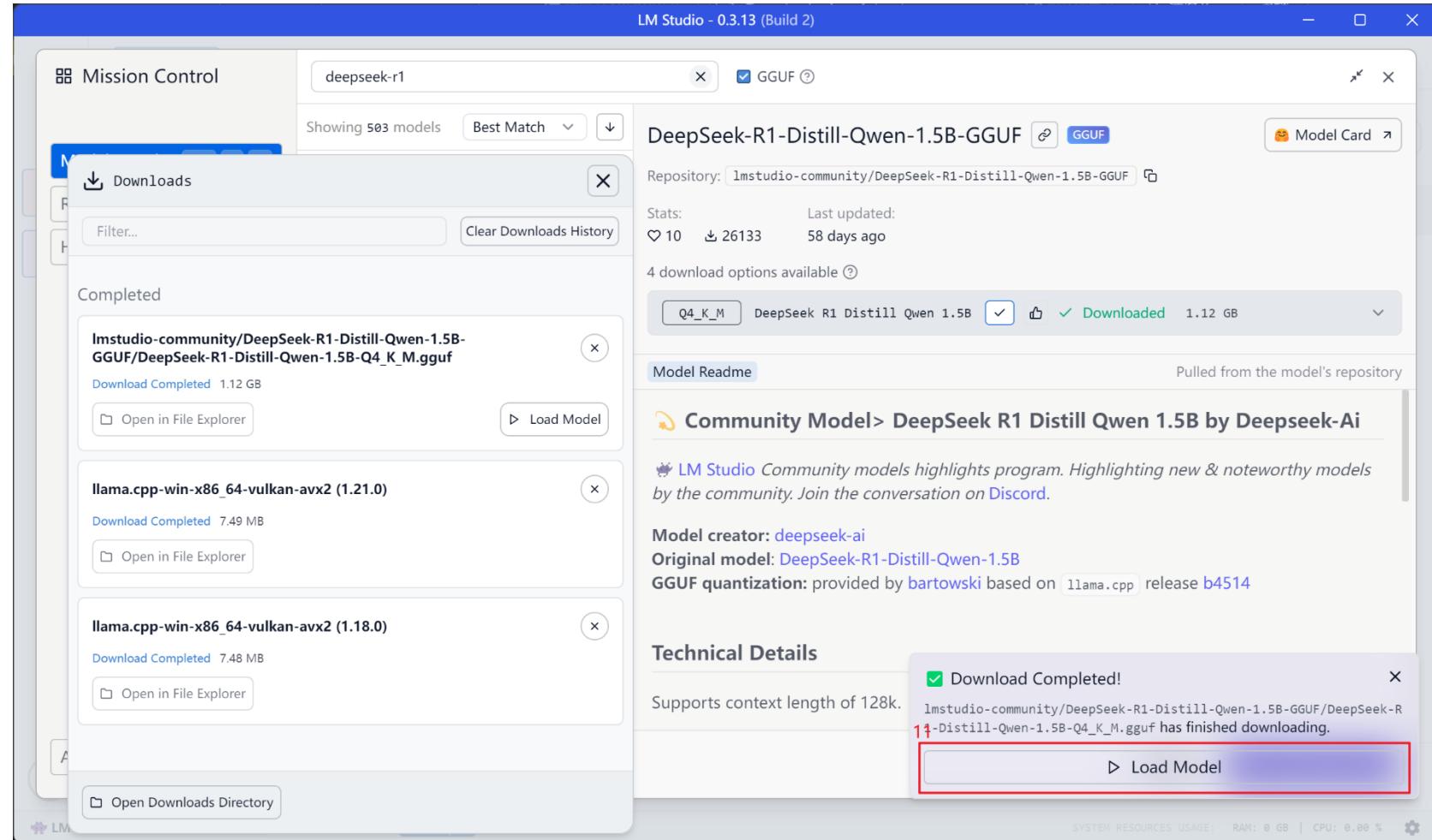
本地化部署DeepSeek蒸馏模型



(4) 下载相应的蒸馏模型



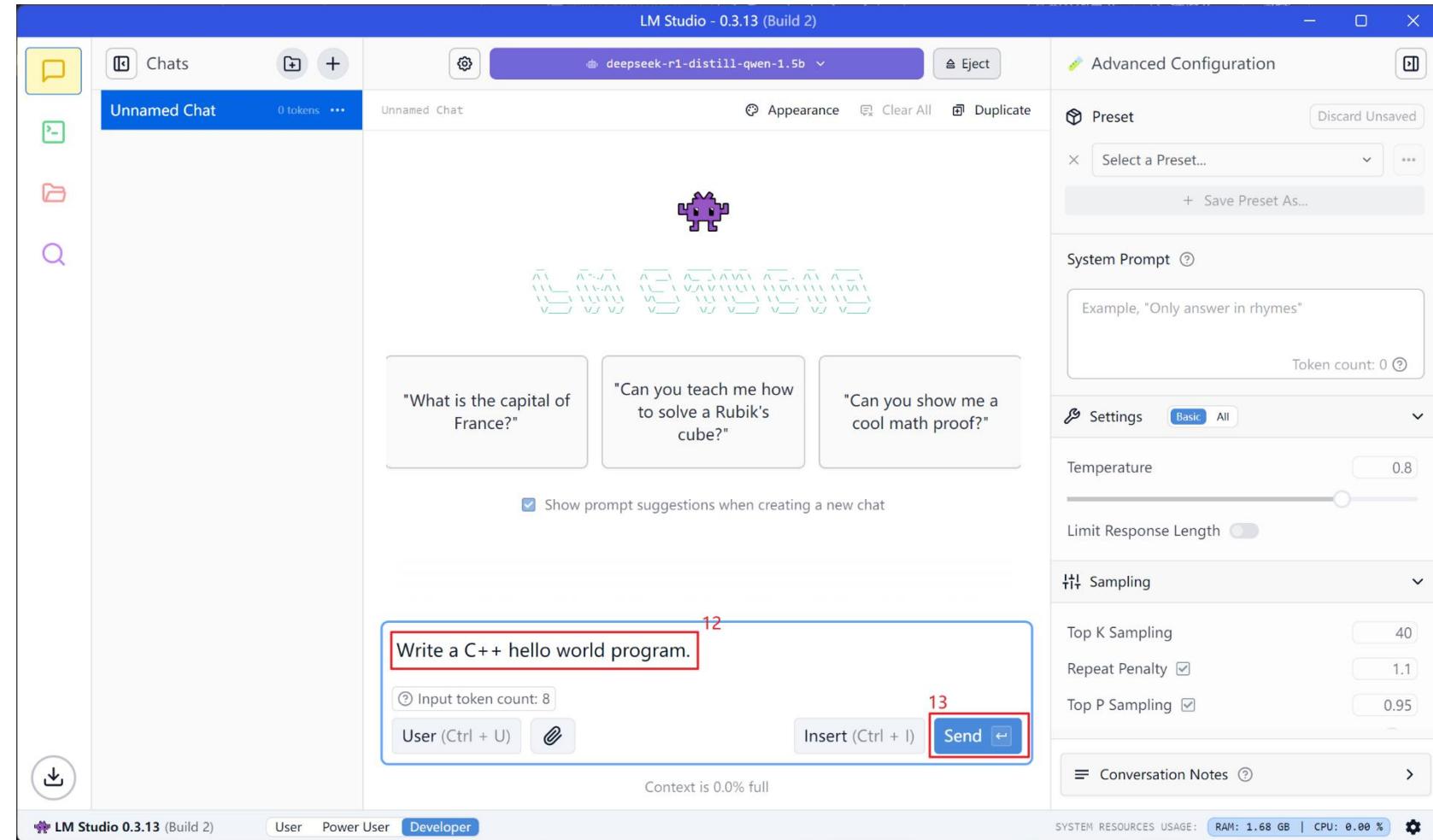
本地化部署DeepSeek蒸馏模型



(5) 把对应下载模型加载到内存



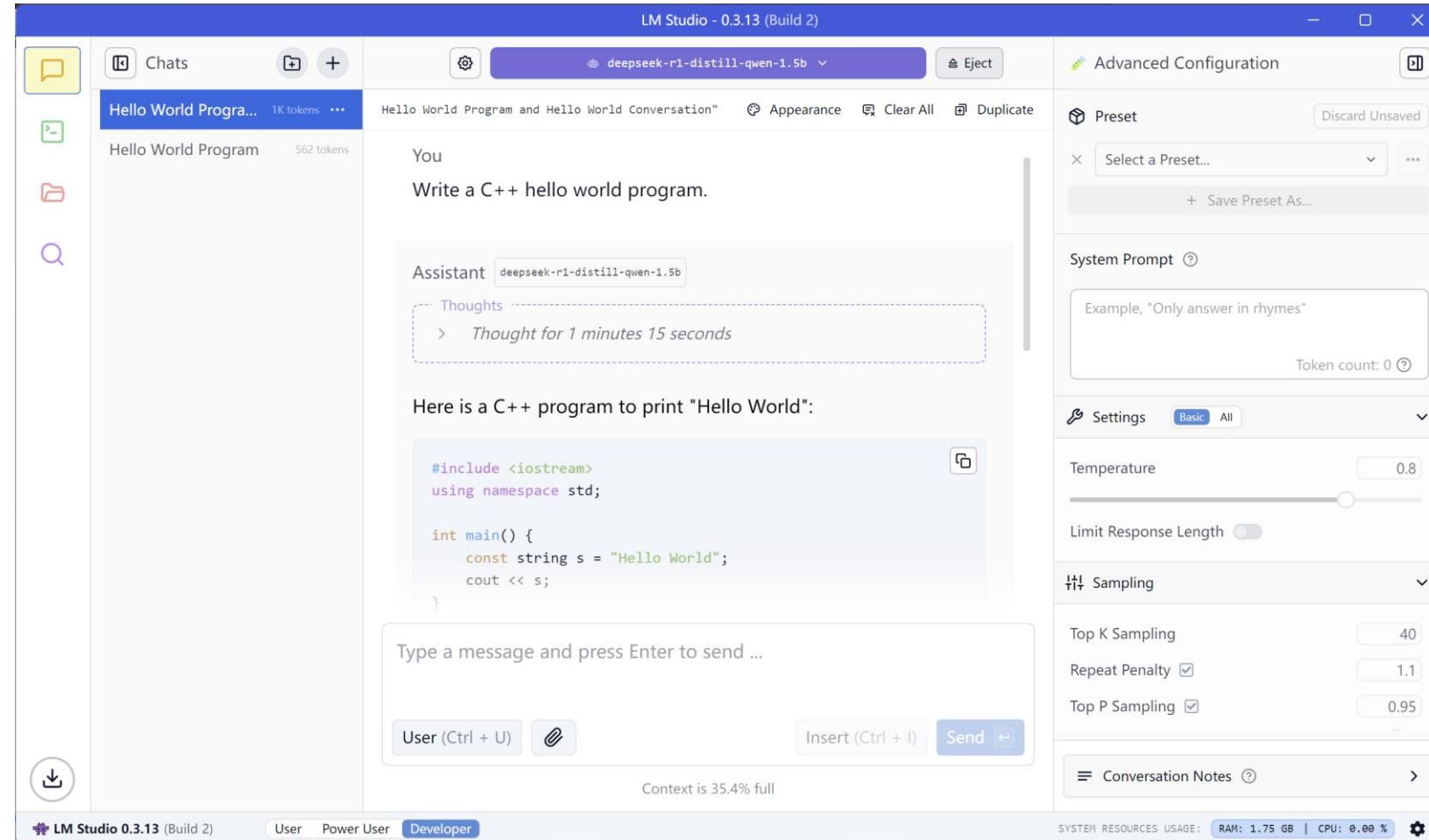
本地化部署DeepSeek蒸馏模型



(6) 开始使用本地模型帮你写程序



本地化部署DeepSeek蒸馏模型



(7) 本地模型帮你写的Hello World



拥抱 DeepSeek!



本地部署不了，怎么办？

蒸馏模型有点笨，怎么办？

想开发专属大模型应用，怎么办？



内容提纲



- ✓ **大模型DeepSeek历史简介**
 - 从ChatGPT到DeepSeek
- ✓ **本地化部署DeepSeek模型**
 - 理解DeepSeek-R1/V3模型参数
 - 本地化部署DeepSeek蒸馏模型
- ✓ **调用DeepSeek的API服务**
 - 开发专有大模型APP: LatexChecker



使用 DeepSeek 的 API

A screenshot of a search results page from a web browser. The search bar at the top contains the text "deepseek". Below the search bar are filter options: "All", "Images", "News", "Videos", "Shopping", "Short videos", "Forums", "More", and "Tools". The "All" option is underlined with a thin white line. The main content area displays search results. The first result is a link to "DeepSeek" with the URL "https://www.deepseek.com". This link is highlighted with a red rectangular border. To the right of the link, there is a snippet of text: "DeepSeek, unravel the mystery of AGI with curiosity. Answer the essential question with long-termism." Further down the page, another snippet of text is visible: "Hangzhou De...".

deepseek

All Images News Videos Shopping Short videos Forums More Tools

 DeepSeek
<https://www.deepseek.com> :

DeepSeek 1

DeepSeek, unravel the mystery of AGI with curiosity. Answer the essential question with long-termism.

Hangzhou De...



使用 DeepSeek 的 API



© 2025 杭州深度求索人工智能基础技术研究有限公司 版权所有

浙ICP备2023025841号

浙公网安备 33010502011812 号

研究

DeepSeek R1
DeepSeek V3
DeepSeek Coder V2
DeepSeek VL
DeepSeek V2
DeepSeek Coder
DeepSeek Math
DeepSeek LLM

产品

DeepSeek App
DeepSeek 网页版
开放平台 2
API 价格
服务状态

法务 & 安全

隐私政策
用户协议
反馈安全漏洞

加入我们

岗位详情



使用 DeepSeek 的 API



用量信息

所有日期均按 UTC 时间显示，数据可能有 5 分钟延迟。

充值余额

¥58.78 CNY

3

[去充值](#)

余额预警已开启 ([去设置](#))

赠送余额 [查看有效期](#)

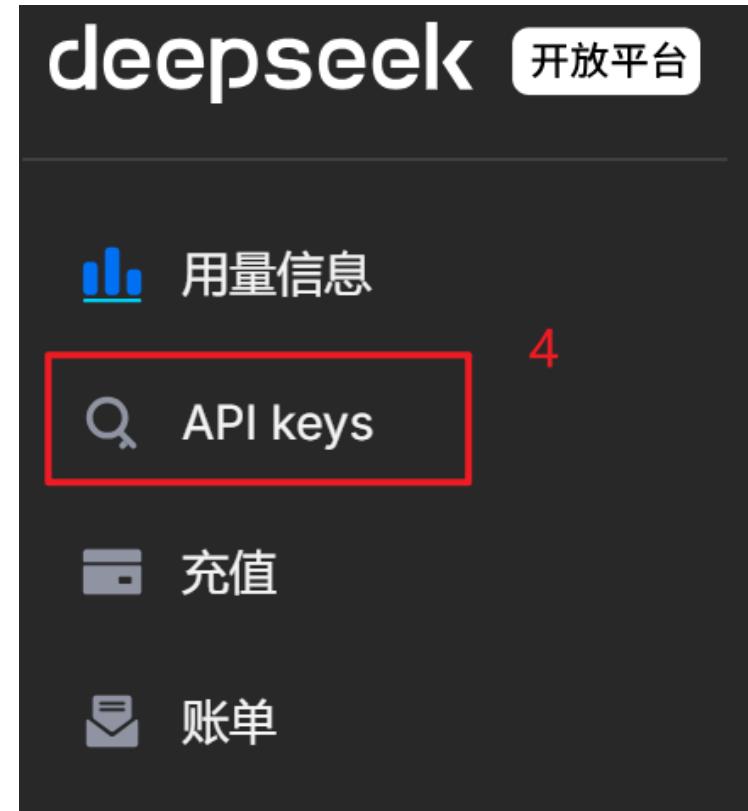
¥0.00 CNY

本月消费

¥0.21 CNY



使用 DeepSeek 的 API





使用 DeepSeek 的 API



API keys

列表内是你的全部 API key, API key 仅在创建时可见可复制, 请妥善保存。不要与他人共享
为了保护你的帐户安全, 我们可能会自动禁用我们发现已公开泄露的 API key。我们未对 20:

名称	Key
code	sk-a7351*****d5e8

5 创建 API key

创建 API key

请将此 API key 保存在安全且易于访问的地方。出于安全原因, 你将无法通过 API keys 管理界面再次查看它。如果你丢失了这个 key, 将需要重新创建。

6 sk-8e4033d6ea8647b3b82954ef281f03f7

7 关闭

8 复制

创建 API key

6 名称

7 test

8 取消

9 创建



使用 DeepSeek 的 API



- DeepSeek的API其实就是一个HTTP的接口
- 需要API key来认证你的身份
- messages由system开始，user和assistant交替出现。

```
curl https://api.deepseek.com/chat/completions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer <DeepSeek API Key>" \
-d '{
    "model": "deepseek-chat",
    "messages": [
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": "Hello!"}
    ],
    "stream": false
}'
```



开发专有大模型APP



有了DeepSeek API之后我们可以干什么？



从零开始做语法检查应用



如何不写一行代码造一个论文语法检查软件？

以下内容均原创于DBG Group李润中同学 (START)



从零开始做语法检查应用



(0) 明确需求: 把论文中Latex源码的论文做语法检查

Please improve the prompt "The following input will be given in LaTeX format. It is an academic paper in computer science concerning databases and deep learning. You need to check the correctness of the grammer in the input. Please take it carefully. For each incorrect expression, give the output in the following format:

<original>the original expression in the given input<\original>

<reason>why the original expression is wrong<\reason>

<corrected>the expression should used<\corrected>"

(1) 请DeepSeek帮忙增强改语法错误的提示词



从零开始做语法检查应用



Here's an improved version of your prompt with For each grammatical error identified, provide the output in the following structured format:

Prompt:

You are an expert in English grammar and academic science. The following input is provided in LaTeX focusing on databases and deep learning. Your grammatical errors, including but not limited to sentence structure.

For each grammatical error identified, provide the

```
<original>  
The original incorrect expression from the input.  
</original>  
  
<reason>  
A clear and concise explanation of why the expression is grammatically incorrect.  
</reason>  
  
<corrected>  
The corrected version of the expression that adheres to proper grammar rules.  
</corrected>
```

Ensure that your analysis is thorough and accurate. Focus only on grammatical issues and avoid making changes related to stylistic preferences or technical content unless they directly impact grammatical correctness. If the input is grammatically correct, simply state: "No grammatical errors found."

DeepSeek改进后的提示词



从零开始做语法检查应用



写一个Python程序。程序的作用是检查Latex中的语法错误。程序的要求如下：

1. 使用PyQt/PySide来构建图形界面

整个程序只有一个界面，上半部分是一个文本框，可以粘贴latex程序；下半部分是一个列表，每个列表有三项，分别是错误的原文，错误的原因，改正后的内容；最下方是一个让用户填入API key的lineEdit控件，和一个提交按钮，提交之后使用DeepSeek的对文本框中的内容进行处理，处理后的结果展示在下面的列表中。

2. 使用DeepSeek API来处理Latex程序，DeepSeek API使用的官方文档如下：

...

```
# Please install OpenAI SDK first: `pip3 install openai`  
  
from openai import OpenAI  
  
client = OpenAI(api_key="<DeepSeek API Key>", base_url="https://api.deepseek.com")  
  
response = client.chat.completions.create(  
    model="deepseek-chat",  
    messages=[  
        {"role": "system", "content": "You are a helpful assistant"},  
        {"role": "user", "content": "Hello"},  
    ])
```



(2) 请DeepSeek帮忙写一个Python程序调用API来上述语法检查



从零开始做语法检查应用



已深度思考（用时 59 秒）

以下是完整的Python程序实现：

```
python

import sys
import re
from PySide6.QtWidgets import QApplication, QMainWindow, QVBoxLayout, QWidget, QLineEdit, QPushButton
from PySide6.QtCore import Qt, QThread, QTimer
from openai import OpenAI

SYSTEM_PROMPT = """You are an expert in the field of computer science. The following academic paper focusing on databases and identifying any grammatical errors, in the article usage, and sentence structure.

For each grammatical error identified, <original>
The original incorrect expression from"""

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()

        self.setWindowTitle("语法检查应用")
        self.setGeometry(100, 100, 800, 600)

        self.layout = QVBoxLayout()
        self.setLayout(self.layout)

        self.splitter = QSplitter(Qt.Horizontal)
        self.layout.addWidget(self.splitter)

        self.tex_widget = QWidget()
        self.tex_widget.setLayout(QVBoxLayout())
        self.tex_widget.layout().addWidget(QLabel("输入框"))
        self.tex_widget.layout().addWidget(QLabel("结果表格 (1/3)"))
        self.splitter.addWidget(self.tex_widget)

        self.info_widget = QWidget()
        self.info_widget.setLayout(QVBoxLayout())
        self.info_widget.layout().addWidget(QLabel("详细信息区域 (2/3)"))
        self.splitter.addWidget(self.info_widget)

        self.url_label = QLabel("Base URL: [输入框]")
        self.key_label = QLabel("API Key: [输入框]")
        self.button = QPushButton("按钮")
        self.url_label.setBuddy(self.url_line)
        self.key_label.setBuddy(self.key_line)
        self.layout.addWidget(self.url_label)
        self.layout.addWidget(self.key_label)
        self.layout.addWidget(self.button)

        self.url_line = QLineEdit()
        self.key_line = QLineEdit()

        self.url_line.textChanged.connect(self.check_text)
        self.key_line.textChanged.connect(self.check_text)

    def check_text(self):
        url = self.url_line.text()
        key = self.key_line.text()
        if url and key:
            response = OpenAI().analyze(text=SYSTEM_PROMPT + self.tex_widget.layout().itemAt(1).widget().text(), url=url, api_key=key)
            print(response)

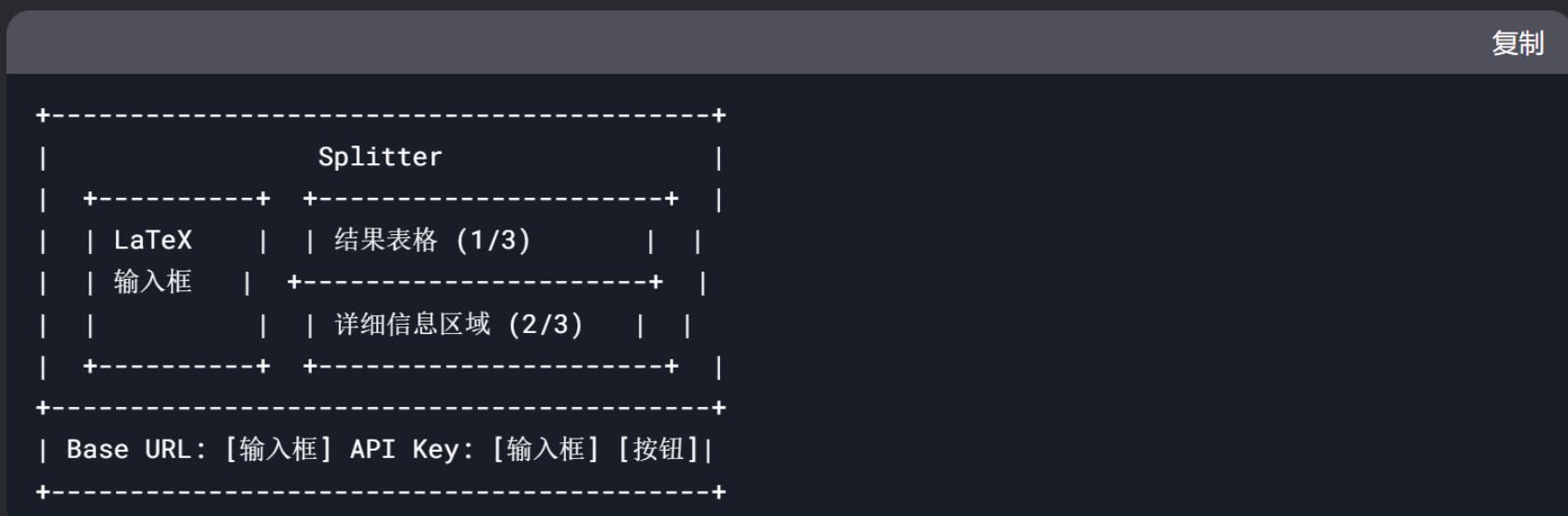
if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = MainWindow()
    window.show()
    app.exec()
```

- 将底部控制栏从右侧移出，直接添加到主布局底部
- 确保base_url、API key和按钮始终位于窗口最下方

3. 布局比例优化：

- 使用QSplitter的垂直分割确保内容区域和控制栏的比例分配
- 调整窗口初始化大小为1200x800以获得更好的显示效果

修正后界面结构：



DeepSeek生成语法检查的Python代码



从零开始做语法检查应用



如何把上述的python程序打包成exe

已深度思考 (用时 9 秒) ▾

以下是使用PyInstaller将Python程序打包为EXE的详细步骤：

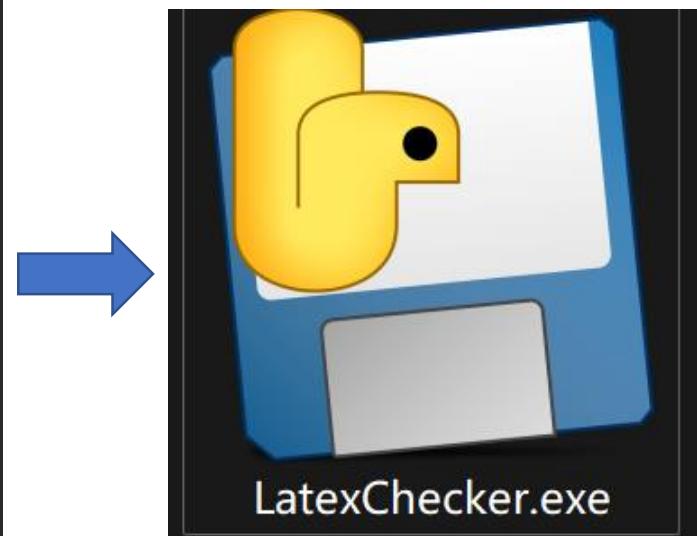
1. 安装打包工具

推荐使用PyInstaller，先安装依赖：

```
bash
pip install pyinstaller
pip install openai
pip install pyside6
```

2. 创建打包配置文件 (可选)

在程序目录创建 **build.spec** 文件：

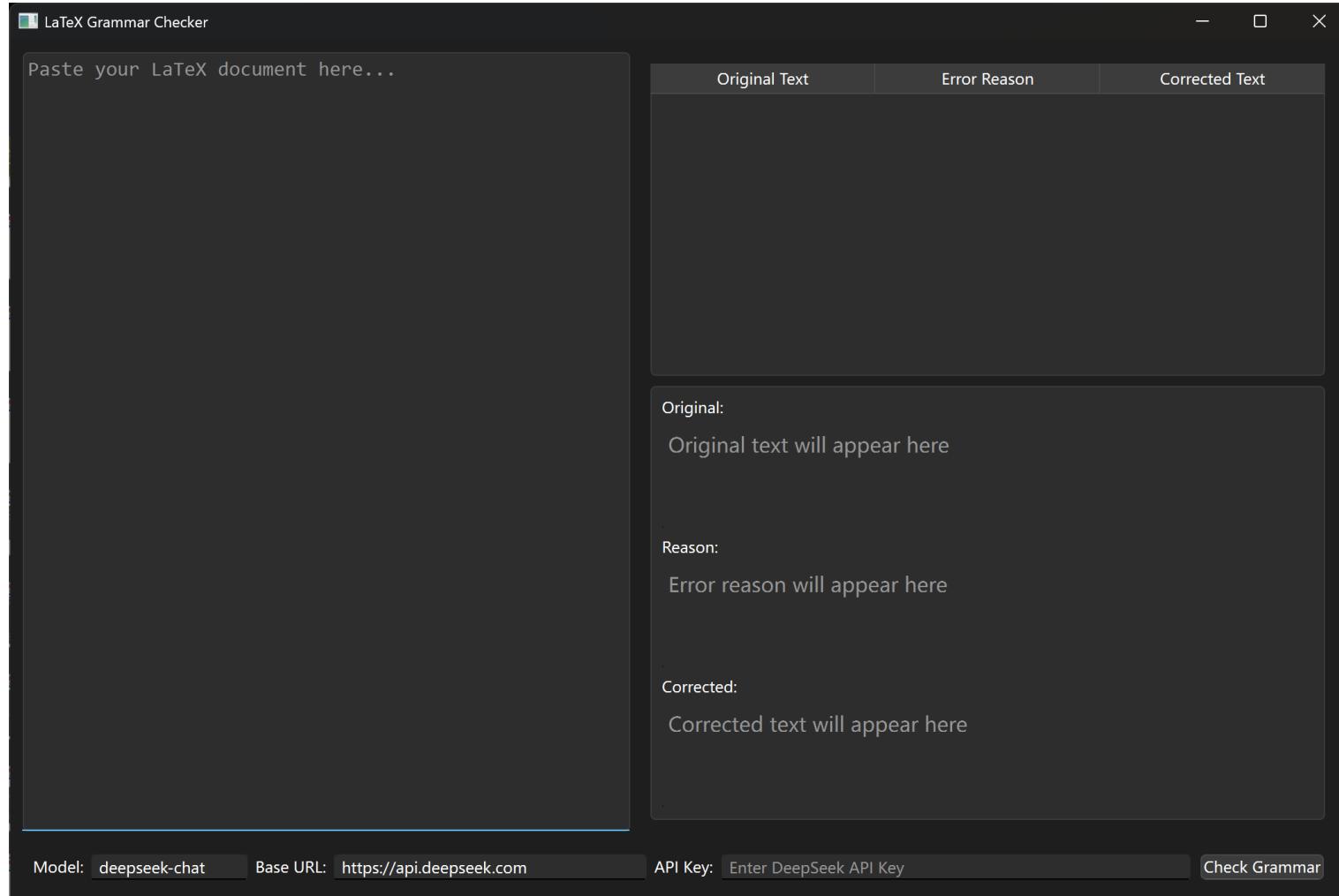


https://pan.baidu.com/share/init?surl=KnIBkHg99aeb_kHEm1qYsw&pwd=tyxm

(4) 请DeepSeek把上述Python代码打包成exe



从零开始做语法检查应用



(5) 在windows上直接使用的界面展示



从零开始做语法检查应用



LaTeX Grammar Checker

```
\section{Tree-Mamba Plan Comparator}\label{sec:comparator}

\begin{figure}
    \centering
    \includegraphics[width=0.95\columnwidth]{figures/comparator.pdf}
    \caption{Selecting procedure of candidate plans}
    \label{fig:model_architecture}
\end{figure}

% \rvsn{With the generated execution plans by the order-centric plan explorer,
% % \sysname{} next selects the best one among these plans via learned models.}

\rvsn{After generating execution plans via the proposed order-centric plan explorer, \sysname{} selects the best plan from these candidates using learned models.} The selecting procedure is depicted in Figure~\ref{fig:model_architecture}. Initially, the generated plans $P_1$, $P_2$, and $P_3$ are input of the featurizer. The featurizer converts a query plan into a tree of vectors, where each vector is an encoded representation of the corresponding node. Subsequently, the tree of vectors will be used by a learned selective model $M$ to predict their relative costs $M(P_1)$, $M(P_2)$, and $M(P_3)$. Finally, the plan with the lowest cost, i.e., $P_1$, is selected as the final execution plan. In the selective model, a plan representation network embeds the tree of vectors into a hidden representation of the plan. An MLP is used subsequently to predict the cost of the plan using its hidden representation.
```

Original Text	Error Reason	Corrected Text
1 The selecting procedure is ...	"The selecting procedure" is ...	The selection procedure is ...
2 Initially, the generated plans ...	"are input of the featurizer" is ...	Initially, the generated plans ...
3 The featurizer converts a quer...	The phrase "a tree of vectors" ...	The featurizer converts a quer...
4 Subsequently, the tree of ...	The phrase "the tree of vector..."	Subsequently, a learned ...
5 Finally, the plan with the lowe...	The phrase "i.e., \$P_1\$" is ...	Finally, the plan with the lowe...
6 In the selective model, a plan ...	The phrase "embeds the tree ..."	In the selective model, a plan ...
7 An MLP is used subsequently ...	The phrase "An MLP is used ..."	Subsequently, an MLP is used ...
8 The crux of the plan comparat...	The phrase "plays a supportiv..."	The crux of the plan ...
9 The state-of-the-art learned ...	The phrase "The state-of-the-..."	State-of-the-art learned ...
10 Meanwhile, many learned ...	The phrase "Meanwhile, many..."	Meanwhile, numerous learned...
11 As Figure~\ref{fig:node_mixer}...	The phrase "nodes 1,2,6, and ..."	As Figure~\ref{fig:node_mixer}...

Original:
The selecting procedure is depicted in Figure~\ref{fig:model_architecture}.

Reason:
"The selecting procedure" is grammatically incorrect. The correct form should use the gerund "selecting" as an adjective modifying "procedure," which requires the definite article "the."

Corrected:
The selection procedure is depicted in Figure~\ref{fig:model_architecture}.

Base URL: <https://api.deepseek.com>

API Key:

填入你刚
购买的API

(6) 语法检查工具使用效果图



从零开始做语法检查应用



Athena: An Effective Learning-based Framework for Query Optimizer Performance Improvement (Shepherd Submission)

Paper ID: 1605

Abstract

Recent studies have made it possible to integrate learning techniques into database systems for practical utilization. In particular, the state-of-the-art studies hook the conventional query optimizer to explore multiple execution plan candidates, then choose the optimal one with a learned model. This framework simplifies the integration of learning techniques into the database system. However, these methods still have room for improvement due to their limited plan exploration space and ineffective learning from execution plans. In this work, we propose Athena, an effective learning-based framework of query optimizer enhancer. It consists of three key components: (i) an order-centric plan explorer, (ii) a Tree-Mamba plan comparator and (iii) a time-weighted loss function. We implement Athena on top of the open-source database PostgreSQL and demonstrate its superiority via extensive experiments. Specifically, We achieve 1.75x, 1.95x, 5.69x, and 2.74x speedups over the vanilla PostgreSQL on the JOB, STATS-CEB, TPC-DS, and DSB benchmarks, respectively. Athena is 1.74x, 1.87x, 1.66x, and 2.28x faster than the state-of-the-art competitor Lero on these benchmarks. Additionally, Athena is open-sourced and it can be easily adapted to other relational database systems as all these proposed techniques in Athena are generic.

Table 1: Summary of learned optimizer enhancer

Method	Plan Space		Representation Method	Training Objective	
	Direct	Diversity		Time	Rank
Bao	No	Medium	No	Yes	No
Lero	No	Low	No	No	Yes
Athena	Yes	High	Yes	Yes	Yes

query optimizer in the DBMS via an end-to-end learned query optimizer, e.g., Neo [27], Balsa [50], and LOGER [6]. In particular, the learned optimizer consists of a search-based plan generator and a learned value network. (III) *Learned optimizer enhancer*: the general idea of the learned optimizer enhancer is that it generates multiple query plan candidates, which probably have good performance, then selects the best one among them via a learned model. It enhances the query optimization performance by utilizing both the existing query optimizer and learning-based models [26, 54].

In this work, we focus on the learned optimizer enhancer as it is more practical to improve the query optimization performance [26]. The major reasons are two-fold: (i) it combines both the advantages of conventional query optimizer and learned models. Thus, it not only improves the optimization performance but also is robust to the schema, data, and workload changes; (ii) it shows its superiority in terms of both tail performance and training cost over *learned*



Submission files:

④ Athena1951.pdf

Revision Files:



Athena_revision_summary_2.pdf

④ Athena_revision_3.pdf

Accept
Meta-Reviews

(7) 显著提升生产效率,SIGMOD论文改后就中了(End)



南方科技大学

SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY



谢谢 !

DBGroup @ SUSTech

Dr. Bo Tang (唐博)

tangb3@sustech.edu.cn

