

大模型的探索与实践

Introduction to Large Language Models

§ 1.2 大模型基础

Basics

滕佳烨
上海财经大学
www.tengjiaye.com

回顾 Recall

大语言模型

- 大 -> 训练数据大、模型大
- 语言 -> 对话能解决很多问题

大模型的工具

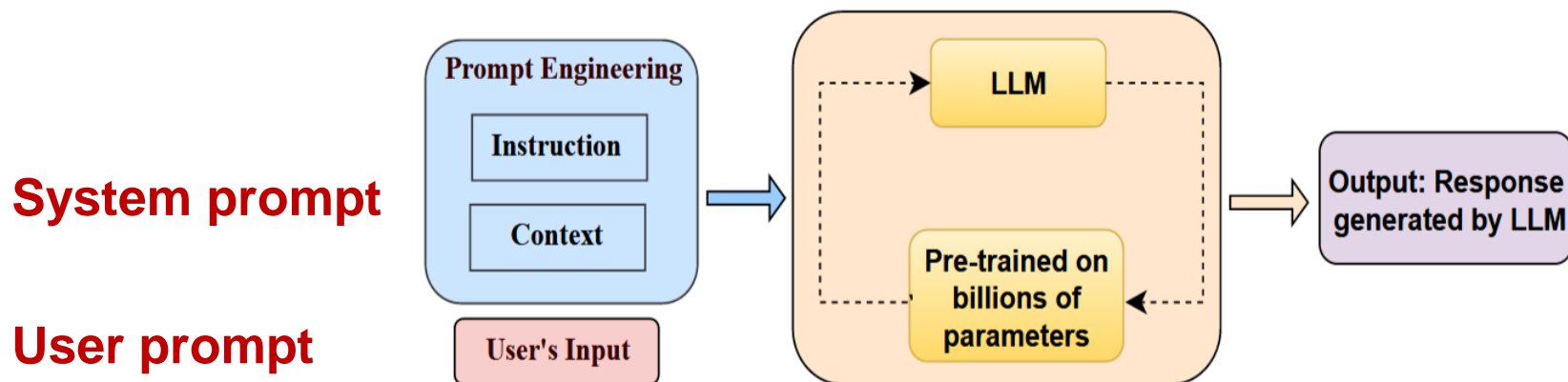
- 尝试使用Trae
- 很多任务可以由代码完成，而大模型能够参与写代码工作 (Vibe Coding)

今天的任务：大模型的提示词技术、练习使用Github

大模型元素 Elements

提示词 Prompts

- 提供给模型的初始上下文 (Context) 或条件 (Condition), 用于引导模型生成特定类型的输出。
- 提示词工程 (Prompt Engineering) 成为提升预训练大模型面向下游任务的重要技术。
- 清晰、具体的Prompt是获得高质量输出的关键。



A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications
<http://arxiv.org/abs/2402.07927>

提示词 Prompts

如何设计提示词？

- **结构化提示：**

- 分配角色
- 明确任务与目标
- 提供完整上下文
- 设定格式与规则

(user story)

示例：

角色

你是一名科技记者。

上下文

这里有一份关于“量子计算”的技术报告：[粘贴报告内容]

任务与目标

请为这份报告撰写一篇新闻稿摘要，目标读者是广大普通民众，他们对量子计算几乎没有了解。

规则

- 摘要长度控制在300字以内。
- 避免使用复杂的专业术语。
- 开头要有一个吸引人的标题。

在大模型中，越详细的prompt越容易获得自己想要的答案

提示词 Prompts

如何设计提示词？

- **上下文学习 (In-Context Learning):**

- 基于大模型上下文学习的能力
- 展示一个或多个完整的“输入 -> 输出”例

示例： 将客户评论分类为“正面”、“负面”或“中性”。

示例1:

评论：“这个产品太棒了，完全改变了我的生活！”

分类：正面

示例2:

评论：“快递包装有点破损，但产品还能用。”

分类：中性

现在，请对以下评论进行分类：

评论：“等了半个月才到货，客服回复还很慢，非常失望。”

分类：

提示词 Prompts

如何设计提示词？

- **思维链(Chain-Of-Thought):**

- 执行需要多步逻辑、规划或抽象思考的复杂任务的能力。
- 属于推理(reasoning)能力的一部分
- 许多任务暗含了推理问题
 - 简单的四则运算 $(1+4)*5+6$
 - 复杂的关系推理 父亲的叔叔的儿子的舅舅的奶奶的侄儿的孙子是谁

示例：

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903

提示词 Prompts

如何设计提示词？

- 应避免：

- 模糊表述：指令不具体，给模型留下了过多的解释空间，导致结果不可控。

例子：“写点关于人工智能的东西。”（写什么？技术原理？社会影响？还是科幻故事？）

修正：“请写一篇800字的文章，介绍人工智能在医疗诊断领域的三个主要应用。”

提示词 Prompts

如何设计提示词？

- 应避免：

- 模糊表述：指令不具体，给模型留下了过多的解释空间，导致结果不可控。
- 歧义指令：使用了有多重含义的词语或句子，模型可能会误解。

例子：“给我苹果的信息。”（是苹果公司 Apple Inc.，还是水果苹果？）

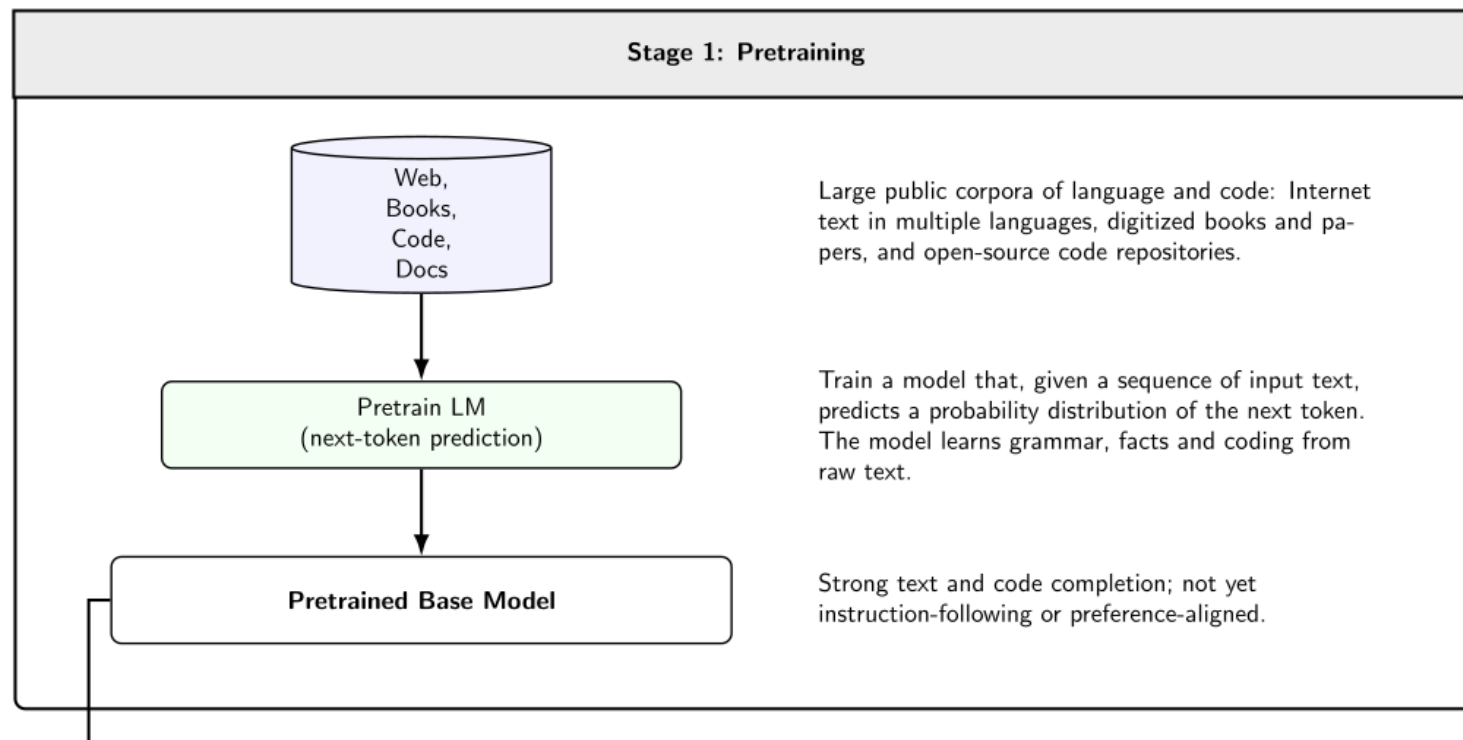
修正：“请提供关于苹果公司（Apple Inc.）的最新股价和市值信息。”

打开一个大模型软件，试一试？

大模型的前沿方向 Frontier Directions

预训练 Pre-train

在海量无标注数据上，通过自监督等学习目标让模型自主学习语言规律、世界知识等通用能力，为后续任务适配奠定基础的过程。

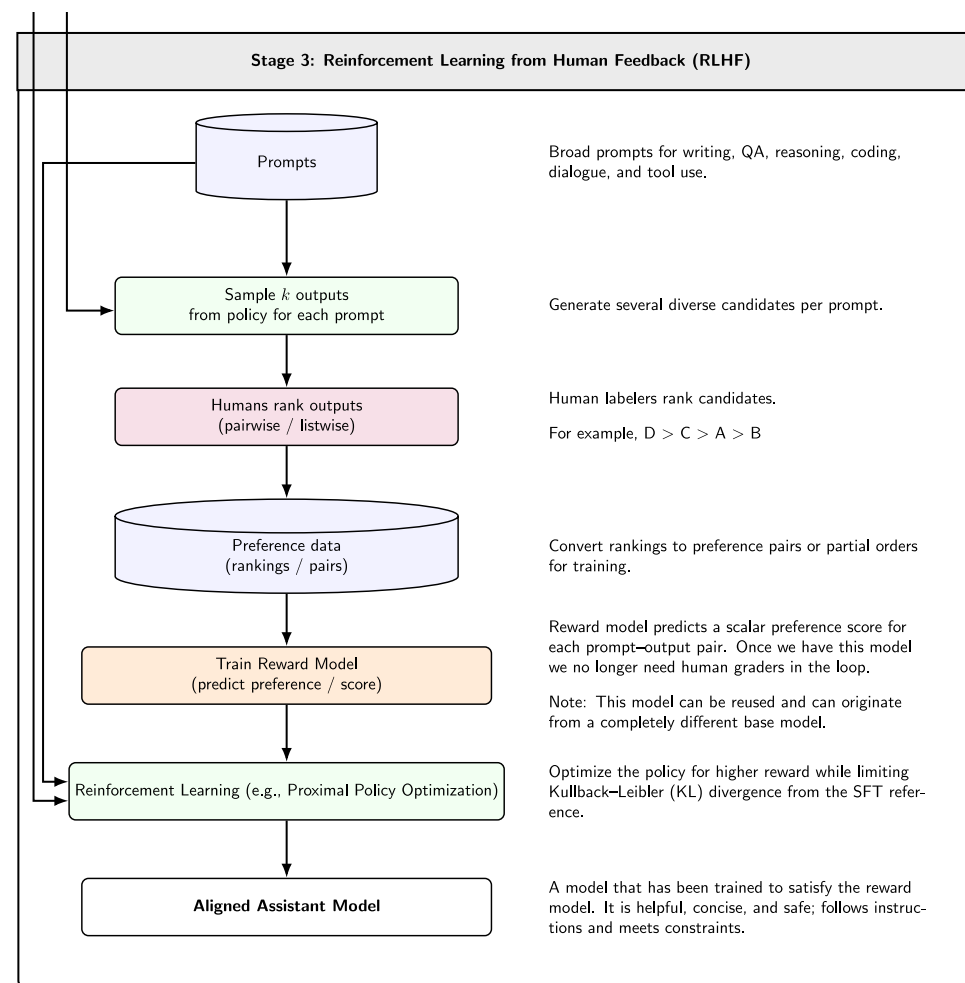


https://en.wikipedia.org/wiki/Generative_pre-trained_transformer

大模型的前沿方向 Frontier Directions

后训练 Post-train (SFT)

在微调模型基础上，利用人类对模型输出的偏好反馈构建奖励模型，再通过强化学习进一步优化模型以使其输出更贴合人类价值观和偏好的过程。

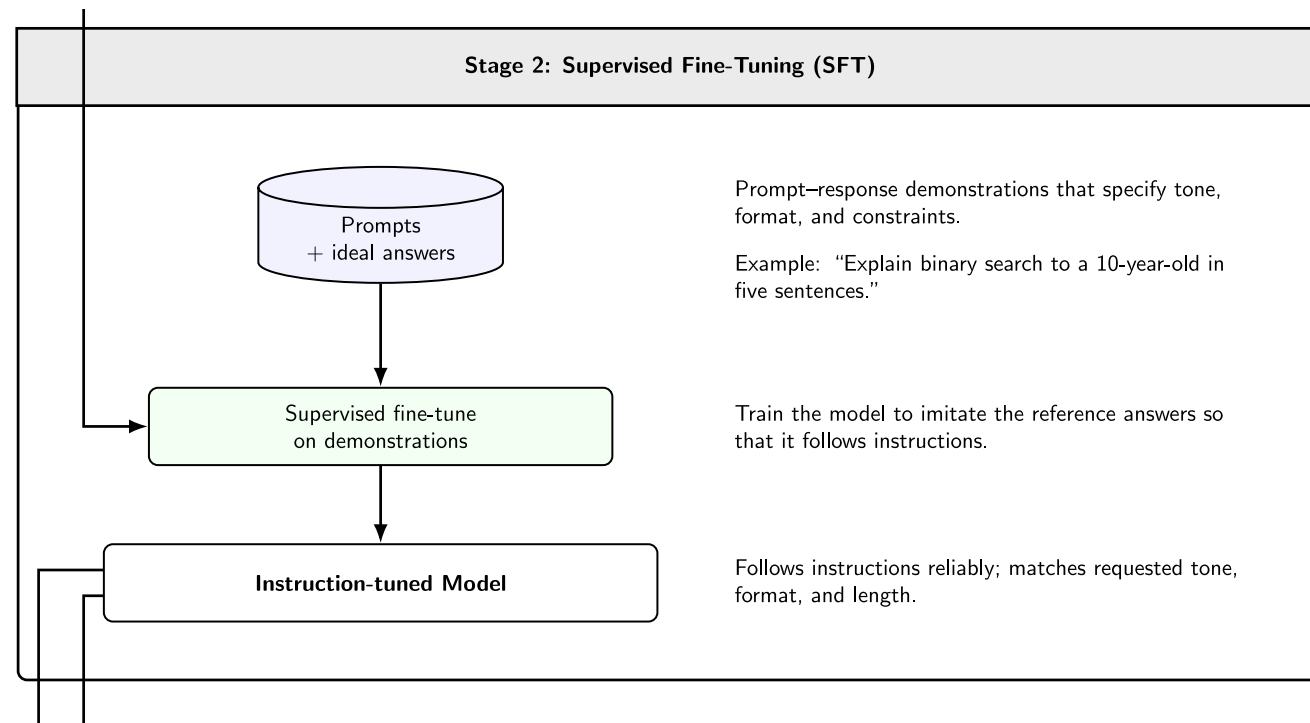


https://en.wikipedia.org/wiki/Generative_pre-trained_transformer

大模型的前沿方向 Frontier Directions

后训练 Post-train (RLHF)

在预训练模型基础上，使用人工标注的高质量任务数据进行针对性训练，以让模型适配特定任务需求并对齐人类意图的过程。



https://en.wikipedia.org/wiki/Generative_pre-trained_transformer

大模型元素 Elements

多模态 Multimodality

处理和生成多种类型数据的能力，如文本、音频、图像、视频等。

生成guacamole图片



✓ a photo of **guacamole**, a type of food.

× a photo of **ceviche**, a type of food.

× a photo of **edamame**, a type of food.

× a photo of **tuna tartare**, a type of food.

× a photo of **hummus**, a type of food.

图生文

*guacamole 鳄梨酱

图片已创建



文生图

一个好用的代码托管平台 - Github

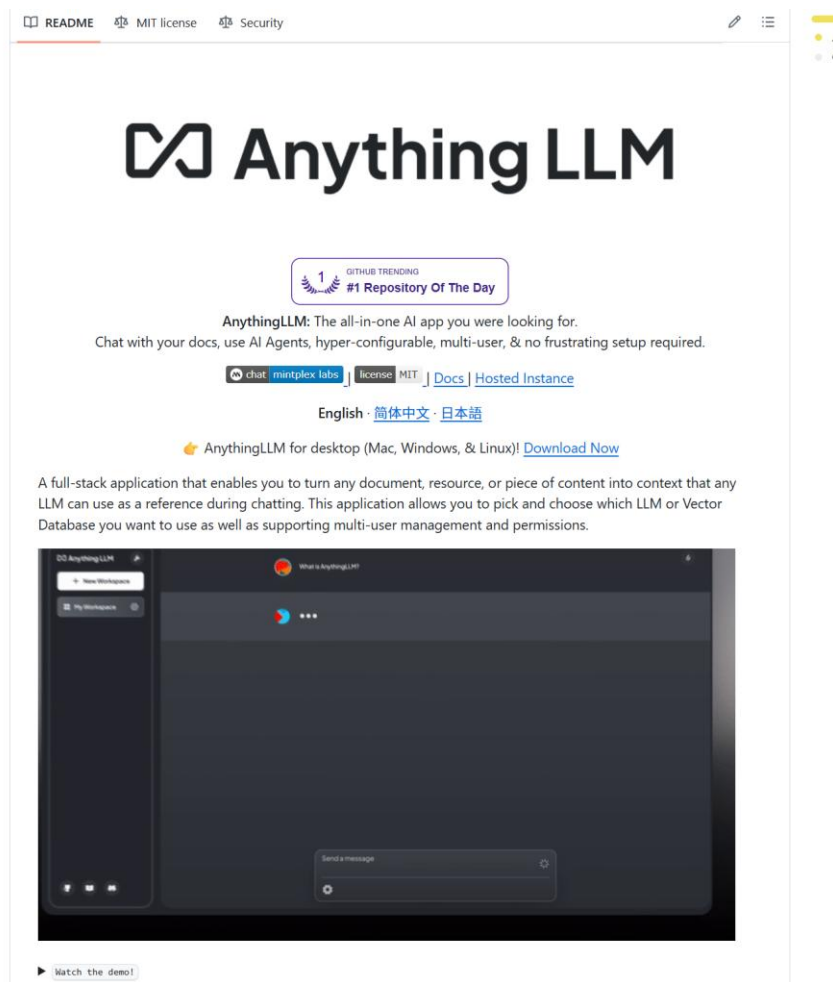
<https://github.com/>

Github

The screenshot displays the GitHub repository page for `Mintplex-Labs / anything-llm`. The repository is public and has 46.7k stars, 4.7k forks, and 329 watchers. The main content area shows a list of files and their commit history, including `.devcontainer`, `.github`, `.vscode`, `browser-extension`, `cloud-deployments`, `collector`, `docker`, `embed`, `extras`, `frontend`, `images`, `locales`, `server`, `.dockerignore`, `.editorconfig`, `.gitattributes`, `.gitignore`, `.gitmodules`, `.hadolint.yaml`, `.nvmrc`, `.prettierrc`, `.prettierrc`, `BARE_METAL.md`, `CONTRIBUTING.md`, `LICENSE`, `README.md`, `SECURITY.md`, and `eslint.config.js`. The right sidebar contains the 'About' section, 'Releases' (latest: AnythingLLM 1.8.4), 'Sponsor this project' (Mintplex-Labs), 'Packages' (anything-llm), 'Contributors' (163), and 'Deployments' (1).

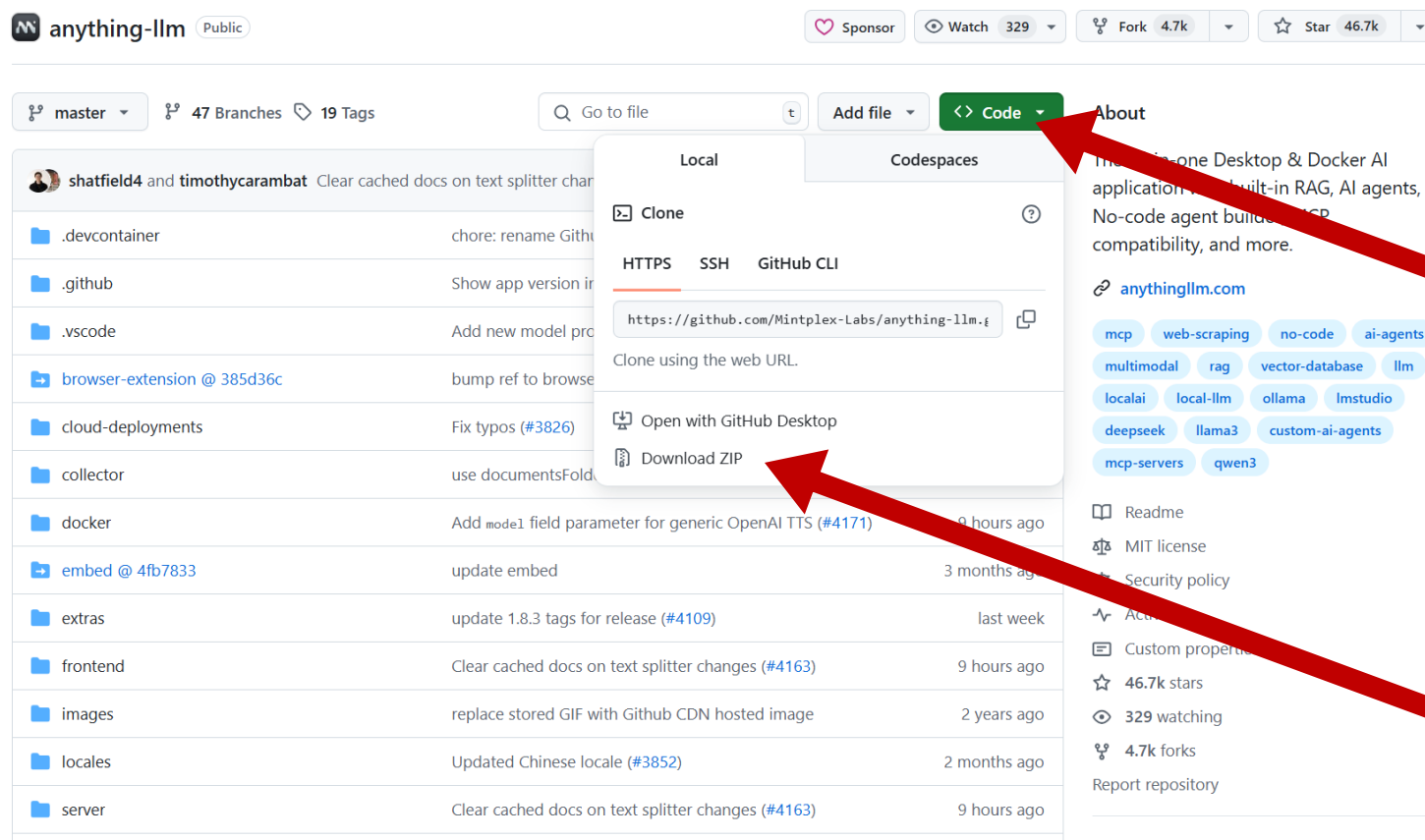
<https://github.com/Mintplex-Labs/anything-llm#>

Github: readme



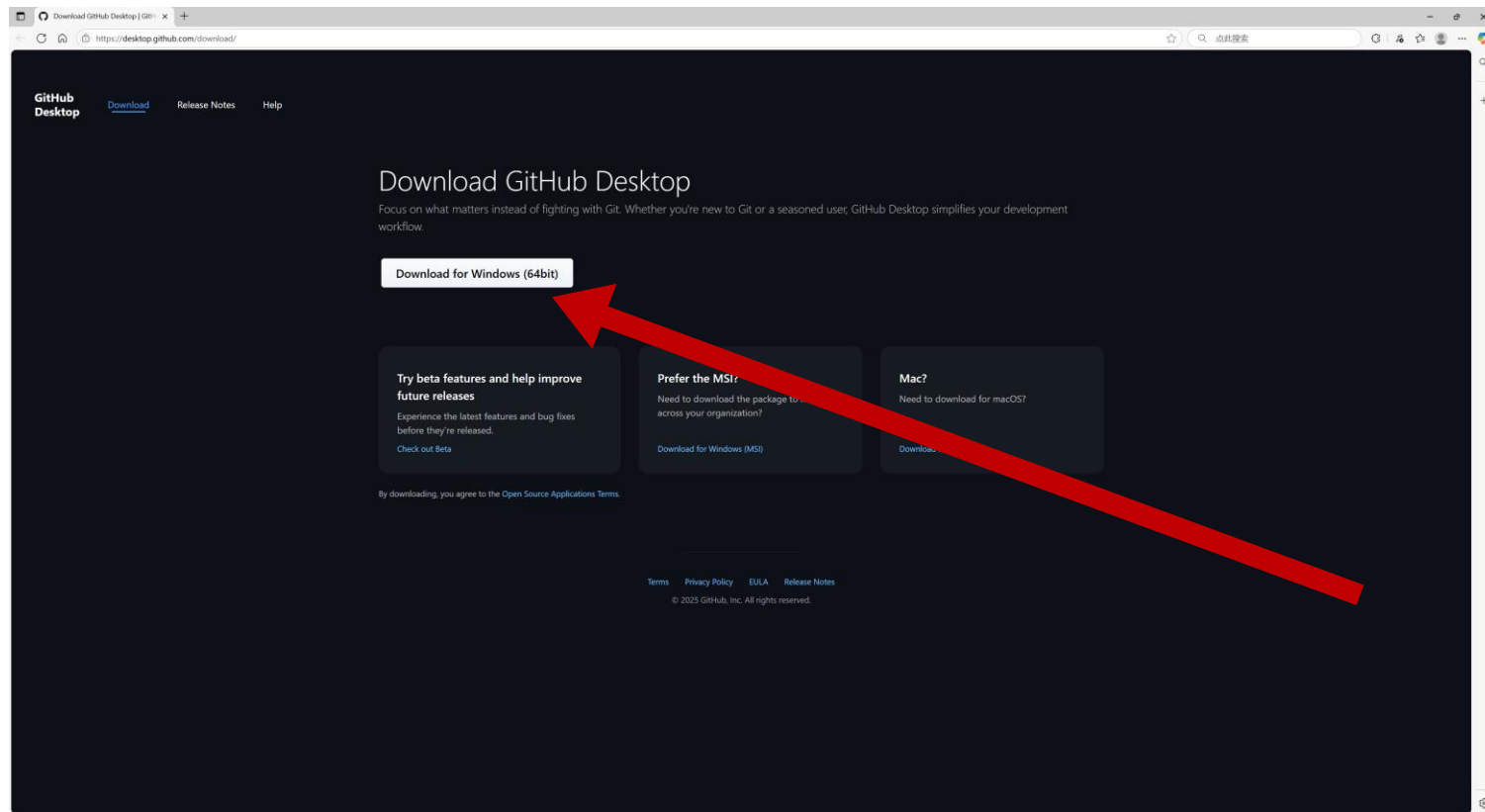
<https://github.com/Mintplex-Labs/anything-llm#>

Github



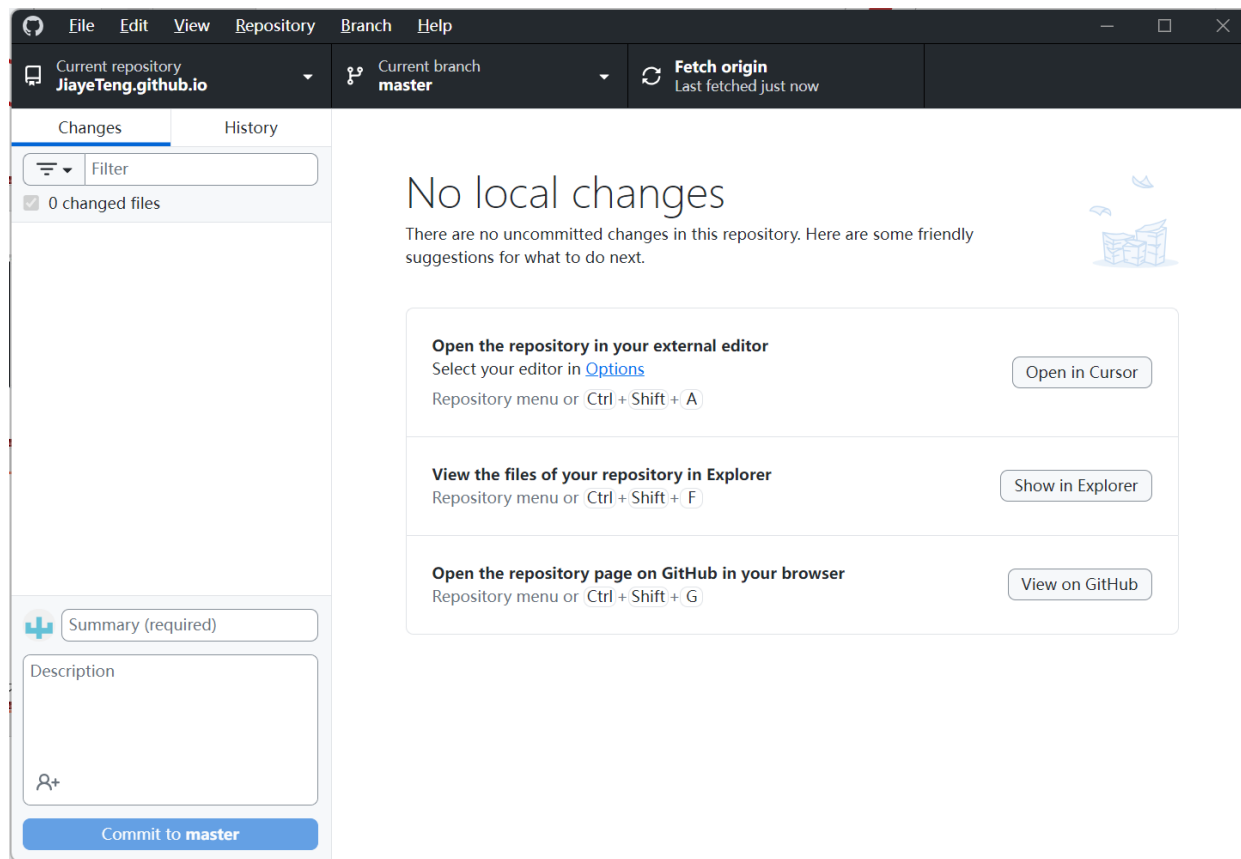
<https://github.com/Mintplex-Labs/anything-llm#>

Github 桌面版



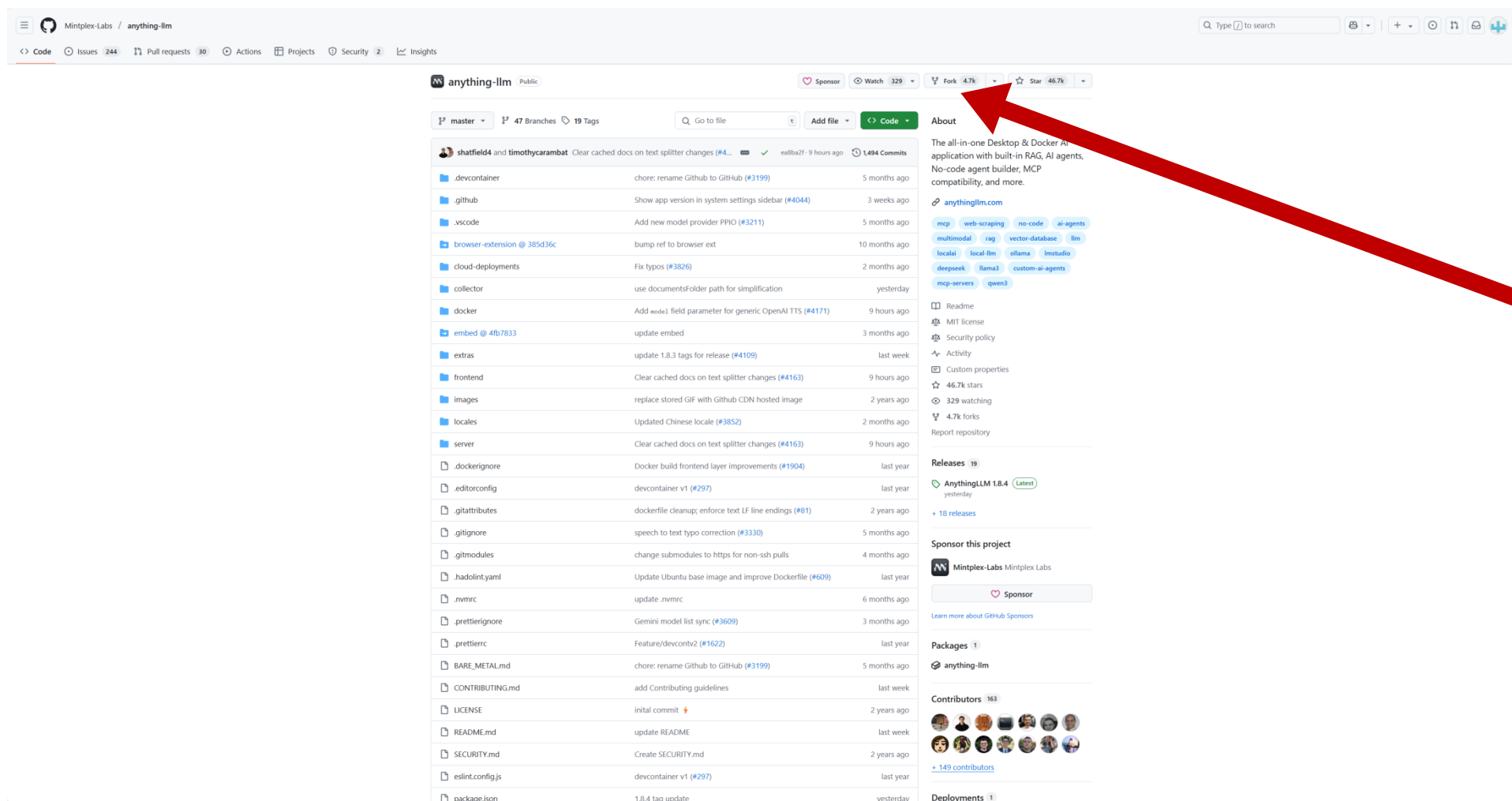
<https://desktop.github.com/download/>

Github 桌面版



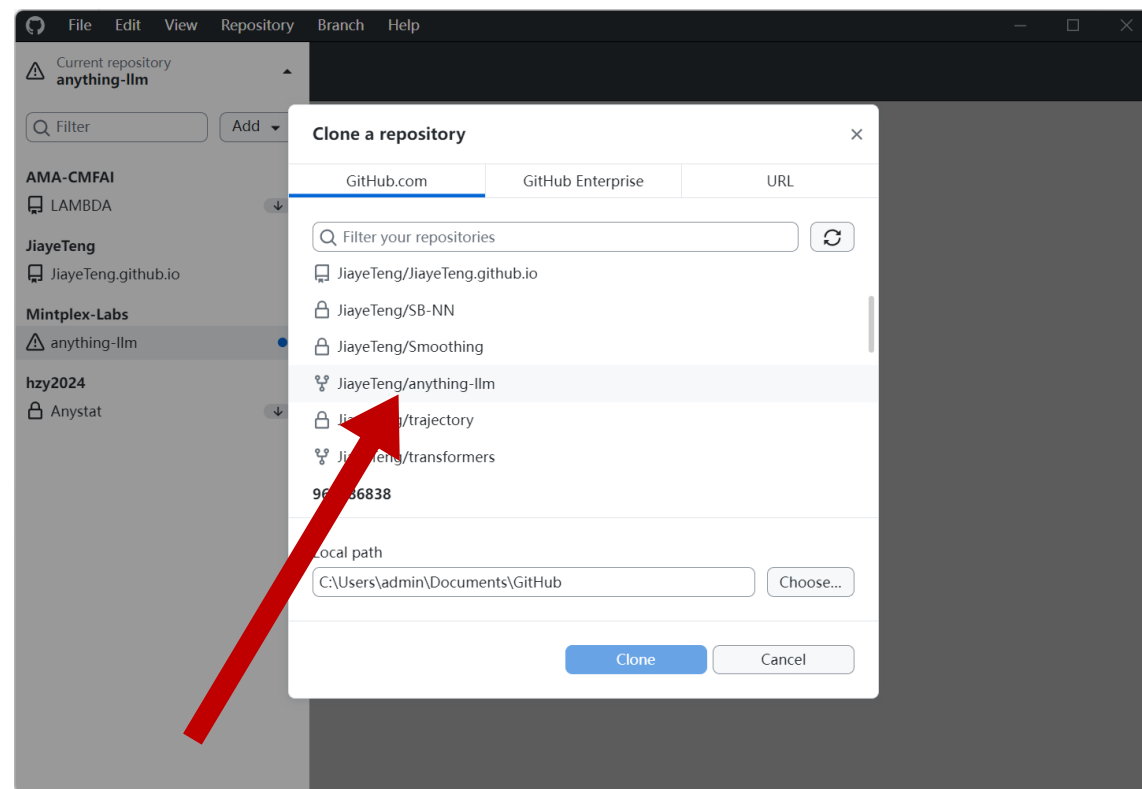
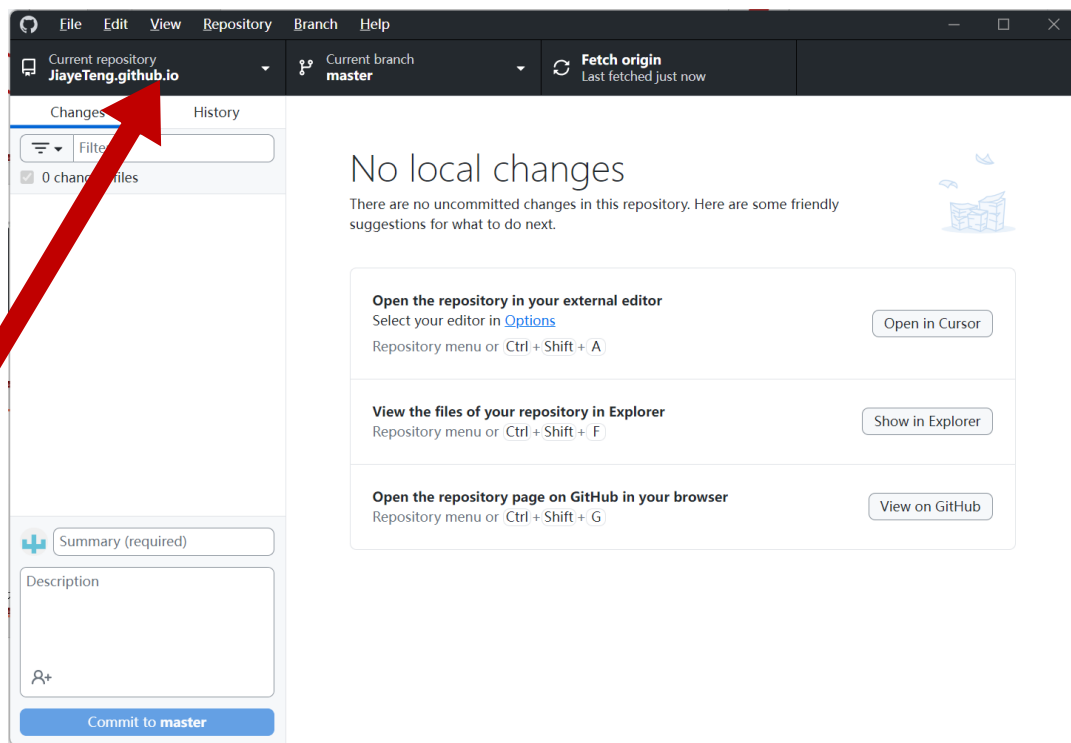
核心：连接云端的代码（github）与本机的代码（本地文件）

Github 桌面版



首先，将其他人的代码创建一个fork（接下来你的操作不会影响其他人云端的代码），全默认即可

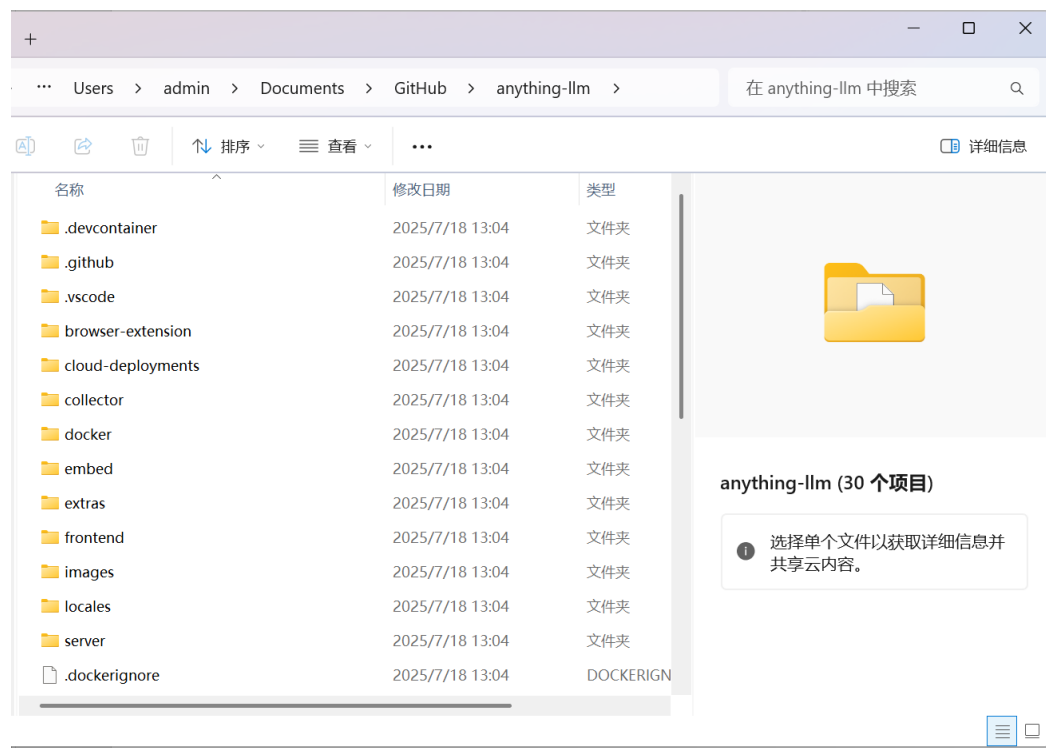
Github 桌面版



add – clone repo – 选择你的repo

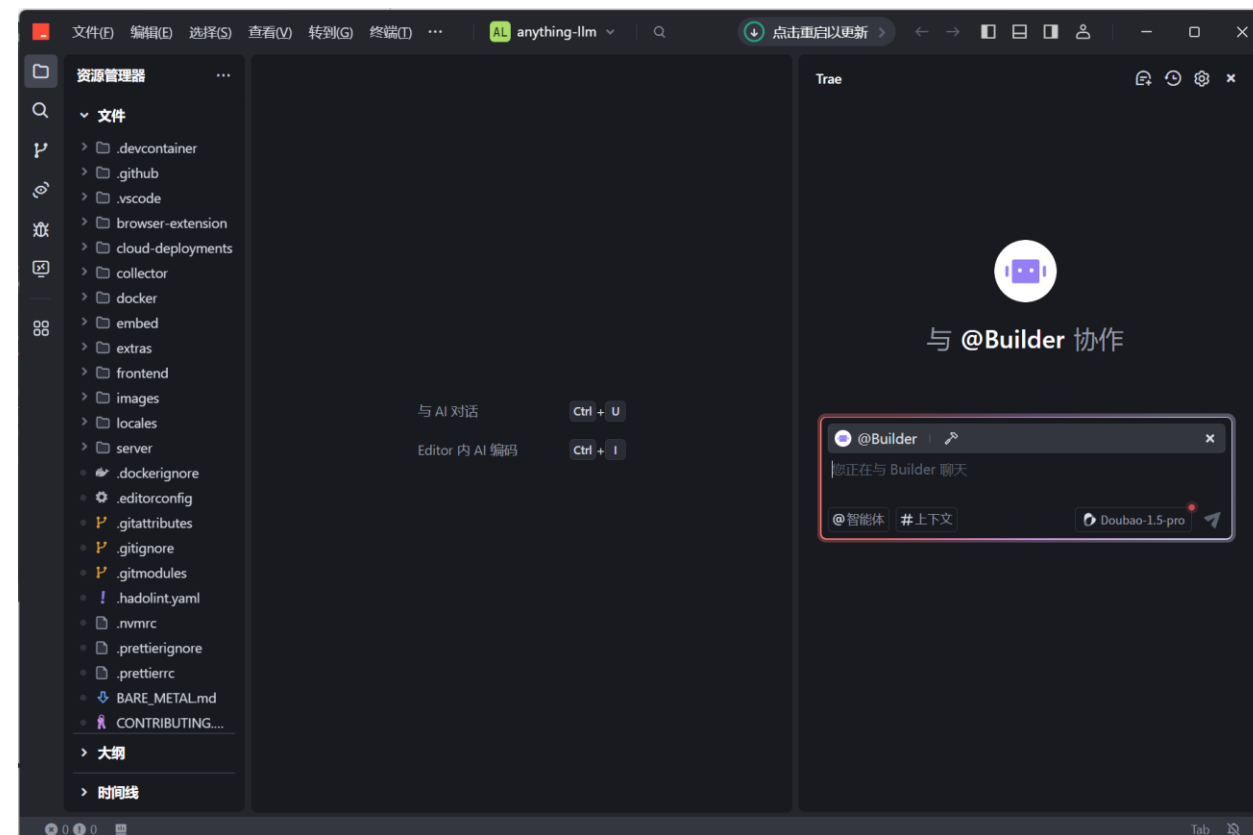
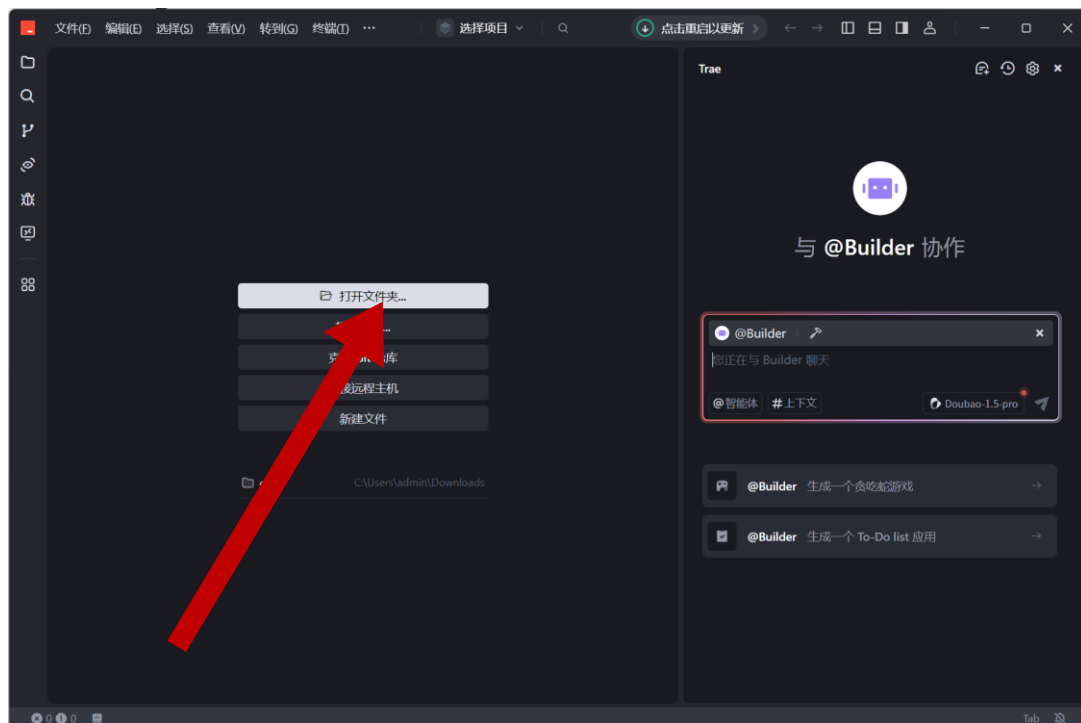
核心：连接云端的代码（github）与本机的代码（本地文件）

Github 桌面版



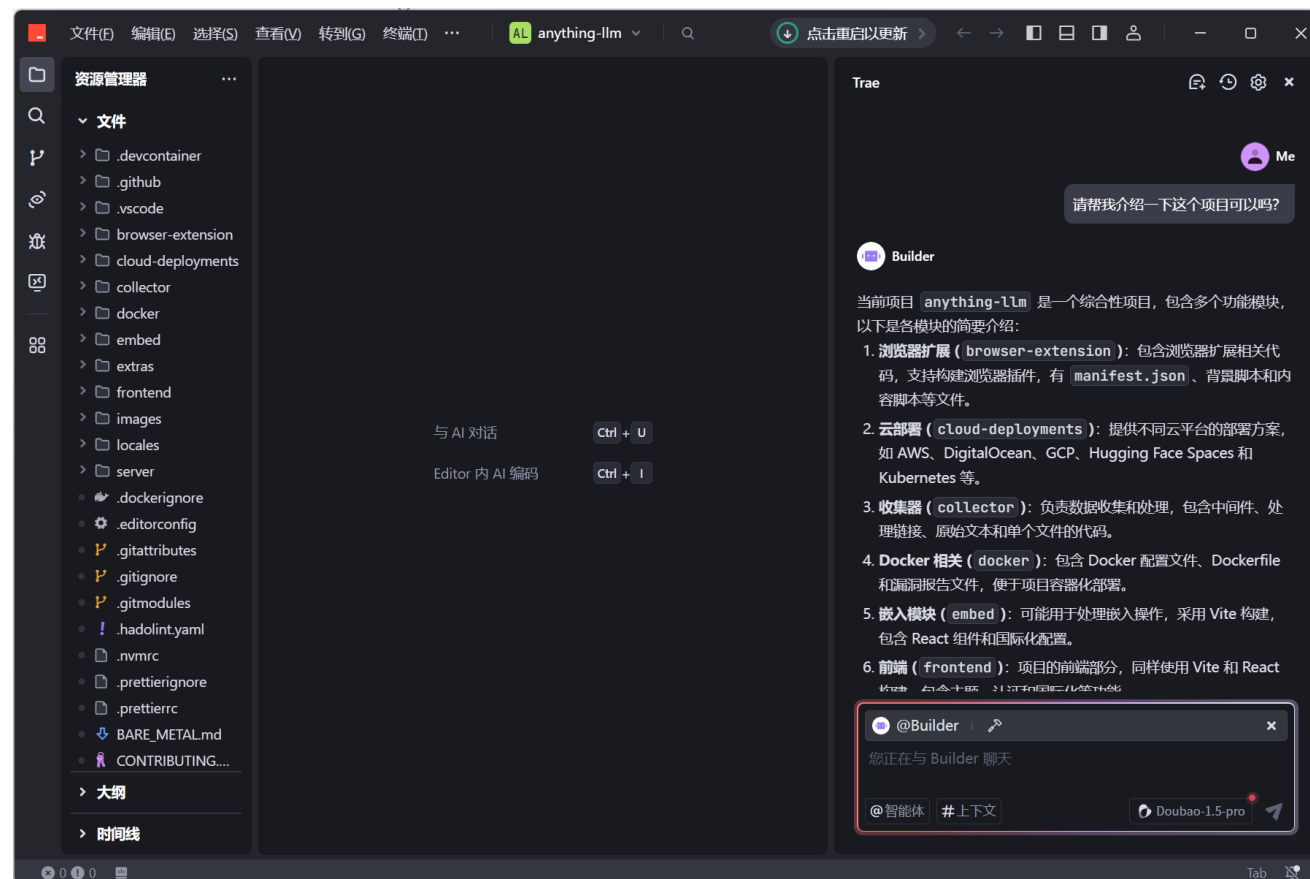
代码在本地进行存储

Github 与 Trae



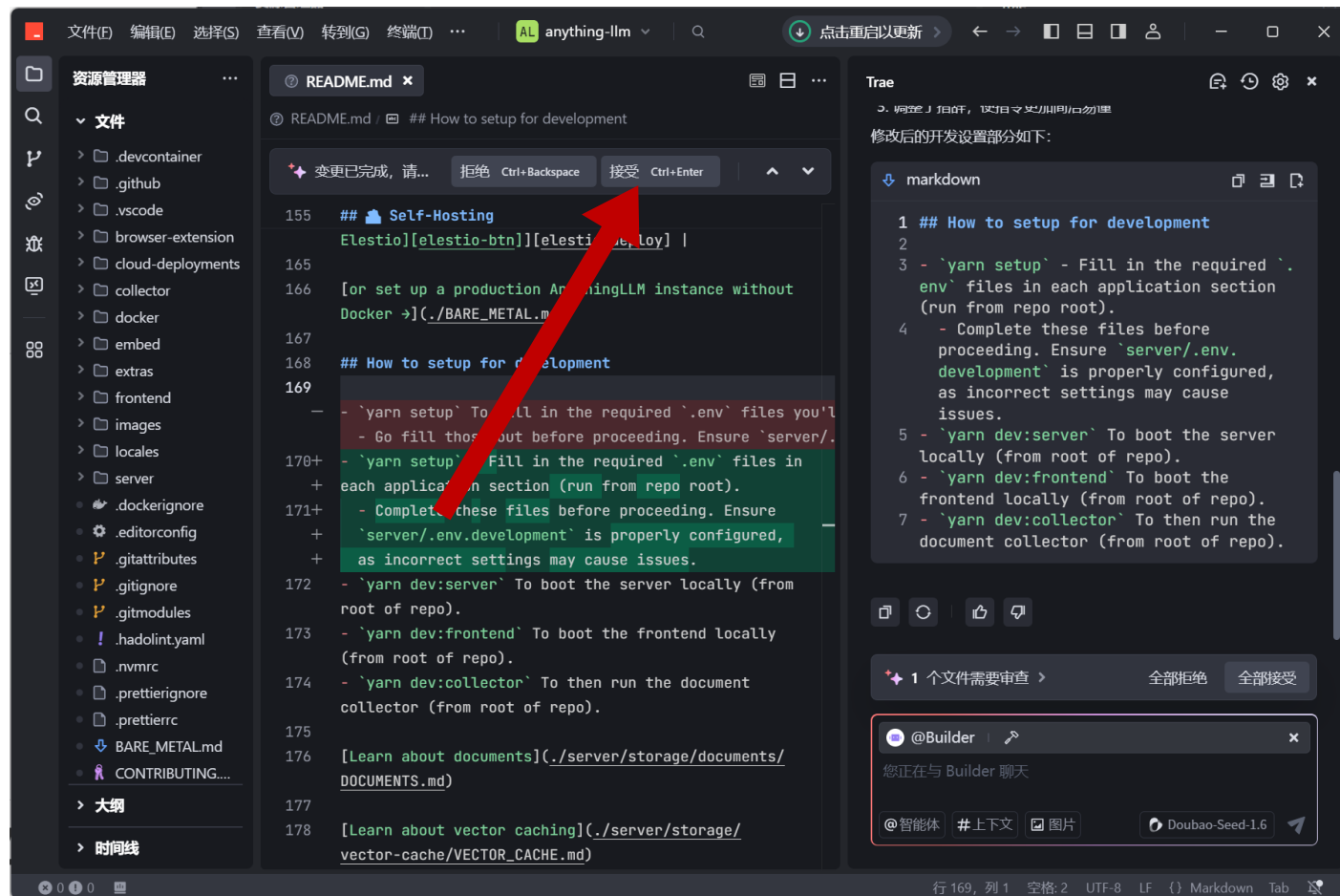
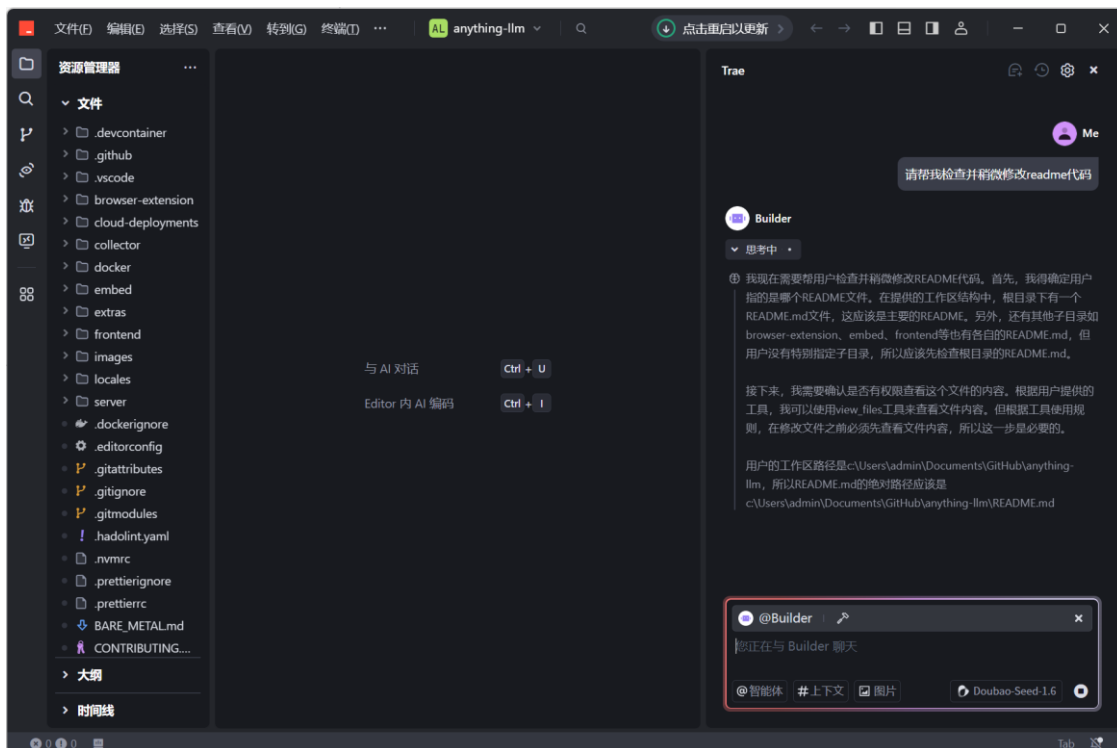
代码在本地进行存储
用Trae打开本地文件夹

Github 与 Trae: 查看文件



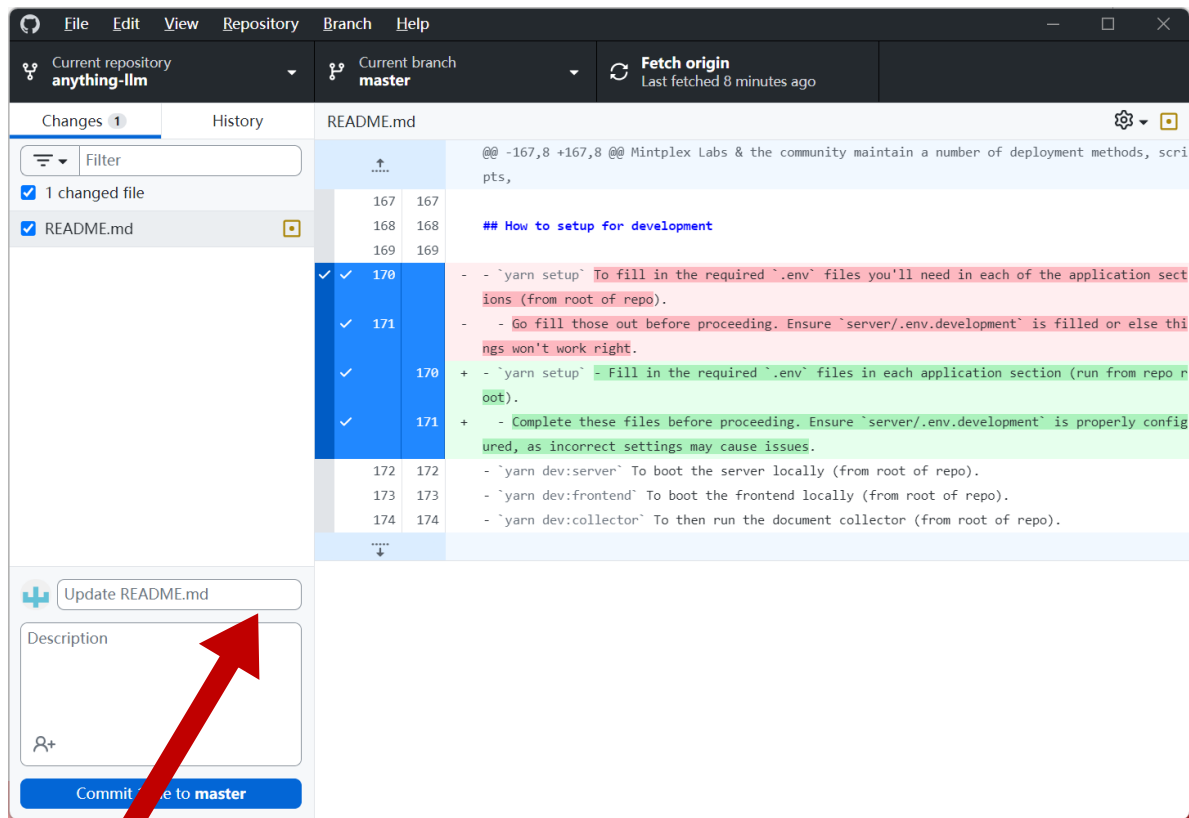
利用AI带你读代码

Github 与 Trae: 修改文件



利用AI修改代码（确定修改后点击确认）

Github 与 Trae: 修改文件

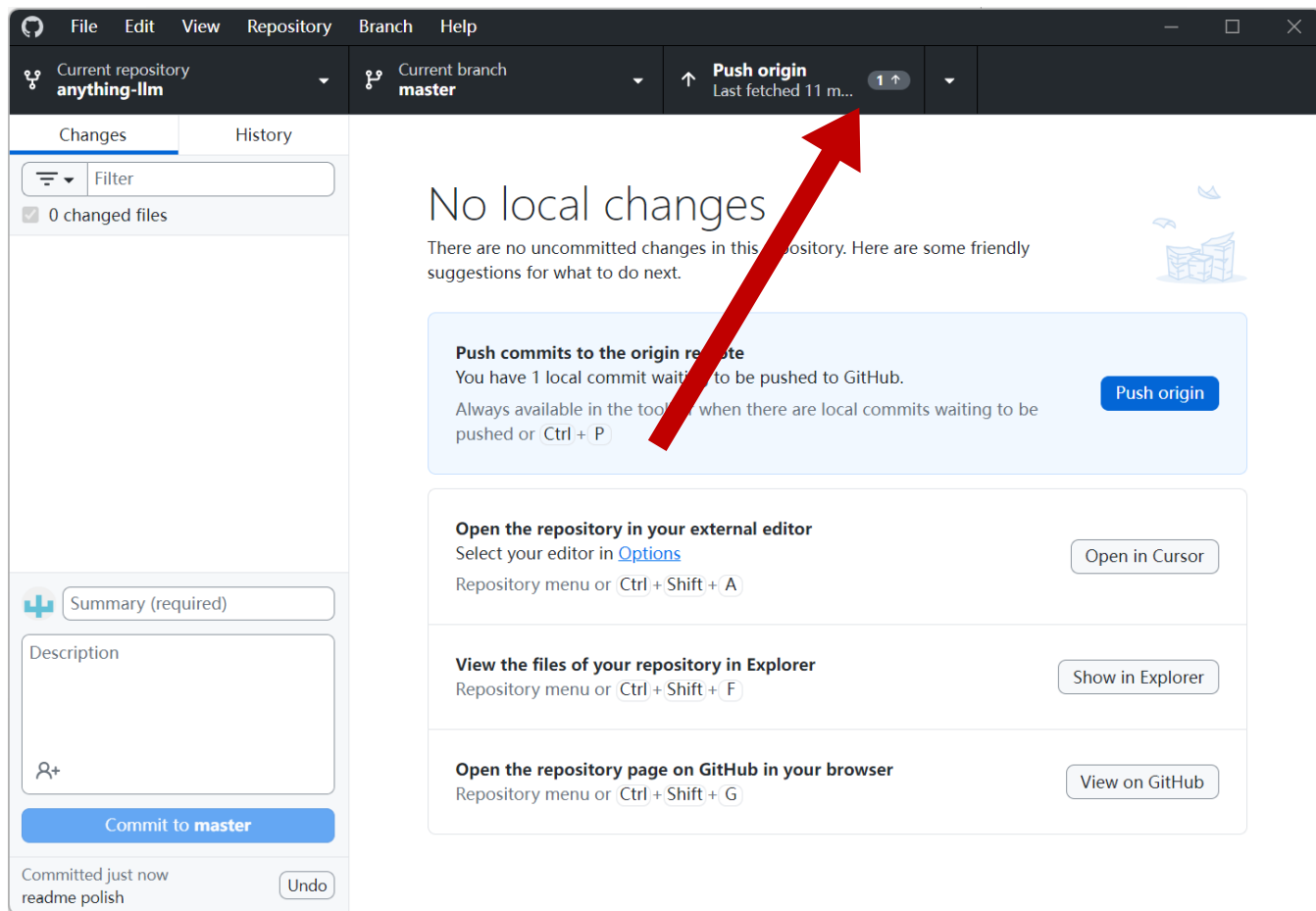


在修改代码的过程中，
任何一个时间点均可以
存一下checkpoint

填入此次checkpoint的
标题和一些描述

将修改后的文件用github存储checkpoint

Github 与 Trae: 修改文件



Push: 将你的代码传入云端
Pull: 将云端的代码（可能你的合作者有所修改）传回本地

将修改后的文件传到github云端

Github

事实上，Github并不**只**是一个代码仓库…

探索更多：<https://www.bilibili.com/video/BV1m4GhzEER3>

总结 Take-away Messages

大模型元素

- 提示词：结构化提示、上下文学习、思维链
- 大模型前沿：预训练、后训练、下游任务、多模态

Github

- 利用Github下载其他人代码
- 利用Github与Trae进行联动

第二次作业：找一个好玩的Github仓库

- 用Trae或Cursor打开并运行代码
- 用内置的AI生成对该仓库的说明
- 用AI在原有代码基础上添加一个小功能