

大模型时代—— 生产力工具转型的技术探索

商汤科技AIGC研究与应用团队负责人 詹明捷

人工智能新范式（AGI）已经到来

AI

AI + 场景应用

AI 新范式（AGI）



大模型使能的AGI与传统AI处理任务对比

零样本多模态思维链

(Zero-shot Multimodal Chain-of-Thought)



AI

明确任务

物体检测
Object Detection



明确任务

文字识别
OCR

明确任务

做决策
是否要减速?

是

AGI

给定图片

Q: 这个图标是什么意思?
我们应该做什么?

AGI

我们一步步思考

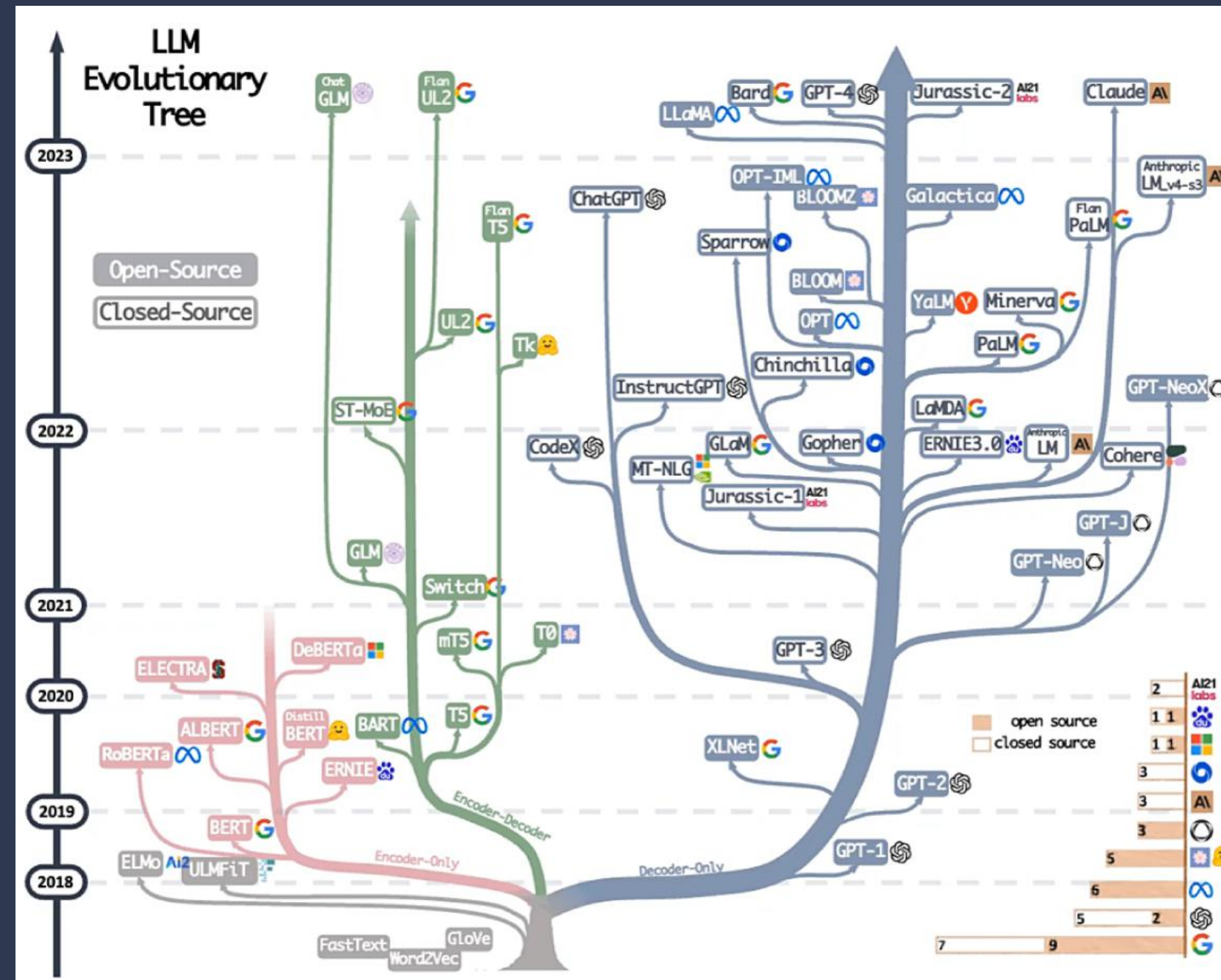
- 30 km/h
- 100m
- 小孩
- 学校区域
- 前方100米有一所学校
- 你需要小心小孩，并将车速降低到30 km/h以下

大模型核心能力



LLM技术路线演化，GPT类模型处于领先

ChatGPT等语言模型发展历程（进化树）



Encoder-Decoder or Encoder-Only (BERT类)

- 训练: Masked Language Models
- 模型类别: Discriminative
- 预训练任务: Predict masked words

Decoder-Only (GPT类)

- 训练: Autoregressive Language Models
- 模型类别: Generative
- 预训练任务: Predict next word

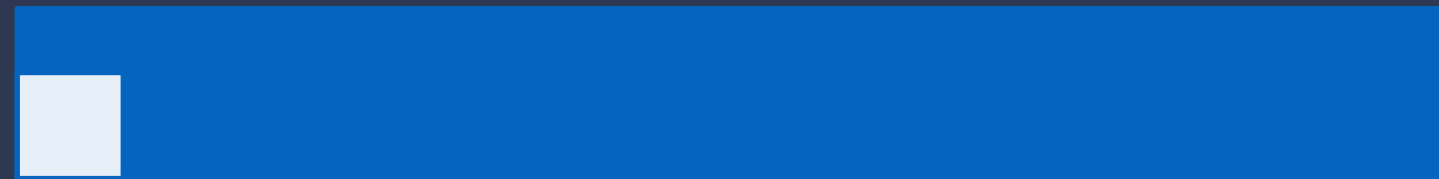
信息来源: The Practical Guides for Large Language Models

大模型关键技术路径

- **Encoder**类模型（以BERT为代表）在过去的方案中经过fine-tune性能会更强。OpenAI也是**坚持自身以生成式任务为目标的技术理想**，终于实现突破
- 随着考察任务的变化，现在大模型更关注zero-shot的文本生成能力（更贴近AGI），**Decoder-Only**类模型（以ChatGPT为代表的）更方便**利用海量的无监督文本数据进行训练**
- Decoder-Only类模型**结构效率更高**，OpenAI以系统工程的角度看待训练，着重训练效率，坚持此路线
- 如今大模型训练成本极高，由于历史原因，各企业组织（如Google）**存在路径依赖**，大多沿用自身已经验证通过的路线。

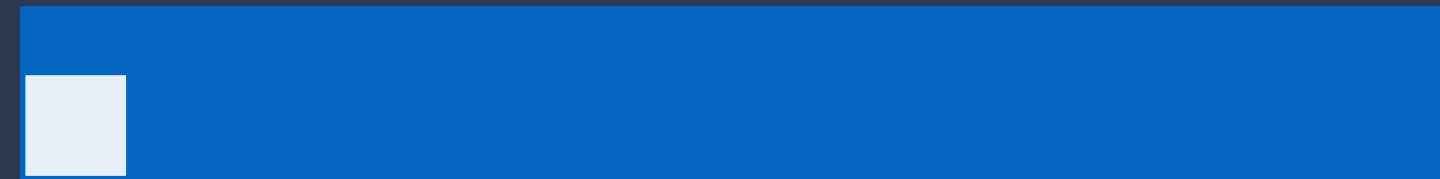
近几年NLP大事件

用好数据



- ChatGPT (OpenAI)
- llama2 (meta)
- Galactica (Meta)
- RETRO (DeepMind)
- WebGPT (OpenAI)
- Flan-PALM (Google)
- T0 (huggingface)

大模型



- GPT3 (OpenAI)
- PALM (Google)
- Megatron-LM (Nvidia)
- Turing (Microsoft)
- GLM-130B (清华)
- 盘古 (华为)
- ERNIE 3.0 (百度)

什么是思维链？

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. ✅

思维链激发语言模型解锁更强的能力，解锁更复杂的推理任务，右侧图绿色部分为思维链

Finetuning tasks

TO-SF

Commonsense reasoning
Question generation
Closed-book QA
Adversarial QA
Extractive QA
Title/context generation
Topic classification
Struct-to-text
...

**55 Datasets, 14 Categories,
193 Tasks**

Muffin

Natural language inference	Closed-book QA
Code instruction gen.	Conversational QA
Program synthesis	Code repair
Dialog context generation	...

69 Datasets, 27 Categories, 80 Tasks

CoT (Reasoning)

Arithmetic reasoning	Explanation generation
Commonsense Reasoning	Sentence composition
Implicit reasoning	...

9 Datasets, 1 Category, 9 Tasks

Natural Instructions v2

Cause effect classification
Commonsense reasoning
Named entity recognition
Toxic language detection
Question answering
Question generation
Program execution
Text categorization
...

**372 Datasets, 108 Categories,
1554 Tasks**

- ❖ A **Dataset** is an original data source (e.g. SQuAD).
- ❖ A **Task Category** is unique task setup (e.g. the SQuAD dataset is configurable for multiple task categories such as extractive question answering, query generation, and context generation).
- ❖ A **Task** is a unique <dataset, task category> pair, with any number of templates which preserve the task category (e.g. query generation on the SQuAD dataset.)

Instruction finetuning

Please answer the following question.
What is the boiling point of Nitrogen?

Chain-of-thought finetuning

Answer the following question by reasoning step-by-step.
The cafeteria had 23 apples. If they used 20 for lunch and bought 6 more, how many apples do they have?

Language model

-320.4F

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 $3 + 6 = 9$.

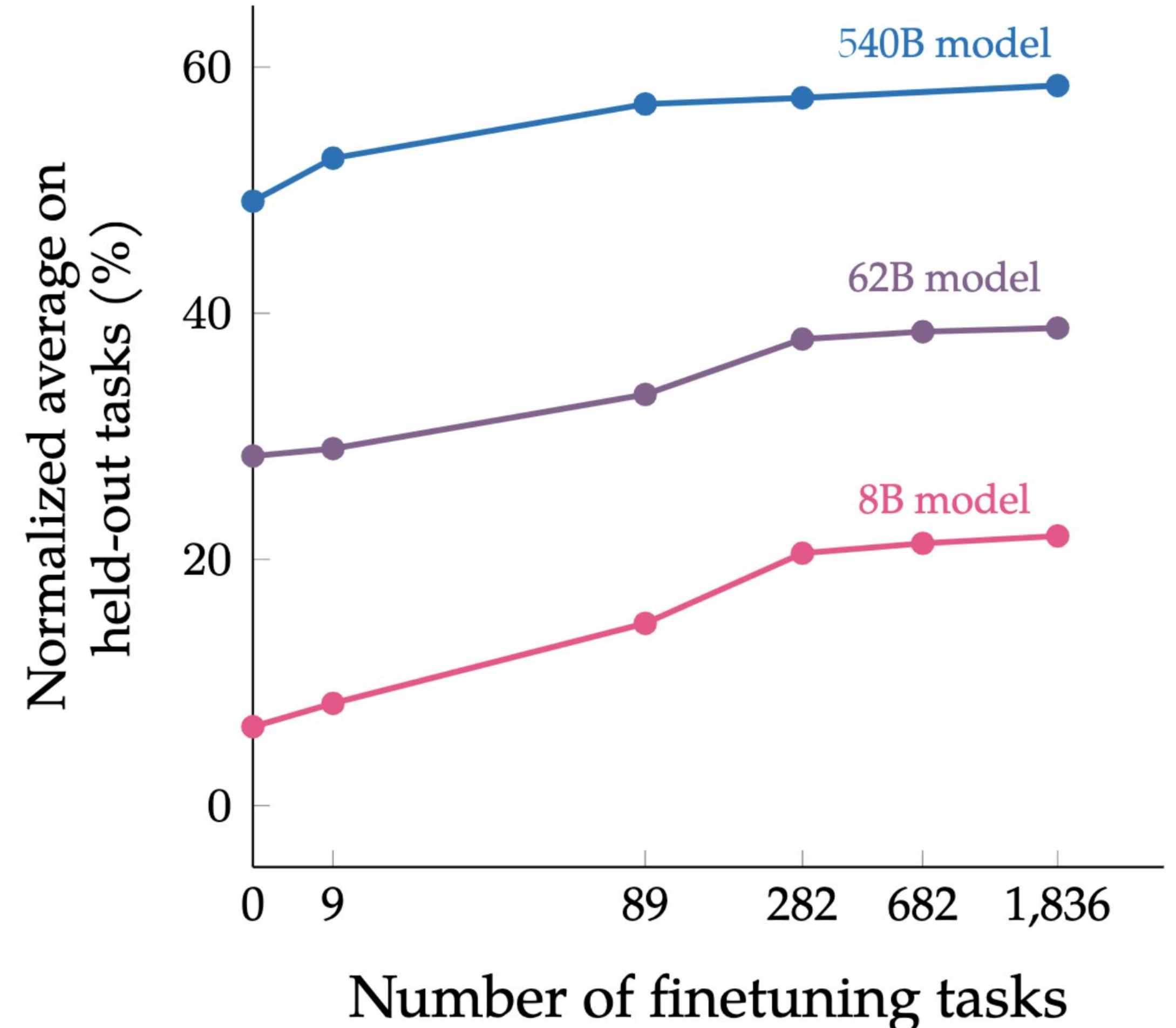
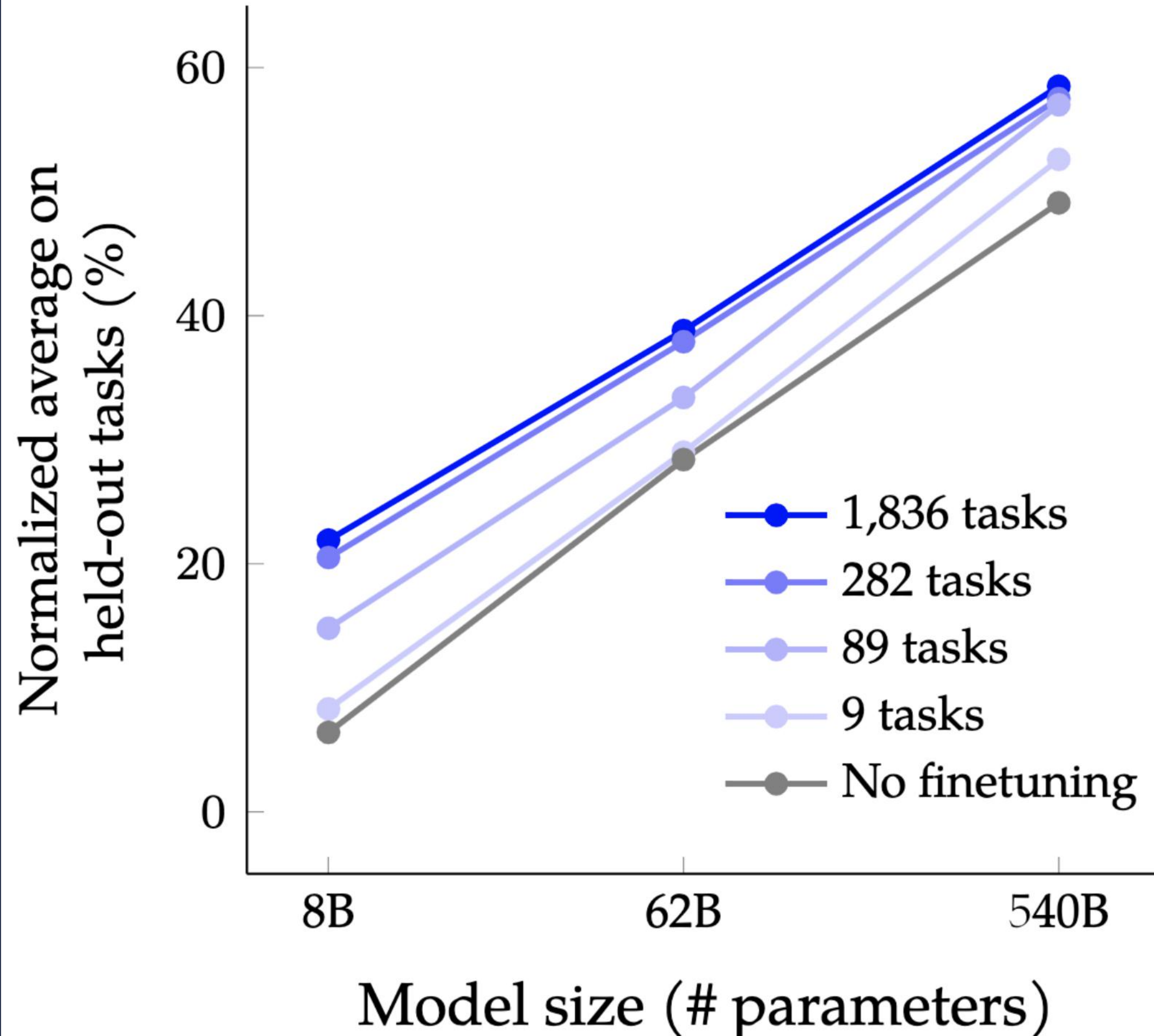
Multi-task instruction finetuning (1.8K tasks)

Inference: generalization to unseen tasks

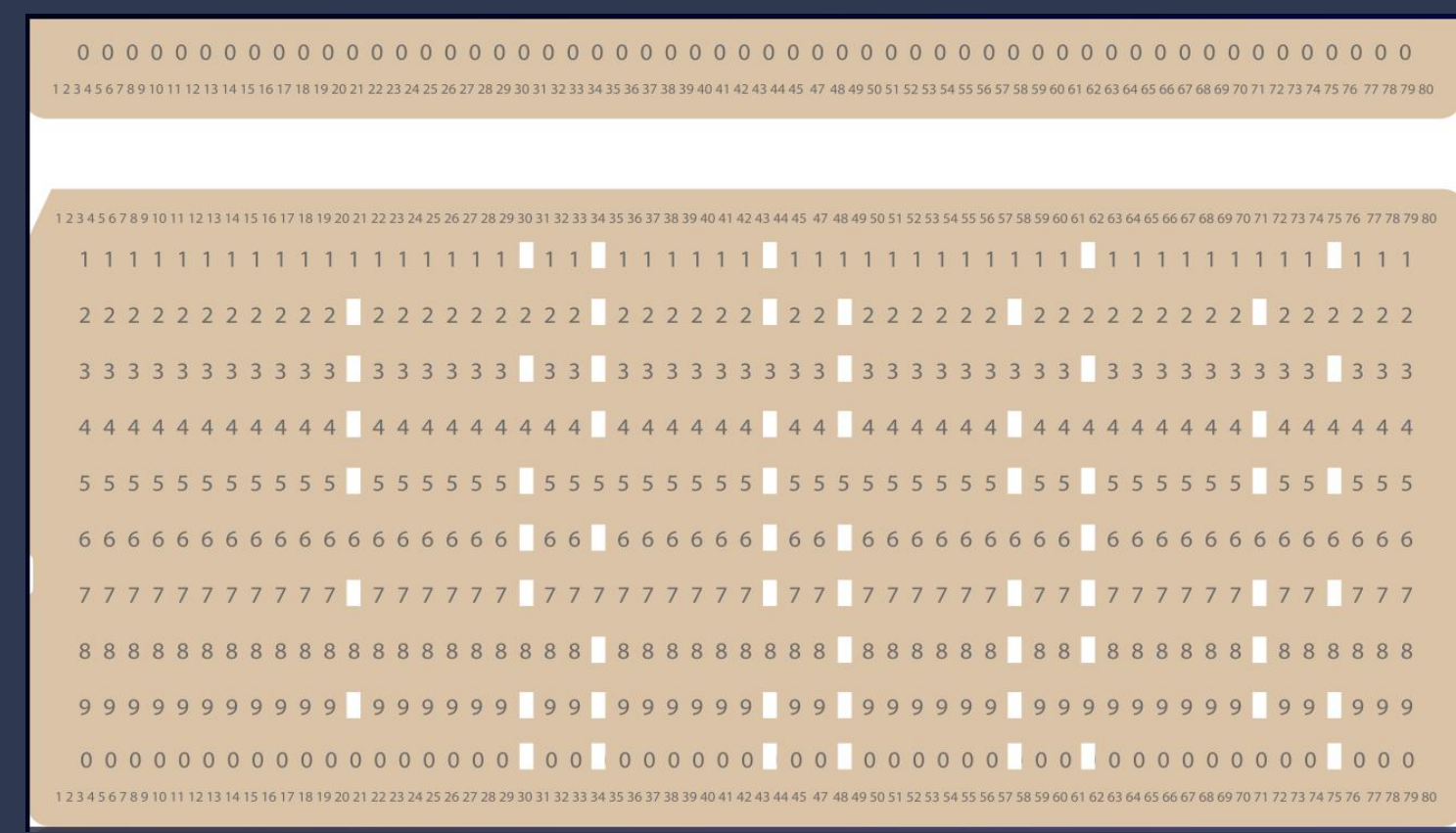
Q: Can Geoffrey Hinton have a conversation with George Washington?
Give the rationale before answering.

Geoffrey Hinton is a British-Canadian computer scientist born in 1947. George Washington died in 1799. Thus, they could not have had a conversation together. So the answer is "no".

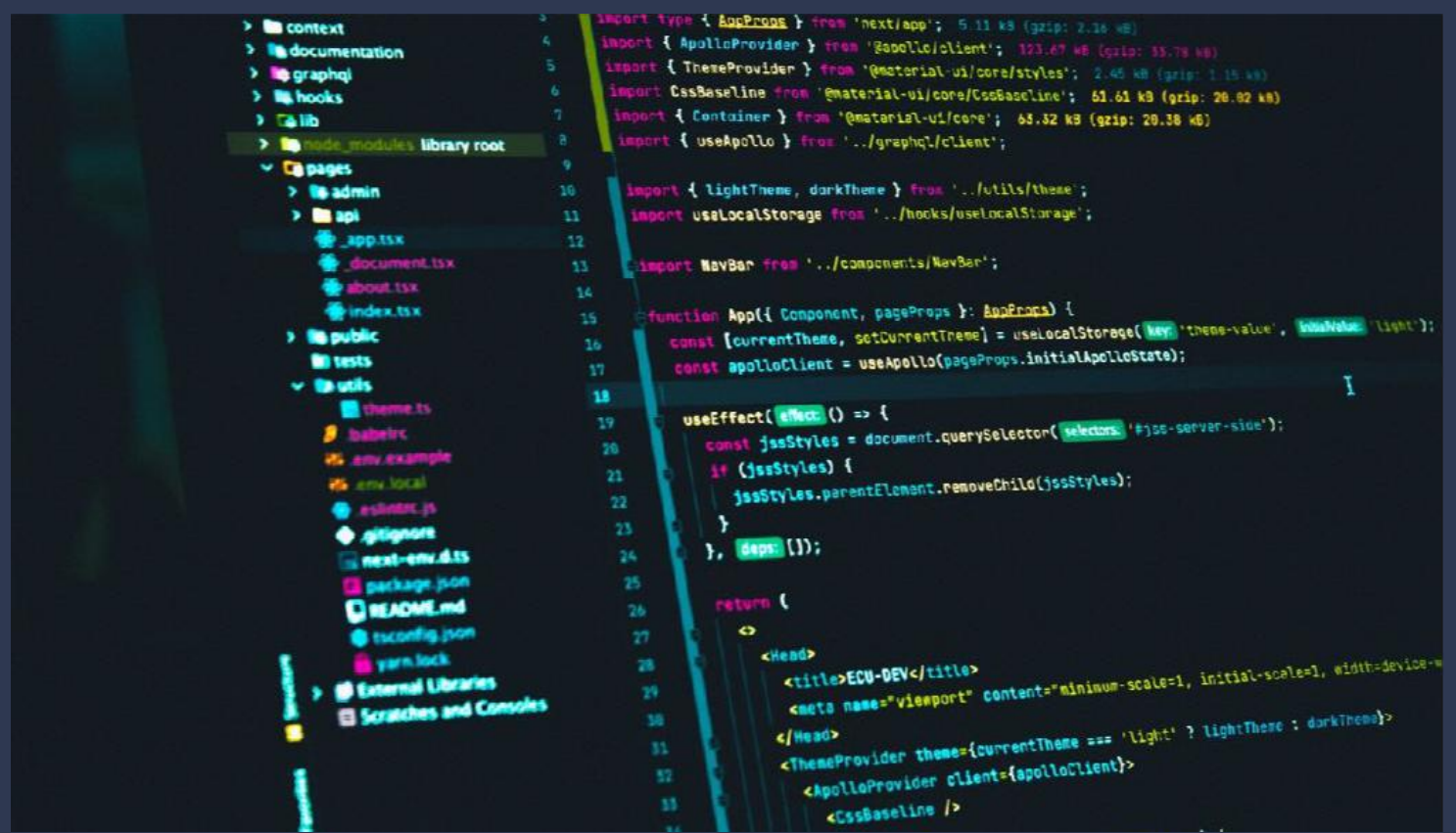
收益分析



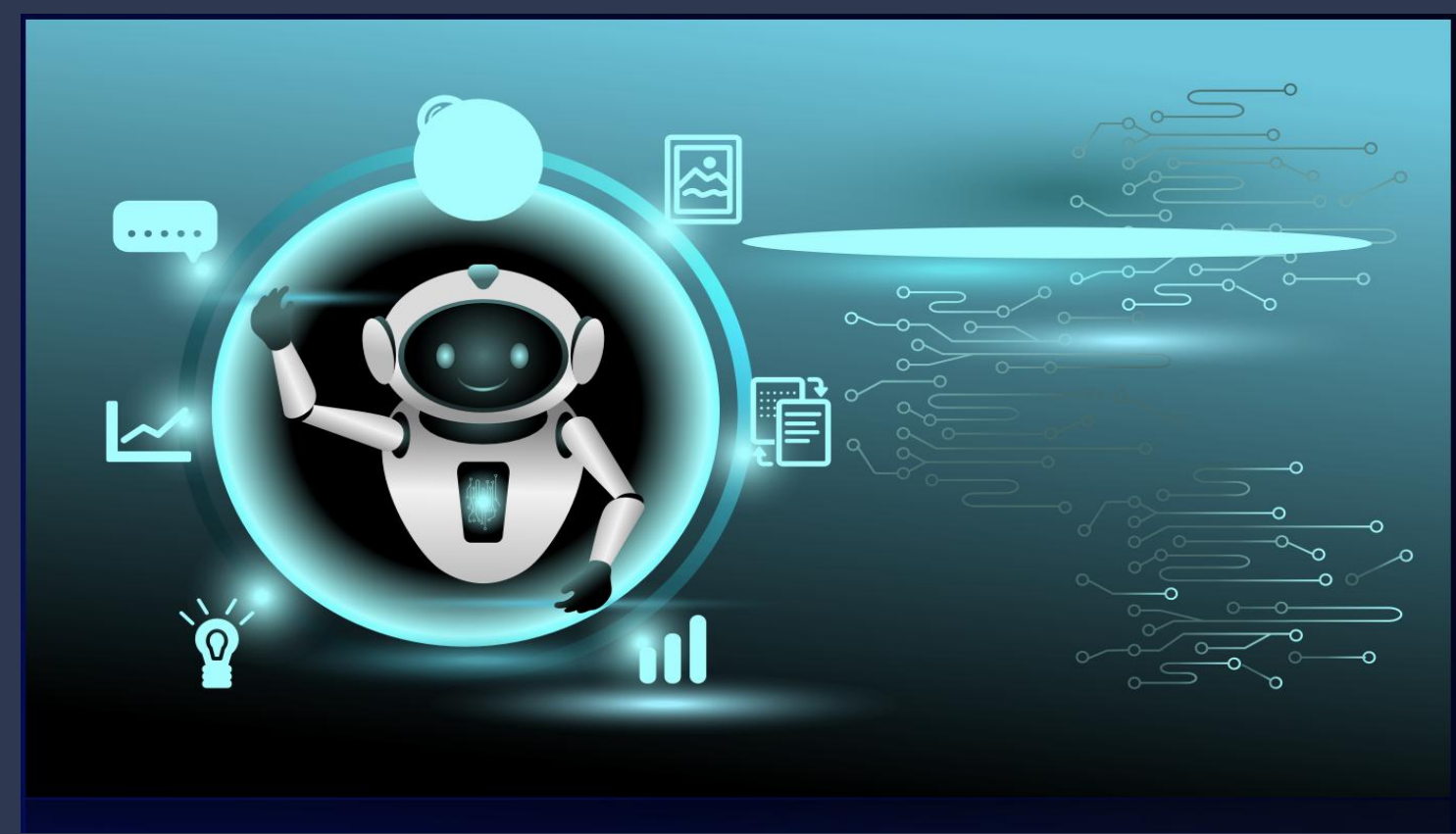
编程工具的发展史



打孔卡片



编程语言



生成式AI

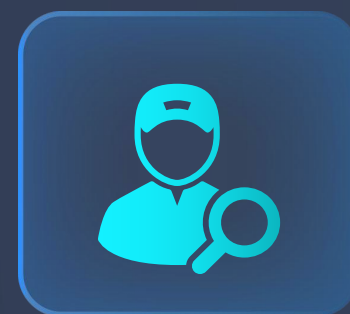
AI是生产生产力工具的工具
AI FOR AI

下一代软件开发范式：AI FOR AI

人工智能大模型

智能化、模块坏、标准化的AI生产工具

开发智能化



流程标准化

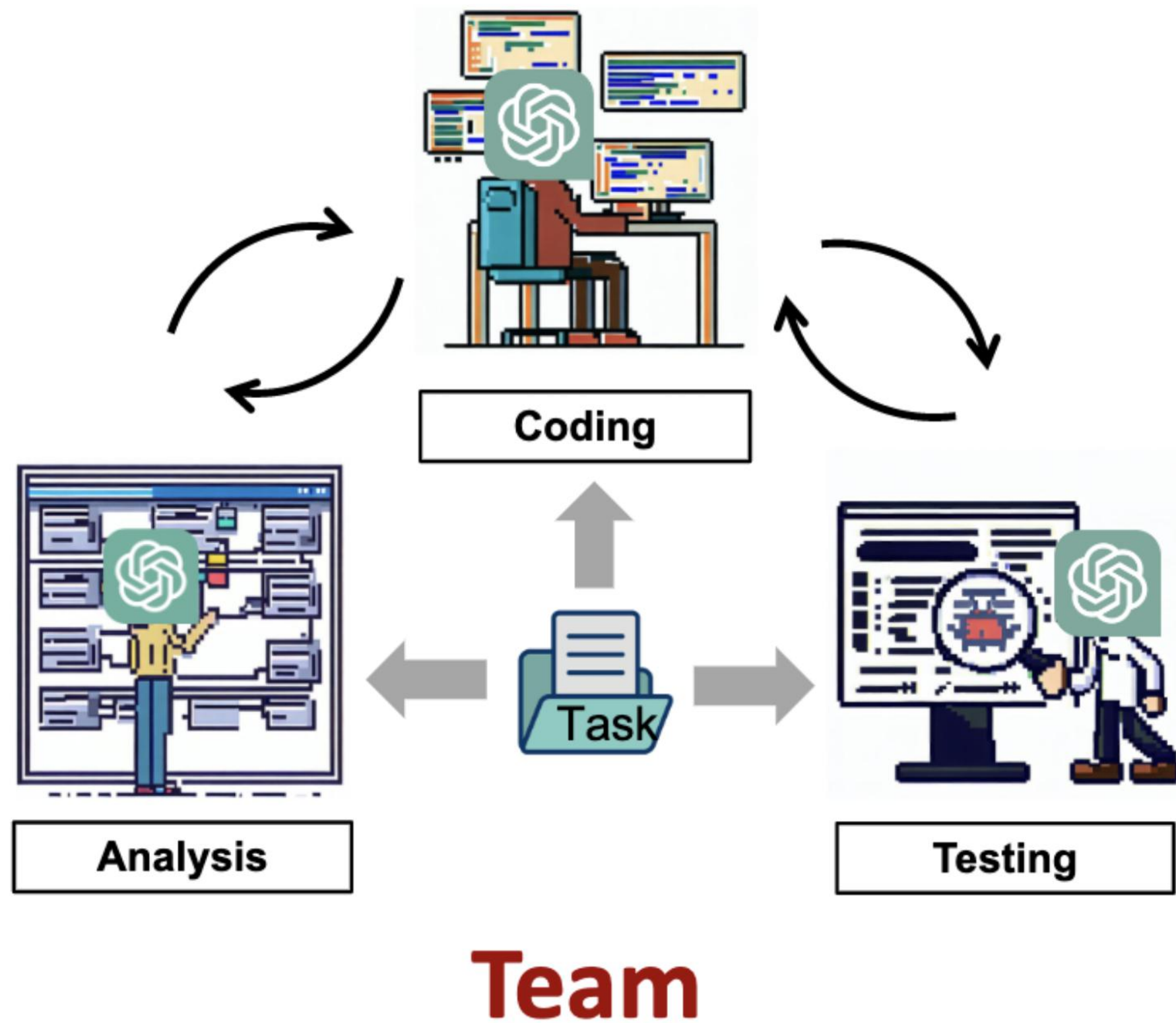


工具模块化



AI辅助开发、一键调度大算力、数据驱动、快速部署

软件开发不同角色：分析师、程序员和测试员

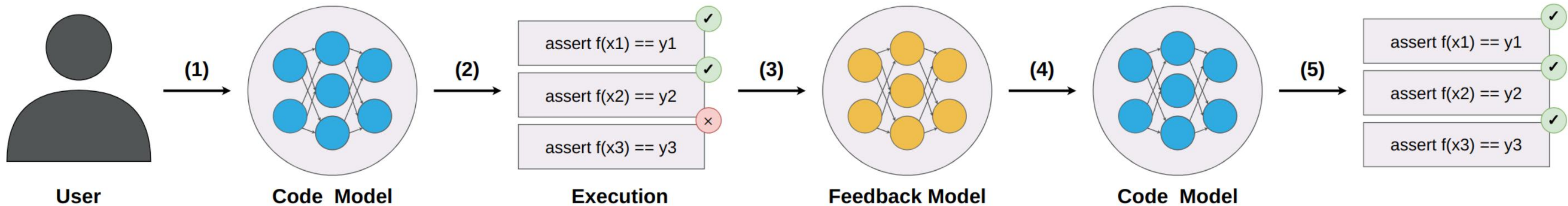


分析师：分析师的目标是制定高层次的 plan 并专注于指导程序员编写程序，而不是深入研究实现细节。给定需求X，分析师将分解为几个易于解决的子任务，以方便程序员直接实施，并制定概述实施主要步骤的 plan。

程序员：作为该团队的核心角色，程序员将在整个开发过程中接收来自分析师的 plan 或来自测试人员的测试报告。将两项主要职责分配给程序员：1. 编写满足指定要求的代码，遵守分析师提供的 plan。2. 修复或细化代码，考虑到测试人员反馈的测试报告反馈。

测试员：测试人员获取程序员编写的代码，随后记录包含各个方面的测试报告，例如功能性、可读性和可维护性。与直接生成测试用例相比，生成测试报告更符合语言模型的倾向。

自动Debug



(1)

Given is a string s representing the day of the week today. s is one of SUN, MON, TUE, WED, THU, FRI, or SAT. After how many days is the next Sunday (tomorrow or later)?

```
# UNIT TESTS
# (EXECUTABLE)
assert f('MON') == 6
assert f('WED') == 4
assert f('SUN') == 7
```

```
def f(s):
    return (7 - ['SUN', ..., 'FRI', 'SAT'].index(s)) % 7
```

(2)

Given input 'SUN', the program returned 0, but the expected output was 7.

(3)

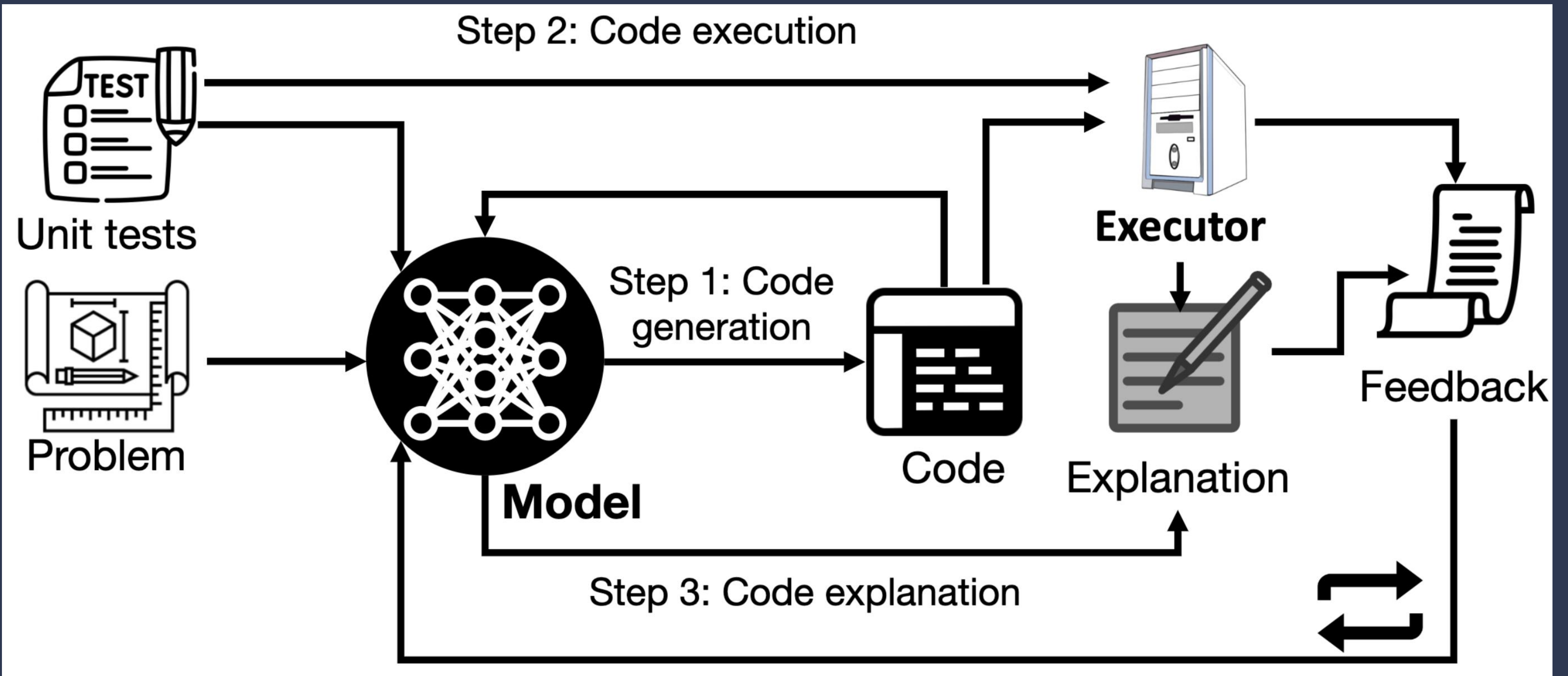
The code does not account for the case where the input is 'SUN' and the output should be 7. This can be fixed by removing the modulo operation.

(4)

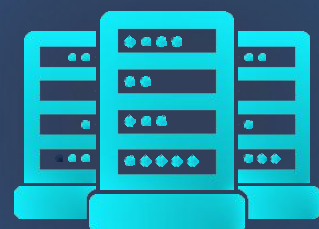
```
def f(s):
    return (7 - ['SUN', ..., 'FRI', 'SAT'].index(s)) # % 7
```

(5)

自动Debug



新二八定律：代码 = 80% AI 生成 + 20% 人工



商汤大语言模型 ➡ 商汤AI开发代码助手 ➡

48.7%
8.5pp ↑

HumanEval 测试集一次通过率 (GPT 3.5 48.1%)

25ms
1.6X ↑

每token代码生成速度

1.9K
243X ↑

推理吞吐量
(Tokens单台每秒)

新二八定律：代码 = 80% AI 生成 + 20% 人工



48.7%
8.5pp ↑

HumanEval 测试集一次通过率 (GPT 3.5 48.1%)

25ms
1.6X ↑

每token代码生成速度

1.9K
243X ↑

推理吞吐量
(Tokens单台每秒)

1.0

代码补全、代码扩写、代码翻译、代码重构、代码修正、注释生成代码、复杂度分析、测试用例生成



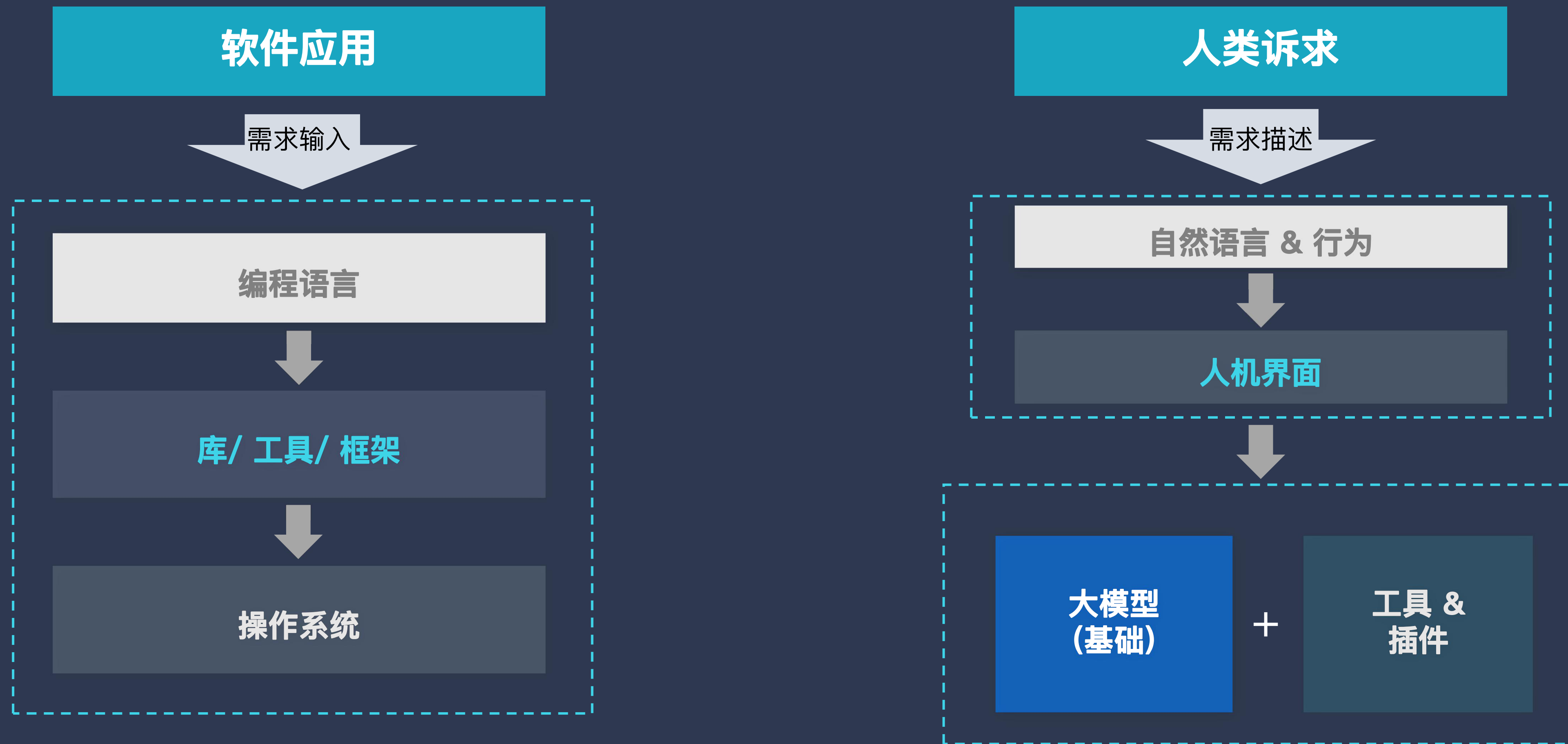
2.0

多轮对话、代码添加注释、代码解释、代码中间补全...

78% ↑
16pp

代码编写效率提升

“新二八定律”带来软件开发新体验



语言与操作系统，是当前人类与世界知识交互的核心媒介依赖：

- 人类需求的表达/生成方式将发生变化。➡ 上层行业应用创新 ➡ **主流行业应用都可以被重新做一遍**
- 人类需求被满足的方式和底层依赖发生变化。➡ **底层编程逻辑变革**

商汤AI代码助手

业界持续推出生成式AI技术延展

Auto-GPT

Plugins

Code Interpreter

CoT

Function Calling

MOE

...

延展：用代码解决视觉问题



Question:
Is the carriage to the right of a horse?

Codex
(Few-Shot Prompting)

In-Context Examples

```
# Image 1: On which side of the picture is the rug?  
img = open_image("Image1.jpg")  
rug_pos_x, rug_pos_y =  
get_pos(img, "rug")  
if rug_pos_x < (LEFT+RIGHT)/2:  
    answer = "left"  
else:  
    answer = "right"  
...
```

Code Generation

```
horse_exists = query(img, "Is there  
a horse?")  
answer = "no"  
  
if horse_exists == "yes":  
    carriage_pos_x, carriage_pos_y =  
        get_pos(img, "carriage")  
    horse_pos_x, horse_pos_y =  
        get_pos(img, "horse")  
    if carriage_pos_x > horse_pos_x:  
        answer = "yes"
```

Execute Code

query(img, "Is there a horse?")

Captions:

1. 'a police horse pulled by a fire policeman in a wagon',
2. 'man riding a horse drawn carriage pulling horse next to a officer', ...

returns "yes"

get_pos(img, "carriage") *get_pos(img, "horse")*



returns 5, 11

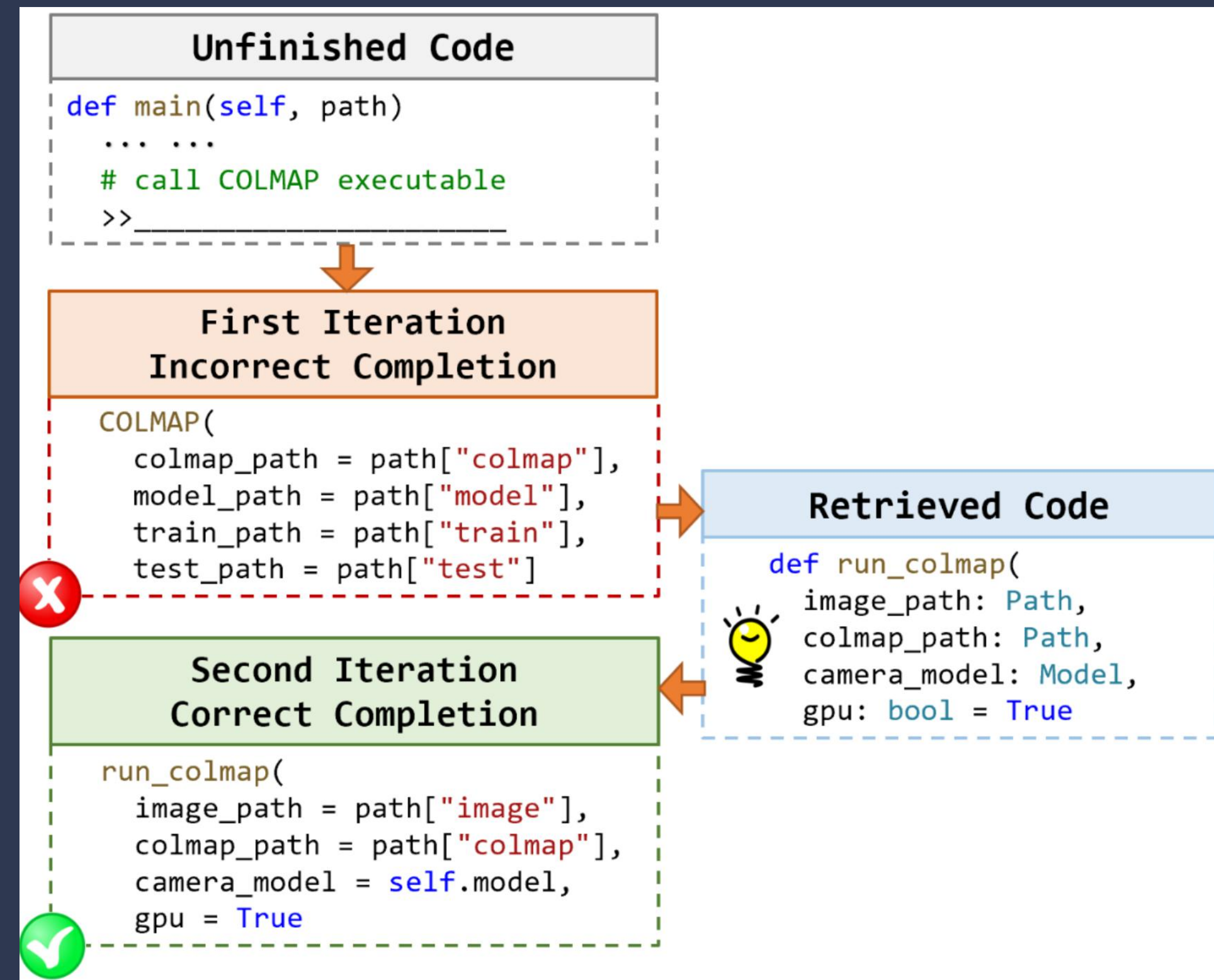
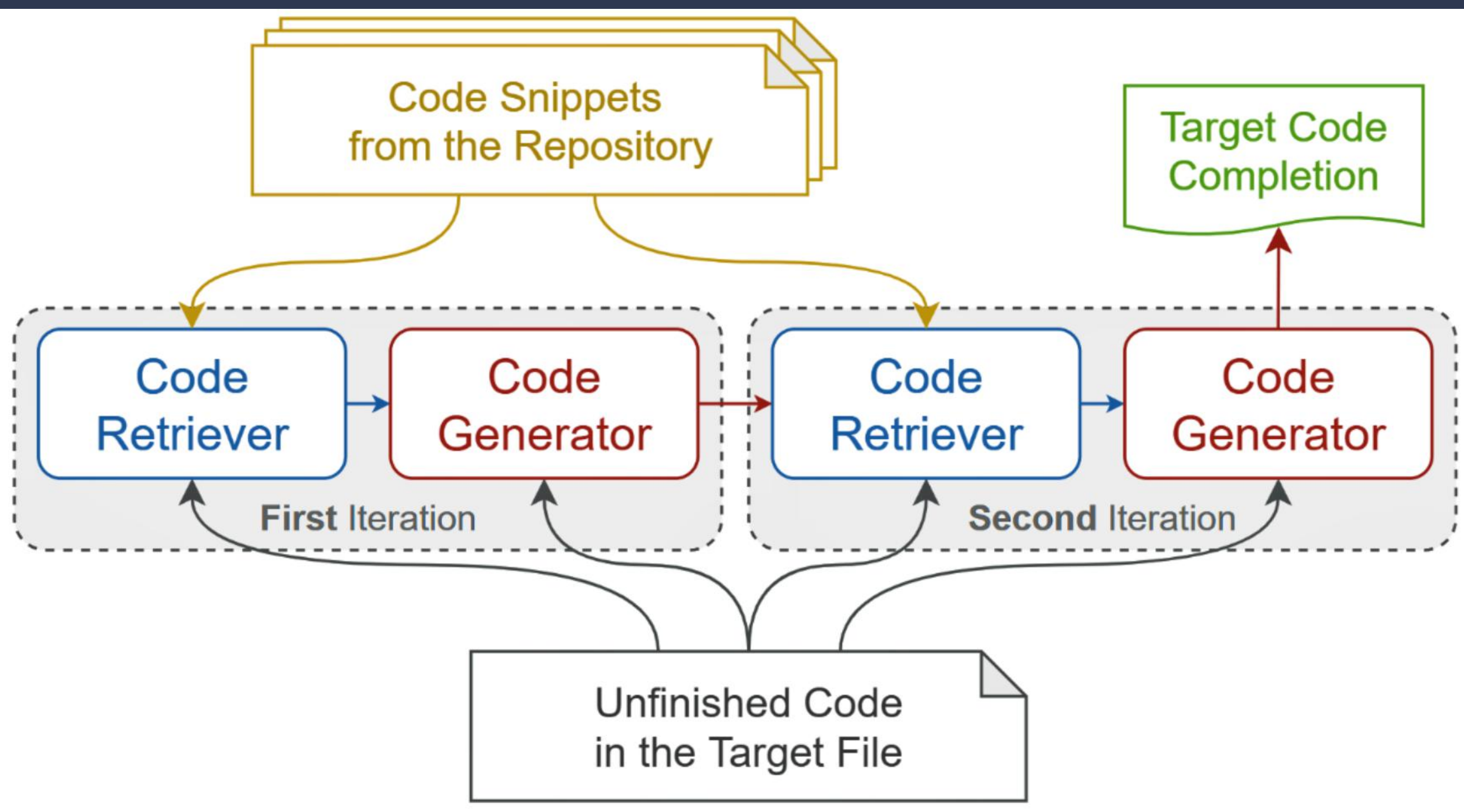


returns 12, 11

carriage_pos_x < horse_pos_x

Answer:
No

延展：基于多仓库融合的代码生成



延展：基于代码解决数学题

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 tennis balls.

`tennis_balls = 5`

2 cans of 3 tennis balls each is

`bought_balls = 2 * 3`

tennis balls. The answer is

`answer = tennis_balls + bought_balls`

Q: The bakers at the Beverly Hills Bakery baked 200 loaves of bread on Monday morning. They sold 93 loaves in the morning and 39 loaves in the afternoon. A grocery store returned 6 unsold loaves. How many loaves of bread did they have left?

Model Output

A: The bakers started with 200 loaves

`loaves_baked = 200`

They sold 93 in the morning and 39 in the afternoon

`loaves_sold_morning = 93`

`loaves_sold_afternoon = 39`

The grocery store returned 6 loaves.

`loaves_returned = 6`

The answer is

`answer = loaves_baked - loaves_sold_morning
- loaves_sold_afternoon + loaves_returned`

`>>> print(answer)`

74



THANKS

如果您有兴趣进一步了解商汤AI代码助手，可联系：
sensecode@sensetime.com