

# 《GPT-4，通用人工智能的火花》 论文内容精选与翻译

## 引言：

《通用人工智能的火花：GPT-4早期实验》是3月最重要的一篇文章，引起了广泛的关注和讨论，但是论文长达 154页，中文版本还无人翻译。

本文挑选了论文中的重点结论并进行翻译，虽然已经是精选，但仍然超过万字。但考虑到 GPT5 明年才能面世，这篇文章在今年什么时候看都不晚。

微软的研究院在很早期就接触到了 GPT-4 的非多模态版本，并对齐进行了详尽的测试。这篇论文就是整个的测试过程和结论。不管是测试方法还是结论都非常精彩，强烈推荐看一遍，传送门在此。<https://arxiv.org/pdf/2303.12712v1.pdf>

本文的翻译没有添加任何夸张的修辞（DeepL和ChatGPT贡献也很大），但文中透露的信息本身已足够震撼。

本文的网页链接，欢迎分享转发 <https://orangeblog.notion.site/GPT-4-8fc50010291d47efb92cbbd668c8c893>

## 基本信息：

测试者：Microsoft Research

测试模型：GPT-4早期模型，非多模态版本。

## 基本结论：

尽管是纯粹的语言模型，这个早期版本的GPT-4在各种领域和任务上表现出显著的能力，包括抽象、理解、视觉、编码、数学、医学、法律、对人类动机和情感的理解等等。

**GPT-4的能力具有普遍性，它的许多能力跨越了广泛的领域，而且它在广泛的任务中的表现达到或超过了人类水平，这两者的结合使我们可以说GPT-4是迈向AGI的重要一步。**

虽然GPT-4在许多任务上达到或超过了人类的水平，但总体而言，它的**智能模式明显地不像人类**。

GPT-4只是迈向通用智能系统的第一步。然而即使作为第一步，GPT-4也挑战了相当多的关于机器智能的假设，并表现出涌现的行为和能力，其来源和机制目前还不够清楚。

我们撰写本文的主要目的是分享我们对GPT-4的能力和局限性的探索，以支持我们关于技术飞跃的评估。我们相信，**GPT-4的智能标志着计算机科学领域及其他领域的真正范式转变**。

## 研究方法：

本文的更接近于传统的心理学而不是机器学习，借鉴了人类的创造力和好奇心。我们的目标是生产新的和困难的任务和问题，令人信服地证明GPT-4远远超出了记忆的范围，并且它对概念、技能和领域有深刻和灵活的理解。我们还旨在探究GPT-4的反应和行为，以验证其一致性、连贯性和正确性，并揭示其局限性和偏见。我们承认，这种方法有些主观和不正式，可能无法满足科学评估的严格标准。然而，我们认为这是一个有用的和必要的第一步，以了解GPT-4的显著能力和挑战，这样的第一步为开发更正式和全面的方法来测试和分析具有更普遍智能的AI系统开辟了新的机会。

GPT-4的主要优势在于它对自然语言的掌握无可比拟。它不仅可以生成流畅和连贯的文本，还可以以各种方式理解和处理文本，如总结、翻译或回答一系列极其广泛的问题。此外，**我们所说的翻译不仅是指不同自然语言之间的翻译，还包括语气和风格的翻译，以及跨领域的翻译，如医学、法律、会计、计算机编程、音乐等等**。这些技能清楚地表明，**GPT-4能够理解复杂的思想**。

许多读者心中可能萦绕的一个问题是，GPT-4是否真正理解了所有这些概念，或者它是否只是在即兴发挥方面比以前的模型好得多，而没有任何真正深刻的理解。我们希望在阅读完这篇论文后，这个问题几乎会被反转，让人不禁思考：**真正深刻的理解和**

**即兴临场发挥的差别在哪里？**一个能通过软件工程候选人考试的系统难道不是真正的智能吗？对于【真正深刻的理解】，也许唯一的测试手段，就是看它能否能产生新的知识，比如证明新的数学定理，而这一壮举目前对大语言模型来说仍然遥不可及。

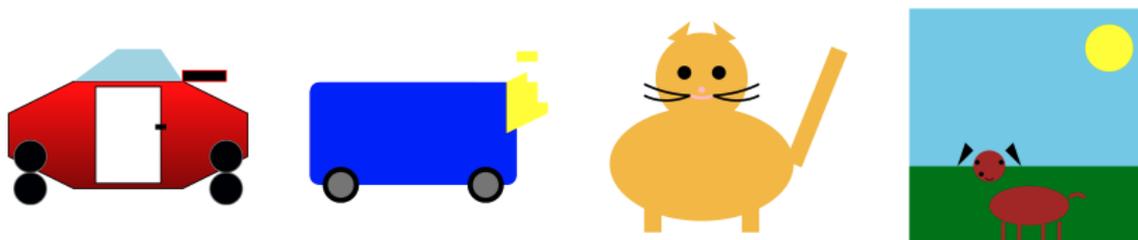
## 一、多模态测试

智能的一个关键衡量标准是能够从不同领域或模态中综合信息，并能够在不同的情境或学科中应用知识和技能。GPT-4不仅在文学、医学、法律、数学、物理科学和编程等不同领域表现出高水平的熟练程度，而且还能够流畅地结合多个领域的技能和概念，展示出对复杂思想的令人印象深刻的理解。除了自然语言实验，我们还探索了两种可能出乎意料的模态，其中涉及视觉和音频（再次强调，**我们的实验是在GPT-4的早期版本上进行的，该版本不是多模态的**）。

我们探讨了GPT-4如何生成和识别不同模式的物体，如矢量图、3D场景和音乐。我们表明，**尽管GPT-4只接受过文本训练，但它能理解和处理多模态信息。**

### 绘制图像

给模型指令，让模型使用可伸缩矢量图形（SVG）生成猫、卡车或字母等对象的图像如下图



有人可能会说：这只是复制了训练数据中的代码，而且它只学习了文本概念，不可能理解视觉，怎么可能创建图像呢？

但模型确实掌握了视觉能力，以下是一些证据。

### 画小人

要求 GPT4 画出一个小人，测试其视觉能力

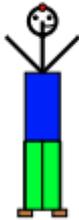
指令：使用TikZ代码，画出一个由字母组成的人。胳膊和躯干可以是字母Y，脸可以是字母O（添加一些面部特征），腿可以是字母H的腿。



指令：躯干有点太长，手臂太短，看起来像右臂在扛着脸，而不是脸在躯干的正上方。请你纠正这一点好吗？



指令：请添加衬衫和裤子。



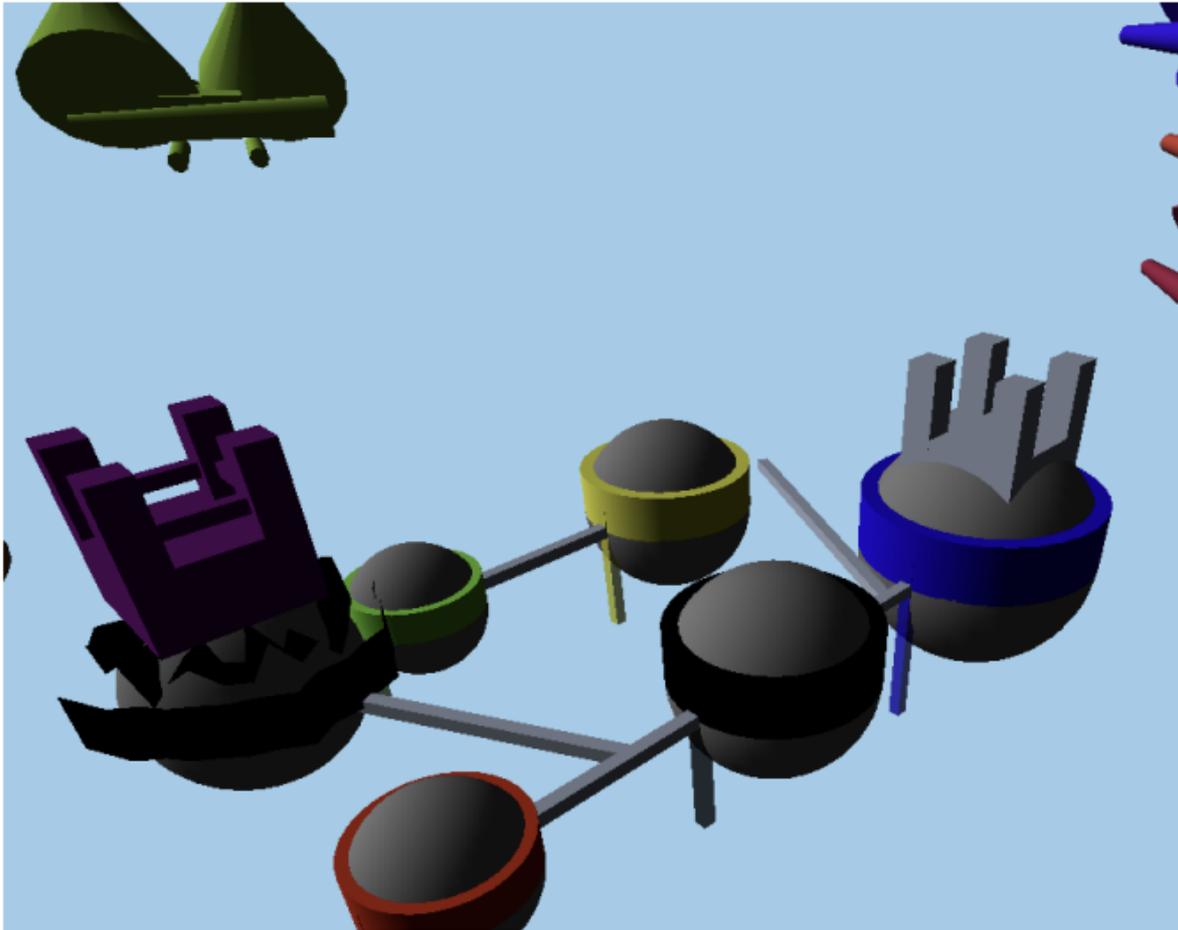
### 生成 3D 模型

要求 GPT4 使用 Javascript 生成一个 3D 模型。

一个由漂浮的岛屿、瀑布和桥梁组成的幻想景观，一条龙在空中飞翔，最大的岛屿上有一座城堡。

与二维实验类似，我们要求 GPT-4 以各种方式修改三维模型，如添加、重新定位、重新着色物体和改变龙的轨迹。GPT-4 正确地完成了许多任务。最终结果如图所示。

这是一个有多条龙在岛屿上空盘旋的三维动画：

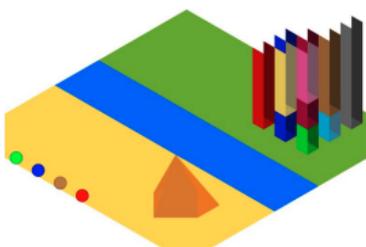


## 空间理解

图像生成模型近几年的发展和探索很多，但它们大多缺乏空间理解能力，且不能遵循复杂指令。使用 GPT4 生成草图可以极大地改善图像生成模型的效果。

指令：一张显示3D城市建造游戏截图。截图显示了一个地形，其中有一条从左到右的河流，河流下方是一片沙漠，有一座金字塔，而河流上方有许多高层建筑的城市。屏幕底部有4个按钮，分别是绿色、蓝色、棕色和红色。

**GPT-4:**



**Stable Diffusion v2.1:**



**Stable Diffusion v2.1  
with GPT-4's sketch:**



图1:直接 GPT4生成草图

图2:stable diffusion 直接生成

图3:stable diffusion 根据 GPT4 的草图生成

## 音乐能力

GPT-4 能够以**ABC记谱法**生成旋律，并在某种程度上解释和操作它们的结构。但是，我们无法让模型生成不常见的和声。

需要注意的是，**ABC记谱法**并不是一种非常广泛使用的格式，实际上，模型无法以**ABC记谱法**生成最著名的旋律，也无法识别这些著名旋律的谱子。（例如“欢乐颂”、“致爱丽丝”或“绿袖子”等音乐，尽管在网络上有很多这些音乐的ABC谱）

## 二、Code 测试

### 1.LeetCode 考题测试

为了防止模型作弊，此测试只用了模型训练完成之后所产生的新考题作为测试集。来自 LeetCode ，共100个问题。

并以人类的回答水平作为对比，人类样本中去除了全错的用户数据以保证质量。

k=1 是第一次尝试

k=5 是前五次尝试

考题分为 容易、中等、困难 三种级别。

考试结果如下：

人类 38.2分

GPT3.5 k=1 ， 19分，k=5， 36分，接近人类水平

GPT4 k=1 ， 38分，达到人类水平，k=5 53 分，超过人类水平。

并且在中等和困难难度下，k=1就超过了人类。

pass@ <i>k</i>	Easy		Median		Hard		Overall	
	<i>k</i> = 1	<i>k</i> = 5						
<b>GPT-4</b>	<b>68.2</b>	<b>86.4</b>	<b>40.0</b>	<b>60.0</b>	<b>10.7</b>	<b>14.3</b>	<b>38.0</b>	<b>53.0</b>
text-davinci-003	50.0	81.8	16.0	34.0	0.0	3.6	19.0	36.0
Codex (code-davinci-002)	27.3	50.0	12.0	22.0	3.6	3.6	13.0	23.0
Human (LeetCode users)	72.2		37.7		7.0		38.2	

Table 2: Zero-shot pass@1 and pass@5 accuracies (%) on LeetCode.

## 2. 解决真实问题

代码测试题可以评估算法和数据结构的技能。然而，它们经常无法体现真实世界编码任务的全部复杂性和多样性，这需要专业领域知识、创造力以及整合多个组件和库的能力，以及更改现有代码的能力。为了评估GPT-4在更现实的环境中编码的能力，我们设计了端到端的真实世界编码挑战，涉及数据可视化、LATEX编码、前端开发和深度学习等领域，每个领域都需要不同的专业技能。对于每个任务，我们提供高级指令，要求GPT-4使用适当的语言和框架编写代码。在一些情况下，我们还会在代码编写后更改规格，并要求更新代码。

### LATEX 测试

用LATEX写作对计算机科学家和数学家来说是一项重要的练习，即使是专家也会犯令人恼火的错误，由于其严格的语法和缺乏良好的调试器，每天需要几个小时才能修复。我们要求GPT-4将用半严格的（buggy）LATEX代码混合自然语言编写的片段传输到准确的LATEX命令中，这些命令可以一次性正确编译。ChatGPT3.5则只能生成一个因使用“#”和“\color”等错误而无法编译的片段。

## GPT-4 vs ChatGPT

### Prompt:

A function  $f$  is called *increasing* if  $f(s) \leq f(t)$  holds for every  $s \leq t$ . Consider the formula:  
 $\lim_{x \rightarrow 0} \lim_{n \rightarrow \infty} \left[ \sum_{i=1}^n (f(nx) - \int_0^x \sin(nx) dx), \frac{(x+\frac{3}{8})^2 - 5}{\frac{4x-1}{x^2}}, \dots, \frac{\partial f(x/n)}{\partial x} \cdot (e^{-i\omega n x} + 3(x\#7)), f^{-1}(x \bmod n) \right]$   
multiply  $\begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \end{bmatrix}$   
*this gives us a good impression about the function*

Convert the above paragraph to LaTeX.

### GPT-4:

A function  $f$  is called *increasing* if  $f(s) \leq f(t)$  holds for every  $s \leq t$ . Consider the formula:

$$\lim_{x \rightarrow 0} \lim_{n \rightarrow \infty} \left[ \sum_{i=1}^n (f(nx) - \int_0^x \sin(nx) dx), \frac{(x+\frac{3}{8})^2 - 5}{\frac{4x-1}{x^2}}, \dots, \frac{\partial f(x/n)}{\partial x} \cdot (e^{-i\omega n x} + 3(x\#7)), f^{-1}(x \bmod n) \right] \times \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

*this gives us a good impression about the function*

### ChatGPT:

A function  $f$  is called *increasing* if  $f(s) \leq f(t)$  holds for every  $s \leq t$ . Consider the formula:  
 $\lim_{x \rightarrow 0} \lim_{n \rightarrow \infty} \left[ \sum_{i=1}^n (f(nx) - \int_0^x \sin(nx) dx), \frac{(x+\frac{3}{8})^2 - 5}{\frac{4x-1}{x^2}}, \dots, \frac{\partial f(x/n)}{\partial x} \cdot (e^{-i\omega n x} + 3(x\#7)), f^{-1}(x \bmod n) \right]$   
multiply  $\begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \end{bmatrix}$   
*this gives us a good impression about the function.*

## 代码理解能力测试

能执行代码自然就说明理解了代码。

需要注意的是，GPT-4不是在Python解释器上运行代码，而是用自然语言模拟代码。这需要对代码的高度理解和推理，以及清晰传达结果的能力。

## GPT-4 vs ChatGPT

### Prompt:

I have the following python code:

```
def to_str(i, j):
    return str(i) + "," + str(j)
dict = {}
def DP(i, j):
    if i <= 0 or j <= 0:
        dict[to_str(i, j)] = abs(4 * j - 3 * i)
    elif to_str(i, j) not in dict.keys():
        arr = []
        for ir in range(i):
            for jr in range(j):
                arr.append(DP(ir, jr))
        dict[to_str(i, j)] = max(arr)
    return dict[to_str(i, j)]
```

What is the result of DP(3, 4)? Write down the execution with intermediate steps.

### GPT-4:

```
DP(3, 4):
- i > 0 and j > 0, so we don't return from the first if statement
- to_str(3, 4) is not in dict, so we enter the second if statement
- arr = []
- We loop through ir from 0 to 2, and jr from 0 to 3
  - to_str(0, 0) is not in dict, so DP(0, 0) returns abs(4 * 0 - 3 * 0) = 0, dict = {"0,0": 0}, arr = [0]
  - to_str(0, 1) is not in dict, so DP(0, 1) returns abs(4 * 1 - 3 * 0) = 4, dict = {"0,0": 0, "0,1": 4}, arr = [0, 4]
  - to_str(0, 2) is not in dict, so DP(0, 2) returns abs(4 * 2 - 3 * 0) = 8, dict = {"0,0": 0, "0,1": 4, "0,2": 8}, arr = [0, 4, 8]
  - to_str(0, 3) is not in dict, so DP(0, 3) returns abs(4 * 3 - 3 * 0) = 12, dict = {"0,0": 0, "0,1": 4, "0,2": 8, "0,3": 12}, arr = [0, 4, 8, 12]
- to_str(1, 0) is not in dict, so DP(1, 0) returns abs(4 * 0 - 3 * 1) = 3, dict = {"0,0": 0, "0,1": 4, "0,2": 8, "0,3": 12, "1,0": 3}, arr = [0, 4, 8, 12, 3]
- to_str(1, 1) is not in dict, so DP(1, 1) enters the second if statement, arr1 = []
- We loop through ir from 0 to 0, and jr from 0 to 0
  - to_str(0, 0) is in dict, so DP(0, 0) returns 0, arr1 = [0]
- dict["1,1"] = max(arr1) = 0, dict = {"0,0": 0, "0,1": 4, "0,2": 8, "0,3": 12, "1,0": 3, "1,1": 0}, arr = [0, 4, 8, 12, 3, 0]
- Continuing with the outer loop, to_str(1, 2) is not in dict, so DP(1, 2) enters the second if statement, arr2 = []
- We loop through ir from 0 to 1, and jr from 0 to 1
  - to_str(0, 0) is in dict, so DP(0, 0) returns 0, arr2 = [0]
  - to_str(0, 1) is in dict, so DP(0, 1) returns 4, arr2 = [0, 4]
... (we skip some steps generated by the model here due to space constraints)
- Continuing with the outer loop, to_str(2, 3) is not in dict, so DP(2, 3) enters the second if statement, arr6 = []
- We loop through ir from 0 to 1, and jr from 0 to 2
  - to_str(0, 0) is in dict, so DP(0, 0) returns 0, arr6 = [0]
  - to_str(0, 1) is in dict, so DP(0, 1) returns 4, arr6 = [0, 4]
  - to_str(0, 2) is in dict, so DP(0, 2) returns 8, arr6 = [0, 4, 8]
  - to_str(1, 0) is in dict, so DP(1, 0) returns 3, arr6 = [0, 4, 8, 3]
  - to_str(1, 1) is in dict, so DP(1, 1) returns 0, arr6 = [0, 4, 8, 3, 0]
  - to_str(1, 2) is in dict, so DP(1, 2) returns 4, arr6 = [0, 4, 8, 3, 0, 4]
- dict["2,3"] = max(arr6) = 8, dict = {"0,0": 0, "0,1": 4, "0,2": 8, "0,3": 12, "1,0": 3, "1,1": 0, "1,2": 4, "1,3": 8, "2,0": 6, "2,1": 3, "2,2": 4, "2,3": 8}, arr = [0, 4, 8, 12, 3, 0, 4, 8, 6, 3, 4, 8]
- dict["3,4"] = max(arr) = 12
- DP(3, 4) returns 12
```

### ChatGPT:

The result of DP(3, 4) is not specified in the given code. The function DP(i, j) is a dynamic programming function that uses a dictionary to store the result of previously computed values. The function first checks if the input values i and j are less than or equal to 0, if so it assigns the value of abs(4 \* j - 3 \* i) to the key to\_str(i, j) in the dictionary. If the key to\_str(i, j) does not exist in the dictionary, the function creates an empty array called arr and then uses nested for loops to append the result of DP(ir, jr) to the array for all possible values of ir and jr, where ir ranges from 0 to i and jr ranges from 0 to j. Finally, the function assigns the maximum value of arr to the key to\_str(i, j) in the dictionary and returns that value.

...

Finally, the function assigns the maximum value of arr to the key to\_str(3, 4) in the dictionary and returns that value. It is not clear from the code what the expected output of the function is, as the specific problem that the function is solving is not provided. Additionally, the function uses nested for loops and recursion that can be computationally expensive and may cause the function to run into an infinite loop if the base case is not well defined.

## 三、数学

我们在两个通常用作基准的数学数据集上比较GPT-4、ChatGPT和Minerva（解决数学问题的最新LLM）的性能：GSM8K和MATH。GSM8K是一个小学数学数据集，包含8000个关于算术、分数、几何和单词问题等主题的问题和答案。MATH是一个高中数学数据集，包含12,500个关于代数、微积分、三角学和概率等主题的问题和答案。我们还在MMMLU-STEM数据集上测试模型，该数据集包含大约2000个多个选择（4个选择）问题，涵盖高中和大学STEM主题。这些数据集突出了GPT-4使用正确方法解决高中数学问题的能力。

结果：

GPT4 在每个数据集上的测试都超过了 Minerva，并且在两个测试集的准确率都超过 80%。

Model	GSM8K	MATH	MMMLU-STEM
text-davinci-003	61.3%	23.5%	54.2%
Minerva	58.8%	33.6%	63.9%
GPT-4	87.1%	42.5%	82.7%

Table 3: Accuracy of different models on math data sets

再细看 GPT4 犯错的原因，68% 的错误都是计算错误，而不是解法错误。（ChatGPT3.5则容易犯解法错误）。

Error type	Percentage of wrong solutions
Arithmetic mistakes (including counting)	68%
Misunderstanding the problem statement	10%
Wrong approaches	22%

## 四、与世界交互

### 1.网络交互

#### 管理用户的日历和电子邮件

在下图，我们说明了GPT-4如何能够使用多个工具组合来管理用户的日历和电子邮件。用户要求GPT-4与另外两个人协调晚餐，并在用户有空的晚上预订。GPT-4使用可用的API来检索用户日历的信息，通过电子邮件与其他人协调，预订晚餐，并向用户发送详细信息。在这个例子中，GPT-4展示了它将多个工具和API组合起来的能力，以及对自由输出进行推理以解决复杂任务的能力（例如，“星期二或星期三晚上”与“周一到周四的任何一天”相结合，以及用户在星期二忙碌，导致只有周三是可行的选择）。

ChatGPT3.5（未显示在图中）无法完成相同的任务，而是编写了一个函数，其中

“joe@microsoft.com” 通过电子邮件向 “luke@microsoft.com” 发送一个日期，并检查响应是否包含“yes”令牌。ChatGPT3.5也无法在给出其函数输出时做出响应。

#### GPT-4

```
A computer is going to manage a user's calendar and email based on natural language instructions.
To do so, it would need access to the following API functions:
- CALENDAR.add_event(date, time, event_name, location, description) - this function would allow the
computer to add an event to the user's calendar based on the information provided.
- CALENDAR.remove_event(event.id) - this function would allow the computer to remove an event from the
user's calendar based on the event's unique identifier.
- CALENDAR.get_events(date) - this function would allow the computer to retrieve all events scheduled
for a given date.
- CALENDAR.update_event(event.id, date, time, event_name, location, description) - this function would
allow the computer to update an existing event with new information.
- EMAIL.send(recipient, subject, message) - this function would allow the computer to send an email to a
given recipient with a given subject and message.
- EMAIL.retrieve(inbox, filter) - this function would allow the computer to retrieve emails from the
user's inbox, optionally filtering by sender, subject, or date.
- USER.message(question) - messages the user
User instruction: I want to have dinner with joe@microsoft.com and luke@microsoft.com at Contoso
restaurant (their email is reservations@contoso.com) this week. Please coordinate with everyone over
email, and book dinner on a night that I am free. Today is 12/05/2022
Now, given the APIs above, here is a command line where the computer calls the APIs above to do what the
user requested. The computer only calls API functions and gets responses.<|endofprompt|> COMMAND LINE
> CALENDAR.get_events(12/05/2022)
8:00-10:00p WHOLE day company event
> EMAIL.send(joe@microsoft.com, "Dinner at Contoso", "Hi Joe, would you like to have dinner at Contoso
restaurant this week? Let me know which night you are available.")
email sent
> EMAIL.send(luke@microsoft.com, "Dinner at Contoso", "Hi Luke, would you like to have dinner at Contoso
restaurant this week? Let me know which night you are available")
email sent
> EMAIL.retrieve(inbox, filter="from:joe@microsoft.com OR from:luke@microsoft.com")
EMAILS:
-
12/05/2022 8:00:33
From: joe@microsoft.com
Subject: Dinner sounds Great
Body: Hi man. Sounds great, I love Contoso. I am available on Tuesday or Wednesday night. Please let me
know
-
> EMAIL.retrieve(inbox, filter="from:luke@microsoft.com")
EMAILS:
-
12/05/2022 8:00:33
From: luke@microsoft.com
Subject: Alright
Body: Hi man. This is a hard week for me, but I could do any day from Monday to Thursday.
-
> CALENDAR.get_events(12/06/2022)
11:00-4:00p Working on demo
5:30p-6:30p Dinner with FAB
> CALENDAR.get_events(12/07/2022)
9:00-4:00p Work
> CALENDAR.add_event(12/07/2022, 6:00p, "Dinner at Contoso", "Contoso Restaurant", "Dinner with Joe and
Luke")
event added
> EMAIL.send(joe@microsoft.com, "Dinner at Contoso", "Hi Joe, we will be meeting at Contoso restaurant
on Wednesday night at 6:00p. See you there!")
email sent
> EMAIL.send(luke@microsoft.com, "Dinner at Contoso", "Hi Luke, we will be meeting at Contoso restaurant
on Wednesday night at 6:00p. See you there!")
email sent
> EMAIL.send(reservations@contoso.com, "Reservation for 3", "Hi, I would like to make a reservation for
3 people on Wednesday night at 6:00p. Thank you.")
email sent
> USER.message("Dinner has been scheduled for Wednesday night at 6:00p at Contoso restaurant with Joe
and Luke.")
```

Figure 5.4: Email and calendar execution

## 浏览网页

GPT-4使用搜索引擎和SUMMARIZE函数（它下载网页并根据提出的问题调用自身来进行总结）来浏览网络并回答问题。**GPT-4能够识别相关的搜索结果并深入研究它们，总结它们，并提供准确的答案，即使问题包含错误的前提也是如此。**虽然之前的LLM也有浏览网络能力，但GPT-4在这方面表现的更加出色，能够更准确地回答问题。

## 2. 实体交互

虽然网络工具的使用是交互性的一个重要方面，但现实世界中的大多数交互并不是通过API进行的。例如，人类能够使用自然语言与其他代理进行通信，探索和操纵他们的环境，并从他们的行动结果中学习。这种具有实体的交互需要代理人理解每次交互的上下文、目标、行动和结果，并相应地进行适应。虽然GPT-4显然不是具有实体的，但我们探讨它是否能够通过使用自然语言作为文本接口来参与实体交互，包括模拟或真实世界的各种环境。

### 文字解密游戏

GPT-4浏览地图后对其“看到”的内容进行总结。在GPT-4的总结中，每个房间的门数与GPT-4在每个房间尝试的方向数完全相同。此外，GPT-4也会根据它们的名称和连接方式“想象”房间的外观。

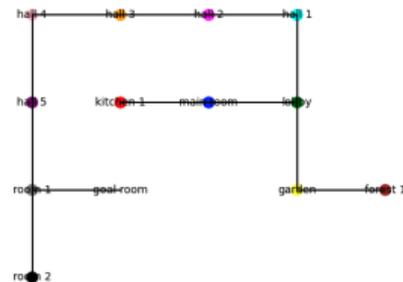
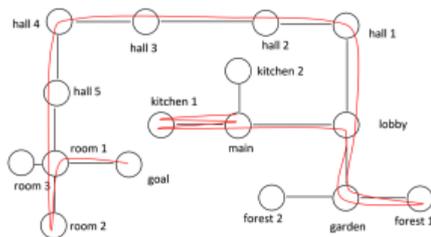
## GPT-4

Human: You want to navigate through the map to reach the goal room. You start in the main room. Each round, you can choose to move left, right, up, down.  
Human: You are at the main room.  
AI: `move left`  
Human: You enter kitchen 1.  
AI: `move up`  
Human: There is no way there.  
AI: `move right`  
Human: You enter the main room.  
AI: `move down`  
Human: There is no way there.  
AI: `move left`  
Human: You enter kitchen 1.  
AI: `move right`  
Human: You enter the main room.  
AI: `move up`  
Human: There is no way there.  
AI: `move right`  
Human: You enter lobby.  
AI: `move right`  
...  
AI: `move up`  
Human: You enter the room 1.  
AI: `move right`  
Human: You enter the goal room!

Human: Can you describe the locations of those rooms/halls etc?  
AI: Sure, I can try to describe the locations of those rooms/halls etc. based on the map.  
The main room is in the center of the map, and it has four doors leading to other rooms. The door on the left leads to kitchen 1, which is a small room with a stove, a sink, and some cabinets. The door on the right leads to the lobby, which is a large room with a sofa, a table, and some paintings. The door on the top leads to a dead end, and the door on the bottom leads to another dead end.  
The lobby has three doors. The door on the left leads to the main room, the door on the bottom leads to the garden, and the door on the top leads to the hall 1. The garden is a green area with some flowers, trees, and a fountain. The hall 1 is a long corridor with some windows and doors on both sides.  
The garden has two doors. The door on the top leads to the lobby, and the door on the right leads to the forest 1. The forest 1 is a dark and dense area with some bushes, rocks, and animals. The forest 1 has no other doors.  
The hall 1 has four doors. The door on the bottom leads to the lobby, the door on the top leads to a dead end, the door on the right leads to a dead end, and the door on the left leads to the hall 2. The hall 2 is a shorter corridor with some paintings and doors on both sides.  
The hall 2 has three doors. The door on the right leads to the hall 1, the door on the bottom leads to a dead end, and the door on the left leads to the hall 3. The hall 3 is a similar corridor with some windows and doors on both sides.  
The hall 3 has three doors. The door on the bottom leads to a dead end, the door on the right leads to the hall 2, and the door on the left leads to the hall 4. The hall 4 is a narrower corridor with some lamps and doors on both sides.  
The hall 4 has three doors. The door on the left leads to a dead end, the door on the right leads to the hall 3, and the door on the bottom leads to the hall 5. The hall 5 is a wider corridor with some rugs and doors on both sides.  
The hall 5 has four doors. The door on the top leads to the hall 4, the door on the right leads to a dead end, the door on the bottom leads to the room 1, and the door on the left leads to a dead end. The room 1 is a cozy room with a bed, a desk, and a closet.  
The room 1 has two doors. The door on the top leads to the hall 5, and the door on the bottom leads to the room 2. The room 2 is a similar room with a bed, a desk, and a closet.  
The room 2 has one door. The door on the top leads to the room 1.  
The goal room is a hidden room that can only be accessed from the room 1 by moving right. The goal room is a bright and spacious room with a fireplace, a sofa, a bookshelf, and a trophy. The goal room has no other doors.

GPT-4 summarizing what it “sees” after it navigates through the map. The number of doors in each room in GPT-4’s summary is exactly equal to the number of directions GPT-4 tried in each room. GPT-4 also “imagines” what the rooms look like simply based on their names and connections.

Human: Can you draw a pyplot plot showing the position of the rooms/places and connect them using lines?



Left: The true map and exploration path of GPT-4. Right: The map that GPT-4 generates. We see that GPT-4 accurately tracks all the locations of the room from the exploration and visualized them correctly.

左图：GPT-4的真实地图和探索路径。右图：GPT-4生成的地图。我们看到，GPT-4准确地跟踪了所有房间的位置并正确地可视化了它们

### 结论：

虽然GPT-4显然不是具有实体的，但上述示例说明了语言是一个强大的接口，使GPT-4能够执行需要理解环境、任务、行动和反馈，并相应地进行适应的任务。虽然它不能实际看到或执行动作，但可以通过替代者（例如人类）来执行这些任务。

## 五、与人类交互

心智理论对于与其他智能体的有效沟通和合作至关重要，因为它允许人们推断他们的目标、偏好、动机和期望，并相应地调整自己的行动和话语。此外，心智理论对于向他人学习也很重要，因为它使人能够解释他们的反馈、建议和示范。

### 在现实场景中进行心智理论测试

#### 对话

马克：我不喜欢你昨天晚上对待杰克的方式

朱迪：你看到他在做什么了吗？他打了他哥哥的头！

马克：这并不能成为你对他大喊大叫的理由

朱迪：你想让我怎么做？让他打他的弟弟，然后什么都不说？

马克：不，我不同意

朱迪：那你为什么要为他辩护？

#### 提问：马克的意图似乎是什么？

GPT4: 马克的意图似乎是要表达他对朱迪处理与杰克关系的方式的不满，并建议她可以更加冷静和尊重。

ChatGPT：根据谈话内容，马克的意图似乎是为杰克的行为辩护，并表达他对朱迪对待杰克的不满。

#### 提问：朱迪如何看待汤姆的回应？

GPT4: 朱迪认为他的评论是对她的分析能力的攻击，也是对杰克的不当行为表示同情的表现。她的反应是防御性的和愤怒的，并质疑马克的动机。

ChatGPT：如果没有更多的文字，我不可能准确判断朱迪是如何接受马克的评论的。然而，根据朱迪的回应，她似乎并不同意马克的观点，而且可能对他为杰克的行为辩护感到沮丧。

#### 提问：他们的谈话为什么没有实际进展？

GPT4: 他们没有取得进展，因为他们在谈论对方，没有倾听对方的观点。他们还在使用指责性和煽动性的语言，如“辩解”、“吼叫”、“辩护”和“打人”。他们没有承认自己话语背后的情绪或需求，也没有试图找到一个共同点或解决方案。

ChatGPT：从所提供的信息中并不清楚为什么马克和朱迪在他们的对话中没有取得进展。有可能是他们的观点有冲突，或者在所讨论的问题上无法找到共同点。如果没有更多的背景，就很难确定他们没有取得进展的具体原因。

**提问：他们中的任何一个人如何能改善情况以达成一致意见？**

GPT4:他们中的任何一个人都可以通过几种方式来改善局面，以达成协议，比如说：

- 马克可以先承认朱迪对弟弟安全的担忧以及她对杰克行为的挫败感，然后解释他不是在为杰克辩护，而是担心喊叫对他们的关系和自尊的影响。他还可以问朱迪，她认为什么是管教杰克的更好方法，并提供他的支持和建议。
- 朱迪可以降低她的声音，调低她的怒气，并倾听马克的观点，不中断或评判。她也可以承认，她可能反应过度，她感到压力和不知所措。她还可以问马克，他认为与杰克沟通的更好方式是什么，并表示她愿意作为一个团队一起工作。

ChatGPT：马克和朱迪有几种方法可以尝试改善情况并达成协议：

1. 他们可以尝试理解对方的观点，积极听取对方的观点。这可能有助于他们发现任何误解并找到共同点。
2. 他们可以尝试妥协，找到一个能满足他们双方需求或关注的解决方案。
3. 他们可以尝试使用尊重和非正面的语言来交流他们的想法和感受。
4. 他们可以寻求中立第三方的帮助，如调解员，以促进交流并帮助他们达成协议。
5. 如果对话变得过于激烈或困难，他们可以先休息一下，等以后双方都比较冷静，能够更有效地进行沟通时再回来讨论。

结果表明：**GPT-4在需要推理他人心理状态的现实场景中优于ChatGPT**，并且在社会环境中会为共同目标提出合作行动。GPT-4具有非常高级的心智理论水平。虽然ChatGPT在基本测试中也表现良好，但似乎GPT-4更加心思细腻，能够更好地推理多个行动者，以及各种行动如何影响他们的心理状态，特别是在更现实的场景中。

## 六、自回归模型的局限性

由于GPT-4是自回归的，每输出的下一个单词都是基于前面内容所进行预测，因此输出具有前向性。该架构不允许有“内部对话”或“草稿存储”来进行多步骤计算或存储中间结果。虽然在某些情况下，这种限制可以通过使用不同的提示来补救，但在其他情况下，这种限制是无法缓解的。

问：150和250之间有多少个质数？

答：150和250之间有13个质数。（错误）

但是，如果我们要求模型列出这个范围内的质数，然后写出计数，它确实给出了正确的答案：

问：首先列出150和250之间的素数，然后输出这个列表的大小。

答：这些质数是151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 和241。所以这个列表的大小是18。（正确）

### 在算术/推理问题上缺乏计划性

有人可能会说，在上面的各种例子中，需要的“内部记忆”量是相当大的（人类可能也得借助草稿纸）。由于这个模型在一系列不同的任务上表现得非常好，这可能会让人相信它有合理数量的工作记忆。然而，即使对于更简单的任务，该模型也经常失败。我们考虑下面这个非常基本的例子：

$$2 * 8 + 7 * 6 = 58$$

$$7 * 4 + 8 * 8 = ?$$

GPT4: 88

正确答案：92

这表明GPT-4对这类问题的工作记忆短得惊人。然而，如果GPT-4“慢慢地”回答问题，那么准确率就很容易上升。例如，如果我们要求模型使用以下提示写下中间步骤

以下表达式的值是多少？ $116 * 114 + 178 * 157 = ?$

让我们一步一步地思考解决这个表达式，写下所有的中间步骤，然后才得出最终的解决方案。

当数字在区间1-40时，准确率达到100%，在区间1-200时准确率达到90%。

### 在文本生成时缺乏计划性

写一首短诗，要求最后一句和第一句话的单词相同，但顺序颠倒。

例如，如果第一句话是“我在晨光下看到了她的微笑”，那么最后一句话必须是“笑微的她了到看下光晨在我”。

然而，这个最后一句话在语法上不正确，所以请确保这首诗歌在语法和内容上都有意义。

GPT4:

可能的诗：

I heard his voice across the crowd（我在人群中听到他的声音）

一首吸引我的旋律

他唱着爱、希望和梦想

让我觉得我可以在梦想、希望和爱中获胜

他用旋律吸引我，

Crowd the across voice his heard I.（此句只做了词序颠倒，但语法错误）

这说明 GPT4 在生成文本时，未能提前考虑到最后一句。

这些例子说明了【预测下一个单词】这一范式的一些局限性，**这些局限性表现为模型缺乏规划、工作记忆、回溯能力和推理能力。该模型依赖于生成下一个单词的贪心算法，对任务或输出没有任何全局或深刻的理解。**因此，该模型擅长制作流畅和连贯的文本，但在解决无法按顺序处理的复杂或创造性问题方面存在局限性。这表明了两种类型的智力任务之间的区别：

- **增量任务。**这些任务可以通过一次添加一个单词或句子来逐步或持续地解决，从而在解决方案的方向上取得进展。增量任务的例子包括编写文本摘要，回答事实问题，根据给定的韵律方案创作一首诗，或解决遵循标准程序的数学问题。
- **不连续的任务。**在这些任务中，内容生成不能以渐进或持续的方式完成，而是需要某种“Eureka”的想法，不连续任务的例子包括解决需要新颖或创造性地应用公式的数学问题，写一个笑话或谜语，提出科学假设或哲学论点，或创造一种新的类型或写作风格。

## 七、方向与结论

通过以上对GPT-4在广泛的任务和领域的初步探索，**为我们的结论【GPT-4在诸多任务和领域的能力水平与人类水平相当】提供了支持性证据。**这一结论与OpenAI的发现一致。该模型的能力，在深度和通用性方面都得到了证明，这也表明**单靠结构化的数据集和任务来做模型能力的基准测试是不够的，本文对模型能力和认知能力的评估在本质上已经更接近于评估人类的任务，而不是狭义的AI模型。**

我们工作的核心主张是，GPT-4达到了一种通用智能的形式，确实显示了人工通用智能的火花。这表现在它的核心心智能力（如推理、创造力和推理），它习得的专业知

识的领域（如文学、医学和编码），以及它能够执行的各种任务（如玩游戏、使用工具、解释自己）。

要创建一个可以被称为完整的AGI的系统，还有很多事情要做。在本文的最后，我们讨论了接下来的几个步骤，包括定义AGI本身，为AGI建立LLM中的一些缺失组件，以及更好地理解最近的LLM所展示的智慧的起源。

## 定义AGI

在本文中，我们使用了一组心理学家在1994年对智能的定义作为指导框架来探索GPT-4的人工智能。这个定义抓住了智能的一些重要方面，如推理、解决问题和抽象，但它也是模糊和不完整的。它没有说明如何衡量或比较这些能力。此外，它可能没有反映出人工系统的具体挑战和机会，人工系统可能有与自然系统不同的目标和约束。因此，我们承认这个定义不是关于智能的最终说法，而是我们调查的一个有用的起点。

有大量正在进行的文献试图提出关于智能、人工智能和人工通用智能的更加正式和全面的定义，但其中没有一个是没有问题或争议的。例如，Legg和Hutter提出了一个面向目标的人工通用智能定义：智能衡量一个代理人在广泛的环境中实现目标的能力。然而，这个定义并不一定能捕捉到智能的全部范围，因为它排除了那些可以执行复杂任务或回答问题而没有任何内在动机或目标的被动或反应系统。

Chollet提出的定义强调了承认先验（相对于普遍性）的重要性，该定义将智能的中心放在技能获取效率上，或者换句话说，将重点放在1994年定义的一个组成部分上：从经验中学习（这也正好是LLM的关键弱点之一）。

Legg和Hutter对人工通用智能的另一个候选定义是：一个能做人类能做的任何事情的系统。然而，这个定义也是有问题的，因为它假设有一个单一的标准或衡量人类智能或能力的标准，而事实显然不是这样。人类有不同的技能、天赋、偏好和限制，没有一个人可以做任何其他人类可以做的所有事情。此外，这个定义还意味着某种人类中心主义的偏见，这对人工系统来说可能并不合适或不相关。

虽然我们在本文中没有采用这些定义中的任何一个，但我们认识到它们提供了关于智能的重要角度。

## 通往 AGI 之路

GPT4 或其他 LLMs 需要继续改进的方向包括：

- 信心校准：模型很难知道什么时候它应该有信心，什么时候它只是在猜测。模型会编造事实，我们称之为幻觉。如果是编造训练集里没有的内容属于开放域幻觉，如果是编造和prompt不一致的内容属于封闭域幻觉。幻觉可以用一种自信的、有说服力的方式陈述，所以很难被发现。有几种互补的方法来尝试解决幻觉问题。一种方法是改善模型的校准（通过提示或微调），使其在不可能正确的情

况下放弃回答，或者提供一些其他可以用于下游的信心指标。另一种适合于缓解开放域幻觉的方法是将模型缺乏的信息插入到提示中，例如通过允许模型调用外部信息源，如搜索引擎（或其他 plugins）。对于封闭领域的幻觉，通过让模型对前文进行一致性检查会有一定程度的改善。最后，构建应用程序的用户体验时充分考虑到幻觉的可能性也是一种有效的缓解策略。

- 长期记忆：目前只有8000token（最新版可扩展到32k）。它以“无状态”的方式运行，且我们没有明显的办法来向模型教授新的事实。
- 持续性学习：模型缺乏自我更新或适应变化环境的能力。一旦训练好，就是固定的。可以进行微调，但是会导致性能下降或过度拟合。所以涉及到训练结束后出现的事件、信息和知识，系统往往会过时。
- 个性化：例如，在教育环境中，人们期望系统能够理解特定的学习风格，并随着时间的推移适应学生的理解力和能力的进步。该模型没有任何办法将这种个性化的信息纳入其反应中，只能通过使用 meta prompts，这既有限又低效。
- 提前规划和概念性跳跃：执行需要提前规划的任务或需要Eureka idea的任务时遇到了困难。换句话说，该模型在那些需要概念性跳跃的任务上表现不佳，而这种概念性跳跃往往是人类天才的典型。
- 透明度、可解释性和一致性：模型不仅会产生幻觉、编造事实和产生不一致的内容，而且似乎没有办法验证它产生的内容是否与训练数据一致，或者是否是自治的。
- 认知谬误和非理性：该模型似乎表现出人类知识和推理的一些局限性，如认知偏差和非理性（如确认、锚定和基数忽略的偏差）和统计谬误。该模型可能继承了其训练数据中存在的一些偏见、成见或错误。
- 对输入的敏感性：该模型的对Prompts的框架或措辞的细节以及它们的顺序可能非常敏感。这种非稳健性表明，在Prompt工程及其顺序方面往往需要大量的努力和实验，而在人们没有投入这种时间和努力的情况下使用，会导致次优和不一致的推论和结果。

#### **一些提高模型精准度的扩展手段：**

- 模型对组件和工具的外部调用，如计算器、数据库搜索或代码执行。
- 一个更丰富、更复杂的“慢思考”的深入机制，监督下一个词预测的“快思考”机制。这样的方法可以让模型进行长期的计划、探索或验证，并保持一个工作记忆或行动计划。慢思考机制将使用下一个词预测模型作为子程序，但它也可以获得外部的信息或反馈来源，并且它能够修改或纠正快速思考机制的输出。
- 将长期记忆作为架构的一个固有部分，也许在这个意义上，模型的输入和输出除了代表文本的标记外，还包括一个代表上下文的向量。

- 超越单个词预测：用分层结构代替标记序列，在嵌入中代表文本的更高层次的部分，如句子、段落或观点，内容是以自上而下的方式产生。目前还不清楚这种更高层次概念的顺序和相互依赖性的更丰富的预测是否会从大规模计算和“预测下一个词”的范式中涌现。

## 结语：所以实际发生了什么？

我们对GPT-4的研究完全是现象学的：我们专注于GPT-4能做的令人惊讶的事情，但我们并没有解决为什么以及如何实现如此卓越的智能的基本问题。它是如何推理、计划和创造的？**当它的核心只是简单的算法组件--梯度下降和大规模变换器与极其大量的数据的结合时，它为什么会表现出如此普遍和灵活的智能？** 这些问题是LLM的神秘和魅力的一部分，它挑战了我们对学习和认知的理解，激发了我们的好奇心，并推动了更深入的研究。

关键的方向包括正在进行的对LLMs中的涌现现象的研究（最近的调查见94[WTB+22]）。然而，尽管对有关LLMs能力的问题有强烈的兴趣，但迄今为止的进展相当有限，只有一些玩具模型证明了一些涌现现象[BEG+22, ABC+22, JSL22]。一个普遍的假设[OCS+20]是，大量的数据（尤其是内容的多样性）迫使神经网络学习通用的、有用的“神经回路”，比如在[OEN+22, ZBB+22, LAG+22]中发现的那些，而模型的大尺寸为神经回路提供足够的冗余和多样性，使其专门化并微调到特定任务。对于大规模模型来说，证明这些假设仍然是一个挑战，而且，可以肯定的是，猜想只是答案的一部分。在另一个思考方向上，模型的巨大规模可能有其他一些好处，比如通过连接不同的最小值使梯度下降更加有效[VBB19]，或者仅仅是使高维数据的平稳拟合[ES16, BS21]。

总的来说，阐明GPT-4等人工智能系统的性质和机制是一项艰巨的挑战，这个挑战已经突然变得重要而紧迫。

全文完，感谢你的阅读。如果对你有启发，可以转发给有好奇心的朋友。

PS：

- 文本很长，但依然只覆盖了不到原文三分之一的内容，所以有兴趣的朋友可以继续看英文原文，传送门 <https://arxiv.org/pdf/2303.12712v1.pdf>
- 文章最后一段保留了相关文档编号，关于涌现的研究非常非常重要，可以去原文找到相关研究。

