

关于General AI Agent & Manus的思考

写在前面

一·体验篇

1.1 测试案例描述

1.2 生成结果分析

1.3 执行过程观测

1.4 一些使用建议

二·反思篇

2.1 Manus的优点

2.2 Manus的缺点

2.3 对人的新考验

2.4 Manus的启迪

三·展望篇

3.1 架构趋于健全

3.2 范式兼容并包

3.3 核心服务与人

完结撒花

 **作者** Tianfu Wang | CS学生@USTC

 **最近** 在做些AI Agent的研究与应用

 **联系** tianfuwang.cs@gmail.com

深度试用了Manus近一周，似乎窥探到了General AI Agent进化成可用产品形态的曙光...

写在前面

我所认识的AI Agent

与AI相识是从研究强化学习开始的。那时，Agent的概念便以“四肢机器人”的形象刻入脑海。“与环境进行交互”是Agent与其他AI理念最大的区别。它不仅用大脑理解环境，还要做出决策并用四肢执行，完成既定目标。不同于追求高准确率的感知认知型AI（如预测和识别模型），Agent直面最终执行结果，目标达成与否、质量高低才是其评判指标。

早期，AI的大脑尚不发达，Agent主要专注于特定任务，表现却已令人瞩目。比如，2016年，以CNN做大脑的AlphaGo坐上围棋棋桌，自此柯洁也绕路走；而后Transformer出现，脑容量升级的Agent开车上路，在复杂路况中操控车辆实现无人驾驶。但现实世界远比棋盘复杂，决策行为也不局限于油门刹车，一个能够胜任各种任务的通用Agent依然停留在科幻层面。直到2022年，GPT3.5横空出世，它所展示的强大通用性前瞻着“General AI Brain”的到来。自此AI按下倍速键，LLM基础模型新突破，应用场景新探索，似乎已成每日常态。

与此同时，AGI话题不再被搁置，关于“N年内实现AGI”的断言遍历着20以内的整数（20甚至成了悲观的数字）。在通往AGI的道路上，General Agent被视为重要实现路径之一（更激进的，必由之路）。General意味着AI能够从容应对复杂、动态环境，一方面需要聪颖的大脑深度理解世界，另一个方面依赖精壮的四肢高效执行各种决策。装配大模型作大脑，各类工具作四肢，General Agent可以在各种复杂环境下，主动与之交互、自主实现目标。

我与AI Agent的经历

驯服过古早RL-based Agent，搭建过LLM-based Agent，幻想过General AI Agent。

轻喷声明：技术向Bias拉满且个人水平有限

目录

- 体验篇：测试案例、输出分析、过程观测、使用建议
- 反思篇：Manus的优劣分析与启迪、对人的考验
- 展望篇：基础架构演变、范式产品形态、以人为本

一 · 体验篇

1.1 测试案例描述

下面是我使用Manus时几个典型cases，prompt中文精简版及其他信息如下：

注：我的人工工作流并不一定高效，时间估测也呈极度乐观态度

Case 1: 课题调研报告（对知识的梳理和思考能力）

输入Prompt

- 任务描述：请帮我调研LLM for Education的最新进展
- 预期输出：一份是近一周arXiv论文的总结和思考；另一份是汇报给大BOSS的Research proposal slide
- 额外要求：应当插入必要图片，UI应当专业美观，能Convince大老板
- 上传附件：无

目标的主要评估纬度

- 内容的真实性
- 思考的深入度
- 设计的美观性

我+LLM可能的工作流：

关于arXiv论文总结汇报

1. 逛arXiv选择性阅读并下载有意思的papers（小方向paper不多，~1h）
2. 丢给DS/GPT总结并让其规划slide，然后自我思考完善（~2h，GPT I/O很高，block的是我的脑子：）
3. 去AI PPT工作网站生成一个Slide初稿
4. 开启纯人工模式

关于Research proposal slide

1. ~~DeepResearch下（它也是Agent，作弊不算，重来~~
2. LLM + WebSearch初步了解下领域发展脉络（来回问答，~30min
3. 看看survey paper，刷刷新paper，keyword+search循环迭代（P人属性爆发，未知时间后循环停止
4. 写了些Note，收集了些材料，自我思考+LLM问答（未知时间后流程停止
5. 等待DDL临近了
6. 开启纯人工模式

Case 2: 项目文档制作（对项目级代码的理解能力）

输入Prompt

- 任务描述：请为我的开源项目完善文档网站
- 预期输出：一个项目doc网站
- 额外要求：UI应当专业美观
- 上传附件：一个项目简要介绍PDF

目标的主要评估纬度

- 结构的完整性
- 内容的专业性
- 设计的美观性

我+LLM可能的工作流

1. 浏览代码架构（~2h）
2. 规划文档结构（~2h）
3. 具体到每个文档，代码文件+LLM多轮交互完善文档（X days）
4. 开启纯人工模式

Case 3: 创业商业计划（对长远事项的规划和执行能力）

输入Prompt

任务描述：请帮我构想一个AI教育的创业项目，实现demo，并准备商业计划slide

预期输出：创业想法、产品demo、商业计划

额外要求：UI&UX应当专业美观、能拿到投资人的钱

上传附件：无

目标的主要评估纬度

- 创意的合理性
- 内容的专业性
- 设计的美观性

Team+LLM可能的工作流

1. 市场调研与确定构思（n1 weeks）
2. 原型设计与Demo开发（n2 weeks）
3. 制作商业计划与寻找投资人（n3 weeks）
4. 持续完善产品直到获得投资（n4 weeks）
5. 返回上面任意步骤或完成后停止

1.2 生成结果分析

和Agent理念一样，无关过程细节，我们先直面结果，看看是否能够达到预期。

结果 1: 课题调研报告

流程回放: <https://manus.im/share/myl8jKijCIWOgjsatijYj?replay=1>

Manus首先调研LLM for Education方向, 然后总结思考, 最后成功制作了两份PPT: 一份最近一周的论文汇总, 另一份是汇报给大老板的Research Proposal。

从结构上来看, 两个PPT都展现了很强的逻辑性和学术性。在处理类似任务时, 人们也有可能会用相同的逻辑来制作PPT。总体来看, Manus在PPT逻辑规划上的表现还是不错的。然而, 内容分页安排上有些不协调, 某些页面的内容较少, 而个别页面则有近两页的内容。

Recent Research Advances on LLMs for Education

Agenda

1. Introduction
2. Overview of Recent Research
3. Key Papers Reviewed
 - Student Engagement with AI Agents
 - Affordably Fine-tuned LLMs
 - Bridging the AI Adoption Gap
 - AI-Driven Assessment for LLM Prompting Skills
 - AI Literacy in K-12 and Higher Education
4. Synthesis and Common Themes
5. Implications for Educational Practice

关于内容方面, 第一份一周ArXiv论文的总结中, Manus仅找到了六篇论文, 而实际上当周有数十篇。这主要是因为Manus没有直接在ArXiv官网上进行搜索, 而是使用了谷歌, 从而导致搜索到的内容并不全。具体到每个Slide页面, 对论文的总结是清晰的, 但更多地采用了“Findings”到“Implications”的结构, 没有集中介绍论文的方法和动机, 细节上还有待提升。当然, 也可能Manus是PPT高手, 他知道每页不该放那么多字。

Paper 2: Affordably Fine-tuned LLMs

Key Findings

- **Fine-tuning Improves Performance:** Smaller models (7B parameters) when fine-tuned outperformed larger generic models (70B parameters)
- **Quantization Maintains Performance:** Reduced computational resources with minimal impact on accuracy
- **Resource Efficiency:** Effective fine-tuning accomplished with modest computational resources
- **Content Correlation:** Strong correlation between fine-tuning material and question content accuracy

Paper 2: Implications

Smaller, fine-tuned models can outperform larger generic ones, making advanced AI capabilities more accessible to resource-constrained educational institutions

- Challenges the "bigger is better" assumption in language models
- Provides a practical roadmap for educational institutions with limited resources
- Suggests leveraging existing educational materials (e.g., textbooks) for fine-tuning
- Democratizes access to AI-enhanced education, potentially reducing disparities

至于第二份Proposal，Manus生成的内容看起来相当唬人。它将各种AI前沿技术和教育领域的融合在一起，看似是一个非常有前景的方向。但里面的参考文献依然存在幻觉问题，大部分文献并不存在。非常神奇的是，它在PPT中竟然专门有一页介绍教育学理论，并且在后续的“AI for 教育”技术方向中涉及到了这些理论。知道先找理论来讲故事，这点看来Manus还是一个不错的水论文高手，这一点还是蛮有意思的。

Research Proposal: Advancing LLMs for Education

Advanced Personalization Algorithms

Knowledge Tracing with Transformer Architecture

We propose a novel approach to knowledge tracing that leverages transformer architectures to model complex dependencies between knowledge components:

$$P(c_{i+1} = 1 \mid c_{1:i}, q_{1:i+1}) = \sigma(W_o \cdot \text{Transformer}(E_k(c_{1:i}), E_q(q_{1:i+1})))$$

Probability of correctly answering question q_{i+1} given history of responses $c_{1:i}$ and questions $q_{1:i+1}$

Current Research Status

Traditional knowledge tracing models like Bayesian Knowledge Tracing and Deep Knowledge Tracing have shown promise but struggle with complex interdependencies between knowledge components. Our transformer-based approach addresses this limitation by capturing long-range dependencies and contextual relationships.

Adaptive Content Selection Algorithm

```
# Pseudocode for Adaptive Content Selection
def select_optimal_content(student_model, content_library, learning_objective):
    # Calculate knowledge gaps
    knowledge_gaps = student_model.identify_gaps(learning_objective)
    # Calculate optimal challenge level (Vygotsky's ZPD)
    optimal_difficulty = student_model.current_mastery + student_model.growth_factor
    # Find content that addresses gaps at optimal difficulty
    candidates = content_library.filter(
        knowledge_components=knowledge_gaps,
        difficulty=optimal_difficulty,
        modality_preference=student_model.preferred_modalities
    )
    # Select most relevant content using multi-armed bandit approach
    return thompson_sampling(candidates, student_model.history)
```

Evaluation Metrics

- **Personalization Efficacy Index (PEI)**: Measures alignment between content and learner needs
- **Learning Efficiency Ratio (LER)**: Time to mastery compared to personalized approaches

Research Proposal: Advancing LLMs for Education

Research Focus: Multimodal Learning with LLMs

Multimodal Large Language Models (MLLMs) offer transformative potential for education by processing and generating content across multiple modalities, creating richer and more effective learning experiences.

Key Research Directions

- **Cross-Modal Translation**: Converting between textual and visual representations
- **Multimodal Integration**: Combining information from multiple sources
- **Eye-Tracking Integration**: Using gaze data to enhance learning analytics
- **Multimodal Assessment**: Evaluating learning across different modalities

Current Research Status

Recent studies by Liu et al. (2024) and Zhang et al. (2024) have demonstrated promising results in using MLLMs for educational content generation, but significant challenges remain in ensuring cross-modal coherence and pedagogical alignment. Our research will build on these foundations while addressing critical limitations.

Educational Applications

- Converting textual descriptions to visual diagrams in science education
- Generating explanations for complex visual concepts
- Creating multimodal learning materials from single-modality inputs
- Analyzing student work across multiple formats
- Providing feedback that integrates visual and textual elements

Case Study: Science Education

In preliminary studies, we found that students using multimodal learning materials generated by our prototype system showed a 27% improvement in conceptual understanding compared to those using traditional materials, with particularly strong gains among visual learners and students

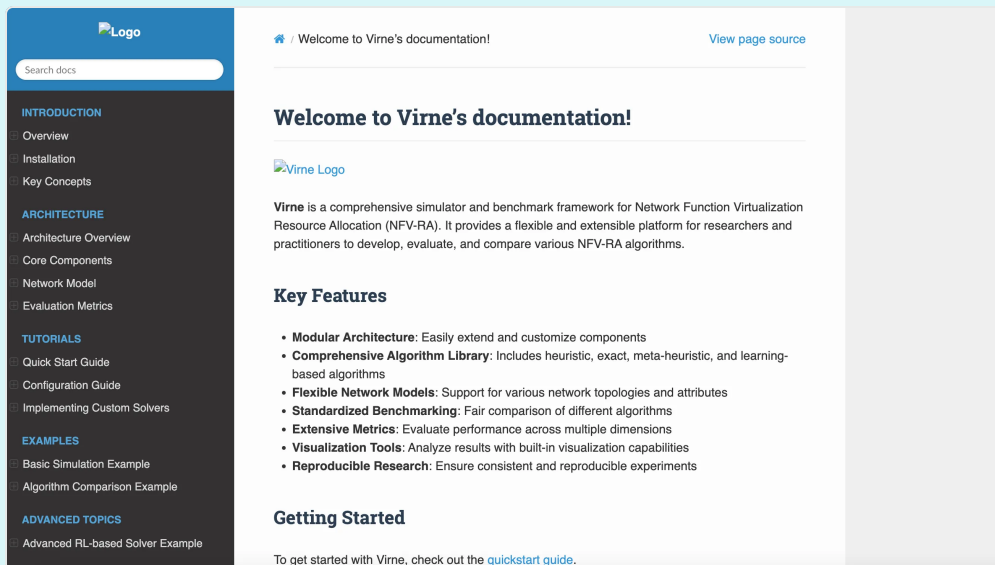
从设计角度来看，这两个PPT的质量都蛮高的。它们的结构和排版非常清晰，色彩搭配也很舒服。网页版的PPT也有些动态效果，其质量应该超过了那些缺乏PPT制作经验或只会套用模板人的作品。

结果 2：项目文档制作

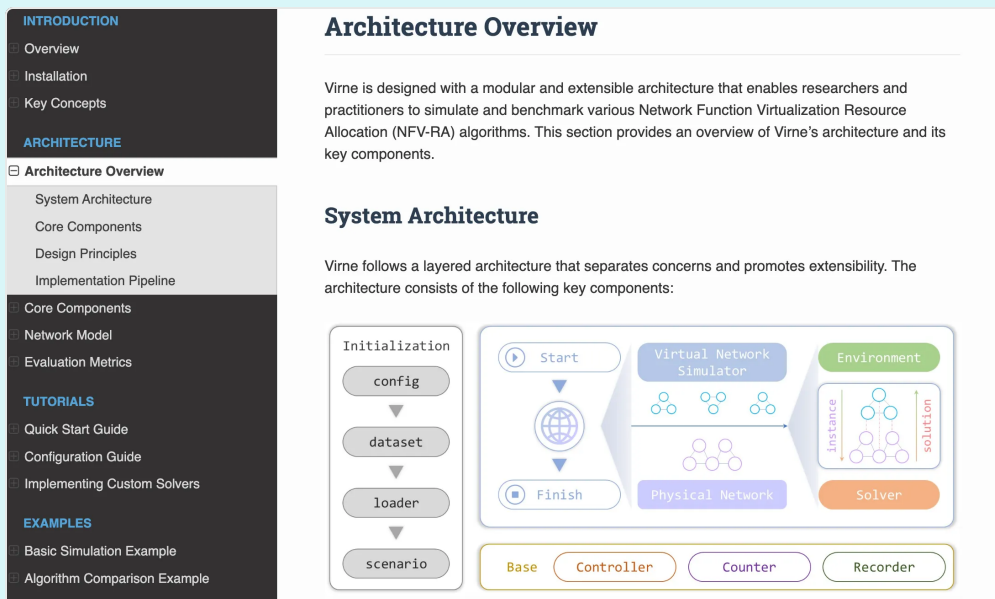
流程回放：<https://manus.im/share/e0fggxvS9O1swK8pQSGGko?replay=1>

Manus首先读取GitHub链接熟悉项目，并阅读我上传的项目简介，然后规划文档结构，笔耕不辍撰写每一个页面，最后生成了一个8MB的HTML压缩包。

从结构上看，Manus生成的文档非常有条理。章节划分很清晰，章节内的小节之间逻辑也十分严密。这需要对代码结构有全局的理解（这个项目应该是有数万行）。从文档规划的质量来看，个人觉得它至少超过了我自己，已经达到了中级工程师的以上水准。



从内容上看，每个章节的信息都是相关的。PDF文档中的公式被正常渲染并显示，内容简明扼要，而且还超出了文档内容，自动补充了一些代码和可能的概念等。Readme.md中部分图片链接正常显示，但出现了幻觉现象导致渲染失败，比如项目Logo。总体来说，内容的可读性和质量是很高。不过，文中的一些Manus自己生成很多代码并不符合原意，这可能是因为我写的代码太烂了，导致它曲解了函数意思：（



- Architecture Overview
- Core Components
- Network Model
 - Basic Network Model
 - Extended Network Model
 - Network Topologies
- Evaluation Metrics
- TUTORIALS
 - Quick Start Guide
 - Configuration Guide
 - Implementing Custom Solvers
- EXAMPLES
 - Basic Simulation Example
 - Algorithm Comparison Example
- ADVANCED TOPICS
 - Advanced RL-based Solver Example
 - Custom Network Topologies

Basic Network Model

In Virne, networks are modeled as graphs with nodes and links representing computing resources and communication channels, respectively.

System Model

In NFV, user network services are virtualized as virtual networks (VNs) and physical infrastructure is represented as a physical network (PN). The system model in Virne consists of:

- **Physical Network (PN):** Represented as a graph $G_P = (N_P, L_P)$, where:
 - N_P is the set of physical nodes (servers)
 - L_P is the set of physical links (connections)
 - Each physical node $n_p \in N_P$ has computing resources $C(n_p)$
 - Each physical link $l_p \in L_P$ has bandwidth resources $B(l_p)$
- **Virtual Network (VN):** Represented as a graph $G_V = (N_V, L_V)$, where:
 - N_V is the set of virtual nodes (services)
 - L_V is the set of virtual links (relationships)
 - Each virtual node $n_v \in N_V$ demands computing resources $C(n_v)$
 - Each virtual link $l_v \in L_V$ demands bandwidth resources $B(l_v)$

- Basic Simulation Example
 - Setting Up the Simulation
 - Visualizing the Results
 - Running the Example
 - Complete Example Code
 - Expected Output
 - Next Steps
- Algorithm Comparison Example
- ADVANCED TOPICS
 - Advanced RL-based Solver Example
 - Custom Network Topologies
 - QoS-Aware Resource Allocation
 - Performance Benchmarking and Analysis
 - Extending Virne with Custom Components
- API REFERENCE
 - Base Module
 - Solver Module
 - Data Module
 - Utils Module

walk through the process step by step, explaining each component and its purpose.

Setting Up the Simulation

First, let's import the necessary modules:

```
from virne.base import BasicScenario
from virne import Config, REGISTRY, Generator
import matplotlib.pyplot as plt
import numpy as np
```

Next, we'll define a function to run our simulation:

```
def run_basic_simulation():
    """
    Run a basic NFV-RA simulation using the Node Ranking Method (NRM) solver.
    """
    print(f"{'-' * 20} Basic Simulation Example {'-' * 20}\n")

    # Step 1: Create configuration
    config = Config(
        solver_name='nrm_rank', # Using Node Ranking Method
        p_net_setting={
            'topology': 'waxman',
            'num_nodes': 50,
            'wm_alpha': 0.5,
            'wm_beta': 0.2,
            'node_attrs': {
                'computing_resource': {
                    'distribution': 'uniform',
                    'low': 50,
```

类似于Case2，我也用Manus尝试生成论文级的PDF。最终它提供了LaTeX代码，渲染出了100多页。论文结构清晰，内容质量高，甚至无错地补充了很多附件里没有出现的解释内容。同样的质量，如果我和GPT拼命肝，可能也要花上十天半个月；而它，仅10分钟（沉默...

结果 3：创业商业计划

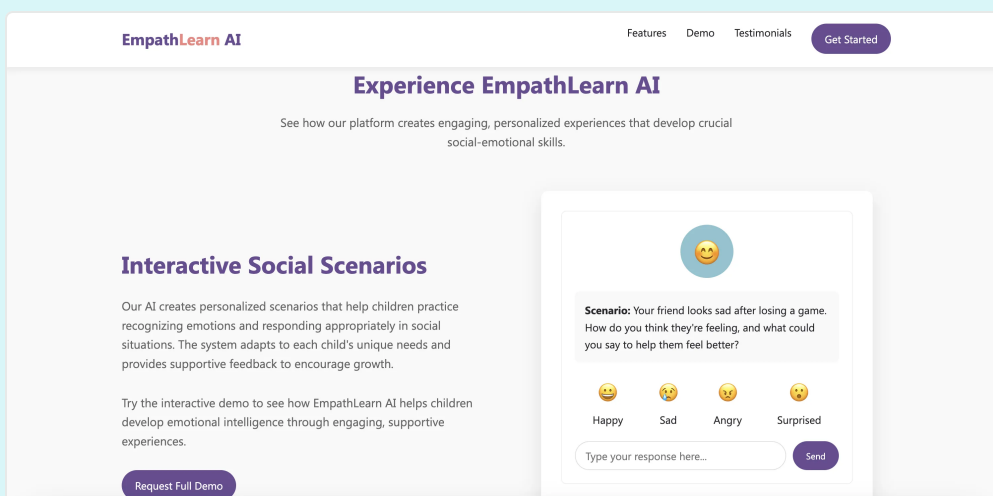
流程回放：<https://manus.im/share/K2bVNSa5mjaqSBMiOD3fYM?replay=1>

这个过程总共执行了三次。前两次，Manus生成的产品市场上已经存在，并且名称相同。在执行第三次时，终于找不到同名产品了。第三次的结果让我很震惊。它竟然融合了我最近正在研究的两个方向。要知道，我从未告诉过它，我在研究LLM情商相关的方向...抑或是这真的是未来。

从结构上来说，整个商业计划书PPT包含了项目概要、市场分析、产品技术、商业模式、竞品分析、团队介绍、财务和风险等内容，还是比较全面的。



从内容上来说，Manus有点儿过于追求大而全。产品中列出了许多技术点，目标用户既包括儿童，也包括父母。不过，如果能聚焦于某些方面，还是很有启发的。此外，由于没有提供关于团队、财务的信息，数据有很强的幻觉效果。



关于设计感，网页整体而言还是比较美观的，但内容不够丰富，很多按钮都不能点击。它只是一个单页应用，离商用水平还差得远，质量明显低于 Cursor。

总结

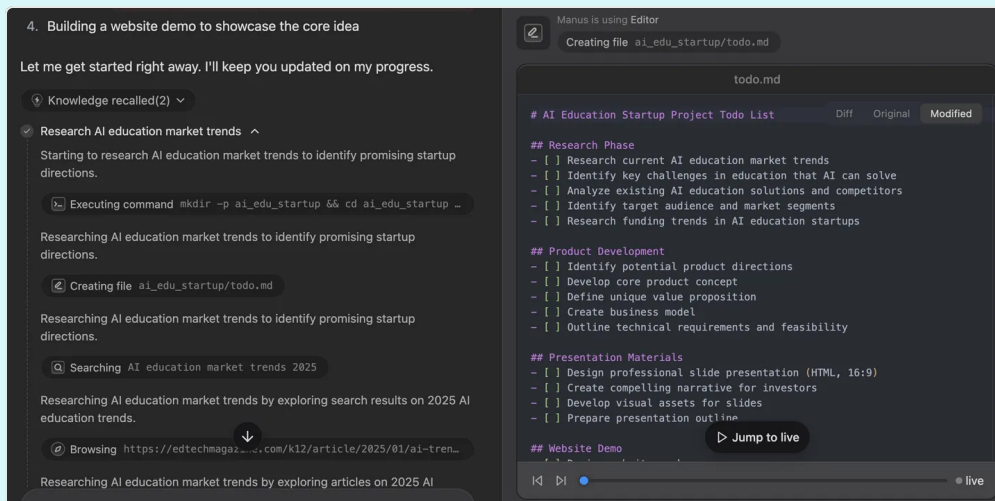
对于结果的评估受预期管理的影响，任务开始我并不对Manus抱很高期望。因为有些任务我可能需要花一天来完成，有些则过于复杂我评估时间都很难。然而，不论个人认为项目多复杂，Manus总能在半小时内给出结果。而且内测居然还是免费的，要啥自行车？人类世界，月薪三千能干到极致，大概率是CEO，现阶段的Manus仍是牛马。

给定相同的Prompt，Manus的效果应该是吊打Proxy的（convergence.ai）。Proxy经常中途退出，输出一段纯文本，给到预期的结果是少数的。这应该是因为Proxy接入的工具更加有限，且规划和工具使用能力欠缺，做不到干脆睡大觉。同时，Manus输出的质量是高于OpenManus的，它通常能plan出更好的实现路径和更高效的选择使用工具。这说明Manus对Prompt还是精心打磨的，也额外了赠送优秀的UI&UX体验。

1.3 执行过程观测

有趣的Planning & Memory实现设计

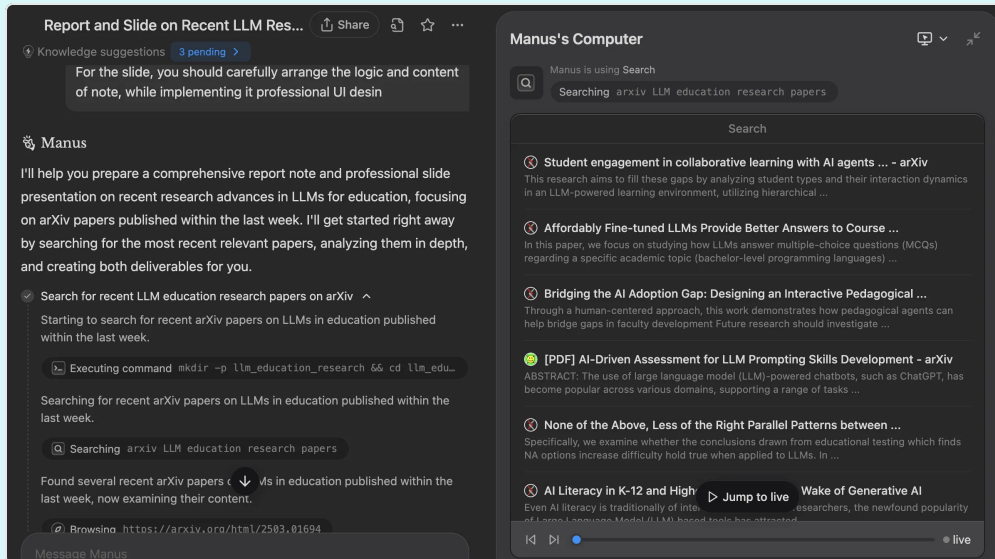
当第一次输入任务prompt后，Manus会规划任务执行路径，并写入 `todo.md` 文件。（Planning是Agent核心能力之一，见Lilian Weng's Blog）。通过planning，Agent解耦复杂任务成子目标，并以望远镜的视角规划出到达最终目标的路径，并可以在执行过程中反思重试，更新计划。Manus应该是采用一种层次化的Planning实现。给定需求，它首先理解用户意图进而生成Long-term Planning（更高层级的步骤）。有意思的是，不同于直接放入prompt作short-term memory，Manus将plan存入文件作long-term memory，以todo-list的方式动态更新完成进度。这样的设计或许可以让Agent更加聚焦于当前子目标，有利于任务执行的稳定性。针对todo-list里的每一个任务，Manus会生成更细粒度的Plan，直接作Action与Sandbox交互。



不过，`todo.md` 文件里面子任务似乎是固定的，而且只能生成一次。初始时一旦生成，执行过程中，这个文件规划的路径并不会动态调整，只会更新是否完成的状态。另外，在同一个chat里，继续追问，即使任务很复杂，Manus并不会再次创建新的 `todo.md` 文件。

和人类执行流程并不完全一致（效率或低或高）

观看Manus执行过程，它的大部分行为都高度的类人，但仍存在一些不一致。多数情况下，我认为它是在以我所能想到高效的工作流程进行的，但也有例外。比如，当指定总结最近一周arXiv论文时，有经验的研究员大都会直接进入arXiv网站进行搜索。但Manus似乎并不首选进入这类专有网站，他直觉性地依赖Google Search工具，在搜索结果中判断是否是arXiv网站、时间是否在一周内。这也最终导致生成的结果是不符合预期的，可能是Manus灵活使用工具的能力还不足。



遇到问题会主动询问人的帮助

在执行过程中，如果遇到难以解决的问题或者重要的抉择，Manus会主动寻求人的帮助和选择。不懂就问不装懂，能够能好的理解用户的意图。另外，像下属汇报工作一样，AB方案式的选择，对人的心理体验也很加分。即使流程全自动化是理想的，但人也怕少了对过程的掌控感。

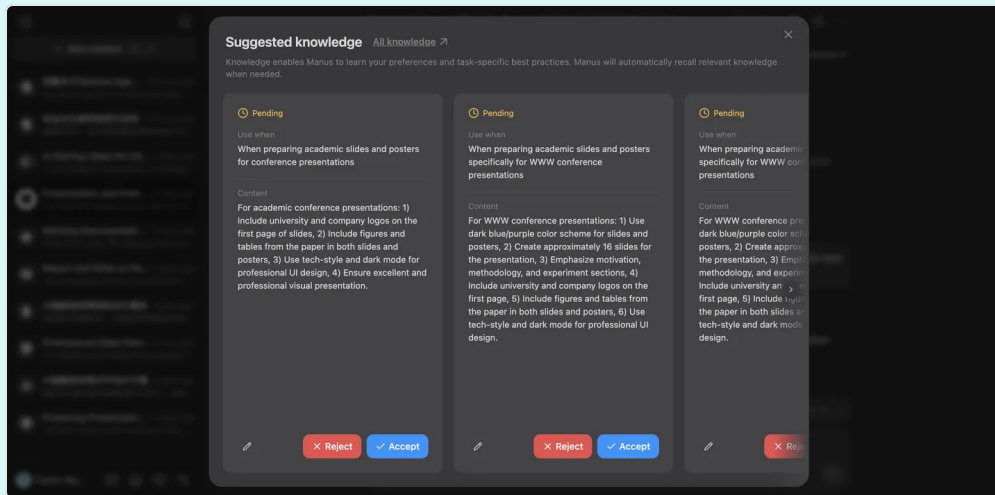
Manus

I'll help you prepare the presentation slides and poster for your WWW conference paper on LLM for Education. I'll create professional, tech-style designs in dark mode with university and company logos, and include the figures and tables from your paper.

To get started, I'll need some information about your paper and the presentation requirements. Let me set up a working environment first.

随时打小抄记录用户偏好作为知识库

在执行过程中，Manus像是位颇有灵性的秘书，他时刻思考着你对执行流程和输出要求的偏好。运行过程中，它主动跑过来问“你是这么想的么”，如果认可，他会巧妙地记下“当X情况下，应该知道Y”（将其加入Suggestion Knowledge，一个User Memory）。可想而知，随着产品使用时间的增加，Manus会更加熟悉用户的偏好，个性化也随之提升，甚至可能比本人还理解自己的需求，越用越好用。（ChatGPT其实也会记录用户偏好到Memory，但总是偷偷摸摸的，这缺少了与用户对齐的环节，记下来的可能是错误的。）

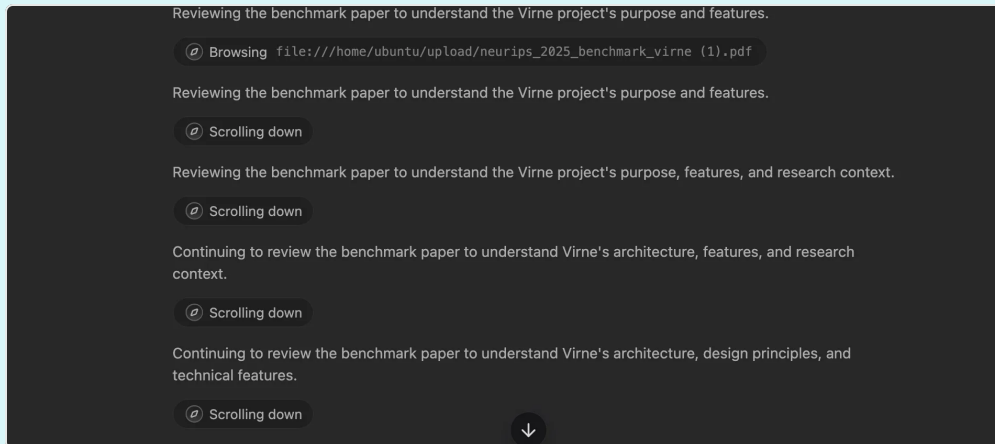


面对网络营销也和人类一样“清澈”

Case 1中，Manus输出的文件图片链接是虚拟机本地的，于是告诉它应该将用公开图片链接。它很聪明的想找免费的图床网站。但很遗憾，Google搜到的top results经常是打着免费的幌子，要求注册或者施舍些极少的免费额度。重试了多次仍未成功，至少它应该还不是一个足够老练的白嫖专家。

真的在用眼睛阅读PDF文件

将上传的PDF抽取出文本图片作LLM的输入，是ChatBot的通用做法。但Manus不吃嗟来之食，似乎更相信自己的眼睛，看完一页翻下一页。利用多模态能力或许可以更好的定位图片与文本之间的关系，但属实低效。



Agent时而也会摸鱼

有时候会发现它卡在一个步骤很久，不清楚它努力尝试还是在摆烂，也可能是在服务器load过重的问题。这时候通常需要输入"Continue"监工push下。如果资源充沛的情况下，仍然存在这个问题，可能也需要让Agent适时学会“放弃”。

1.4 一些使用建议

1. 需要对任务有清晰的要求，往往会得到更符合预期的结果
2. 过程中，如果它没有按照预期的方式执行，可以随时打断更正其行为（目前基座模型意图理解能力有限，人的任务描述也不一定清晰。这个机制很有意思，随时更正，避免Manus打白工）
3. 如果任务开始很久后，仍没有收到完成提示，建议查看下Manus，它可能需要你的帮助或决定。
4. Manus多轮对话有上限，当轮数过多时，会提示context过长，需要重新开启新对话。
5. 如果预期输出是Slide的话，相交于.ppt格式，通常指定HTML格式可以得到UI & UX设计感更强的结果（原因可能是，相较于使用工具，Manus所用的基座模型——Claude更擅长Coding）
6. 衔接新对话的方式并不丝滑，如果有“一键新开”功能就好了（比如，自动总结当前对话重点和将结果文件作附件，当作新对话的输入）

二 · 反思篇

暂且以Manus指代General AI Agent初阶形态。最初论文审稿的时候，师兄教导我说，优缺点至少各三个，正分可以多写个优点，负分多写个缺点。在这里不成立。

2.1 Manus的优点

弥补大模型从知道到做到之间的差距

基座大模型在各类Benchmark SOTA日日刷新，但他们始终被桎梏双手，停留在问答层面。Agent则赋予LLM四肢，使其可以理解环境并与之进行交互，自动化目标实现过程。输出不再局限于文本，而可能是多媒体文件、物联网设备的开关。于人而言，我们可以从步骤级别的多轮问答解脱出来，也不需整理中间结果，一键直达目标。LLM一定程度上取代了人的脑力劳动，Agent则进一步的省略了人的繁琐操作。

Agent没有完美主义，执行力拉满

于人而言，做事最难的是开始。缺乏清晰度和行动力，面对复杂事项，人可能永远都不会去做。但Agent是行动的巨人。给定需求，它快速思考规划布局，最后生成一个从零到一的毛坯房。无论质量好坏，人都能快速获得初始解，增加清晰度，进而思考下一步的优化方向。

展示了General AI Agent产品形态的可能性

Computer use/ GUI Agent的概念对于AI研究者来说并不陌生，是近期的研究热点，但在Manus之前它只活在实验室和开源社区。Manus首次把它们从论文里拽了出来，以友好的交互摆在普通用户面前，它展示了一种大众化General AI Agent产品形态的可能性。输入任务，直达结果，且中间流程可视化，Sandbox可灵活访问。如ChatGPT定义着LLM ChatBot的产品理念一样（几乎其他所有LLM厂商的ChatBot都高度同质化），或许Manus也有机会引领LLM时代Agent的产品形态。

2.2 Manus的缺点

基础模型的局限性依然存在

认知水平决定行动质量，Manus依然继承了LLM基础模型的现有各种问题，如幻觉现象、复杂意图理解不足、长文本信息丢失、多模态能力欠缺等。在执行过程中，这些缺点可能会产生蝴蝶效应，导致最终结果质量低下或明显脱离预期。这些问题的解决依赖着大脑的进化，即基础模型的进一步突破。

工程设计上的优化欠缺降低了体验

在现有模型水平一定的情况下，Manus也需要有更多工程上创新来优化产品体验。在我的使用过程中，有几点特别影响体验。一是Context Length过长而停止。处理过于复杂的任务时，Manus经常执行到一半就因为Context溢出而停止；对于简单任务，多轮交互不到两三轮也是同样的情况。这需要有更高效的Context管理机制，比如Long-term memory & RAG。二是Agent迷失。在任务执行到某个步骤时，经常会卡住，但没有任何提示（如缓慢执行中或负载过高），我通常不敢打断它。除了需要更好的状态提示之外，Manus也需要从系统层面做更多优化以提升高并发下的用户体验。三是中间停止时状态提示不够明显。和Manus工作的同时，人也在做其他事情。执行过程中，Manus可能主动询问人的选择，比如网站URL是否公开。但系统级别提示是不明显的，导致我经常没感知到它在等待我的相应。另外就是按钮上区别也不明显，第一眼直觉我竟然没分辨出需要我点击，在哪儿纳闷了一会。在UI & UX方面，Manus还有很大优化空间。

完全依赖GUI导致低效和高成本

Agent过于类人并非是件好事，因为人是局限的，并不一定高效的。Agent无需双眼看世界，敲击鼠标滑动屏幕，它完全可以用机器语言与环境高效沟通。相比之下，Claude的MCP展示了一个Agent通信交互新的范式，比GUI更加高效，比API更加泛化，更期待用MCP交互的Agent实现。

通用但并不好用，效率质量提升空间巨大

正如产品所定义的，Manus确实表现出了一定的通用性，能够像人一样做多种任务。然而，由于基座模型的局限性和工具的有限性，它在许多任务上的执行效率和输出质量仍然不高。首先，他所规划的路径未必是最优的，可能不如人类专家最高效的工作模式，导致了效率较低。其次，他在特定任务的体验上，比如网站编程或研报撰写，所生成的网站质量和内容深度，都不如Cursor和DeepResearch。当任务聚焦于这类高频任务时，使用垂类Agent体验明显更好，速度更快，质量更好。

与用户私有文件和设备脱节

Manus仍然生活在云端，与用户的私有文件和设备相对脱节，需要用户手动筛选并上传相关资料。这种割裂增加了用户的负担，需要较强的信息筛选和定位能力，或者就只能让Manus在信息缺失的情况下任务上下文理解受限。本地化的大脑确保数据隐私，OS层面的Agent更有潜力成为贴身秘书，更加顺畅地获取并利用用户的私有信息，减少了人的负担，输出也会更满足预期。

2.3 对人的新考验

Manus给的结果不如人愿？有没有可能人自己的问题...

一个残酷的现实是，Manus的出现对使用者的要求提升了。Manus的输出质量是由AI能力和使用者思维共同决定的。人们可能在单步决策时思路足够清晰，因此在与 ChatBot 互动时还能感受到一定的掌控感。General AI Agent更加长远，从输入需求直达结果，清晰的要求有助于获得预期的输出。然而，在任务开始时，人们很难拥有足够的清晰度，有时甚至不清楚自己真正想要的是什么。作为 AI 的使用者，人是否能清楚地描述需求，能否提供相关文件，都将显著影响最终结果。人做Agent的老板并不容易，仍需要在下达命令时就考虑到特定要求，否则只能抱怨AI依然是智障。此外，人的输入影响着Agent输出质量，这也进一步反馈到个人进步上，可能会进一步拉大人与人之间的差距...

我是谁，我从哪里来，将到哪里去 以及未来社会的变革

郁郁 555...

2.4 Manus的启迪

关于技术——创新岂止于底层模型

“套壳”是对Manus最多的指控之一。在大模型时代，基础模型SOTA频登中文顶刊，吊足了大家对底层技术进步的胃口。抱着同等的预期，AI极客们希望Manus复刻如DeepSeek般模型层的技术突破，失望而归，于是直言“套壳”。这种观点显然忽视了技术服务于人的本质，好用才是核心。把复杂的CUDA算子封装起来，Kreas和Pytorch等深度学习库降低了AI代码的实现门槛，推动LLM的出现；更利益相关的，深度学习时代下Data Mining研究也大都“套壳”机器学习模型与方法，但这也促进了AI技术在各种场景下的应用落地。更上一层，Manus将LLM和Computer Use封装成产品，真正让普通人能够更便捷地使用AI。套壳可以普及概念，加速发展，又有何不可？实现一个好用的AI Agent产品也绝非易事，它不仅需要在产品理念上的套壳，更需要工程技术上的优化与创新。即使达到现在的水平，Manus仍是需要做不少技术活的。至于Open-manus/ OWL要实现类似输出质量和用户体验，他们应该还需要不少个三小时。

关于产品——技术不是唯一的壁垒

技术不是唯一的产品壁垒，技术力与产品力并非总成正比。作为AI极客，对于General AI Agent，即使不讲好故事，技术本身就充满了想象。然而，普罗大众对于AI技术的使用能力相对较低。如何将友好易用的交互界面摆在用户面前，体现着团队对于用户产品体验的理解和打磨的差距。即使技术门槛是低的，Manus快速上线，灵活迭代，做到这种可用状态，也反映了执行力重要性。此外，用户粘性、营销策略、团队管理、商业模式也都是产品竞争力的一部分，回归于互联网时代的模式。

关于商业——天下武功，唯快不破

这句话也摆在Manus创始人肖弘的办公室里。对于Startup而言，何时选择暴露产品是一个艰难的决定。过早公开可能导致用户体验不够理想，大公司也可能快速复制并宣传。而如果公开得太晚，市场赛道可能早已挤满了竞争产品。Manus的选择时机恰到好处。Cursor/ DeepResearch展示了垂直领域Agent令人惊艳的生产力和体验，对迈向初阶General AI Agent的共识已达成，OpenAI在二月底甚至提前预告了其雏形。赛道起点，大家都在等待信号枪指令，默念着技术要成熟再成熟（大模型还不够强，Computer Use技术储备可以更好，MCP生态尚不完善...）。这时候，Manus抢跑了，成为了第一个吃螃蟹的人。在某种程度上，GPT在语言模型领域，SORA在视频生成领域，而Manus在通用AI Agent产品赛道上抢先一步，斩获了General AI Agent的冠名。可以预见，XXX-Manus家族将迅速扩张。此外，资金纷至沓来，也有助于Manus后续产品优化及商业扩展。

关于营销——已成热点通用替罪羊

第一天关于Manus的评价确实有所夸张。即使在技术层面将Manus并不能与DeepSeek并称，我依然坚信Manus是General AI Agent迈向可用形态的一个里程碑式的创新。GPT3.5的出现把真正智能聊天机器人普及大众，问题和答案的距离只需一个回车键。General Agent则更近目标一步，直接交付结果，省去了繁琐的中间步骤。但不可否认，其技术层面的创新并没有GPT或R1时刻那么惊艳。真正恐怖地是不到二十四小时，舆论竟然在黑白之间翻面...我相信当时拿到邀请码进行内测的人应该是极少数。但自媒体总想抢发震惊的观点，正面说完了，既然能用的人是少数，干脆再来一遍反面的。不过另一个角度看，黑红也是红，Manus再赢。

三 · 展望篇

将最初的“幻想”替换成展望，分享下自己对AI Agent短期内发展的看法

3.1 架构趋于健全

更小且精明的大脑

Agent干什么，是由大脑决定的（现在是LLM）。基础模型的快速演变可能会使之前提到的缺点逐渐消失。此外，聪明并不是大脑唯一的追求。随着大模型的能力逐渐可以转移到小模型上，小模型逐渐会成长，逐步实现与大模型相当的能力，就像计算机从房间里的庞然大物演变成手掌大小一样。那时的Agent会更加轻巧，分布在真实与虚拟世界各个角落。

更健壮灵活的四肢

与Manus相比，人可以使用成千上万的工具，面临特定任务，能从中找到最适合的来完成。未来，Agent在工具使用方面也将更加发达。根据不同情况找到最佳工具，能够更快速和高质量地实现目标。工具不仅限于软件控制，还可以是硬件操作。Agent将可以像人一样灵活的做事，以及做超越人类能力范围的任务。

更高效的通信语言

受限于人理解和记忆，电脑从控制台进化成了GUI，更加易于使用。但这种基于视觉的理解和基于GUI的交互也导致了效率低下的问题，初衷是对于人类大脑的妥协。回归本源，Agent完全可以摆脱GUI的通信模式，通过自己的语言与世界进行更高效的沟通和交互。比如，Claude推出的MCP协议已提供了一种可能性。未来Agent直接通过MCP中间态的数据进行沟通交互，只需把最终结果呈递给用户。中间数据服务与Agent，人只需浏览结果。

可进化的学习机制

知识渊博但从不从实践中接受反馈，终归是纸上谈兵，这是“理论结合实践”的重要性。此外，高质量的决策并不依赖于全知全能的大脑，况且现实世界往往是部分可观测的。相反，真正的智慧在于如何在不确定和信息不完整的情况下，做出最佳选择。即使大模型做大脑，Agent也需要进入以目标实现为导向的练兵场接受试炼，收集经验，积累智慧（比如用强化学习进行训练，从经验中学习。顺带一想，Barto和Sutton的图灵奖奖金，他们会打算咋花呢）。试想不久的将来，World Model对真实世界的模拟已足够逼真，一个智商爆表Agent在其中游历，用机器语言与环境高效交互，做各种尝试来收集经验，最后用强化学习更新自己的大脑，或许那时真正的AGI已至。

3.2 范式兼容并包

工作流 vs 全自动

工作流强调预定义的自动流程，Agent则更依赖全自动的Planning。然而，事事都去重新思考规划是不必要的。就像车间的流水线一样，如果已经有了非常高效成熟的任务分解，又何必重新规划。另外，至少在Manus目前而言，Planning依然是线性的TODO List。而现实世界更复杂的任务，它可能会存在For循环或While True，这对基础模型的Reasoning&Planning能力提出了更高的挑战。对于人类高频任务，基于专家级别的经验，我们可以直接构造出最优的Workflow。Agent无需思考，效率更高，也可以完成更复杂的任务，目前而言是更优的。当然，为所有任务编写Workflow并不现实。对于更多的非高频任务，依然需要依靠AI Agent的General能力来去处理长尾。

垂类 vs 通用Agent

和人类社会的明确分工相似，垂类Agent更专注于在特定场境或任务下的表现，而通用Agent则强调对所有任务的适应性。即便大模型预训练预料足够丰富，但在某些场景中（如医学、金融），领域特定的知识仍显不足。另外，将知识灵活应用于实战，也需要对大模型精细的调教。这凸显了垂类Agent的优势。它们通常在特定领域数据上进行了微调，并对特定场景做了针对性适配，比如Workflow的设计、工具的使用等。目前来看，既然已经存在更高效的垂类Agent，比如编写代码的Cursor和撰写研报的DeepResearch，那么似乎Manus也不需要做到通才。或许未来，一个General Agent雇佣着多个偏科的T型人才——垂类Agent。当下属能力不足无法解决的时候，它再自己上手处理，这样会更加高效且成本更低。当然，大脑通过MOE针对不同任务分区，开启相应的模式，垂类Agent的优势将逐渐消失。

云端 vs 端边云协作

云端的Agent开辟虚拟机，终归有点自娱自乐。而端侧的Agent在用户本地生活，能够直接访问用户隐私数据、操作用户设备，甚至调动用户各类其他设备，将有更大的想象空间。有了用户的私有数据，Agent可以更好地理解用户，甚至比他自己还理解。即使用户的描述不够清晰，端侧Agent更综合地推测出用户潜意识里的意图。同时它自己快速定位到相关文件，避免用户的输入，更程度的解放人。此外，统一调度各种私人的设备，贴身秘书或将人人可有。

3.3 核心服务与人

实现性价比的权衡

在最高阶的General AI出现之前，Trade-off应该也会一直存在，比如运行成本与结果好坏、决策质量与执行效率、灵活性与稳定性，等等。在模型能力一定情况下，如何平衡这些Trade-off，做到更优的用户体验，也将是区别开各种General AI Agent产品的核心因素之一。

更友好的人机协作模式

AI的目标是服务于人，应当主动去适应人的思维。更多的意味着AI主动降维兼容人的局限性，比如目标清晰度低、思维过于发散等。而且，端到端地从任务描述到交付结果确实足够高效，然而也无形中增加了人的失控感。全自动化黑盒流程看似完美，但人类往往更需要的是过程中适度参与感以及掌控权。即使能力再强，真正聪明的AI应该会体贴地降低身段，以更符合人类自然直觉的方式与我们交互。

可缓解人类焦虑的AI

AI能力越强，人越需要对其有掌控力，价值观对齐是必须的（之前太愚昧了，并不重视这个方向，现在的我逐字学习）。此外，如何主动取信于人，也是AI Agent必须思考的事情。当AI在解放生产力的同时，也不可避免地加剧了人类的焦虑（毕竟人类的进步速度远低于AI）。未来，作为“强者”的AI，也应当主动去兼容“弱者”——人类。强烈建议AI Agent开始工作前，先抄写《人性的弱点》，狐狸扮演小白兔来安抚下人类脆弱的心灵。

完结撒花

从试用Manus的第一天便开始Plan这篇笔记，每天抽出至少三四个小时来写，历时6天，终于撒花，可见至少我这个人在写作上是低效的。或许未来某一天，AI可以感知并理解我的脑电波，梳理我跳跃的海马体，以我的风格撰写出这篇文章。希望那时的AI可以轻度剥削我的脑力（这个阶段来看这还是幻想）。

Tianfu Wang

2025/03/14