

科研素养训练 I

4. 英文论文解析

效率工具与技术文档

宋庆禹 厦门大学

0323, 2026

第一部分：英文文献阅读示例

IMRaD 结构

现代科学论文的通用底层骨架（沙漏结构）：

I

Introduction

(引言)

为什么要做这个研究？
大概做了什么东西？

M

Methods

(方法)

我是怎么做的？

R

Results

(结果)

我发现了什么？

a D

Discussion

(讨论/结论)

这有什么意义？

“非线性阅读法”第一遍：鸟瞰图 (5分钟)

目标：判断这篇论文值不值得你花时间。

只读三个部分：

- 1 Title (标题)：** 它是不是我要找的方向？
- 2 Abstract (摘要)：** 核心缩影，快速了解背景、方法和结果。
- 3 Conclusion (结论)：** 直接拉到最后看大结局。

决策点： 读完这三部分，如果你觉得没用，立刻关掉，去寻找下一篇。

“非线性阅读法”第二遍：抓骨架 (15分钟)

目标：像X光扫描一样，掌握论文的核心逻辑和主要创新点。

阅读重点：

- ✓ 仔细阅读 Introduction 的最后两段（通常是核心贡献总结）。
- ✓ 扫视所有的图表、插图和图注（Figures & Tables）。
- ✓ 看各个章节的小标题。

跳过：

- ✗ 跳过所有的数学证明和复杂的实验细节。

“非线性阅读法”第三遍：扣细节 (因人而异)

❗ **前提：**这篇论文对你极度重要，你需要复现它，或者在它的基础上做后续研究。

🔍 **行动：**一字一句死磕 Methods，推导公式，检查数据集，甚至跑它的开源代码。

大一提示：**现阶段你们很少需要进行到第三遍，掌握第一、二遍，能快速获取有效信息，就已经超越同龄人了。**

文献阅读打卡模板

每读完一篇论文，用这4个问题拷问自己（尝试用自己的话写下来）：

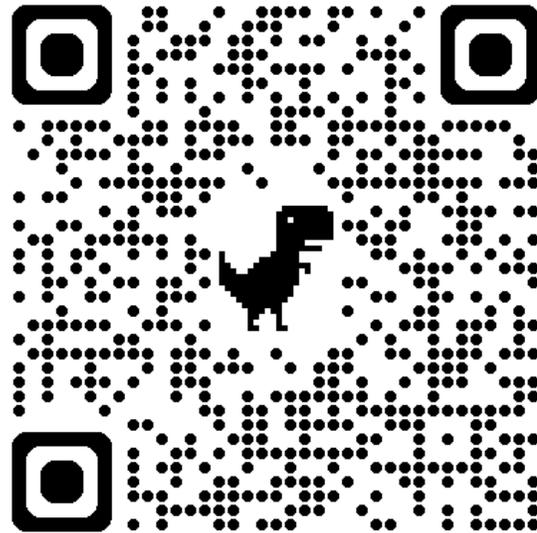
1. 这篇论文试图解决什么痛点？
2. 核心创新点是什么（一句话概括）？
3. 结论或结果是什么？
4. 对我的启发或疑问是什么？

如果做不到这4点，这篇论文等于没读。

文献阅读练习:

Faster Progress Bars: Manipulating Perceived Duration with Visual Augmentations

Chris Harrison Zhiquan Yeo Scott E. Hudson
Human-Computer Interaction Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
{chris.harrison, zyeo, scott.hudson}@cs.cmu.edu



ABSTRACT

Human perception of time is fluid, and can be manipulated in purposeful and productive ways. In this note, we propose and evaluate variations on two visual designs for progress bars that alter users' perception of time passing, and "appear" faster when in fact they are not. As a baseline, we use standard, solid-color progress bars, prevalent in many user interfaces. In a series of direct comparison tests, we are able to rank how these augmentations compare to one another. We then show that these designs yield statistically significantly shorter perceived durations than progress bars seen in many modern interfaces, including Mac OSX. Progress bars with animated ribbing that move backwards in a decelerating manner proved to have the strongest effect. In a final experiment, we measured the effect of this particular progress bar design and showed that it reduces the perceived duration among our participants by 11%.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors

Keywords: Progress bars, percent-done indicators, perception, perceived performance, induced motion.

INTRODUCTION

Progress bars [8], typically used to visualize the progression of an extended operation, are prevalent in current user interfaces. In desktop systems, advanced users often multitask during these periods. However, it is not uncommon for advanced users to watch an install finish or file transfer complete – especially if they are waiting on that operation. Anecdotally, novice users tend to anxiously monitor their progress bars, in hopes that some error does not occur. In non-desktop applications (e.g., ATMs, ticketing kiosks, and some mobile device platforms), novice and expert users alike have no choice but to watch progress bars frustratingly inch their way across the screen. No matter how objectively fast we make these operations, it is typically the subjective speed that mars the user experience [11]. Indeed, a core tenet of HCI is to improve user satisfaction.

Previous research has shown that the perceived duration of progress bars can be manipulated by changing how they

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
Copyright 2010 ACM 978-1-60558-929-9/10/04...\$10.00.

move (e.g., pauses, accelerations) [4]. We extend this exploration to the manipulation of visual attributes. Following a series of head-to-head comparisons of perceived duration for different visual styles, we conclude with an experiment that quantitatively assesses the perceptual improvement over the ubiquitous, solid-color progress bar.

This work adds to the nascent field of time design [5,9] – a discipline that looks at how temporal aspects of interactive systems can be structured and manipulated to improve the user experience. It is argued that subjective time is not only the most readily manipulated, but also the most important [11]. After all, our perception is our reality. Finally, with good design, such benefits can often be realized immediately and essentially for free (i.e., we do not have to *make* faster computers to make computers *feel* faster).

STUDY 1: PULSATING PROGRESS BARS

Frequency variations in rhythmic stimuli have been shown to affect peoples' perception of time [7]. We hypothesized that this effect could be used to reduce the perceived duration of progress bars. To test this, we designed several variations of a progress bar that used a sinusoidal visual pulsation, causing the progress bars' fill color to vary between light blue and blue.

Study Design

To investigate how pulsation can be used to manipulate perceived duration, we recruited 20 participants (7 female, mean age 23) to evaluate five distinct behaviors we had

Behavior Name	Start Frequency (0% Progress)	End Frequency (100% Progress)
Constant	1.1 Hz	1.1 Hz
Slow Increasing	1.1 Hz	1.17 Hz
Fast Increasing	1.1 Hz	1.25 Hz
Slow Decreasing	1.1 Hz	0.95 Hz
Fast Decreasing	1.1 Hz	0.8 Hz

Table 1. The five pulsating progress bar behaviors.



Figure 1. The study interface.

I - Introduction (引言：讲好一个故事)

INTRODUCTION

Progress bars [8], typically used to visualize the progression of an extended operation, are prevalent in current user interfaces. In desktop systems, advanced users often multi-task during these periods. However, it is not uncommon for advanced users to watch an install finish or file transfer complete – especially if they are waiting on that operation. Anecdotally, novice users tend to anxiously monitor their progress bars, in hopes that some error does not occur. In non-desktop applications (e.g., ATMs, ticketing kiosks, and some mobile device platforms), novice and expert users alike have no choice but to watch progress bars frustratingly inch their way across the screen. No matter how objectively fast we make these operations, it is typically the subjective speed that mars the user experience [11]. Indeed, a core tenet of HCI is to improve user satisfaction.

Previous research has shown that the perceived duration of progress bars can be manipulated by changing how they

move (e.g., pauses, accelerations) [4]. We extend this exploration to the manipulation of visual attributes. Following a series of head-to-head comparisons of perceived duration for different visual styles, we conclude with an experiment that quantitatively assesses the perceptual improvement over the ubiquitous, solid-color progress bar.

This work adds to the nascent field of time design [5,9] – a discipline that looks at how temporal aspects of interactive systems can be structured and manipulated to improve the user experience. It is argued that subjective time is not only the most readily manipulated, but also the most important [11]. After all, our perception is our reality. Finally, with good design, such benefits can often be realized immediately and essentially for free (i.e., we do not have to *make* faster computers to make computers *feel* faster).



大背景：进度条主观时长 可以通过 更改进度条变化方式 所改变。



指出缺陷：我们想看看能不能通过视觉效果的变化达到同样的目的。



我的贡献：我们通过一系列的实验（不同视觉风格），论证了这个结论：确实比纯色、普通进度条更好。

M - Methods (方法：细节的做法描述)

STUDY 1: PULSATING PROGRESS BARS

Frequency variations in rhythmic stimuli have been shown to affect peoples' perception of time [7]. We hypothesized that this effect could be used to reduce the perceived duration of progress bars. To test this, we designed several variations of a progress bar that used a sinusoidal visual pulsation, causing the progress bars' fill color to vary between light blue and blue.

Study Design

To investigate how pulsation can be used to manipulate perceived duration, we recruited 20 participants (7 female, mean age 23) to evaluate five distinct behaviors we had

Behavior Name	Start Frequency (0% Progress)	End Frequency (100% Progress)
Constant	1.1 Hz	1.1 Hz
Slow Increasing	1.1 Hz	1.17 Hz
Fast Increasing	1.1 Hz	1.25 Hz
Slow Decreasing	1.1 Hz	0.95 Hz
Fast Decreasing	1.1 Hz	0.8 Hz

Table 1. The five pulsating progress bar behaviors.

Pulsating 震动的
Sinusoidal 正弦的



内容：设计不同填充速度的“呼吸”进度条（浅蓝-蓝），观测人对他们观感上的区别。

实验设计：找20个实验人员，去评估5组不同的实验。

M - Methods (方法：细节的做法描述)

created (Table 1). To identify perceptual differences, we employed a simple interface that presented two progress bars to the participant (Figure 1), each of which took five seconds to complete. Progress bars were run sequentially – when the first one finished, the second one began. Sequential, rather than parallel presentation was necessary to hide the fact that the progress bars were actually the same duration. Although five seconds is a somewhat short duration for a progress bar, it was a necessary trade off to run the number of trials we desired and still maintain user attention for the complete experiment. However, as you will see in discussion of the results from our final experiment, we believe that these types of effects can scale to at least somewhat longer durations.

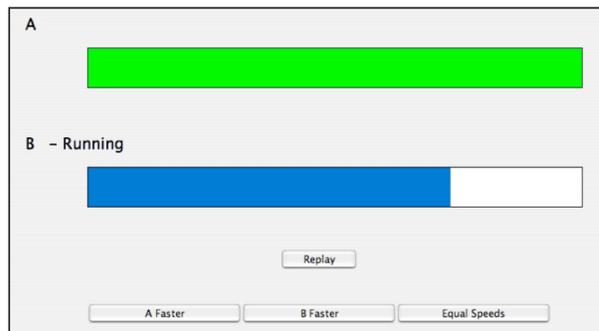


Figure 1. The study interface.

After watching the two progress bars run to completion, participants selected whether they believed the first progress bar was faster, the second progress bar was faster, or that the progress bars were equal in duration. The next trial began once an answer was provided. Participants could replay the progress bars if desired.

All distinct ordered pairings of the five pulsation behaviors were shown to participants, for total of 25 trials. This included testing behaviors against themselves, and both orderings of every pair, the results of which are combined to control for order effects. Presentation order was also randomized to control for time order errors (see [3], p52).

步骤：

1. 依次播放两个进度条，各5秒钟。
2. 两两对比，选择哪个进度条更快/一样快。
3. 25种对比组合

M - Methods (方法：细节的做法描述)

Results and Discussion

We employed a simple preference metric [4] to compare the performance of different behaviors, which is calculated as an average of the following individual scores: +1 if the first behavior is preferred (i.e. perceived to be faster), -1 if the second is preferred, and 0 if the participant believed the two to be equal in duration. For example, in 40 comparisons of Fast Increasing vs. Slow Decreasing, 16 participants felt the former was faster, 6 participants felt the latter was faster, and 18 believed the two progress bars were equal in duration. Thus, the mean preferences score is $(16 - 6) / 40 = 0.25$. The positive value indicates a preference towards Fast Increasing. To derive our statistical measures, we employed a two-sided sign test of the null hypothesis that each function was equally likely to be preferred.

核心：统计每个进度条大家的打分

具体流程：

1. 两两对比：被选中的加一分，没选中的扣一分；觉得速度一样的不得分。
2. 统计均值。

R - Results (结果：用图表说话)

Results and Discussion

We employed a simple preference metric [4] to compare the performance of different behaviors, which is calculated as an average of the following individual scores: +1 if the first behavior is preferred (i.e. perceived to be faster), -1 if the second is preferred, and 0 if the participant believed the two to be equal in duration. For example, in 40 comparisons of Fast Increasing vs. Slow Decreasing, 16 participants felt the former was faster, 6 participants felt the latter was faster, and 18 believed the two progress bars were equal in duration. Thus, the mean preferences score is $(16 - 6) / 40 = 0.25$. The positive value indicates a preference towards Fast Increasing. To derive our statistical measures, we employed a two-sided sign test of the null hypothesis that each function was equally likely to be preferred.

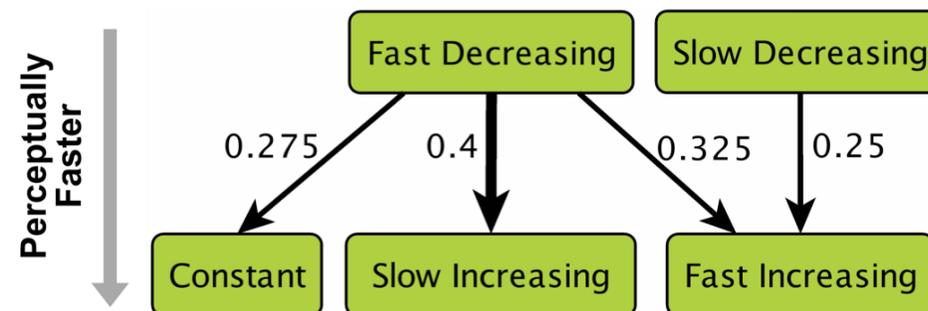


Figure 2. Statistically significant preferences between the five pulsation behaviors. Thin lines indicate significance at $p < 0.05$; thick at $p < 0.001$. Mean preference scores are labeled on the edges.

结果：统计检验

具体流程：

1. 粗线：有超过 95% 把握的事件。
2. 细线：有超过 99.9% 把握的事件。
3. 数值：平均得分差异。

文献阅读打卡示例

每读完一篇论文，用这4个问题拷问自己（尝试用自己的话写下来）：

1. 这篇论文试图解决什么痛点？ 验证不同进度条显示效果，对人观感的影响。
2. 核心创新点是什么（一句话概括）？ 首次通过真人实验，对比测试不同呼吸灯颜色切换速度给人带来的快慢上的视觉误导，推荐“更好”的进度条设计。
3. 结论或结果是什么？ 越来越快的呼吸灯，观感上速度最快。
4. 对我的启发或疑问是什么？ 实验人员太少，只有20个。

如果做不到这4点，这篇论文等于没读。

第二部分：效率工具与技术文档

日常痛点场景



痛点1：文档排版地狱

用Word写实验报告，插入一张图片，整个文档的排版瞬间崩溃，格式调整耗费大量时间。



痛点2：文件传输灾难

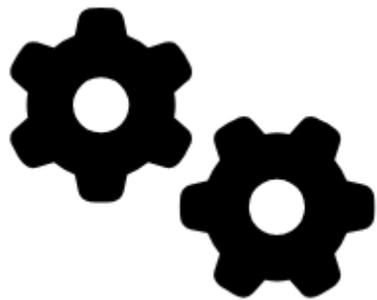
依靠微信互传文件，版本混乱，最终收到《结课报告-最终版-打死不改版-绝对最终版.docx》。



痛点3：知识黑洞

收藏夹里躺着几百篇干货，用时却找不到；
电脑桌面上堆满命名随意的TXT和PDF。

什么是“效率工具”？



思维方式的物化

科研效率的核心原则

- **格式与内容分离**：专注写作本身。
- **历史可追溯**：拥抱试错，随时回滚。
- **知识网状链接**：打破信息孤岛。

三大效率工具



Markdown

像写代码一样写文档，告别繁琐的排版。让内容回归纯粹。



GitHub

科研资产的安全舱与时光机。
理解版本控制的顶级理念。



Notion / 知识库

构建你的第二大脑，让知识相互链接，产生不可思议的复利。



Markdown 入门

抛弃繁琐，回归纯粹的文本创作。

Word (富文本) vs Markdown (纯文本)

Word (所见即所得)

- 把文本和格式 (字体、颜色、大小) 混合存储在复杂的文件中。
- 极度容易因版本 (2016 vs 2019) 和平台 (Mac vs Win) 不同而发生排版错乱。
- 文件体积庞大, 包含大量冗余数据。

Markdown (纯文本)

- 最纯粹的文本格式。通过简单的符号标记格式。
- 最终由渲染器生成漂亮的排版。无论在任何设备或网页上, 永远保持一致。
- 文件极小 (几KB), 极易进行版本控制。

Markdown 的诞生与哲学



- **发明者:** John Gruber (2004年)。
- **设计初衷:** 易读易写。在纯文本状态下依然具有良好的可读性，并且可以轻松转换为HTML或PDF。
- **为什么理工科学生必备?**
它是全球开发者社区、GitHub、Jupyter Notebook、各类博客系统和学术论坛的通用语言。

核心工具推荐：Typora 与 腾讯文档



本地神器 Typora (收费)

所见即所得的 Markdown 编辑器，极简界面，支持海量排版主题，强大的公式和代码块支持。
非常适合写个人实验报告和论文初稿。



云端协作 腾讯文档 (免费)

支持在线 Markdown 语法，适合小组科研项目共同编辑，实时保存。跨越设备阻碍，随时随地开启高效协同写作。

语法 1：标题体系

使用说明：

使用 # 来控制标题层级。

注意：# 后面必须加一个空格！

一级标题 (对应论文大标题)

二级标题 (对应章节)

三级标题 (对应子章节)

我是标题

我是标题

我是标题

语法 2：文本强调

B 加粗：使用双星号包裹 → 重点结论 `**重点结论**`

I 斜体：使用单星号包裹 → 专有名词 `*专有名词*`

~~S~~ 删除线：使用双波浪号包裹 → 错误数据 `~~错误数据~~`

`</>` 行内代码：使用反引号包裹（键盘Tab上方）。用于标注变量名或短代码片段。例如 ``x = 10``

语法 3：无序列表与有序列表

无序列表

使用 - 或 * 加空格。

用于并列的知识点归纳。

- 数据预处理
- 特征工程

有序列表

使用数字加英文句点 1. 。

用于实验步骤或操作流程，支持 Tab 缩进嵌套。

1. 清洗数据
2. 训练模型

语法 4：插入链接与引用

插入链接

超链接语法：[显示文本](URL链接)

例如：[厦门大学官网](https://...)

实现文本与网址的分离，保持页面整洁。

引用区块

使用 > 加空格。

用于引用文献原话或强调特别提示。

| As Einstein said...

语法 5：插入图片（核心痛点解决）

语法规范

`![图片描述](图片路径或URL)`

(注意比超链接多了一个感叹号!)

技术优势

图片在本地只是一个路径链接，纯文本文件体积极小，绝对不会像Word那样动辄几十MB引发崩溃。

Typora 进阶

可以直接复制粘贴图片，Typora会自动将其保存到指定文件夹并生成Markdown语法，彻底解放双手。

`![[image-20260322223704383]](image-20260322223704383.png)`

科研素养训练 I

4. 英文论文解析 效率工具与技术文档

宋庆禹 厦门大学

0323, 2026

语法 6: 代码块

使用三个反引号 ` 包裹多行代码。
可以在第一行反引号后指定语言 (如 python, c, java) ,
编辑器会自动进行**语法高亮**!

```
python import pandas as pd import numpy as np
```

```
import pandas as pd import numpy as np # 数据清洗函数 def  
clean_data(df): df = df.dropna() return df
```

语法 7: 创建表格

使用竖线 `|` 分隔列，使用横线 `-` 分隔表头和内容。

```
| 样本编号 | 测试结果 | 备注 | | :--- | :---: | ---: | | 001 | 成功 | 正常 |
```

(冒号控制左对齐、居中、右对齐。建议复杂表格使用快捷键生成)

样本编号	测试结果	备注
001	成功	正常
002	失败	电压异常

语法 8: LaTeX 数学公式

科研离不开公式。

Markdown 原生完美支持 LaTeX 数学公式渲染。

- **行内公式**: 使用单个 `$` 包裹。例如: `$E=mc^2$`
- **独立公式**: 使用双 `$$` 包裹, 会居中大字号显示。

渲染效果演示:

$$E = mc^2$$

Markdown 画图功能拓展 (Mermaid)

高级技巧

使用 Mermaid 语法, 直接用纯文本画出实验流程图、甘特图、时序图。

```
mermaid graph TD
  A[准备样品] --> B(高温加热)
  B --> C{是否熔化?}
  C -->|是| D[进行测试]
  C -->|否| B
```

优势:

修改流程图只需改动文字字符, 无需重新打开庞大的画图软件重新拖拽图形! 极大提升绘图与修改效率。



Typora 进阶：大纲视图与沉浸模式



大纲视图 (TOC)

根据你输入的 # 标题层级，侧边栏会自动生成树状目录。无论是几万字的科研综述还是复杂的实验报告，都能轻松一键导航。



打字机 / 专注模式

开启后，编辑器将永远高亮你当前正在输入的段落，其余文本变灰。让你彻底抛开排版干扰，100%沉浸在思考与写作流中。

一键导出：从 Markdown 到 PDF/Word

导出为 PDF

写好纯文本后，一键渲染导出为带精美排版和目录的 PDF 文件。

直接用于提交作业，任何人打开格式都不会乱。

导出为 Word

遇到只收 Word 格式文件的老派导师怎么办？

通过 Pandoc 等插件，Markdown 可以无损、带格式自动导出为 .docx 文件。

腾讯文档中的 Markdown 协作

-  当小组做大作业时，开启腾讯文档的在线文档。
-  直接在段首输入 `#`、`-`、`>` 等快捷键，体验“边打字边排版”的云端协作。
-  彻底避免了多人各自修改格式产生的“格式冲突灾难”。



实战场景讨论：写一篇实验报告

传统 Word 流程

1. 先写一段字
2. 选中标题，去菜单栏找字号、加粗
3. 插入一张图片，文字突然跑到上一页
4. 疯狂敲击回车键调整段落间距
5. 因为排版心烦意乱，中断思路

现代 Markdown 流程

1. 双手不离键盘，一气呵成敲击文本
2. 用 # 顺手标明标题
3. 用 贴上实验分析代码
4. 检查无误，点击“导出为 PDF”
5. 完美收工，保持极致专注

常见的 Markdown 误区



误区一：为了排版而排版

Markdown 的核心价值观是“忽略排版专注于内容”。
千万不要本末倒置，花几天时间去寻找、调试花哨的第三方 CSS 渲染主题。



误区二：死记硬背语法

常用的语法其实只有5-6个（标题、加粗、列表、图片）。实际使用几次就能形成肌肉记忆，记不住的冷门语法随时查阅手册即可。

生态拓展：支持 Markdown 的平台

掌握了 Markdown, 相当于掌握了互联网的通用创作语言。走遍天下都不怕。



GitHub (Readme)



Notion / Obsidian

知乎

知乎 / 简书



Jupyter Notebook

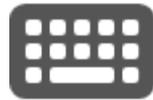
小结：为什么你要学会写 Markdown?



**将大脑宝贵的算力分配给逻辑推演与创新突破
，把机械排版的苦力活全部交给机器。**



——效率科学的核心奥义





GitHub 概念入门

版本控制与开源精神（绝不仅是程序员的专属工具）。

终极灾难场景再现

W 毕业论文_初稿.docx

W 毕业论文_修改版.docx

W 毕业论文_打死不改版.docx

W 毕业论文_最终提交给导师版_v2.docx

痛点分析：

- 你根本不知道每个文件版本到底改了什么内容。
- 如果不小心在“最终版”里删了一大段极其重要的论述并按了保存，很难找回。
- 命名混乱，极易发错文件。

什么是“版本控制” (Version Control)?

严谨定义

一种记录一个或若干文件内容变化,
以便将来查阅特定版本修订情况的**软件工程系统**。

通俗解释

科研文件的“存档”与“时光机”。
你不仅可以随时读档，还能清楚地看到每次存档时
你的修改说明。

Git 与 GitHub 的区别



Git: 底层引擎

运行在你电脑本地的版本控制工具（命令行软件）。
本课程无需深入学习复杂命令，只需知道它是驱动一切的核心技术。



GitHub: 托管平台

基于 Git 技术的在线代码与文档托管平台。
也被称为全球最大的“知识宝库”与“同性交友网站”。

类比：Git 就像你用来拍照的相机引擎，GitHub 则是你展示和备份照片的 Instagram / 朋友圈。

核心概念 1: Repository (仓库)

简称 Repo。

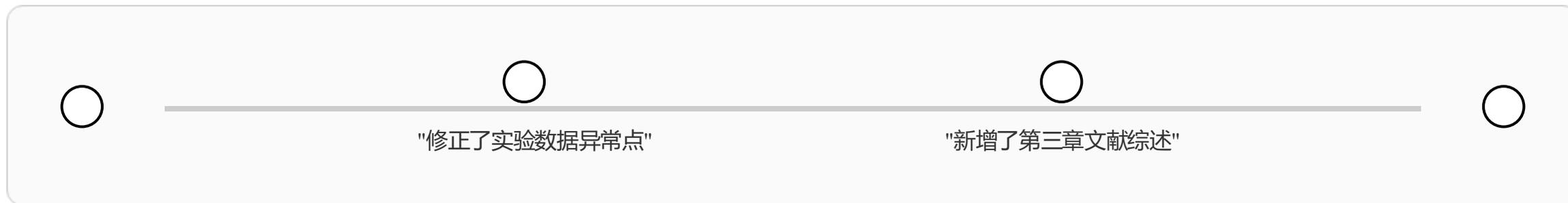
你可以把它理解为存放一个独立科研项目、一门核心课程的大作业、或者一本专著的所有文件（包括代码、数据、Markdown文档）的“终极超级文件夹”。

每一个 Repo 都有自己独立、完整的历史修改记录。



核心概念 2: Commit (提交记录)

不仅是“保存(Save)”，更是“打包并打上标签”。



每次 Commit, 你都需要写一句话说明 (Commit message) 。

通过 Commit 列表, 导师可以清晰看到你是如何一步步推进科研工作的。

核心概念 3: Branch (分支)

“平行宇宙”的概念。

你想尝试一种极其大胆的新算法（或者重写论文的某一核心章节），但怕搞砸了影响原来的稳定版本？

你可以开一个“分支”。

在分支上随便改，成功了就“合并 (Merge)”回主线；失败了就直接丢弃，主线毫发无伤！



为什么科研人员需要使用 GitHub?

-  **可重复性研究 (Reproducibility):** 将论文的原始数据和分析代码开源到 GitHub, 是顶级期刊 (如 Nature, Science) 的强烈建议甚至硬性要求。
-  **防灾备份机制:** 电脑丢了? 硬盘坏了? 只要 push 到 GitHub, 你的科研心血永远不会丢失。
-  **极佳的电子名片:** 你的 GitHub 主页上那面绿色的“代码提交墙”, 是展现你学习毅力和专业技能的最强名片。

什么是“开源” (Open Source)?

开源不只是“免费”，其核心是

“开放源代码与底层逻辑，允许他人审阅、修改和分发”。

**如果说我看得比别人更远些，那是因为我站在
巨人的肩膀上。**

—— 艾萨克·牛顿

现代科研和人工智能 (AI) 的飞速发展，完全建立在开源社区的无私奉献与全球协作之上。

科研真的做到开源了吗？

- 网络领域公开代码很少
 - 近十年，不到50%

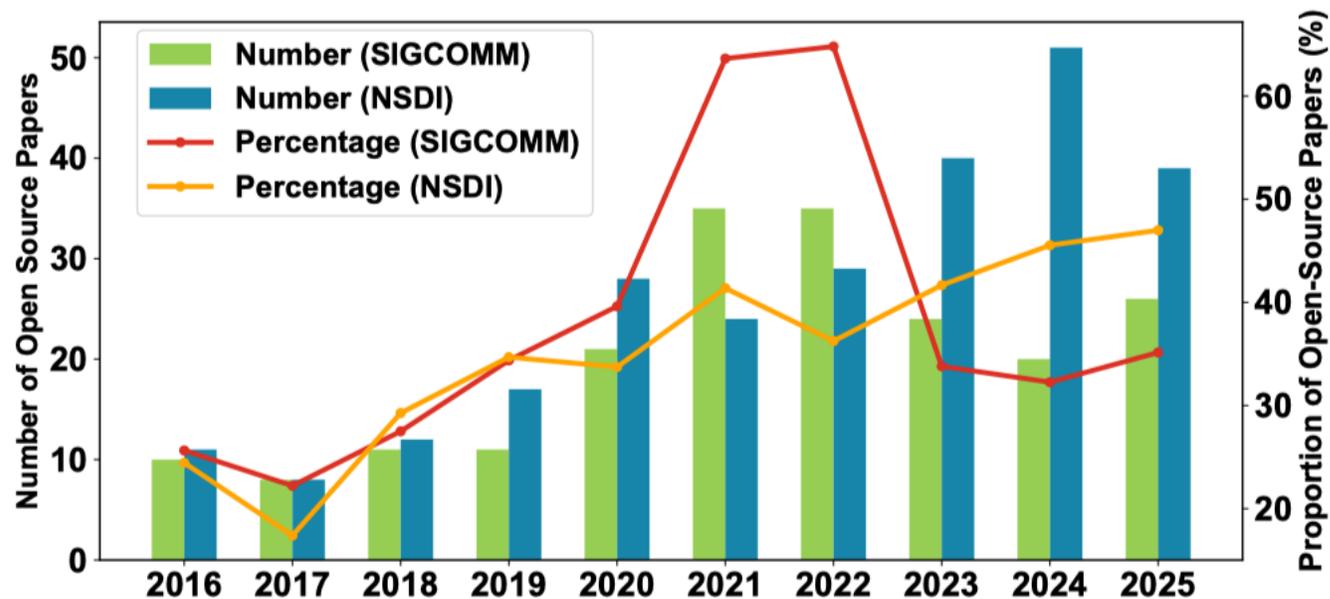
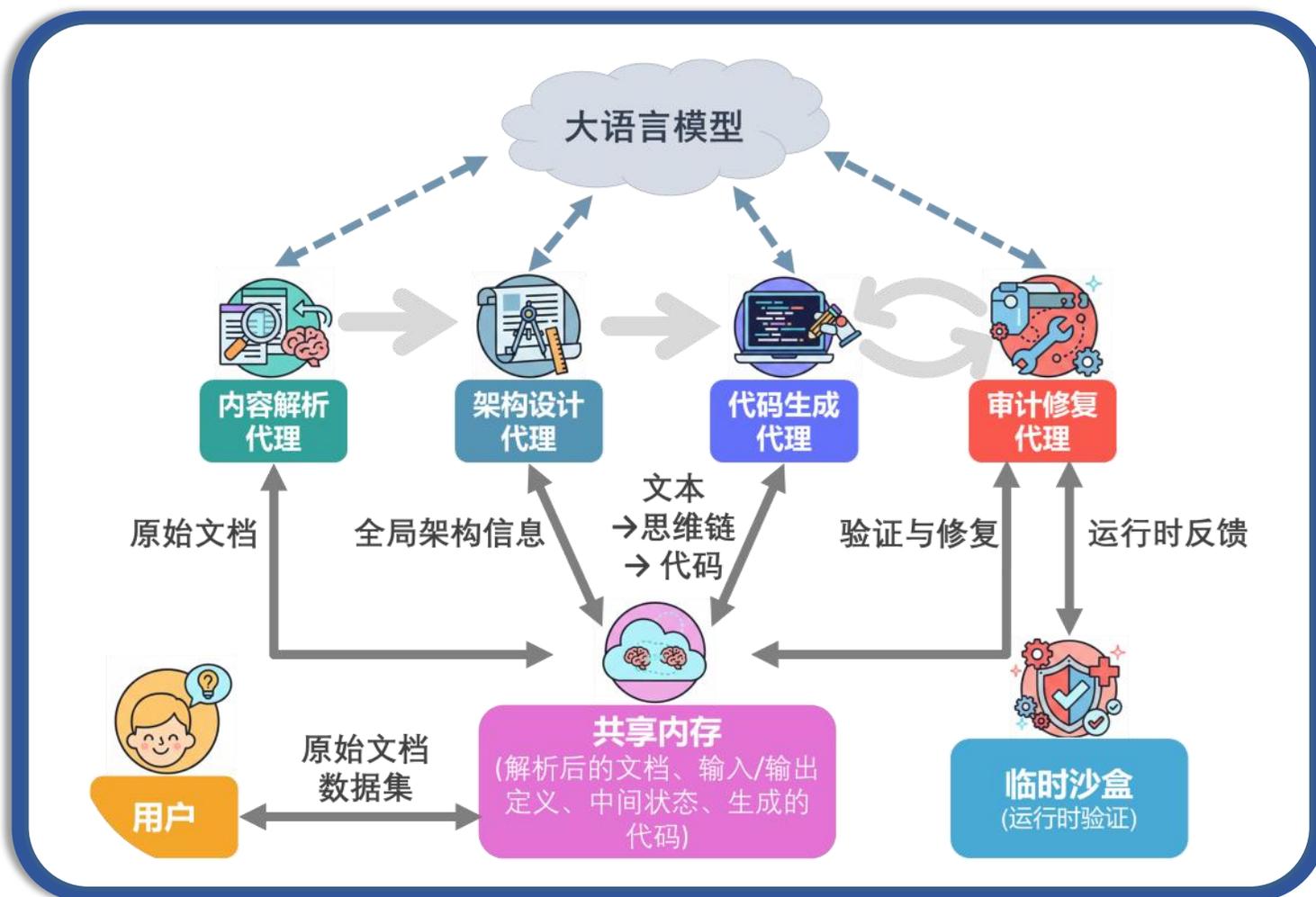


Figure 1: Statistics of SIGCOMM and NSDI Papers with an Open-Source Prototype from the Authors (2016-2025).

自动化论文代码复现工具

Leveraging Large Language Models for Automated Reproduction of Networking Research Results

<https://arxiv.org/abs/2509.21074>



四大核心智能体：

- 内容解析
- 架构设计
- 代码生成
- 审计修复

关键基础设施：

- 共享内存
- 临时沙盒
- LLM 驱动

工作流：

文档解析→架构设计
→代码生成→审计迭代修复

GitHub 的社交属性：Star 与 Fork



Star (标星)

相当于微博的点赞或收藏。遇到优秀的开源项目、厉害的论文复现代码，点个 Star，既是对原作者的鼓励，也方便自己日后查找。



Fork (复刻)

把别人优秀的仓库“完整克隆”到自己的账号下，作为自己研究的起点。“你的代码很好，现在它是我的起跑线了。”

GitHub 的项目管理：Issues

Issues 不是单纯的“问题提问”

它是科研项目的“讨论区与任务板”。

- 遇到别人代码里跑不通的 Bug，提一个 Issue 寻求帮助；
- 记录自己下周要做的实验计划，也可以给自己提一个 Issue 并勾选进度；
- 导师可以通过 Issue 给你分配科研阅读任务。



带你逛逛 GitHub 界面 (1) - 仓库主页



清晰的文件目录区与下方的 Markdown 渲染区，构建了极度专业的项目门面。

带你逛逛 GitHub 界面 (2) - 历史记录

Commit: 修正了数据分析逻辑

```
- y_pred = model.predict(X_train)
```

```
+ y_pred = model.predict(X_test)
```

红色（删除的代码）和绿色（新增的代码）的对比界面，直观感受“代码审查(Code Review)”的严谨过程。

给小白的建议：如何下载资料？

很多同学第一次上 GitHub，对着满屏英文找不到下载按钮。

1 新手方案

寻找页面右侧鲜艳的绿色 **Code** 按钮，点击后在下拉菜单中选择 **Download ZIP**。解压即可得到全部文件。

2 进阶方案

打开本地终端命令行，输入：

```
git clone <仓库URL>
```

未来科研工作者的必备技能。

小结：拥抱开源与版本控制



不要害怕自己的代码或文章写得烂而不愿公开。

开源本身就是一种终身学习的态度。



养成随时 Commit 的习惯，让你的每一步科研努力都有迹可循。





笔记与知识管理

从“收藏从未停止”到“构建第二大脑”。

大学生的“知识焦虑”困境



信息过载

每天看到大量知乎干货、公众号文章、学术文献，习惯性“马克（Mark）”。



信息孤岛

Word 笔记在D盘，PDF 在C盘，网页收藏夹在浏览器，彼此毫无关联。



输出匮乏

看的多用的少，写期末论文时大脑一片空白，收藏的内容完全提取不出来。

传统“文件夹树状图”思维的局限性

✗ 传统方式：树状分类

严格按照文件夹层层嵌套（如：大一上 -> 高等数学 -> 第一章）。

致命缺陷：真实世界的知识不是线性的。比如一篇“机器学习在材料学中的应用”论文，该放计算机文件夹还是材料学文件夹？强行单一分类必然导致找不到文件。

✓ 现代方式：网状互联

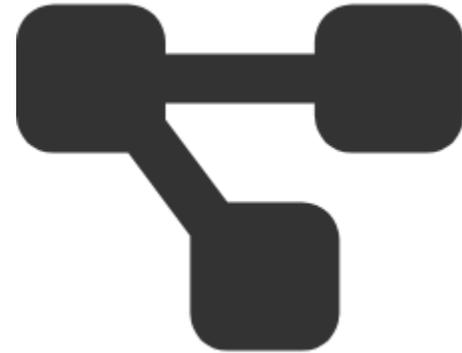
知识点像大脑神经元一样，通过网络相互连接，不再受限于单一维度的层级树状图。打破物理文件夹的束缚。

现代知识管理理念：双链与网状结构

通过“**标签 (Tags)**”和“**双向链接 (Backlinks)**”把零散的知识点串联起来。

卡片盒笔记法 (Zettelkasten) 核心：

- 用自己的话写下原子化的知识点。
- 抛弃分类焦虑，主动寻找新笔记与旧知识的连接点。
- 最终知识将像网络一样自下而上地生长出来。



知识管理神器：Notion



All-in-one workspace

将笔记记录、任务管理、数据库搭建完美融合的革命性工具软件。

为什么推荐？

- 向拥有教育邮箱（.edu.cn）的学生免费提供 Pro 权限。
- 极简黑白颜值，排版极其自由。
- 支持无限层级的页面嵌套。

Notion 核心概念 1: 万物皆 Block (区块)

彻底打破 Word 中固定“页面”与“行段”的概念限制。

一段普通文本

Block

待办事项

Block

在 Notion 中, 上述全都是独立的“块 (Block)”。

你可以通过鼠标拖拽 (Drag & Drop), 像搭积木一样随意拼装组合, 实现极度自由的排版布局。

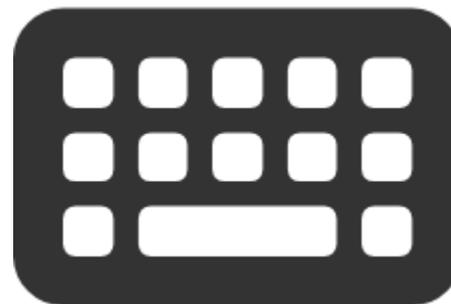
Notion 核心概念 2: Slash Command (斜杠指令)

极客级的沉浸体验:

写笔记时再也无需动鼠标去繁杂的菜单栏里寻找格式。

只需在键盘上输入 `/` (斜杠), 即可瞬间唤出所有功能菜单。

- 输入 `/h1` 变成大标题
- 输入 `/image` 插入本地图片
- 输入 `/math` 插入公式 (完美底层支持LaTeX)



Notion 核心概念 3: Database (多维数据库)

Notion 最强大的杀手锏，同一份底层数据可一键切换多种展现形式：



Table (表格视图)

用来记录大量学术文献的元信息（作者、年份、期刊）。



Board (看板视图)

按进度列（Todo, Doing, Done）
管理你的科研项目。



Gallery (画廊视图)

以图片为封面，展示实验室拍下的照片或设计图纸。

实战场景 1：搭建文献阅读追踪库

痛点： 读了一百篇英文文献，完全混淆。 **解决：** 搭建 Notion 多维数据库。

论文题目 (Title)	作者/年份	研究方法 (Tags)	阅读状态
Deep Learning in Genomics	Smith (2022)	CNN, 基因测序	已精读 已精读
Attention is all you need	Vaswani (2017)	Transformer, NLP	阅读中 阅读中

最强大的是，每一行点开，里面还能用 Markdown 写下万字详细的阅读笔记。

实战场景 2：个人任务与课程管理

搭建个人主页大盘

将混乱的大学生活高度结构化、项目化。

左侧：每周课程表与常规事务。

中间：各科作业的 Deadline (结合日历视图强提醒)。

右侧：社团、竞赛等课外活动安排。

告别大脑过载

将所有必须做的事情放进系统。

释放大脑内存，专注于“执行”而非“记忆”。

掌控感是缓解学业焦虑的唯一良药。

如何用好知识管理工具？



1. 收集与加工

不要做无脑的“搬运工”，复制粘贴毫无意义。必须用自己的话重述重点（使用 Markdown）。



2. 整理与连接

定期回顾（如每周日），给笔记打上标签体系，主动建立不同学科笔记之间的网状关联。



3. 输出与创造

切记：知识库的最终目的，是帮助你更高效地写出自己的高质量论文、科研报告或博客。

替代方案与补充：国内文档生态

Notion 的局限性

服务器部署在国外，部分网络环境下加载图片缓慢；且官方界面目前主要为纯英文，有一定上手门槛。

国内平替优质选项

腾讯文档 / 语雀 / 飞书。

腾讯文档绝对优势：团队协作属性无敌，微信生态完美打通，极其适合几个同学快速组建课程大作业的“共享资料库”。

核心工具定位区分

工具没有绝对的好坏，关键在于是否匹配你的应用场景。



Word

最终定稿交付（如毕业论文打印提交前的最终排版格式微调）。



腾讯文档/飞书

多人实时在线编辑、高频沟通与快速收集信息的协作场景。



Notion

个人深度思考、长期的知识沉淀与庞大体系化结构的搭建。

警惕“差生文具多”



警惕工具沉迷陷阱：

花整整3天时间去网上搜罗极度复杂的 Notion 模板，配置好看的页面封面图和动态图标，最后却只花了10分钟去写真正的笔记内容。

核心法则：大道至简

工具越简单越好。如果你觉得 Notion 太复杂影响了思考，哪怕只在本地电脑建立一个普通的文件夹，里面装满你手敲的 Markdown (.md) 纯文本文件，只要坚持记录，也绝对足够支撑你读到博士毕业。

小结：让时间成为你的朋友



大一开始建立规范的数字知识库，初期看似繁琐费事。但当你在大三写综述、大四写毕业论文、准备保研面试时，你将拥有一座别人无法企及、随时调用的私人智慧金矿。

总结：科研效率的“黄金三角”

这三者共同构成了数字时代的科研基础素养：



1. Markdown 掌控内容

实现排版与内容分离，极大提升个体的日常书写效率与文本质量。



2. GitHub 掌控历史

建立严密的版本控制思维，培养溯源防灾与拥抱开源共享的极客习惯。



3. Notion 掌控体系

告别孤岛与碎片化，构建第二大脑，沉淀个人学术智慧的复利。

进阶学习资源推荐



文档与手册

Markdown 官方教程中文版
(随时查阅的语法字典)



视频教程

B站搜索: 《GitHub 入门与实践》
B站/少数派: Notion 基础操作指南
与大学生专属模板分享。



底层逻辑必读书目

《卡片笔记笔记法》
(Sönke Ahrens 著, 彻底重塑
你的知识管理底层思维逻辑)

课后作业4

- 题目

- 任务一 (Markdown):** 使用腾讯文档, 用 Markdown 语法重写你最近的一份实验报告 (需包含标题、加粗、列表、图片、公式)。记录:

1. 原始Markdown文本。
2. 对应渲染后的内容。

- 任务二 (GitHub):** 注册一个 GitHub 账号, 探索界面。记录如下实验流程:

1. 你的Github主页。
2. 搜索你专业相关的英文关键字。
3. 给一个开源项目点亮 Star。

- 提交格式及方式:

一个Word文件, 发送助教邮箱: bochengli03@stu.xmu.edu.cn

- 截止日期: 2026年3月30日 (周一) 11:59 PM

Q & A

任何问题都可以问，因为这里没有标准答案。

Thank You