科研素养训练 I

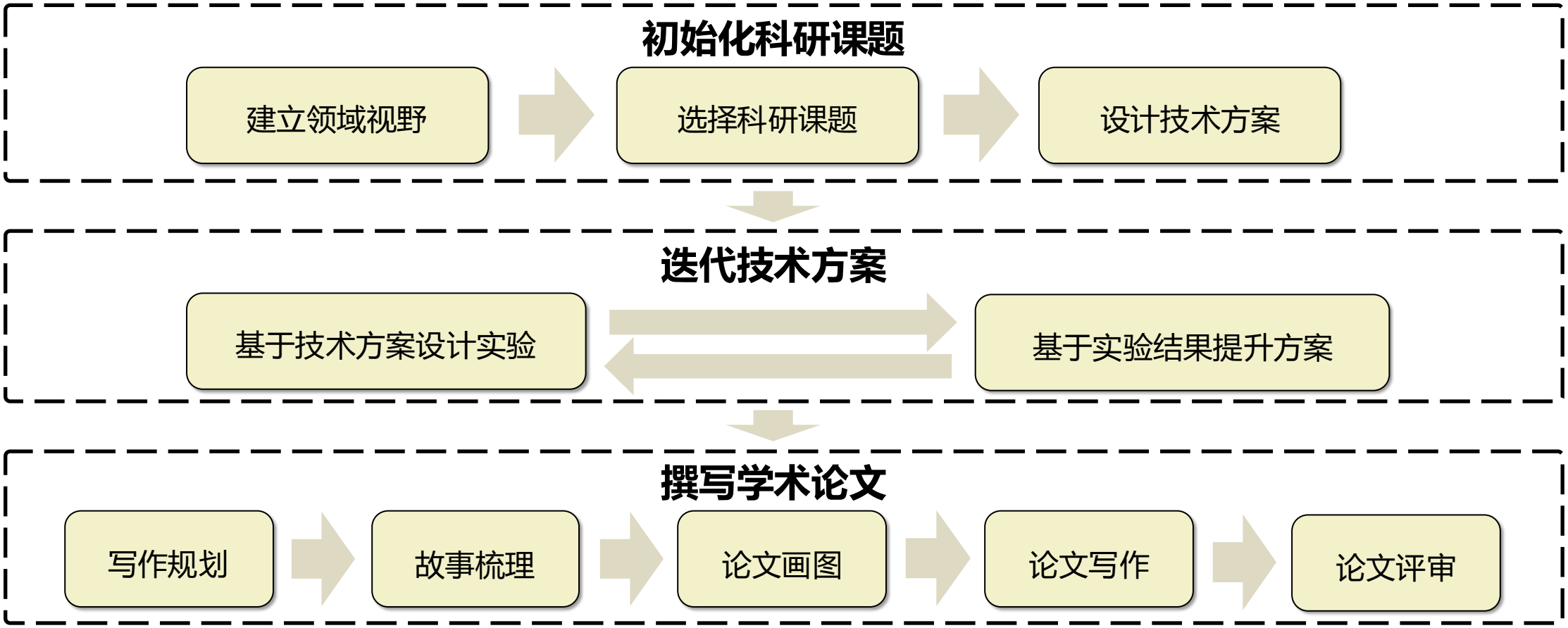
5. 科研选题

宋庆禹 厦门大学

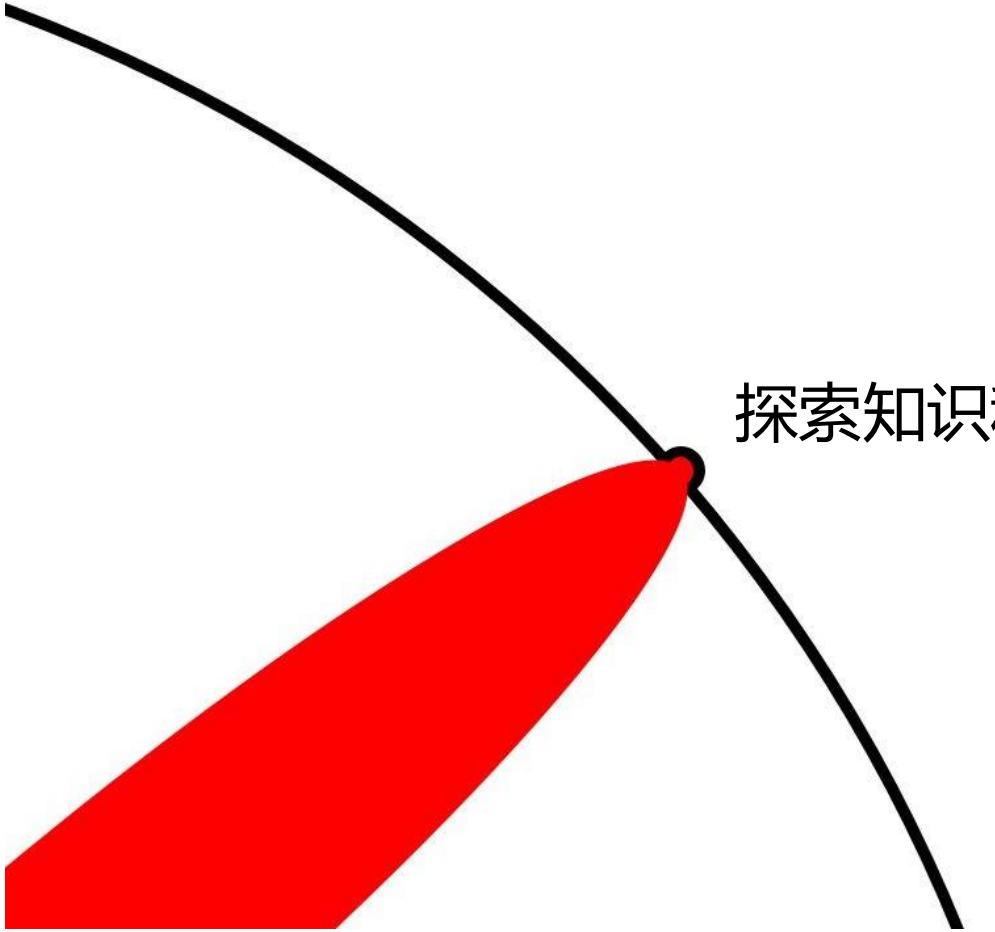
0330, 2026

Credit to GAMES 003 @ ZJU

科研的常规流程


初始化科研课题
建立领域视野
选择科研课题
设计技术方案
迭代技术方案
基于技术方案设计实验
基于实验结果提升方案
撰写学术论文
写作规划
故事梳理
论文画图
论文写作
论文评审

科研是什么？

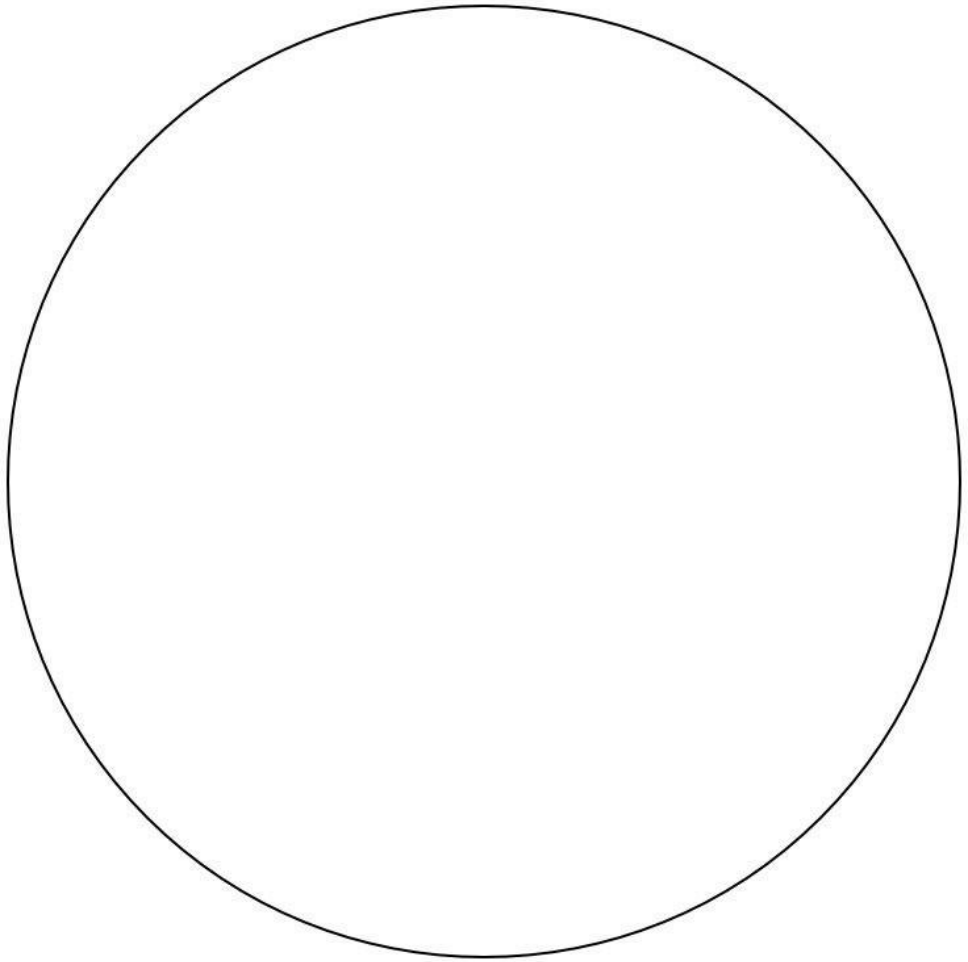


探索知识和技术的边界

“The illustrated guide to a Ph.D.” by Matt Might

科研是什么?

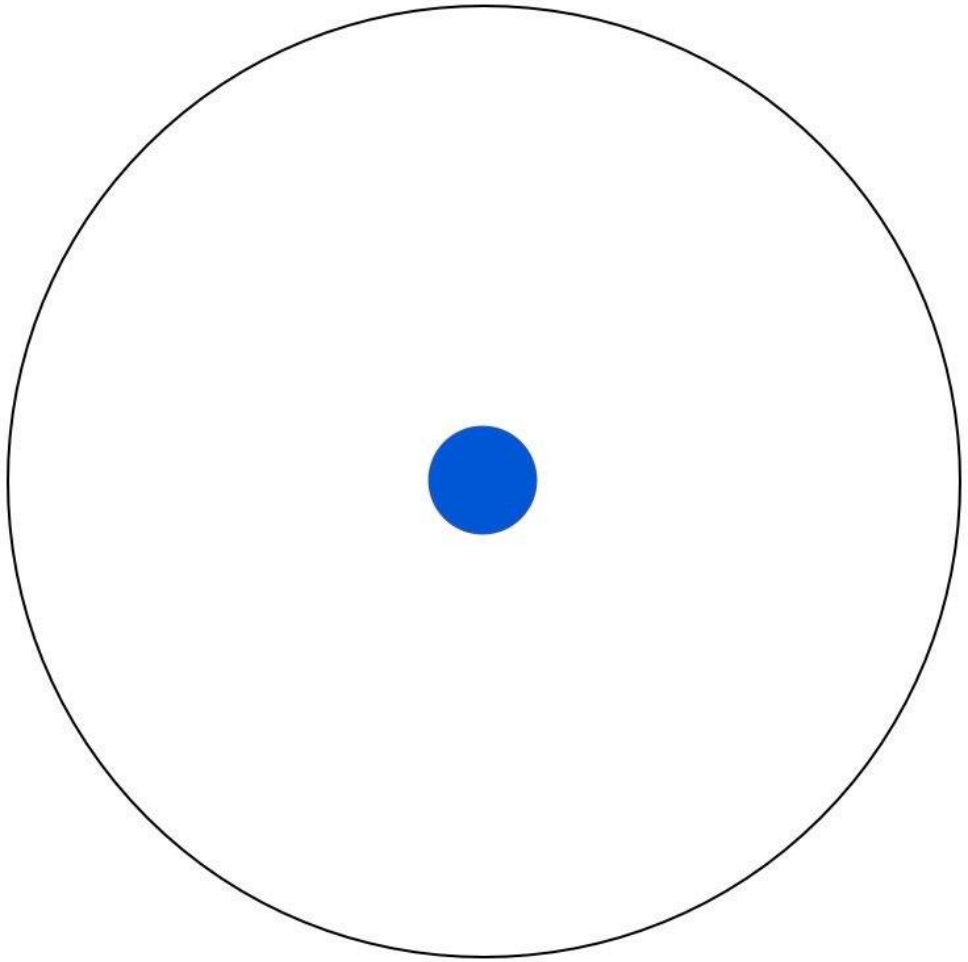
Imagine a circle that contains all of human knowledge:



"The illustrated guide to a Ph.D." by Matt Might

科研是什么?

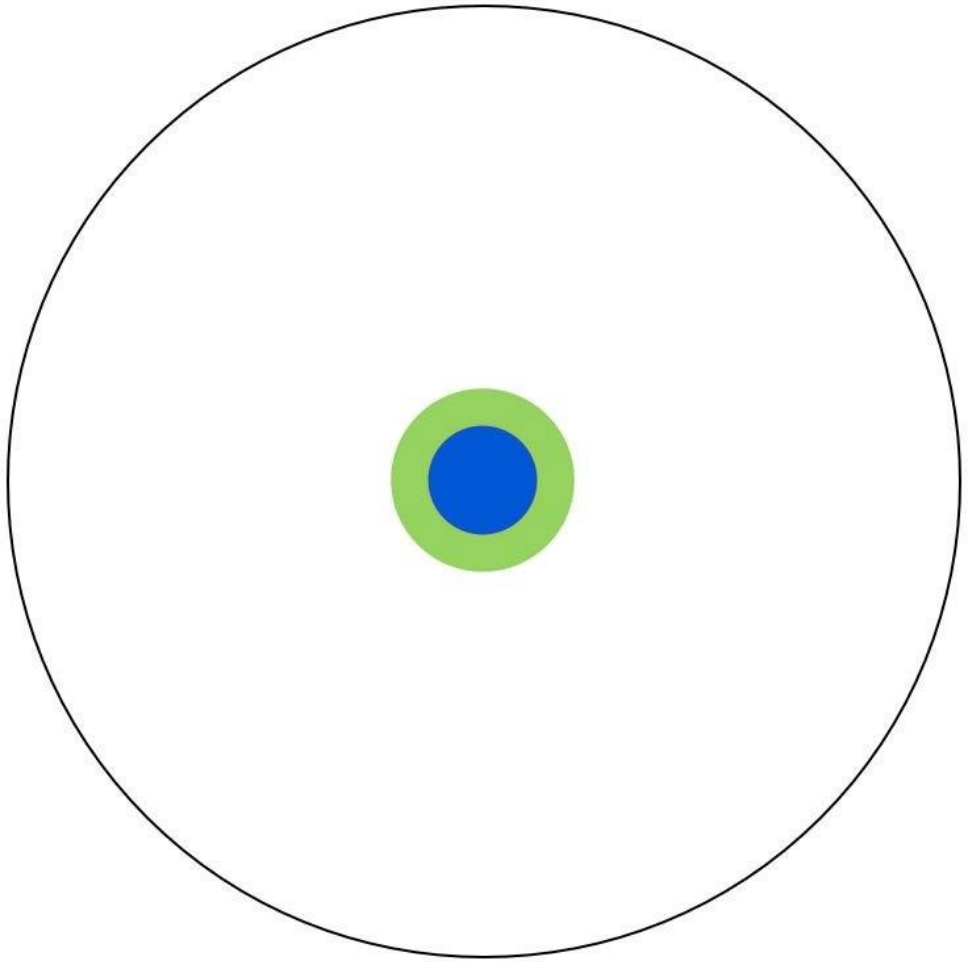
By the time you finish elementary school, you know a little:



"The illustrated guide to a Ph.D." by Matt Might

科研是什么?

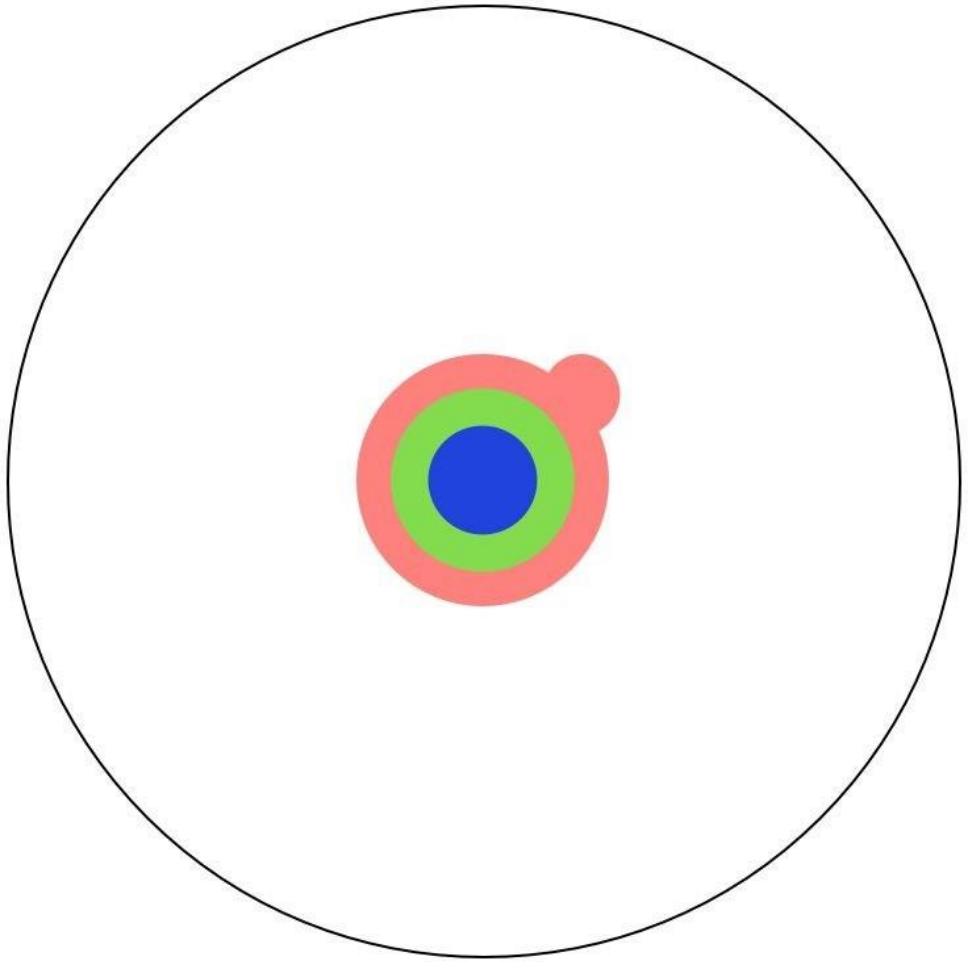
By the time you finish high school, you know a bit more:



"The illustrated guide to a Ph.D." by Matt Might

科研是什么?

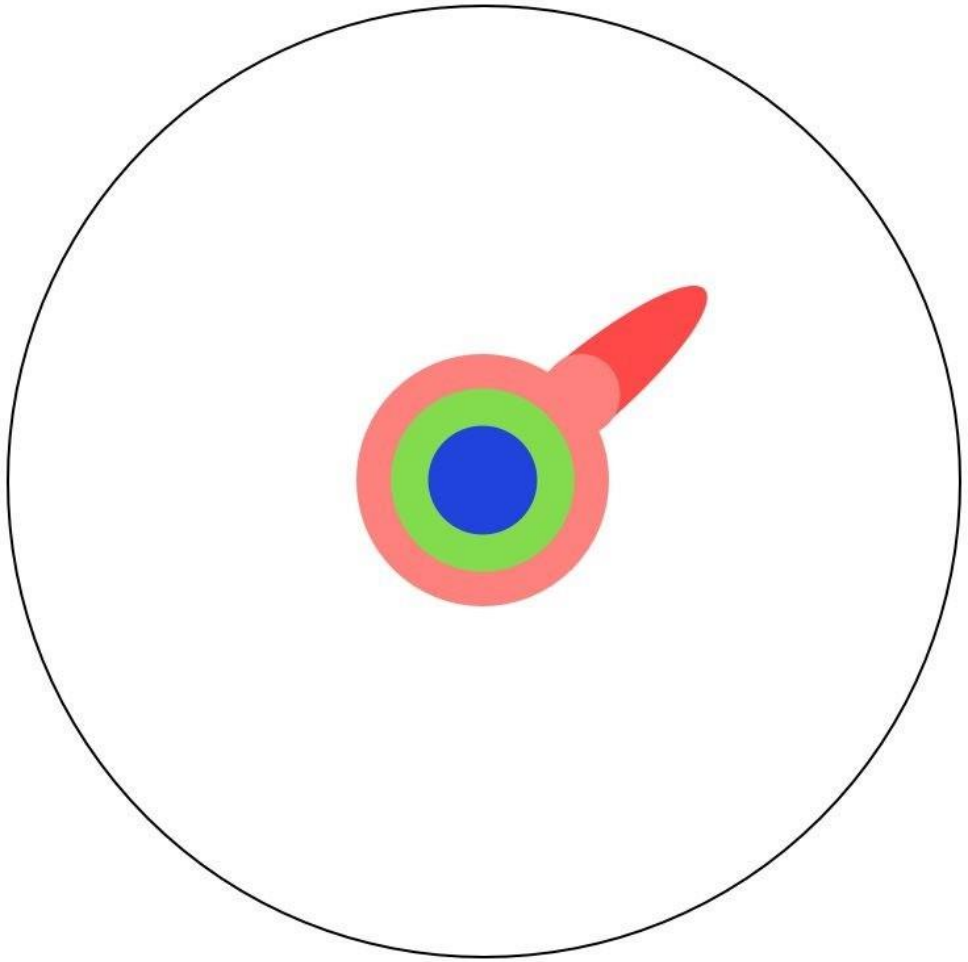
With a bachelor's degree, you gain a specialty:



"The illustrated guide to a Ph.D." by Matt Might

科研是什么?

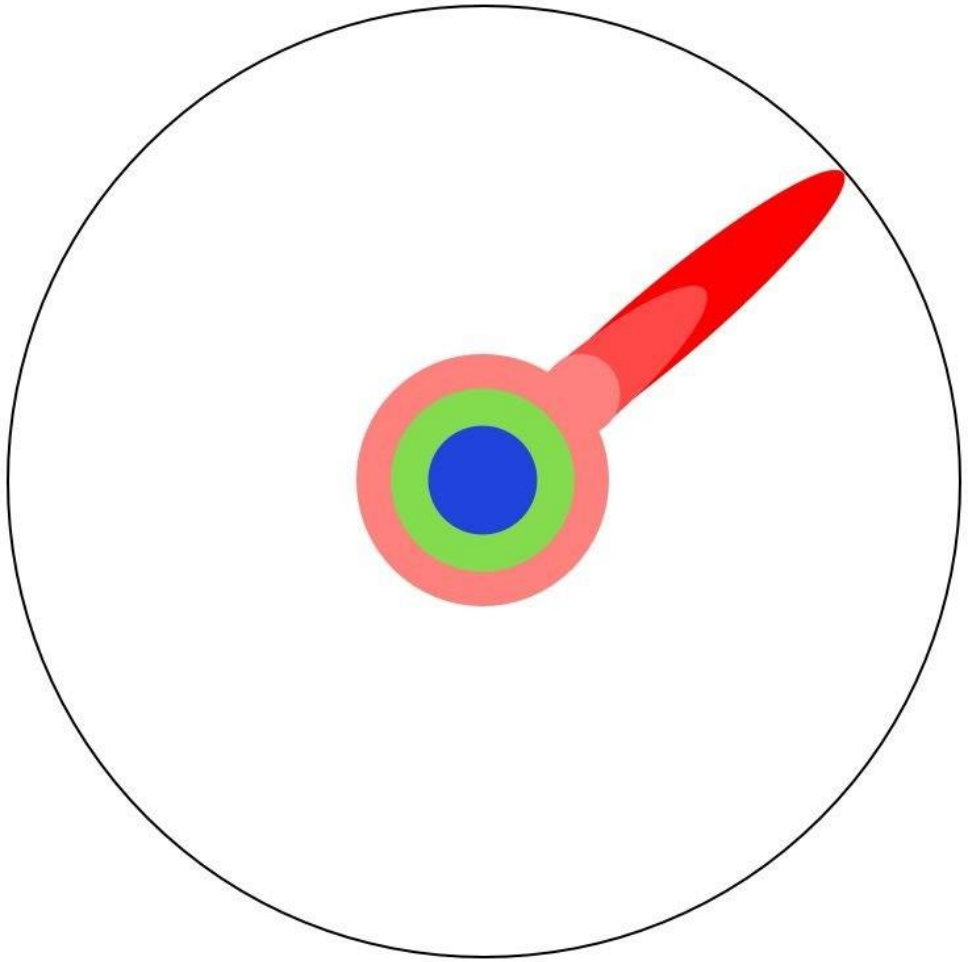
A master's degree deepens that specialty:



"The illustrated guide to a Ph.D." by Matt Might

科研是什么?

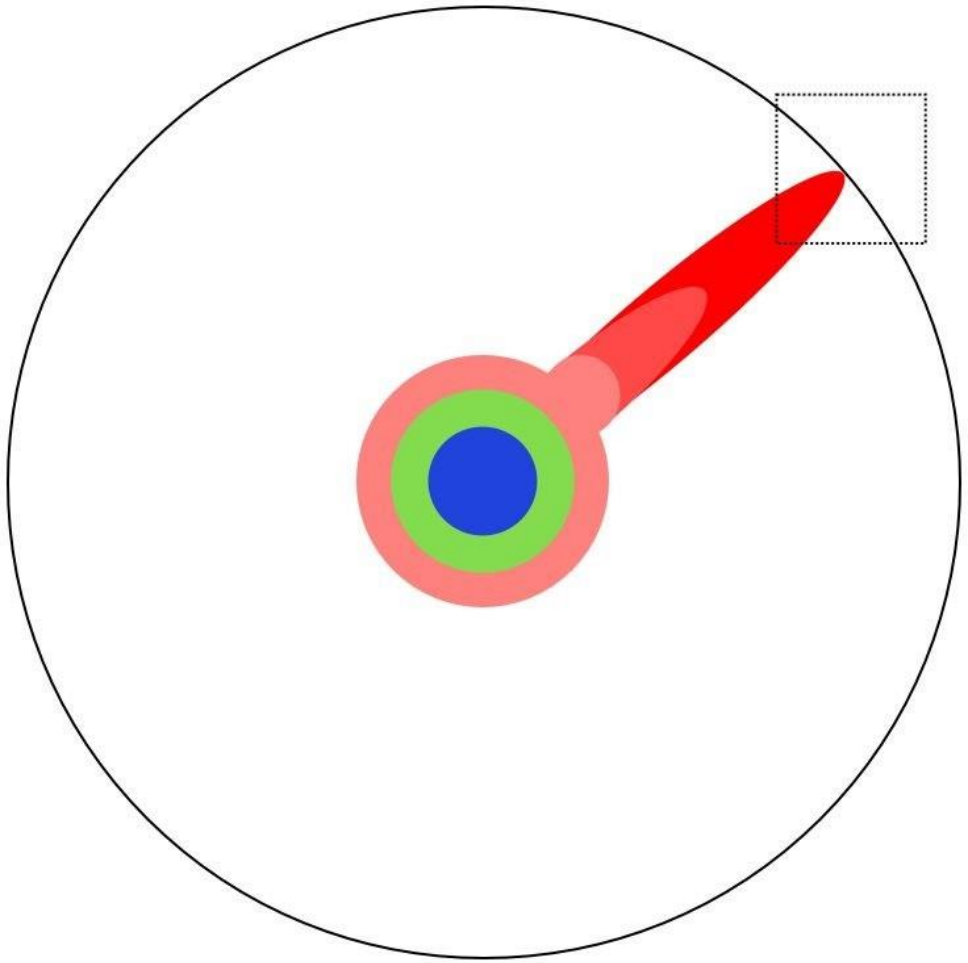
Reading research papers takes you to the edge of human knowledge:



"The illustrated guide to a Ph.D." by Matt Might

科研是什么?

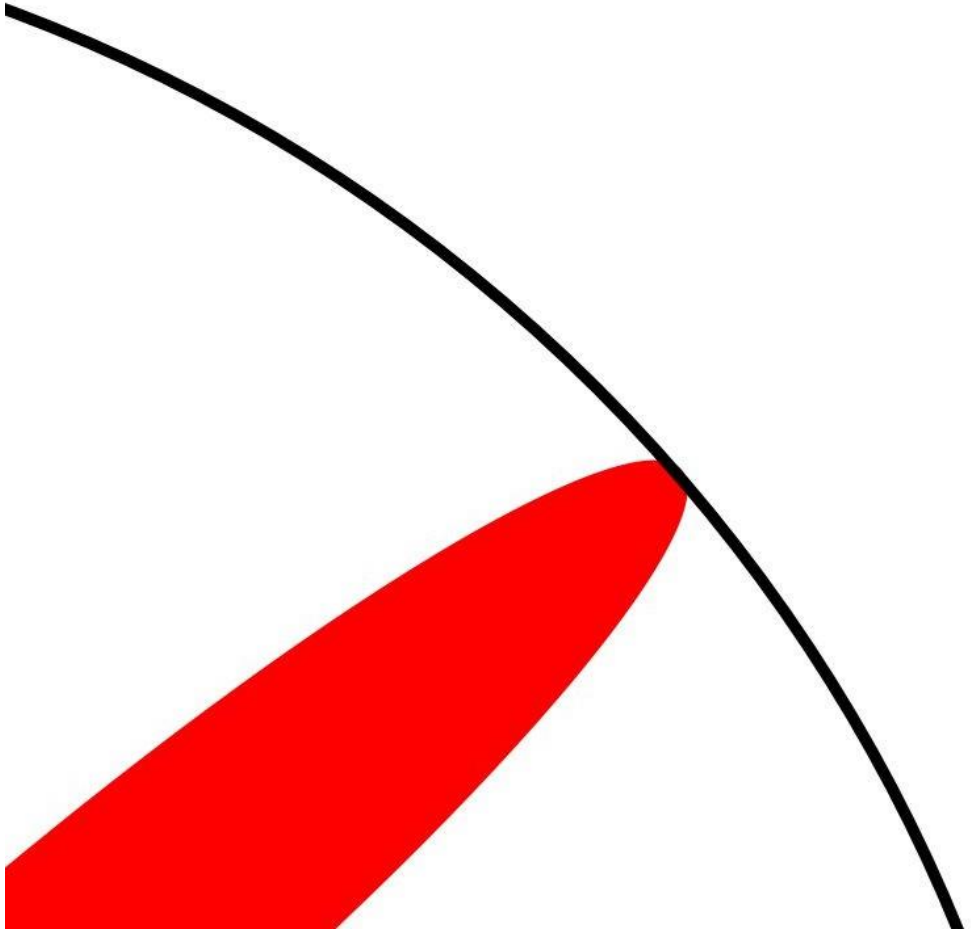
Once you're at the boundary, you focus:



"The illustrated guide to a Ph.D." by Matt Might

科研是什么?

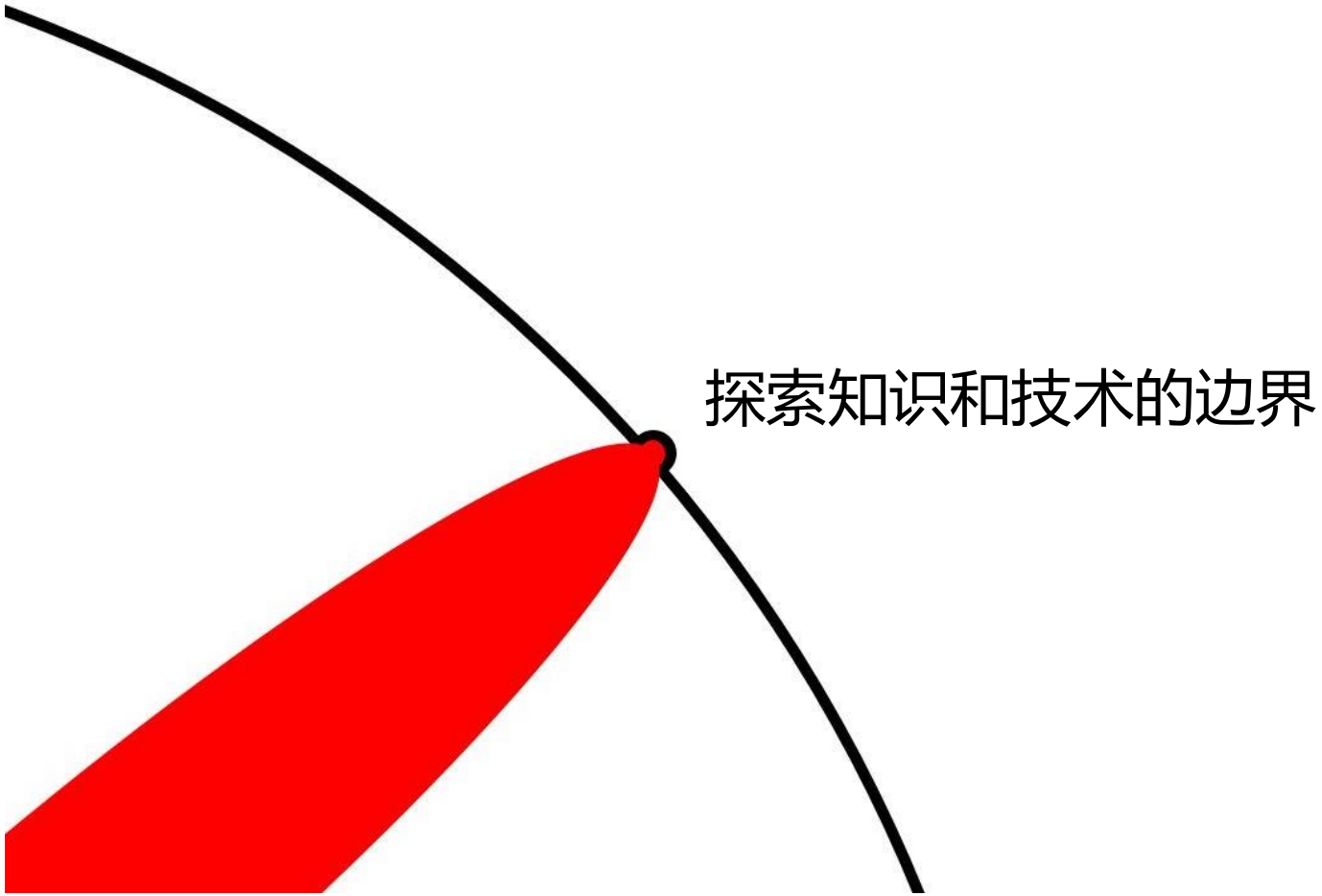
You push at the boundary:



"The illustrated guide to a Ph.D." by Matt Might

科研是什么?

Until one day, the boundary gives way:



“The illustrated guide to a Ph.D.” by Matt Might

为什么选题很重要？

杨振宁讲“学术成功的关键因素”

为什么选题很难？

- 选择课题需要从“已知”推测“未知”，伴随着很大的不确定性：

1. 我选择的课题可能是一条“死路”

  - **花了半年基于某个特定假**设推导算法，最后发现基础数学模型存在悖论；
  - **或者使用的数据集存在**严重的数据泄露，导致之前**的好**结果全是假象。

本人例子：MathL2OProof Project，发现基于Induction的证明思路受限于没法写出NAG算法，**没法开展**

解决思路：换成简化版的Gradient Descent

- **拥抱不确定性：建立正确的心理预期（open-minded）**
- **降低不确定性：不断更新、加深自己对问题的认识**

为什么选题很难？

- 选择课题需要从“已知”推测“未知”，伴随着很大的不确定性：
  2. 我选择的课题可能有很多“分叉路”（一开始想做A，最后做成了B）

本人例子：Learning-Only L2O MIMO Project中，一开始想用强化学习，最后选择了L2O

IEEE INFOCOM
IEEE International Conference on Computer Communications
20–23 May 2024 // Vancouver, Canada

- **拥抱不确定性：建立正确的心理预期（open-minded）**
- **降低不确定性：不断更新、加深自己对问题的认识**

为什么选题很难？

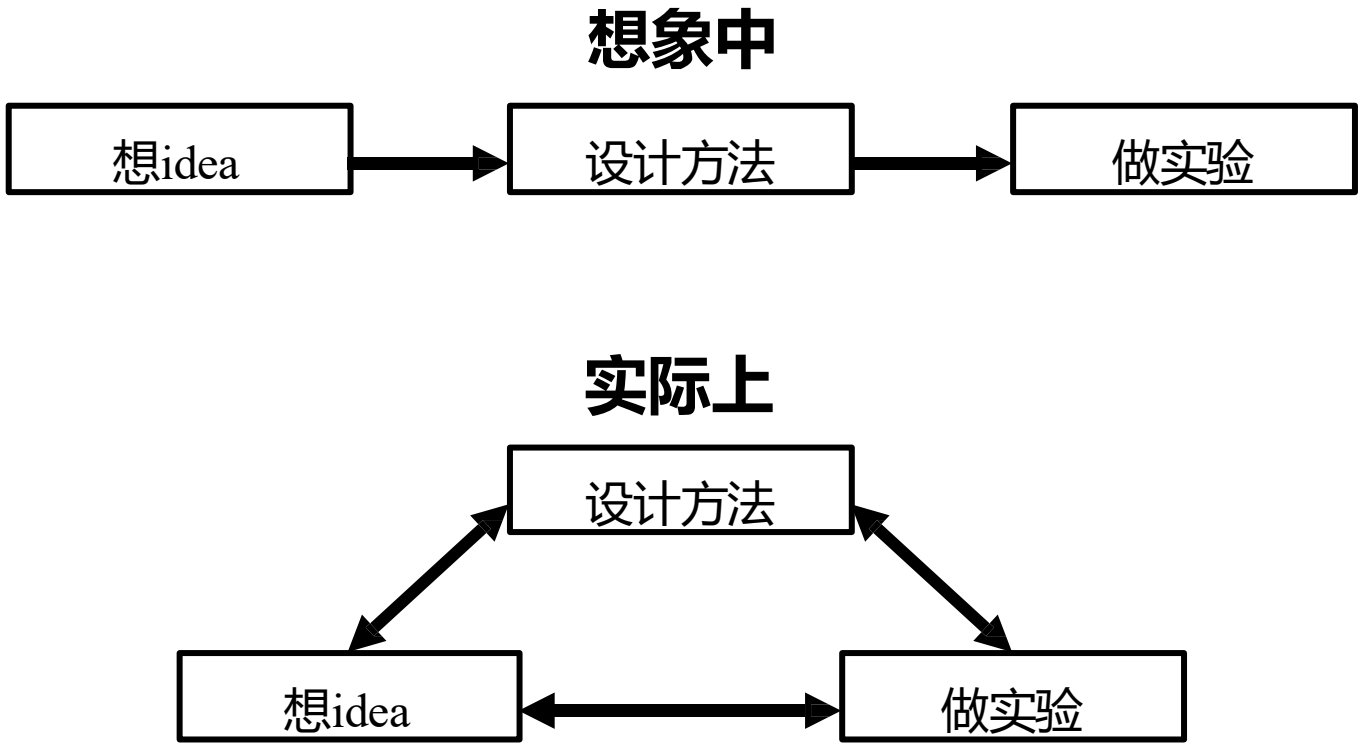
- 选择课题需要从“已知”推测“未知”，伴随着很大的不确定性：
  3. 其他人可能在同一条路上，可能先达到终点
  4. 整个道路的布局可能发生巨大的变化(出现新的技术突破)

  例如，LLM的出现颠覆了传统NLP的很多领域：机器翻译

- **拥抱不确定性：建立正确的心理预期（open-minded）**
- **降低不确定性：不断更新、加深自己对问题的认识**

寻找科研课题本身也是科研

- 它是科研最关键的环节之一
- 它常常会贯穿一个项目的始终



课题选择的关键因素

课题选择非常因人而异

- 自身情况
  - 自己的兴趣和背景
  - 自己所处的阶段（刚入门/有一定经验？）
  - 自己的目标（发表一篇顶会/深耕一个问题）
- 大环境
  - 领域的发展阶段
  - 领域的活跃程度
  - 领域的发展潜力
- 小环境
  - 导师的风格(自由探索/给特定方向？)
  - 实验室的情况（在某方向有一定积累/起步阶段？）
  - 合作者情况（有师兄师姐带/独立探索？）

选择课题的关键因素

- 兴趣/喜好 – 决定了做得开不开心
- 可行性 – 决定了能不能做出来
- 影响力 – 决定了做出来了关注度有多高

选择课题的关键因素

- 兴趣/喜好 – 决定了做得开不开心
- 可行性 – 决定了能不能做出来
- 影响力 – 决定了做出来了关注度有多高

兴趣与热情是好的科研不可或缺的元素

- 科研没有高低贵贱之分
- 兴趣很大程度上也决定了能不能做出来，以及做出来的影响力有多大
- 两种不同状态：
  - 当做一个自己兴趣一般的项目时：当一天和尚撞一天钟
  - 当做自己兴趣浓厚的项目时：日思夜想，不自觉花很多时间在上面

如何找到研究兴趣和热情?

- 首先，要了解自己
- 了解自己擅长什么
  - 因为擅长所以喜欢
  - 因为喜欢所以擅长
- 了解自己对什么风格的问题感兴趣
  - 有的人追求实用性：
    做有重要实际应用的问题；提升效果和速度
  - 有的人追求新颖和有趣：
    做别人没做过的问题；酷炫的效果
  - 有的人喜欢数学严谨性：
    喜欢能用数学解释的问题
  - 通过阅读论文和学术交流，可以大致判断自己对哪类风格的问题兴趣

如何找到研究兴趣和热情?

- 浓厚的兴趣建立在对一个问题有过亲自探索、产生了深刻理解的基础上
  - 例子：Learning to Optimize这个topic是我在做完一个无线通信领域Project过程中发现的
- 多与他人交流，了解大家对什么感兴趣

- 时而没有热情是十分正常的（不是必需品）
  - 找到一个自己很感兴趣的课题是一件很幸运的事情
  - 重要的是一直保持好奇心和求知欲

选择课题的关键因素

- 兴趣/喜好 – 决定了做得开不开心
- 可行性 – 决定了能不能做出来
- 影响力 – 决定了做出来了关注度有多高

可行性的影响因素

- **课题的研究空间/难度/竞争程度**

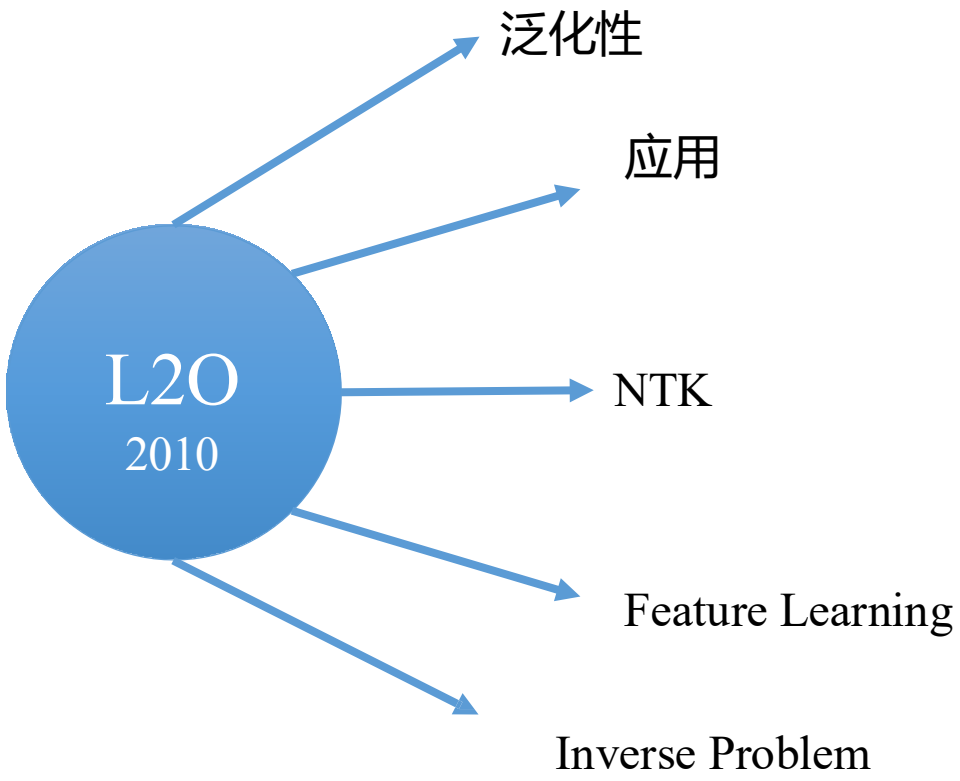
  示例：
  - 研究空间：这是一个新兴领域（如Neural Rendering），尚有大量未解决的基础问题。
  - 难度：该算法需要极高的数学推导能力，超出了我目前的水平。
  - 竞争程度：许多顶尖实验室都在研究此课题，抢发论文的风险很高。
- **我的背景/能力/阶段**

  示例：
  - 背景：我是计算机视觉（CV）背景，但此课题主要涉及系统工程。
  - 能力：我熟悉底层代码实现，可以快速复现基线方法。
  - 阶段：我是一年级博士生，可以探索具有高风险、长周期的课题。
- **我能获取到的资源和支持（计算资源/合作者/指导）**

  示例：
  - 计算资源：此课题需要训练超大模型（如GPT-4级），但我只有两块1080Ti。
  - 合作者：我可以和一位擅长强化学习（RL）的同学合作，优势互补。
  - 指导：我的导师非常支持这个新方向，并能引荐相关领域的专家。

可行性——课题的研究空间

- 新兴的领域内课题研究空间通常较大，越成熟的领域和问题研究空间通常较小



- 衡量方法：针对一个方向试想一下，看自己能想到多少个不同的研究方法/角度

可行性——课题的难度

- 课题的难度并不等同于问题本身的难度

- 难度 = 相较于前人的工作做出更好的效果的难度
  - 越成熟的领域难度越高（例：图像分割）
- 难度 = 相较于前人的工作挖掘新的角度的难度
  - 即使领域不是很成熟，也有可能大家探索的角度已经比较全面了
- 难度是相对的，我们可以在做项目的时候调整课题的难度
  - 如果目标太难，可以适当降低难度（简化问题设定）
  - 每个方法都有自己的优缺点，想办法将自己方法的优点发扬光大，调整问题设定

可行性——课题的竞争程度

- 大家越关心的问题竞争通常越高 (例：LLM Agent应用)
- 越容易想到的课题竞争度也通常越高 （例：LLM Agent协作机制）
- 应当尽量避免竞争激烈的问题
  - 但如果有比较独特的角度或者优势（跟自己的背景很相关/手快/合作者给力），可以一试

可行性——我的背景/能力/阶段

- 我编程能力、数理基础怎么样?
- 我对所研究领域是否有充分的了解?
- 我对科研流程的把握能力怎么样?

可行性——我的背景/能力/阶段

- 对于刚入门的同学：
  - 还需锻炼基本的科研技能
  - 尽可能选择门槛比较低的、难度适中、风险较低的课题
  - 目标：培养科研技能、建立科研的信心
  - 建议可以找一篇自己喜欢的、代码维护比较好的论文，深入理解方法，并在此基础上尝试对其进行改进
  - 尽可能选择自己能够得到足够支持和指导的课题

- 对于有一定科研经验的同学：
  - 可以选择深挖某个问题里的难点，尝试解决核心问题
  - 也可以适当探索一些别的方向，丰富自己的技能和知识储备，更有可能产生好的idea

影响可行性的其他因素

- 问题定义的清晰程度：能否用简洁的语言描述我的核心研究问题?
  - 我要解决一个什么问题?
  - 我想的方法为什么能解决这个问题?
  - 解决了带来什么?
- 指导：我的导师能否给我有意义的指导，如果不能，我能否找到其他人来指导我?
- 反馈环：我能否及时得到反馈，还是要等几周/几个月才知道?
- 时间：这个项目所需的时长是否符合我的时间安排?
- 资源：是否有足够的计算资源、数据集支持我想做的课题?
- 灵活性：如果我想到的idea不work，有多少plan-B？有多容易“回收”我已探索的　成果?

选择课题的关键因素

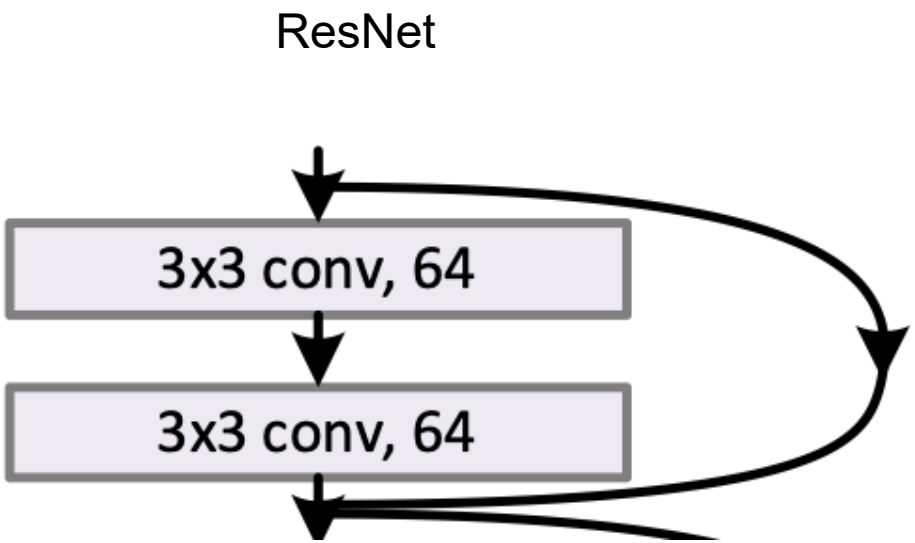
- 兴趣/喜好 – 决定了做得开不开心
- 可行性 – 决定了能不能做出来
- 影响力 – 决定了做出来了关注度有多高

影响力

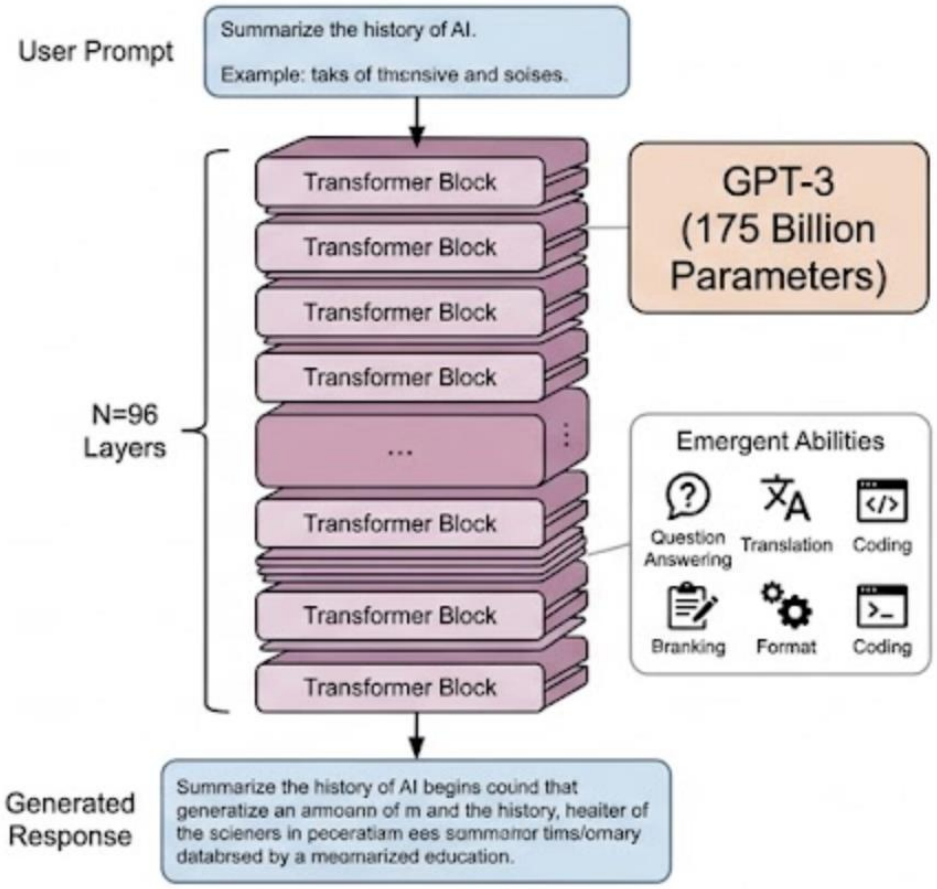
- 工作的影响力 = 一个领域的总体关注度 x 我的工作的显著程度
- 越重要/越普适/越热门的问题关注度越高
- 重要的问题例子：
  - 对应关系（correspondence）："What are the three most important problems in computer vision?" Takeo Kanade: "Correspondence, correspondence, correspondence!"
  - LLM Agent输出决策可靠性
- 相对小众的问题例子：
  - 网络系统领域的代码自动生成
- 思考：哪些人群会对我的课题感兴趣？
  - 生成系统级别的可用代码：系统相关研究领域的人
  - 更省tokens：商业领域的人（省钱）

影响力

- 影响力有不同的表现形式：
  - 很有用：解决核心的问题/挑战
  - 很特别：给大家留下深刻的印象



Deep Residual Learning for Image Recognition. CVPR 2016



Language Models are Few-Shot Learners, NeurIPS 2020

Picture from Gemini

如何扩大自己工作的影响力？

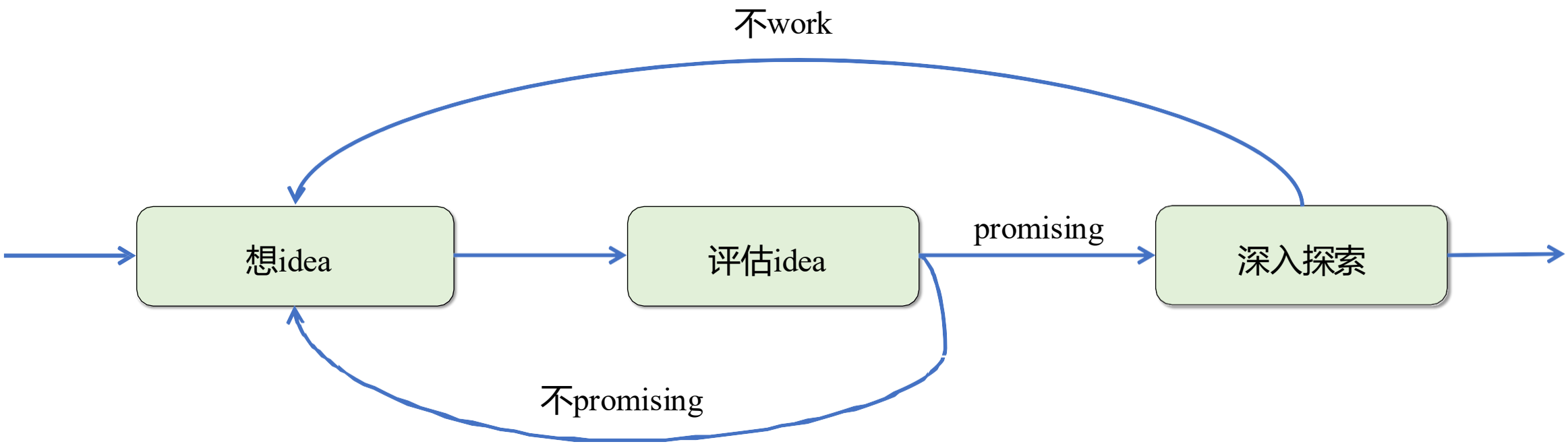
- 根本：提升工作的质量
- 但是更好的呈现也会让人更容易关注到你的工作
  - 提升论文写作
  - 及时开源代码
  - 做好项目主页
  - 更好的可视化来突出自己的工作的优势和特点

如何选择课题？

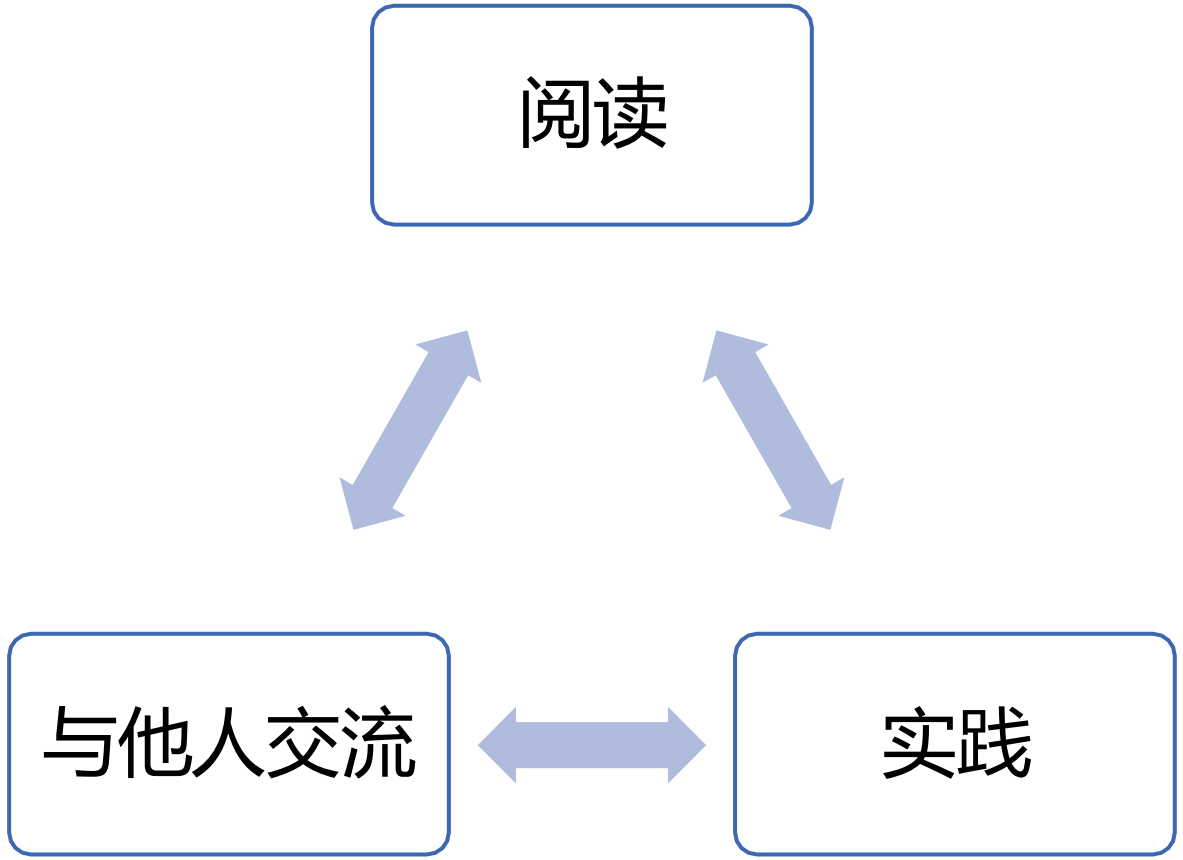
前提：建立对领域的认知

- 整理一个领域的发展脉络
  - 找出每一年的关键论文 （citation/阅读related work）
  - 整理出论文之间的联系
  - 总结过去每个阶段大家在重点解决哪些问题
  - 判断当前这个领域还有哪些重要的、可解决的问题

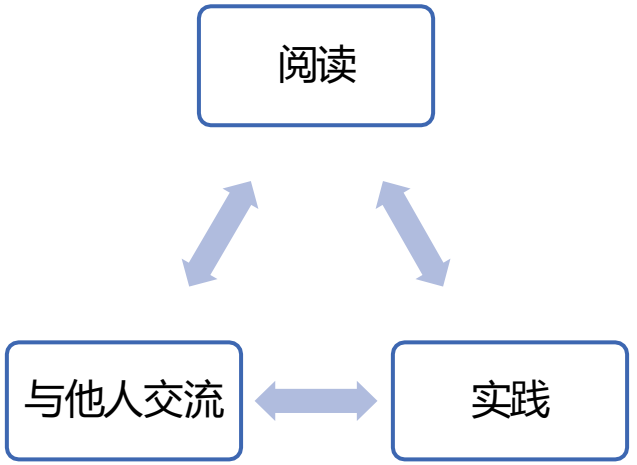
寻找课题是一个循环往复的过程



想idea的过程



想idea——阅读



- 对待一篇具体的工作，带着问题去阅读：
  - 这个工作有哪些不足的地方？
  - 这个工作开启了哪些新的可能性？
  - 这个工作跟我想解决的问题有什么关系？
  - 误区：完全关注在一个工作不足的地方；完全相信一个论文的结论

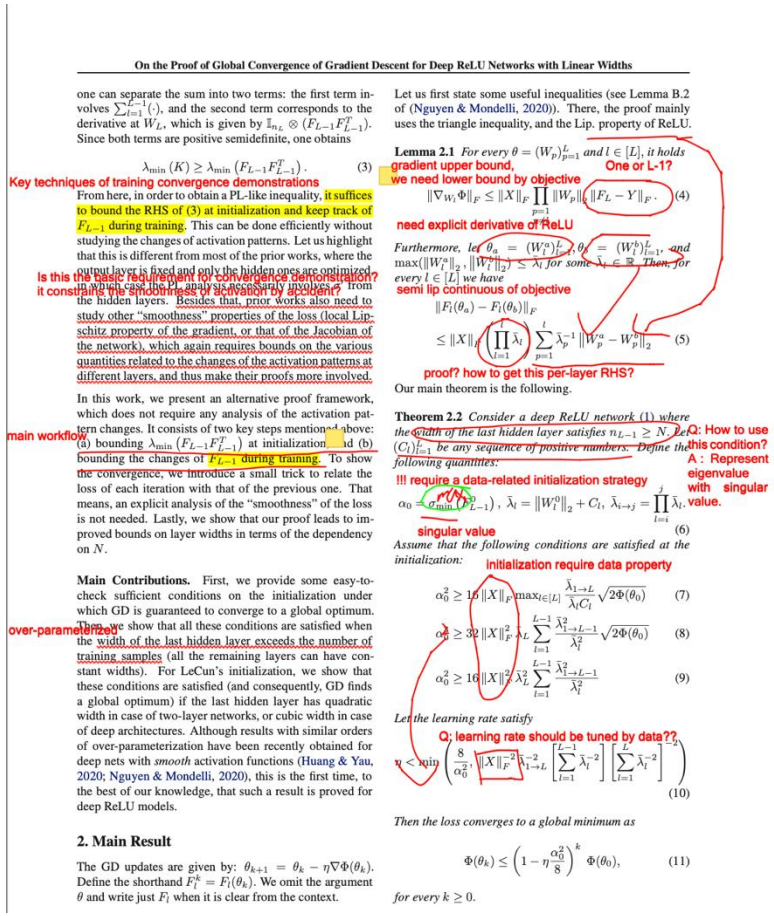
- 广泛阅读：
  - 横向比较，整理脉络，识别缺口

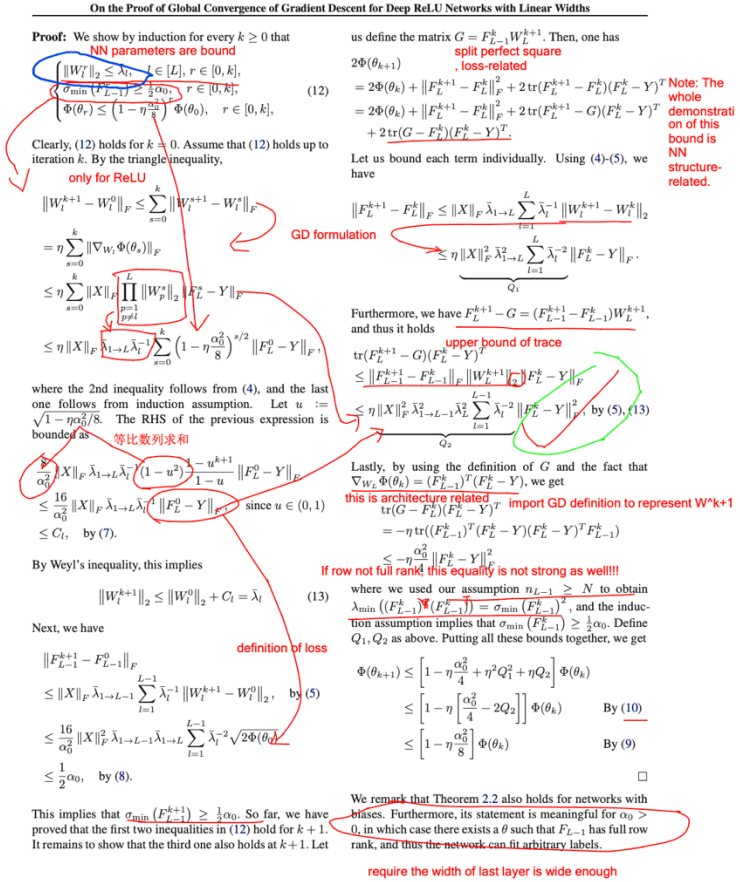
- 阅读经典论文：
  - Alyosha Efros: Read old papers. History does not repeat, but it rhymes

历史不会重演，但总是惊人的相似

阅读的个人案例

- MathL2OProof 期间，读到一篇ICML 2021年的paper，浅层MLP解二次规划问题可以通过拆解完全平方公式的方式证明





阅读的个人案例

- GoMathL2O brainstorm期间，阅读了UT Austin Atlas Wang ICML, 2023的paper，L2O的必要条件，在不同case都被证明了，但是泛化性没人讨论

Towards Constituting Mathematical Structures for Learning to Optimize
ICML 2023

Towards Constituting Mathematical Structures for Learning to Optimize

Jialin Liu [*1] Xiaohan Chen [*1] Zhangyang Wang [2] Wotao Yin [1] HanQin Cai [3]

Abstract

Learning to Optimize (L2O), a technique that utilizes machine learning to learn an optimization algorithm automatically from data, has gained arising attention in recent years. A generic L2O approach parameterizes the iterative update rule and learns the update direction as a black-box network. While the generic approach is widely applicable, the learned model can overfit and may not generalize well to out-of-distribution test sets. In this paper, we derive the basic mathematical conditions that successful update rules commonly satisfy. Consequently, we propose a novel L2O model with a mathematics-inspired structure that is broadly applicable and generalized well to out-of-distribution problems. Numerical simulations validate our theoretical findings and demonstrate the superior empirical performance of the proposed L2O model.

1. Introduction

Solving mathematical problems with the help of artificial intelligence, particularly machine learning techniques, has gained increasing interest recently (Davies et al., 2021; Charton, 2021; Polu et al., 2022; Drori et al., 2021). Optimization problems, a type of math problem that finds a point with minimal objective function value in a given space, can also be solved with machine learning models (Gregor & LeCun, 2010; Andrychowicz et al., 2016; Chen et al., 2021a; Bengio et al., 2021). Such technique is coined as *Learning to Optimize (L2O)*.

As an example, we consider an unconstrained optimization problem $\min_{x\in\mathbb{R}^n} F(x)$ where F is differentiable. A classic algorithm to solve this problem is *gradient descent*:

$$x_{k+1} = x_k - \alpha_k \nabla F(x_k), \quad k = 0, 1, 2, \cdots,$$

where the estimate of x is updated in an iterative manner, $\alpha_k > 0$ is a positive scalar named as step size, and the update direction $\alpha_k \nabla F(x_k)$ is aligned with the gradient of F at x_k. Instead of the vanilla gradient descent, (Andrychowicz et al., 2016) proposes to parameterize the update rule into a learnable model that suggests the update directions by taking the current estimate and the gradient of F as inputs

$$x_{k+1} = x_k - d_k(x_k, \nabla F(x_k); \phi), \ k = 0, 1, \cdots, K-1, \quad (1)$$

where ϕ is the learnable parameter that can be trained by minimizing a loss function:

$$\min_{\phi} \mathcal{L}(\phi) := \mathbb{E}_{F\in\mathcal{F}}\Big[\sum_{k=1}^{K} w_k F(x_k)\Big], \quad (2)$$

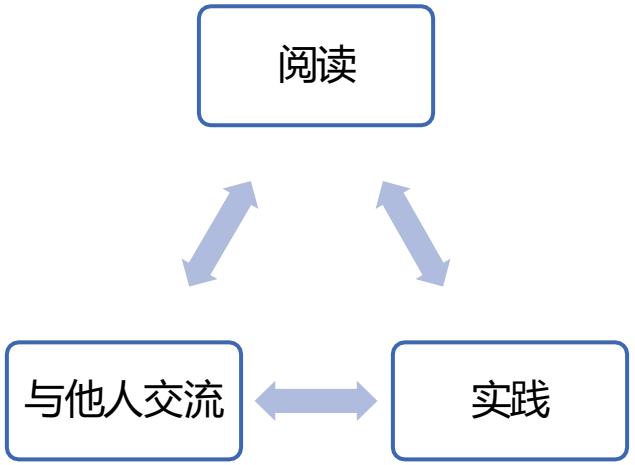
where $\mathcal{F}$ is the problem set we concern and $\{w_k\}_{k=1}^K$ is a set of hand-tuned weighting coefficients. Such loss function aims at finding an update rule of x_k such that the objective values $\{F(x_k)\}$ are as small as possible for all $F \in \mathcal{F}$. This work and its following works (Lv et al., 2017; Wichrowska et al., 2017; Wu et al., 2018; Metz et al., 2019; Chen et al., 2020; Shen et al., 2021; Harrison et al., 2022) show that modeling d_k with a deep neural network and learning a good update rule from data is doable. To train such models, they randomly pick some training samples from $\mathcal{F}$ and build estimates of the loss function defined in (2). Such learned rules are able to generalize to unseen instances from $\mathcal{F}$, i.e., the problems similar to the training samples. This method is quite generic and we can use it as long as we can access the gradient or subgradient of F. For simplicity, we name the method in (1) as *generic L2O*.

Generic L2O is flexible and applicable to a broad class of problems. However, generalizing the learned update rules to out-of-distribution testing problems is quite challenging and a totally free d_k usually leads to overfitting (Metz et al., 2020; 2022). In this paper, we propose an approach to explicitly regularize the update rule d_k. Our motivation comes from some common properties that basic optimization algorithms should satisfy. For example, if an iterate x_k reaches one of the minimizers of the objective $F(x)$, the next iterate x_{k+1} should be fixed. Such condition is satisfied by many

[*]Equal contribution [1]Alibaba Group (U.S.) Inc, Bellevue, WA, USA [2]Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA [3]Department of Statistics and Data Science and Department of Computer Science, University of Central Florida, Orlando, FL, USA. Correspondence to: HanQin Cai <hqcai@ucf.edu>.

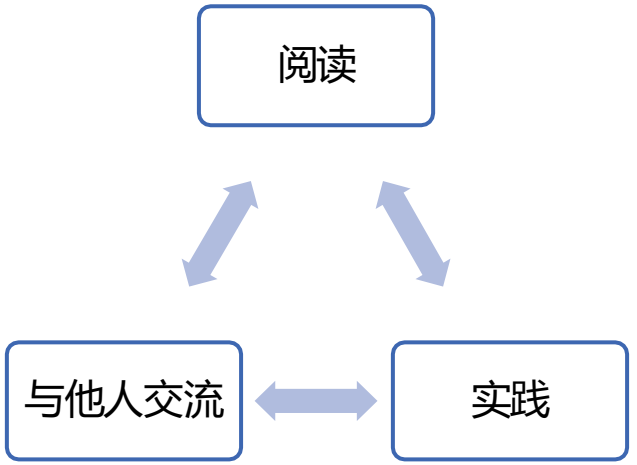
Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

想idea——与他人交流



- 寻找跟自己研究兴趣相同/技能互补的同学一起讨论idea
- 跟老师/资深的研究者交流 – 更有宏观视野
- 无目的的闲聊会有意外的收获
  - 帮助你了解领域动态
  - 可能诞生research ideas

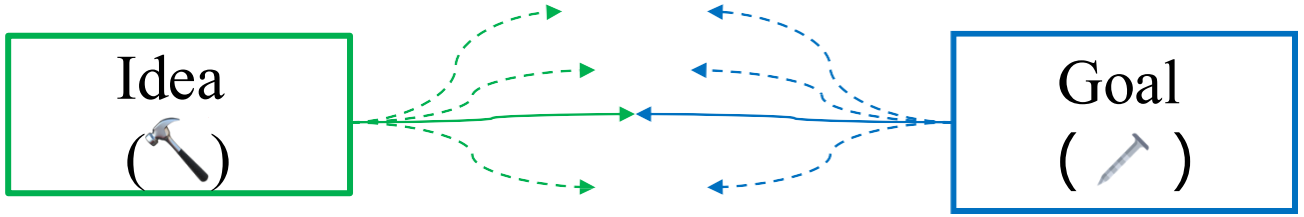
想idea——实践



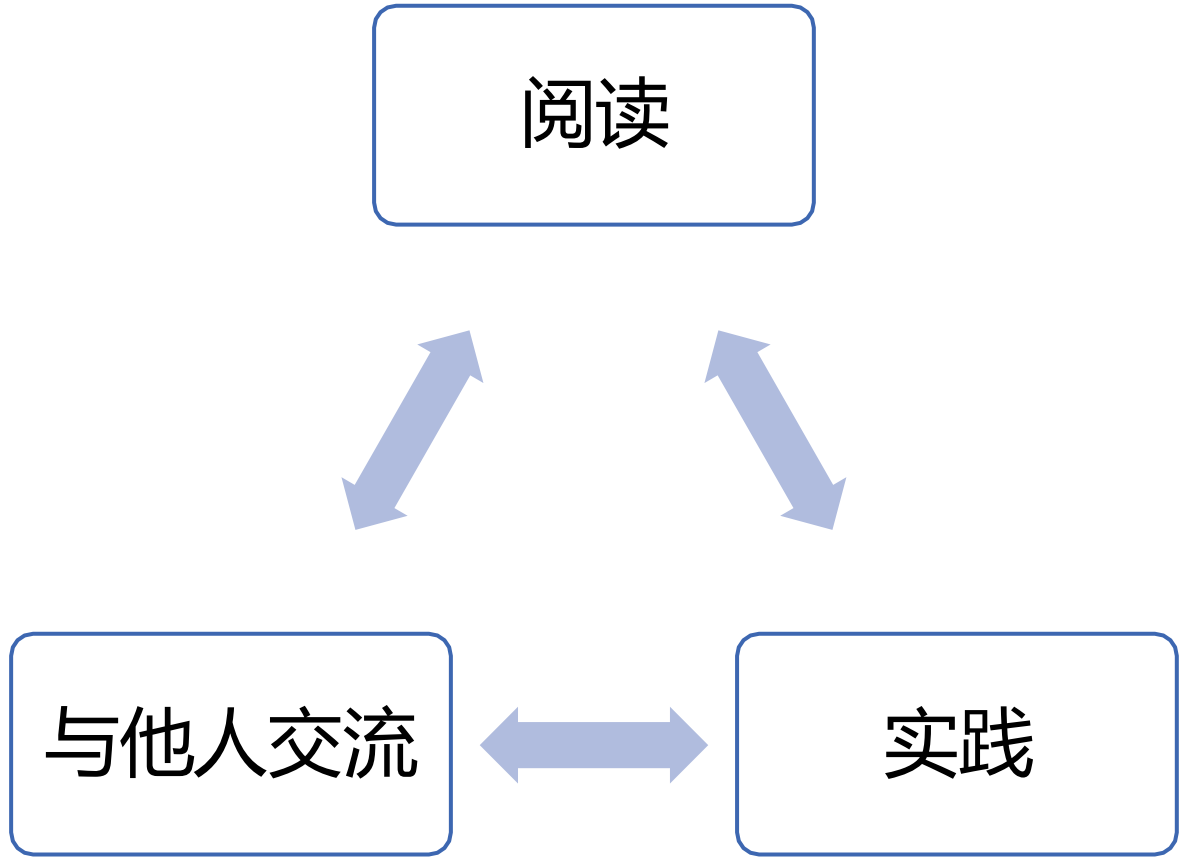
- 上手跑代码，深入理解一个方法的机理
- 理解它的不足、可以提升的地方
- 提出新的解决方案来提升效果

想idea的两种方式

- Goal driven：
  - 有目标，寻找实现目标的方法
- Idea driven：
  - 想到一个新的idea，看效果/看能解决什么问题



评估idea的过程



评估idea的标准

新颖性

可行性

影响力

评估idea——novelty (beauty)

- 越令人意想不到的，就越让人耳目一新
- 越简单有效的idea，潜力越大
- 最好的idea让人读后感觉：
  - 这个好make sense，这个问题就应该这么解决！我之前怎么没想到?!

Michael Black. **Novelty in Science** A guide to reviewers.

关于novelty的参考看法

- 能给领域带来新的增益（信息/知识/技术）的工作都是novel的工作
- 所有工作一定程度上都是A+B
  - 因为所有的工作都需要站在巨人的肩膀上，都会有前人工作的影响

- 但不同的是，这个A+B的组合是否是：由你要解决的问题所驱动的一种自然（甚至必然）的解决方案
- 以及，这个A+B的组合是否是大家在类似的情景下已知能work的

培养研究品味(research taste)的练习方法

1. Write down a list of research ideas. Have a mentor you respect rate each idea 1-10. Discuss ideas where you disagree with them after reflection.
2. Pay attention when other people try ideas you've had. How did the results compare with your expectations?
3. Interview researchers around you on their taste. Why do they work on the problems they do? How do they pick problems? What's their "big picture" of research?
4. Read books about the history of science. Reflect on why some researchers focused on important directions their contemporaries ignored.
5. Critically consider your research taste, and the community taste around you. Your taste is likely very influenced by your research cluster (your collaborators, advisor, etc)

https://colah.github.io/notes/taste/

评估idea——可行性

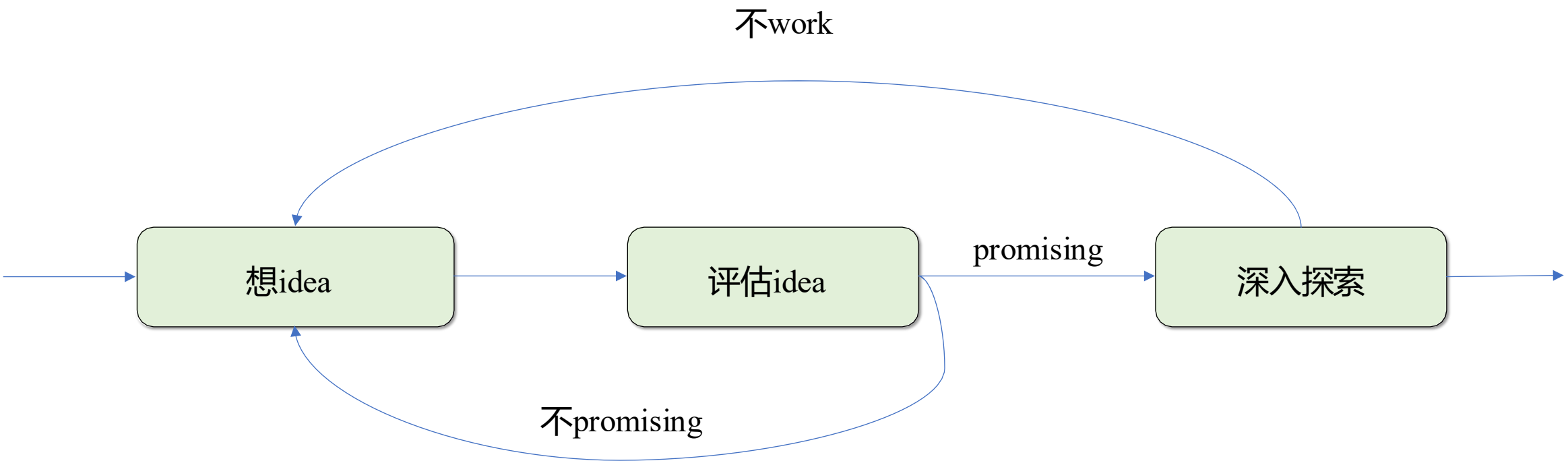
- (阅读/与他人交流)：我想做的问题有没有人已经做过了？
- (阅读/与他人交流)：我想的方法有没有人在其他的领域尝试过？是否有过成功经验？
- (实践)：设计最简单的实验来快速验证核心idea

评估idea——影响力

- 如果一切顺利，所有的难点都得以解决，最终的效果是否足够惊艳?
- 如果我跟别人讲述最终可能实现的效果，他们是否对其感到兴奋?

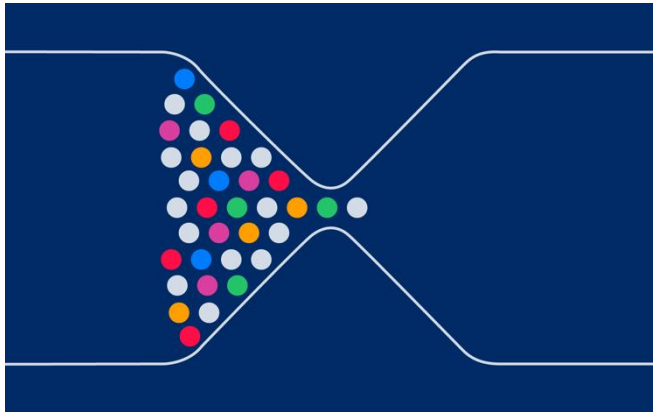
如何执行？

- 有多种策略：
  - 有的人同时对不同的方向和ideas进行初步尝试，选出最promising的一个深入探索
  - 有的人会谨慎想好一个大概的方向，在这个方向内根据实际的反馈对idea进行调整和提升



什么时候应当放弃一个idea?

- 有硬伤:
  - 核心idea已经被人做过了
  - 核心的idea不work
  - 核心的contributions不够，列出contributions并判断这些contributions是否达到目标期刊/会议的标准（求助有经验的人）
- 陷入瓶颈（不要完全放弃）：
  - 尝试所有能想到的解决方案，总结他们不能解决问题的原因，这些原因是不是致命原因?
  - 跳出来看，是不是思路不对？考虑对问题进行重构
  - 是不是有一个特定的模块缺失？发现/等待更好的工具出现



一些tips

- 常见错误：对领域了解不够、对前沿的跟进不够，导致研究的方向明显不能达到SOTA而不自知
  - 没有在最好的方法的基础上进行改进
  - 没有用最先进的工具
- 解决方案：多读多问

一些tips

- 常见错误：在深入探索一段时间后放弃一个项目，转而开始做一个不相关的项目
  - 浪费了在一个方向上积累的insights和技能
- 解决方案：对这个项目进行解构，尽可能找到一个相关的项目，“重复利用”在这个项目里学到的知识和积累的技能

一些tips

- 维护一个文档，记录自己所有的research ideas
- 记录本身可以帮助自己理清思路
- 方便对不同的ideas进行比较和融合

Thank you!