

Learning Research

Jiahao Xiang¹

¹Hengyang Normal University

2024 年 10 月 19 日

Motivation: 对于接触 Research 一年多，还是小白的我来说，不具备一套高效的方法论。这次分享，我们对常年混迹与顶会的，一位浙大的大佬（彭思达）分享的 learning_research 进行学习，在大佬的输入下，输出一下我们学习到的内容，希望能够对大家也有所帮助。

大佬思想使用该颜色块标注

https://github.com/pengsida/learning_research

我们的想法

汇报 slide: https://github.com/jiahaoxiang2000/TempWrite/blob/master/slided/learning_research.pdf

Table of Contents

① 找问题

② 解决问题

③ 做实验

④ 写论文

找问题

一阶段

这个阶段追求**广度**，了解一些基础的概念和算法。不要求深度，不要求掌握/熟悉算法所有的细节。这个阶段的目的是让你对大方向有一个大概的了解，知道有哪些算法，知道这些算法的**大概原理**，知道这些算法的应用场景。

二阶段

这个阶段追求**深度**，追求掌握某一篇论文的细节（算法细节、代码实现细节）。这个阶段的目标是构建某一个科研细分方向的算法基础，了解一篇**论文**是怎么做出来的（寻找科研问题、想 idea、做实验、写论文）。

找问题

当来到二阶段时，一类问题已经明显了，一类为旧的 issue，我们阅读的文献；二类为新的 issue，属于开创新的贡献。

解决问题

三阶段

在有了一定算法基础以后，开始在实验室的指导下做一个自己一作的 Project。这个阶段的目标是通过**实践**来学习一篇论文是怎么做出来的。

想 idea

想点子的过程，就是尝试去解问题的过程。找找旧的解法，看看有没有可以改进的地方，或者能不能引入一些新的思路。

杨植麟认为

技术的本质就是对方法做**组合**，把小的技术组合成大的技术，把老的技术组合成新的技术。

想 idea

具体的想 idea 的流程 (Goal-driven research)

1. general goal

一般而言, general goal 容易定义, 但制定 roadmap 需要对领域有深刻的理解。可以通过构建 literature tree 来建立起对该领域的认知。

literature tree

- 收集相同方向的论文。
- 通过阅读论文, 梳理出当前方向已有的 milestone tasks, 并标记提出该task的第一篇论文 (1 类 novelty)。
- 将论文根据 milestone tasks 进行归类。梳理出有代表性的 pipelines, 并标记提出该pipeline的第一篇论文 (2 类 novelty)。
- 根据 pipeline 再细分到 novel module, 归类论文 (3 类 novelty)。加一些 module 改进已有 pipeline 地工作 (4 类 novelty)
- 随着自己对领域的理解, 增加新的 milestone tasks。

novelty 的分类

创新性越高，它所能影响的文章数量就越多。1 类 milestone task, 2 类 novel pipeline, 3 类 novel module, 4 类旧 module 改进已有 pipeline 地工作。i.e. 创新性很大程度上，影响文章录用的 level。

2. 选题

根据 novelty-tree 列出的 roadmap，选择有研究空间的 task，调研这个 task 有没有重要的 technical challenge。选题是对一个 research project 影响最大的一步，而不是后面的想方法。

3. why not work reason?

从技术层面上分析现在的 pipelines 在某个 task 上不 work 的原因，在 pipeline module 层面思考可能的原因，然后在 pipeline 层面或 high-level idea 层面思考可能的原因。

4. Innovation

- 通过看论文、思考、做实验、与人讨论等方式，发现当前的 pipeline 满足不了哪个指标。找到的“问题”。
- 1) 寻找有没有解决相似“问题”的论文，看看这些论文里有没有分析导致“问题”的技术原因。2) 从论文获得的知识。总结这些论文的分析，形成自己的一套思考。3) 从有经验的人身上蒸馏知识。4) 从实验获得知识。

找点子

点子函数 $f: g \mapsto q$, 其中 g 为具有泛化性的知识, i.e. 可以用在多个问题上, q 为具体的问题。我们的目标是找到一个函数 f , 使得 q 尽可能优。

1) 知 $f_1: g \mapsto q_1, q_1 \approx q_2$, 求 $f_2: g \mapsto q_2$ 。2) 和 3) 知 q , 找 g , 求 f 。4) 知 q, g , 求 f 。2) 和 3) 难, 1) 简单一些。

5. 实验验证创新点。

Goal-driven research 在 research 产出方面的好处

Goal-driven research 的风格是追求重要的任务，试各种方法把这个任务做 work。通过一些条件的 relax，总可以把一些重要的任务做出一些 work 的结果。这样 project 有**产出的保证**。

有些人喜欢追新技术，一味的把新技术调 work。但我们这方向是实验科学，不通过大量实验难以确定一个技术是否真的 work，导致这种科研方式风险性太大了。

当出现新锤子的时候

非常值得拿新锤子来做自己 roadmap 上的某一个 milestone task，这样容易做出有影响力的工作。如：

- Transformer 出来的时候，拿来搞 LoFTR
- NeRF 出来的时候，拿来搞 Neural Body
- Stable diffusion 出来的时候，拿来搞 DreamFusion、DreamBooth

实验室的帮助

- 开箱自带重要的科研问题 task。
- 丰富的 Review 经验。
- 防止进入 local minimum，路走窄。
- 避免 technical flaw，路走死。
- 帮改进其提出的创新。

做实验

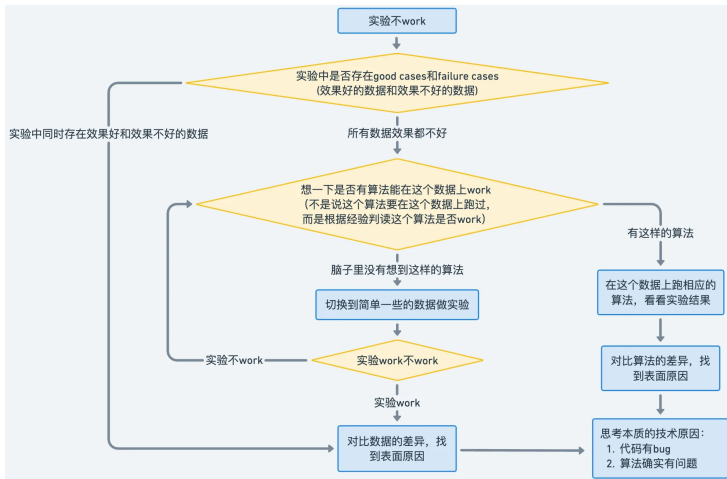


图 1: 如何找到当前实验不 work 的原因

做实验

not work 怎么办

- ① 搜集当前实验的 failure cases
- ② 搜集当前实验的 good cases (效果好的实验结果), 或者找到一个能 work 的实验版本。方式: 降低问题复杂度, 遍历问题点。
- ③ 分析 “work 的实验版本” 和 “不 work 的实验版本” 之间存在 performance gap 的技术原因。列出尽量多的技术原因。
- ④ 实验验证上一步中提出的技术原因。快速迭代。
- ⑤ 针对导致 failure cases 的技术原因, 提出解法。(需要建立自己的武器库, 知道学术界都有哪些技术。可以通过构建 literature tree 来帮助建立武器库。)要经常性地确认自己在正确的方向上: 当前的算法思路真的对吗。要避免陷入 local minima。建议经常找同学交流讨论。

写论文

写论文的步骤

- 画一个清楚的 pipeline figure 的草图（**理清楚方法的流程步骤**）。
- **梳理论文的 story**（倒推，然后正推。详细做法见 Introduction 部分）。整理成 story 以后，写一个 introduction 的思路。同时列出要做的 comparison experiments。
- **写 method，同时做实验。**
- 改 introduction 和 method，同时做实验。
- 实验做差不多以后，写 experiment。
- 写 related work。
- Review 论文。改论文的 introduction、method 和 experiment。
- 写 abstract，取论文名字。
- **反复 review 论文，改论文。**

写论文

把论文做得漂亮、美观，让人第一印象觉得这篇论文很高级。

- 好看的 teaser figure、pipeline figure。
- 好看的表格和结果图。
- 整齐的排版。

段落写作原则

- 一段文字只讲一个 Message，并表达清楚，不要把几个 Messages 杂糅在一起。
- 一段文字开头第一句就要让读者知道这段在说什么。

Flow, WILLIAM ZINSSER, On Writing Well

Good writing have an aliveness that keeps the reader reading from one paragraph to the next.