

研究目的：研究和论文应该旨在真正推进计算机视觉领域。这意味着研究者在进行研究时，应不断批判性地问自己：“我的工作是否能定义或重塑未来的某个问题、领域或技术？”研究主题应具有广泛的“客户群”，并提供他们想要使用的解决方案。

关于未来而非过去：优秀的研究项目不应只是超越过去的成果，而应为未来的研究提供灵感和引用。

实例分析：举了一个 CVPR' 09 提交的例子，尽管在 Caltech101 上取得了 95% 的高性能，但因为没有提供新的学习或应用价值而被拒绝。

简洁性原则：方法/算法应尽可能简单、连贯、有原则。过于复杂或需要手动调整许多参数的方法不被认为是有原则的。

研究者的角色：作为世界顶级大学的博士生，研究者应展现出最高水平的智慧。他们不仅是技术人员，而是在通过论文传达思想和知识。

高标准目标：应设定高标准的目标。研究者应问自己：我的论文是否能成为该年度在计算机视觉领域中被人记住的 10 篇论文之一？

审稿过程的随机性：审稿过程虽然随机，但有一个黄金法则：写得不好的论文会得到差评。无论理念、结果、引用如何，写作都至关重要。

研究和论文的成功公式：重要的问题（鼓舞人心的想法）+ 坚实新颖的理论 + 有说服力的分析实验 + 良好的写作 = 杰出的研究 + 优秀的论文。如果这些要素中任何一个薄弱，论文和评审分数都会受影响。

请务必做好实验记录，这样你才能：

- 1) 做对照实验（高中生物里学的，对照实验），分析各个变量对结果的影响；
- 2) 在几个月之后查询一些之前你以为不重要的实验；
- 3) 方便远程协作沟通，密切配合，彼此不用通话都知道实验进展和安排。

好的习惯：

- 1) 把对你当前实验重要的变量，列成栏目，而不是堆在一个 setting 栏里。做实验过程中，你会逐渐意识到重要的变量是什么，这个时候就需要整理表格，打新的栏，分离出重要变量。
- 2) 在做实验之前就打好表格，用表格来 drive 你安排实验，思考实验安排是否合理。
- 3) 不断的思考，整理表格。用整理表格来 drive 你总结提炼对问题的理解。
- 4) 在有问题的時候，设计恰当的实验，快速定位问题在哪儿。比如做 self-supervised learning 实验，有问题的時候，设计实验，假设 ground-truth 是已知的，通过这个实验 setting 来区分是 self-supervised learning 的问题，还是其它的问题。

5) 控制实验规模，就是在控制实验风险。项目初期的重点是放在想法的基本可行性验证（明确比较对象，规模要小），避免无关的结构调整和参数调整（这些是后期冲击点数用的）。

6) 一个好的实验安排，不论实验结果点数高低，都能拿到 insight，每次的实验都是为了之后的实验而进行的设计。需要充分和之前各种实验联合对比，一方面验证自己的理解是否正确，一方面为下一步的实验安排提供更合理的思路方针。列实验表格备注实验结论方便回溯。

7) 确保每个实验都有控制变量的比较对象，只变换一个参数，这样的实验结论是可分析的，如果某个实验和其他任何实验比都换了不止一个参数，就难以分析其中的原因了。从 baseline 一步一步迁移到我们的方法，逐步验证有效性，不要一把梭哈。

8) 对实验现象和实验结果要充分分析，耗费大量的时间（占据 50%甚至更多的工作时间）去理解可能的背后原因，充分利用各种验证性实验（例如通过设定假设来区分不同的实验问题）和分析性探针工具，去进一步剖析实验背后的机理，在此基础之上再思考现在的下一步的实验安排。而不是只看实验结果，哪里点数低就想着怎么涨点，这样就容易有偏差（用解决 A 问题的方法 试图 去解决 B 问题）。

9) 扩大实验 design choice 时，先确定哪些 design choice 对实验有较大影响的，哪些 design choice 对结论影响比较小的。先在 baseline 上确定“对结论影响比较小的 design choice”的基础取值（不至于点数比较离谱，合理即可）。然后针对“对结论影响比较大的 design choice”，可以列出所有可能的排列组合，从中剔除掉明显比较差的组合，对这些排列组合进行优先级的排序（1）promising 的排前面，（2）某些实验如果结果不论好或者不好，都能明显减小后面实验规模的，排前面 （3）控制变量法

10) 所有新实现的，只要不是之前别人的已经被验证过的代码，不论是简单的模型结构，还是复杂点的算子实现，还是 dataloader（Dataloader 给 model 的输入根据反变换判断当前图像是否是对的）等等，**任何一行代码，都不要认为它是对的，再简单的代码，也要假设它是错的，可能是有问题的【防御式编程】**。每次新添加的代码都要验证它的正确性，可以做（1）**unit test（单元测试的代码往往比主实验的代码还要多）**，（2）sanity check，（3）简单实验验证。只有每一步都走扎实了，实验才能稳步推进，否则一旦出现意料外的实验结果，就不清楚是自己的 bug，还是实验本身整个结论，可能还要面临代码回滚，最糟糕的是回滚到特别古老的代码，发现最近跑的实验都是有问题的。

11) 保证你的实验的可复现性，即时 commit

节约 GPU 的 practice:

1) 多起短轮数的实验，偶尔起长轮数的实验。比如 COCO 上，做出 baseline a 之后，起短轮数 3epoch 和长轮数 16epoch 的实验；之后的改进和 debug 的 b, c, d 都在短轮数上做，观察趋势是否符合预期；积累一定改进之后，到 baseline e 再跑长短轮数。

1. 监控 GPU 使用率，看是否有 IO 问题导致 GPU 没用满。

2. 在 A100 上能够 BF16 的时候就 BF16，不过 BF16 对 attention, softmax 等有精度问题，需要特别注意，请教有经验的同学

沟通习惯：

1. 对于偏 junior 的同学，尽可能的提高跟 mentor 的沟通频率，保证沟通的顺畅（特别是实验计划和下一步的计划）。不要担心消息太多，这样才能让所有人都 keep track progress
2. 实验效果不好的情况可能占据项目时间的 95%+，这是正常情况，不要报喜不报忧觉得我闷头再花时间弄一弄看能不能把效果调好，除非是实验跑错了，不然及时 broadcast 结果，方便大家一起来思考

注意规划，保持紧迫感：

1. 对 project 要有规划，不要觉得有 mentor 帮忙把关自己只需要专注开会同步进度+做实验就可以了。在做实验计划的时候就要想清楚，这一步需要做多久，不要钻在细节里出不来觉得我这个问题不解决就不能往下推进，大部分时候一篇 submission 无法做到尽善尽美，需要取舍。如果沟通习惯不好，自己规划的也不好的话会导致来自 mentor 的反馈周期大大拉长，很影响项目在方向上进展。
2. 每个阶段 都要明确这个 project 的目标是什么（story），story 在项目不同阶段可能会变，但要时刻记住现在的目标，并且要检查接下来的实验规划跟 story 是否 align（我跑了这个实验，是否能放进 paper，是否对 story 的完整性有帮助）