



科研训练（第三讲）

科研方法

从机理、解法到实现与证据

晏潇 博士 | 武汉大学 · 武汉数学与智能研究院 | yanxiaosunny@whu.edu.cn



从第二讲到第三讲：候选问题还不是研究方案

第二讲产出

文献比较表 + 候选问题陈述



第三讲任务

机理、解法、 MVP 实现、实验与逻辑审查

候选问题只有经过机理解释、可行性估算和证据设计，才会变成可执行的研究。



科研方法不是“算法、模型和工具清单”

错误起点

“最近大模型很热门，我想用大模型解决这个问题。”

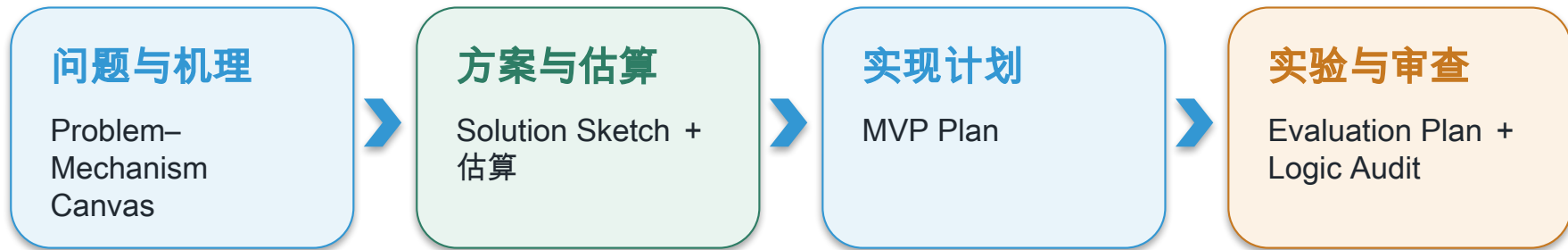
正确起点

“关键机理是什么？什么改变能影响它？需要什么证据？”

技术是候选解法；科研方法决定为什么选它、怎样实现、如何证明。



本讲路线与四项产出



所有内容围绕同一条链：问题 → 机理 → 解法 → 实现 → 实验 → 结论。



先区分：现象、问题、机理和解法

现象

迭代任务运行很慢

研究问题

如何降低迭代任务的跨轮次数据访问开销？

机理

每轮重复读取稳定存储，I/O 开销随迭代累积

解法

跨轮次在内存中复用工作集，并保留可恢复信息

研究顺序不能倒置：不要先有解法，再反过来包装问题。



一个“缺点”为什么还不是研究问题？

缺点描述

“现有方法准确率低。”

研究问题

“在少样本类别中，现有分类器因训练数据分布不均而召回率显著下降；如何改善少样本召回，同时控制总体精度损失？”

- 1 明确场景
- 2 明确可测不足
- 3 提出可能机理
- 4 写出目标与副作用边界



好问题的五个检查标准

重要

结果对谁有影响？

具体

对象、场景和指标明确吗？

可研究

能通过分析、实现或实验推进吗？

可检验

什么证据支持或推翻？

有边界

哪些情况不在本研究范围？

如果问题无法写出可观察的成功与失败条件，就还不能进入方案设计。



用问题边界控制课题规模

对象

研究哪个系统、模型、用户或数据？

场景

在什么任务、规模、环境或约束下？

目标

准备改善哪个主要指标？

资源

可以使用多少数据、时间、算力和人员？

非目标

明确暂时不解决什么？

边界不是削弱创新，而是让创新能够被实现和验证。



为什么值得研究：不要只写“很重要”

影响范围

多少任务、用户或系统受影响？

发生频率

问题多久出现一次？

损失程度

造成多大性能、成本或风险？

现有缺口

已有方案为什么仍不足？

重要性也需要证据：规模数据、真实案例、文献共识或成本估算。



五类常见研究空白

效果 / 质量

结果不够准确、可靠或完整

效率

延迟、吞吐、扩展性不足

开销

算力、内存、通信、能耗过高

适应性

换数据、任务、硬件后失效

认识 / 证据

机理不清、结论冲突或证据不足

高质量问题常来自“特定条件下的失效”，而不是追求抽象的全面提升。



课堂任务：判断哪些是问题，哪些只是方案

A

使用 Transformer 提升分类效果

B

少样本类别召回率显著低于高频类别

C

如何在不降低总体精度的情况下改善少样本召回？

D

设计一种新的注意力模块

课堂任务

3 分钟分类

提交物

现象 / 研究问题 / 技术或方案



参考答案：先把方案语言拿掉

- A 方案 预设使用 Transformer，尚未说明机理与必要性
- B 现象 说明哪里表现差，但尚未形成研究目标
- C 研究问题 明确场景、目标和副作用边界
- D 方案 描述准备做什么，没有解释为什么需要

修改顺序：先写清问题，再根据机理选择 Transformer 是否合适。



为什么必须分析机理？

只看现象

延迟高 → 换更快硬件

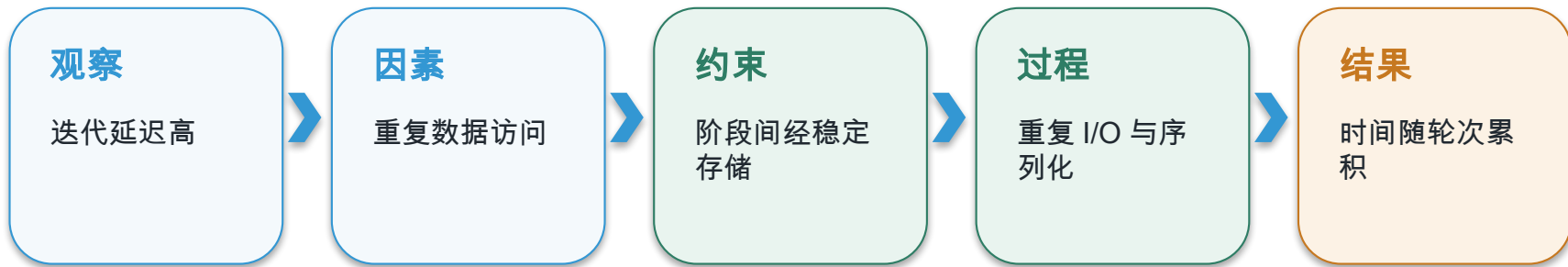
分析机理

重复 I/O 占主要时间 → 改变跨轮数据复用方式

机理决定方案的作用点，也决定实验中应该控制和测量什么。



机理链：从观察走向可干预原因



好的机理描述必须指出：哪个因素，通过什么过程，导致哪个可观察结果。



第一性原理：从不可回避的约束开始

资源

时间、内存、带宽、算力、数据和人力

约束

延迟上限、硬件、隐私、正确性和预算

不变量

必须保持的语义、接口或安全属性

必要条件

方案成立必须满足什么？

第一性原理的产出不是“新技术”，而是一组不可违反的设计条件。



找出方法中的显式假设与隐含假设

假设类型	示例	失效后果
工作负载	任务以批处理为主	迭代与交互场景效率下降
资源	中间结果可以写入稳定存储	I/O 与序列化成为瓶颈
执行	用户函数基本确定	重执行语义可能变弱
规模	集群与数据分区足够均匀	慢任务造成尾延迟



MapReduce 案例：从问题到机理

场景

迭代机器学习需要反复使用同一工作集

设计假设

跨阶段或作业的数据通过稳定存储共享

直接成本

每轮重复读取、解析和序列化

累积效果

I/O 开销随迭代次数增长

候选机理

缺少可容错的跨操作内存复用抽象

案例依据：MapReduce, OSDI 2004；RDD, NSDI 2012.



课堂任务：将信息卡排成机理链

A

多轮重复读取相同工作集

B

迭代任务总延迟高

C

阶段间通过稳定存储共享数据

D

磁盘 I/O 与序列化开销累积

E

每轮计算结束后工作集不能直接复用

课堂任务

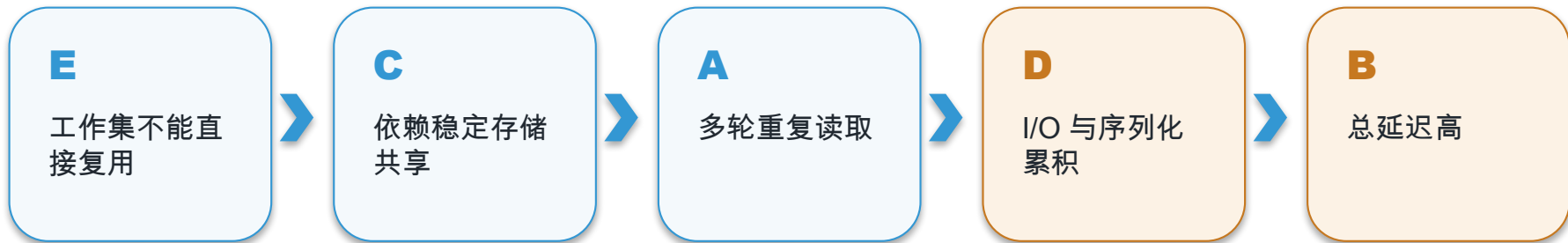
3 分钟排序

提交物

起点 → 过程 → 结果



参考答案：机理链与可干预点



可干预点

改变跨轮数据复用和故障恢复机制，而不是只换更快的磁盘。



研究想法、设计原则与假设并不相同

研究想法

把工作集放在内存里

设计原则

减少跨轮重复 I/O，同时保持容错

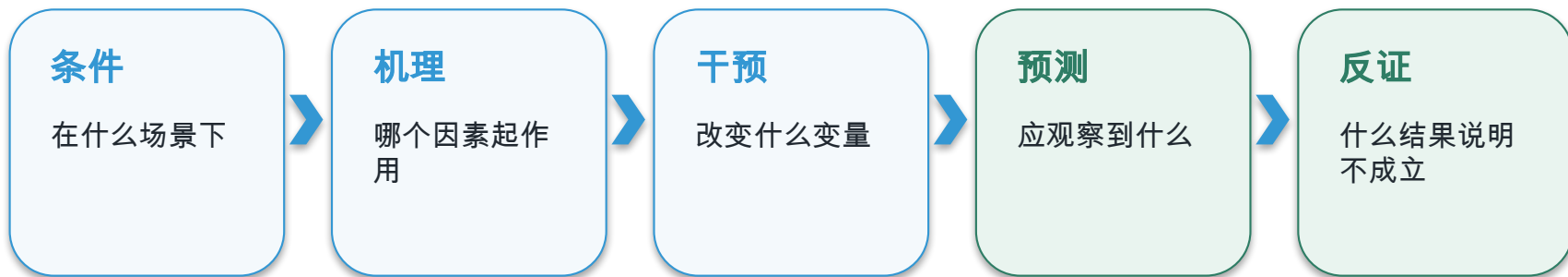
研究假设

若重复 I/O 是主要瓶颈，内存复用应显著降低后续迭代耗时

只有第三种表述能够直接指导实验设计。



可检验假设的基本结构



模板

如果【机理】成立，那么在【条件】下改变【关键因素】，应导致【可观察结果】；如果未出现，则需要修正假设。



可反驳性：研究假设必须允许自己“失败”

不可反驳

“新方法在合适条件下会更好。”
任何结果都可以事后解释。

可反驳

“当工作集可放入内存时，去除重复磁盘读取应使后续迭代耗时下降；若 I/O 占比很低，则假设不成立。”

写假设时，同时写下“什么结果会推翻它”。



课堂任务：用关键词拼装研究假设

条件

工作集可以装入
内存

机理

延迟主要来自重
复 I/O

干预

跨轮复用内存数
据

预测

后续迭代耗时下
降

反证

I/O 占比低或耗
时不变

句式

如果.....，那么.....；如果.....，则需要修正假设。

课堂任务

3 分钟拼装

提交物

一条包含反证条件的假设



参考答案：把机理写成可观察预测

假设

如果迭代任务的主要延迟来自跨轮重复 I/O，那么当工作集可放入内存时，复用内存数据应显著降低后续迭代耗时。

反证条件

如果磁盘 I/O 只占很小比例，或引入缓存后后续迭代耗时没有下降，则该机理判断需要修正。

由此自然得到实验需求：分解耗时、控制工作集大小、比较首轮与后续轮次。



方法必须由问题和机理决定

问题驱动

先确定需要改变的机理，再寻找可以施加干预的方法。

技术驱动

先决定使用某技术，再寻找能包装成应用的场景。

判断方法是否合适：它是否直接改变关键因素？是否产生可验证的预测？



不同研究问题需要不同方法与证据

研究类型	典型问题	主要方法	关键证据
理论	性质、界或可解性	定义、证明、反例	正确性与边界
系统	性能、可靠性、扩展性	架构、原型、测量	基线、开销、规模
机器学习	预测、表示、泛化	模型、数据、训练	效果、消融、鲁棒性
HCI/ 应用	行为、需求、体验	观察、访谈、实验	样本、任务与分析
安全	威胁、攻击、防御	攻击模型、分析、系统	覆盖、成功率与代价

参考：现有课程调研中的 CMU Applied Research Methods 与 Washington Empirical Methods 启发。



从机理出发的六种解法杠杆

删除假设

不再要求固定输入规模

改变表示

从物理数据转为逻辑血缘

复用状态

缓存重复使用的工作集

调整粒度

改变任务、数据或更新粒度

移动计算

计算靠近数据或数据靠近计算

接受权衡

用部分精度换效率或成本

每个候选方案都要写清：改变了哪一个假设或作用过程。



从机理提炼设计原则，再选择具体技术

机理判断

主要延迟来自跨轮重复读取和序列化。

设计原则

跨操作复用工作集，同时保存足够信息以支持故障恢复。

一种具体方案

不可变分区数据 + lineage 重建丢失分区。



不要只保留第一个想法：构思三个备选方案

方案 A

缓存工作集 + 复制容错

恢复快，内存 / 网络开销高

方案 B

缓存工作集 + **lineage** 重算

开销低，长链恢复可能慢

方案 C

缓存 + 周期性
checkpoint

折中恢复成本与检查点开销

比较维度：作用机理 | 理论收益 | 资源开销 | 失败模式 | 实现难度。



方案草图：把想法写成可实现结构

输入

处理什么数据和状态？

核心机制

改变哪个过程或假设？

输出

产生什么结果或系统状态？

预期变化

哪些指标为什么会改变？

代价

新增了什么资源或复杂度？

如果无法解释“机制如何导致指标变化”，**Solution Sketch** 仍然只是功能列表。



为什么编码之前先做数量级估算？

判断收益上限

即使完全消除某项开销，最多能改善多少？

判断资源可行

内存、带宽、算力和数据是否够用？

定位真正瓶颈

主要成本到底在 I/O、计算、通信还是调度？

决定先做什么

优先验证对结论影响最大的未知量。



数量级估算示例：先看方案是否值得做

给定

工作集 100 GB | 磁盘有效带宽 1 GB/s | 20 轮迭代 | 假设每轮完整读取一次

请估算

仅重复读取数据的理论时间约为多少？如果全部消除，收益上限是多少？

参考计算

$100 \div 1 \times 20 = 2000$ 秒。实际收益还受计算、缓存与恢复开销限制。

估算结论： I/O 确实可能是主要矛盾，因此值得先做内存复用的最小验证。



实现目标：验证假设，而不是先完成产品

可以精简

完整 UI | 大规模部署 | 次要功能 | 自动化
包装 | 全部优化

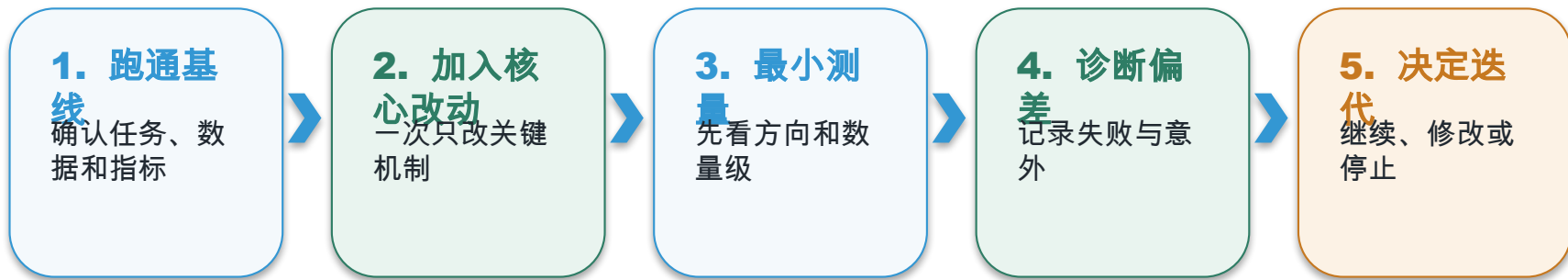
不能省略

核心机制 | 基线 | 正确性检查 | 环境记
录 | 关键测量

20% 的实现完成 80% 的验证：先证明“这个方向值得继续”。



快速迭代：Baseline-first 与单变量改动



每次迭代都应留下：代码版本、参数、环境、结果与下一步判断。



最小实现计划：MVP 也要可复查

模块	最低要求
Baseline	可运行、可重复、结果合理
核心改动	只实现与假设直接相关的机制
正确性	小数据测试、断言或已知结果检查
记录	版本、环境、参数、随机种子和日志
停止条件	何种结果说明不值得继续扩展



实验设计：从研究主张反推证据

Claim 1

方案降低迭代延迟

端到端耗时 + 分阶段耗时

Claim 2

收益来自减少 I/O

I/O 量测量 + 机制消融

Claim 3

额外开销可接受

内存、恢复和管理开销

Claim 4

在目标范围内稳定

不同规模、轮次和故障条件

覆盖原则：每个主要 **Claim** 至少有一组直接证据；每个关键机制至少有一次隔离验证。

参考：现有调研中的 Washington Empirical Research Methods 启发。



一张科学实验设计表需要哪些字段？

主张	实验 / 对照	变量控制	指标与统计	预期反证
降低延迟	Baseline vs 新方案	同数据、资源、轮次	总耗时；中心趋势；误差	耗时无下降
机制有效	关闭内存复用	其他模块不变	I/O 量、阶段耗时	I/O 不是主要来源
开销可控	改变工作集规模	相同任务负载	内存峰值、恢复耗时	开销超过收益
适用范围	多规模 / 故障条件	统一配置	趋势、方差、失败率	仅单一条件有效

至少报告重复次数、中心趋势和波动；统计显著不等于实际意义。



课堂任务：审查一份看似完整的实验

实验描述

新方法在一个数据集上运行一次，准确率从 82% 提升到 84%；旧方法使用默认参数，新方法充分调参，因此证明新方法普遍更优。

A 只运行一次，没有稳定性信息

B 基线与新方法调参不公平

C 单一数据集却声称普遍有效

D 2 个百分点提升是否有实际意义未说明

课堂任务

3 分钟选择

提交物

找出至少两项证据或逻辑问题



常见逻辑谬误：结果正确也可能结论错误

相关当因果

同时变化就声称因果

以偏概全

单一数据 / 场景推广全部

选择性汇报

只展示有利结果

不公平比较

预算、资源、调参不一致

指标偷换

代理指标当成真实目标

混杂变量

多个因素变化却归因一个

事后解释

看结果后才构造假设

缺少反证

只设计支持方案的实验

结论必须严格限制在数据、实验设计和统计证据能够支持的范围内。



完成研究闭环：本讲总结与课后产出

问题 / 机理

解释为何发生



解法 / 估算

判断值不值得做



MVP 实现

快速验证核心机制



实验 / 审查

形成可信证据

课后提交

Solution Sketch | Back-of-the-envelope 估算 | MVP Plan | Evaluation Plan
+ Logic Audit

完成标准

问题有边界；方案作用于机理；实现足以验证；实验覆盖 Claim；结论不过度。

课程设计参考：Stanford、Cambridge、北京大学、Edinburgh 与 Washington 相关调研材料。