大模型推理基础设施

第 11 讲实验方法与可验证性

张书豪

华中科技大学计算机学院

研究生课程课件

教学目标

核心结论

本讲讨论实验设计与验证方法，并说明测量能力本身构成系统研究能力的重要部分。

- 理解 workload、对照组和指标选择。
- 区分端到端指标与局部探针。
- 学会识别不可信实验结论。

实验设计的最小要素

核心结论

一个最小可信实验至少需要 workload、对照组、指标和环境说明四部分。

- 输入分布和服务场景是什么。
- 比较对象和控制变量是什么。
- 主要指标与辅助探针分别是什么。
- 环境和配置是否可复现。

workload 设计的重要性

核心结论

workload 决定实验实际测量的问题对象；若输入分布与服务场景不匹配，结论将整体偏移。

- 长上下文和短上下文应区分。
- 在线动态到达和离线批处理不能混用。
- RAG 和普通对话的链路结构本身就不同。

端到端指标与局部探针

核心结论

端到端指标告诉你用户感受到什么，局部探针告诉你系统里发生了什么。

- TTFT、P99、goodput 属于端到端指标。
- batch 大小、KV 占用、GPU 利用率属于局部探针。
- 研究中通常两类数据都必须报告。

为什么仅报告端到端指标并不足够

核心结论

仅报告端到端数字可以描述结果现象，但通常不足以解释其形成原因。

- P99 上升可能来自队列，也可能来自通信。
- throughput 下降可能来自 KV 容量，也可能来自调度保守。
- 探针指标帮助建立因果解释链。

常见实验陷阱

核心结论

很多看起来漂亮的结果，问题并不在数字，而在实验边界没有说清。

- 只报告最好一次，不报告波动。
- 输入分布不合理，无法代表目标场景。
- 对照组配置不公平或热身不足。

课程项目中的实验设计

核心结论

课程项目中的常见问题在于实验计划模糊，从而导致结果缺乏解释力。

- 先选主指标，再补探针指标。
- 先写预期会否定当前假设的结果。
- 先定义最小可跑场景，再扩到完整 workload。

实验草案的基本结构

核心结论

实验开始之前，应先明确整体结构，而非在执行过程中临时补充设计。

- 主问题和待验证假设。
- 实验组、对照组与控制变量。
- 主要指标、辅助探针和失败标准。

源码穿插：用 nano-vllm-hust 组织最小 benchmark

核心结论

`bench.py` 已经包含最小实验记录需要的配置、warmup 和结果汇总结构。

- 它显式设定随机种子、请求数量、输入长度与输出长度分布。
- 它先执行一次 warmup，再进入正式测量。
- 它最终输出总 token、总时间与 throughput，便于说明最小实验记录格式。

源码穿插：教学版实验任务如何设计

核心结论

结合 `student_exercises/README.md`，可以把实验教学拆成“参考实现阅读 + TODO 补全 + 结果解释”三段。

- 先在参考实现中理解控制点。
- 再在 `student_exercises/bench.py` 或相关练习文件中补全任务。
- 最后要求学生提交结构化实验记录，而不只是运行截图。

讲解：实验方法课的中心不是跑图，而是写出可否定命题

核心结论

如果实验只是为了证明自己是对的，那课程项目和论文都很容易滑向结果导向叙事。

- 好实验首先要写清假设，再写清什么结果会直接否定这个假设。
- workload、对照组和探针的设计，本质上都是围绕这条可否定命题服务的。
- 因此实验方法课训练的是研究纪律，而不是仅仅训练脚本使用能力。

讲解：端到端指标与局部探针为什么必须成对出现

核心结论

只报端到端结果，结论容易悬空；只报局部探针，结果又容易失去服务意义。

- 端到端指标告诉我们用户看到了什么，例如 TTFT、P99、goodput。
- 局部探针告诉我们系统里发生了什么，例如 batch 变化、KV 占用、编译命中。
- 两者结合起来，才能把“现象”变成“因果解释”。

例子：一个看起来完整但其实不可信的实验（1）

核心结论

实验方法课必须先教学生识别坏实验，否则工具越熟练，错误可能越系统化。

表面上很完整

有吞吐图、有延迟表、甚至有消融，但 workload 来源和对照配置没有交代清楚。

例子：一个看起来完整但其实不可信的实验（2）

核心结论

实验方法课必须先教学生识别坏实验，否则工具越熟练，错误可能越系统化。

真正的问题

结果无法回答“为什么变好”以及“在哪些条件下不再成立”。

例子：一个看起来完整但其实不可信的实验（3）

核心结论

实验方法课必须先教学生识别坏实验，否则工具越熟练，错误可能越系统化。

课程结论

一个实验是否可信，不取决于图表数量，而取决于因果链和边界是否被写清。

例子：端到端数字与探针数字不一致时怎么办（1）

核心结论

最有研究价值的情况往往不是数字全都一致，而是局部证据与端到端现象出现张力。

情况一

GPU 利用率更高，但 TTFT 更差，说明等待或同步代价可能在放大。

例子：端到端数字与探针数字不一致时怎么办（2）

核心结论

最有研究价值的情况往往不是数字全都一致，而是局部证据与端到端现象出现张力。

情况二

KV 命中率更高，但 goodput 没提升，说明命中本身未必落在主导瓶颈上。

例子：端到端数字与探针数字不一致时怎么办（3）

核心结论

最有研究价值的情况往往不是数字全都一致，而是局部证据与端到端现象出现张力。

教学意义

探针不是为了给端到端结果背书，而是为了帮助发现矛盾并修正假设。

代码例子：最小 benchmark 模板应包含什么（1）

核心结论

一份合格的 benchmark，不只是能跑，还应显式携带 workload 和环境信息。

代码片段

```
seed(0)
warmup_requests = 10
num_requests = 100
measure(ttft, throughput, p99)
```

代码例子：最小 benchmark 模板应包含什么（2）

核心结论

一份合格的 benchmark，不只是能跑，还应显式携带 workload 和环境信息。

课堂应追问

哪些配置是影响 workload 定义的，哪些配置只是运行时细节？

代码例子：最小 benchmark 模板应包含什么（3）

核心结论

一份合格的 benchmark，不只是能跑，还应显式携带 workload 和环境信息。

系统含义

benchmark 模板的显式化，本身就在减少“实验只存在于一台机器上”的偶然性。

代码例子：最小日志汇总脚本（1）

核心结论

实验课需要让学生看到，图表往往来自对原始日志的二次组织。

代码片段

```
records = load_jsonl("run.log")
p99 = percentile([r["latency_ms"] for r in records], 99)
goodput = sum(r["ok"] for r in records) / total_time
```

代码例子：最小日志汇总脚本（2）

核心结论

实验课需要让学生看到，图表往往来自对原始日志的二次组织。

课堂应追问

如果日志字段设计不对，后面哪些指标将根本算不出来？

代码例子：最小日志汇总脚本（3）

核心结论

实验课需要让学生看到，图表往往来自对原始日志的二次组织。

课程结论

可验证性不只发生在论文写作阶段，而是从日志字段设计就已经开始了。

例子：消融实验最容易做假的地方（1）

核心结论

消融的真正任务是隔离变量，而不是制造更多子图。

伪消融一

一次改了多个控制点，却只关掉其中一个开关就说完成了消融。

例子：消融实验最容易做假的地方（2）

核心结论

消融的真正任务是隔离变量，而不是制造更多子图。

伪消融二

workload 或参数同时变化，导致数字变化无法归因。

例子：消融实验最容易做假的地方（3）

核心结论

消融的真正任务是隔离变量，而不是制造更多子图。

课程要求

每一项消融都要明确写出“被隔离的变量”和“预期影响的指标”。

课堂练习：写一页可信实验卡片（1）

核心结论

实验方法训练最好用一页纸完成：问题、对照、指标、失败条件。

练习一

写出你的 workload 定义，至少包含输入长度分布、输出长度分布和到达模式。

课堂练习：写一页可信实验卡片（2）

核心结论

实验方法训练最好用一页纸完成：问题、对照、指标、失败条件。

练习二

写出一个主指标和两个辅助探针，并说明三者的因果关系假设。

课堂练习：写一页可信实验卡片（3）

核心结论

实验方法训练最好用一页纸完成：问题、对照、指标、失败条件。

练习三

用一句话写出：什么结果出现时，你会承认当前机制没有兑现预期收益？

课堂讨论

核心结论

实验的目标在于判断系统假设是否成立，而非单纯证明预设观点。

- 为什么单次最好结果价值很低？
- 什么时候局部探针最能帮助解释端到端现象？
- 哪些结果一旦出现，就说明当前机制假设应该被放弃？

本讲总结

核心结论

缺乏规范实验设计时，机制难以转化为可信的系统结论。

- 验证方法本身就是研究贡献的一部分。
- 下一讲将把这些方法落到真实代码与复现实验上。

对应 Tutorial

核心结论

本讲对应 Tutorial 10。先独立作答，再对照下一页参考答案。

- 文件：build/tutorials/Tutorial_10_ 实验方法与验证.pdf
- 动手运行、日志记录与提交要求统一查看 build/experiments/ 和对应 experiment 源目录下的 assignment_spring-2026/。