

168 天多模态大模型科研学习详细指南

从 Python 零基础到独立复现 实验设计和论文投稿

适合读者 尚未系统学习 Python 和深度学习 希望逐步进入多模态与大模型科研

建议投入 每周六天 每天二至三小时 约三百至四百小时

最终成果 一个可复现的多模态项目 一份规范实验报告 一篇论文初稿

这份手册先补齐原计划默认具备的基础，再进入 28 天多模态专项训练，最后用十四周完成复现、选题、实验和写作。达到科研能力不能靠背模型名称，而要形成一条可验证的工作链：提出问题，建立基线，控制变量，记录实验，分析失败，写出能被复核的结论。

使用原则 不追求每天看完最多内容。每个阶段只有在验收任务通过后才进入下一阶段。硬件不足时优先使用小模型、子数据集、冻结编码器或云端按时计费实例，不从训练七十亿参数模型起步。

从零基础到第一篇论文初稿

二十四周只沿一条主线推进 每个阶段都有可验证的作品



阶段证据

脚本与图表 训练记录 迷你 CLIP 可复现基线 论文与代码

如何使用这份手册

- 1. 先在交互网页完成入门测评，网页会建议从第几周开始。零基础建议从第 1 周开始。
- 2. 每天按理解概念 运行代码 完成小作业 写学习记录四步推进。当天代码没有运行成功，不把观看视频算作完成。
- 3. 每周保留半天复盘。把报错、实验参数、关键图表和失败原因写进日志。
- 4. 每个阶段完成验收门槛。若不能口头解释并独立运行，就回到对应周补课。
- 5. 第 17 周开始只选择一个研究方向。科研阶段的目标是把一个小问题做扎实，而不是同时追逐所有热门模型。

总路线

阶段	时间	学习重点	通过标准
第一阶段	第 1 至 2 周	Python 与工具	能独立写脚本 读数据 调试报错
第二阶段	第 3 至 4 周	PyTorch 与深度学习	能写训练循环并解释反向传播
第三阶段	第 5 至 6 周	Transformer 与大模型	能说明注意力 Token 和生成流程
第四阶段	第 7 至 10 周	28 天多模态专项	完成迷你 CLIP 与多模态应用
第五阶段	第 11 至 16 周	复现与工程	复现基线 完成消融 形成技术报告
第六阶段	第 17 至 24 周	科研与论文	提出问题 完成实验 写出论文初稿

每周固定节奏

时间	任务	产出
周一至周四	每天四十分钟概念 九十分钟代码 二十分钟记录	四次可运行提交

时间	任务	产出
周五	阅读一篇教程或论文并复述核心图	一页阅读卡
周六	完成本周综合作业与验收	结果图或小项目
周日	休息或补缺 不引入新主题	下周待办

第一部分 Python 与深度学习预备

这一部分只学习后续真正会用到的知识。暂时不需要系统学习计算机组成、复杂算法竞赛或全部高等数学。你需要掌握的是读懂代码、处理张量、训练模型、解释梯度和理解注意力。

周次	主题	必须理解	本周作品
第 1 周	Python 基础	变量 容器 条件 循环 函数 模块 异常	完成 20 个小练习和一个成绩统计脚本
第 2 周	科研编程	NumPy Pandas Matplotlib 文件路径 Git Notebook	读取 CSV 做清洗统计和三张图
第 3 周	PyTorch 张量	Tensor 形状 广播 数据集 DataLoader 模型层	实现线性回归和图像分类数据管道
第 4 周	训练与反向传播	损失 梯度 优化器 正则化 验证集 指标	训练 MLP 并画出损失与准确率曲线
第 5 周	Transformer	Token Embedding 位置编码 注意力 残差 LayerNorm	手写单头注意力并标注张量形状
第 6 周	大语言模型基础	预训练 指令微调 PEFT 推理 RAG 评估	调用小模型 完成提示词和检索实验

第 1 周 Python 基础

学习目标 看懂常见 Python 代码，能把一个任务拆成输入、处理和输出，并在报错时定位到文件和行号。

- 变量和类型：整数 浮点数 字符串 布尔值。重点理解类型决定了可以进行的操作。

- 容器：列表用于有序样本，字典用于键值配置，元组常表示固定形状，集合用于去重。
- 控制流程：if 决定分支，for 遍历样本，while 只在停止条件明确时使用。
- 函数：参数是输入，return 是输出。先写短函数，再组合成数据处理流程。
- 模块和环境：理解 import 包 虚拟环境 requirements 文件和相对路径。
- 调试：先读报错最后一行，再找自己的代码行，打印变量类型和形状，不盲目重装环境。

```
def normalize_scores(scores):
    low, high = min(scores), max(scores)
    return [(x - low) / (high - low) for x in scores]

scores = [63, 78, 91, 84]
print(normalize_scores(scores))
```

练习 编写一个脚本读取学生姓名和成绩，输出平均分、最高分、低于平均分的名单，并把结果保存为文本文件。

验收 不看答案写出一个含函数、循环、条件判断和文件读写的百行以内脚本。

第 2 周 科研编程

NumPy 是数值数组工具，Pandas 处理表格，Matplotlib 画图。科研代码最常见的问题不是语法，而是数据形状、缺失值、重复样本和不可复现。

- NumPy：ndarray 维度 shape 索引 广播 均值 标准差 矩阵乘法。每次计算前先写出期望形状。
- Pandas：read_csv head info isna groupby merge。不要在不检查数据的情况下直接训练。
- Matplotlib：折线图看训练过程，柱状图比较方法，散点图观察关系，混淆矩阵分析错误。
- Notebook 用于探索，脚本用于稳定复现。最终实验必须能从命令行或单一入口运行。
- Git 只先掌握 status add commit log diff。每个能运行的里程碑做一次小提交。

验收 对一个 CSV 完成数据字典、缺失值处理、分组统计和三张带标题、坐标、图例的图，并说明每张图支持什么结论。

第 3 周 PyTorch 张量与模型

PyTorch 的核心对象是 Tensor。它既保存数值，也能记录计算过程。模型本质上是一个带参数的函数，训练就是用数据不断修正这些参数。

概念	小白解释	你必须检查的内容
Tensor	多维数字表	shape dtype device
Dataset	单个样本如何取出	输入和标签是否对应
DataLoader	怎样分批和打乱	batch size 与最后一批

概念	小白解释	你必须检查的内容
Module	可组合的模型函数	输入输出维度
Loss	预测错了多少	定义是否与任务一致
Optimizer	如何调整参数	学习率和参与训练的参数

```
import torch
from torch import nn

model = nn.Sequential(nn.Linear(10, 32), nn.ReLU(), nn.Linear(32, 2))
x = torch.randn(8, 10)
y = model(x)
print(x.shape, y.shape) # [8, 10] -> [8, 2]
```

验收 逐层写出上面网络的输入输出形状，并替换隐藏层宽度和类别数，让代码仍能运行。

第 4 周 反向传播与训练循环

反向传播不是模型自己理解错误，而是链式法则的高效计算。前向传播得到预测和损失；反向传播计算每个参数对损失的影响方向；优化器沿降低损失的方向更新参数。

- 输入数据进入模型，得到预测值。
- 损失函数比较预测值和真实值，得到一个标量。
- 调用 `zero_grad` 清除上一轮累积梯度。
- 调用 `backward` 从损失向前追溯计算图，得到每个参数的梯度。
- 调用 `step` 更新参数。最简单的直觉是新参数等于旧参数减去学习率乘梯度。
- 在验证集上只做前向计算，不更新参数，并保存表现最好的模型。

```
for x, y in train_loader:
    model.train()
    optimizer.zero_grad()
    logits = model(x)
    loss = criterion(logits, y)
    loss.backward()
    optimizer.step()

model.eval()
with torch.no_grad():
    val_logits = model(val_x)
```

常见错误 忘记清零梯度会累积梯度；训练和验证数据混用会泄漏；只看训练准确率会误判过拟合；随机种子、数据切分和库版本未记录会让结果无法复现。

验收 在 MNIST 或 FashionMNIST 上训练一个 MLP，保存损失曲线、验证准确率、混淆矩阵和五个错误样例。

第 5 周 Transformer 概念

Transformer 把输入切成 token，把每个 token 变成向量，再让每个位置通过注意力读取其他位置的信息。多模态模型常把文字 token、图像 patch token、音频片段 token 放进相同或相连的 Transformer 中。

部件	作用	最小理解
Tokenization	把原始输入拆成离散或连续单元	文本词片 图像 patch 视频帧都可视作 token
Embedding	把 token 映射为向量	形状通常是批次乘序列长度乘隐藏维度
位置编码	告诉模型顺序或空间位置	注意力本身不知道先后和方位
自注意力	同一序列内部交换信息	Q 提问 K 被匹配 V 提供内容
交叉注意力	一个模态读取另一个模态	文本问题可读取图像 patch
残差和归一化	稳定深层训练	保留原信息并控制数值尺度

缩放点积注意力可以记作 $\text{Attention}(Q\ K\ V)$ 等于 $\text{softmax}(QK^T/\sqrt{d})$ 再乘 V 。你不需要先会完整推导，但要知道相似度决定读取权重，除以根号 d 用于稳定 softmax。

```
scores = q @ k.transpose(-2, -1) / (q.size(-1) ** 0.5)
weights = scores.softmax(dim=-1)
output = weights @ v
```

验收 用一个含四个 token 的玩具张量手写单头注意力，打印 scores weights output 的形状，并解释每一维代表什么。

第 6 周 大语言模型基础

大语言模型通常以预测下一个 token 进行预训练。指令微调让模型学会按任务形式回答，偏好对齐调整回答行为，LoRA 只训练少量低秩参数，RAG 在生成前检索外部材料。它们解决的问题不同。

概念	解决的问题	不要混淆
预训练	获得通用语言模式和知识	不是针对你的具体任务训练
指令微调	学习遵循输入输出格式	不保证事实完全正确
LoRA	以较少显存适配任务	不是新的基础模型
量化	降低存储与推理显存	可能影响速度或精度

概念	解决的问题	不要混淆
RAG	把可检索资料放进上下文	不改变模型参数
评估	测量特定能力和风险	单一排行榜不等于真实科研价值

验收 用一个可在本机运行的小模型或课程提供的在线环境，固定五个问题，比较零样本提示、少样本提示和带检索材料三种设置，记录输出、耗时和错误类型。

第二部分 28 天多模态专项

这一阶段改编自参考文档的四周结构。每天只要求完成一个核心概念、一个可运行实验和一个验收证据。论文以经典工作为主，最新模型只作为比较对象。若显存不足，使用托管演示、小模型、预计算特征或五百至五千个样本的子集。

第 1 周专项训练

天	主题	学习和实践	验收证据
Day 1	全景与任务	识别检索 生成 VQA 对话四类任务 画系统图	能区分对齐和融合
Day 2	视觉编码器	理解 CNN 到 ViT 跑预训练 ViT 查看 patch 形状	说清 patch token 和 CLS
Day 3	文本编码器	比较 BERT 自回归模型和编码器解码器	解释三种语言建模范式
Day 4	对比学习	手写余弦相似度和 InfoNCE 做玩具实验	损失下降且能解释温度系数
Day 5	CLIP 架构	跑零样本分类 比较不同文本模板	输出准确率和失败样例
Day 6	CLIP 局限	测试计数 空间关系 OCR 和拼写扰动	形成错误类型表
Day 7	迷你 CLIP	冻结编码器训练投影层 计算 Recall at K	训练评估脚本可重复运行

第 2 周专项训练

天	主题	学习和实践	验收证据
Day 8	BLIP	理解 ITC ITM LM 与 CapFilt 跑图像描述	对比生成描述与人工描述
Day 9	BLIP 2	理解 Q Former 和冻结模型桥接	画出输入输出张量流

天	主题	学习和实践	验收证据
Day 10	图文检索	实现 CLIP 粗排与可选精排	报告 Recall at 1 5 10
Day 11	视觉问答	测试感知 OCR 空间和知识类问题	至少分析十个错误
Day 12	图像描述	比较 BLEU CIDEr CLIPScore 与人工判断	说明指标盲点
Day 13	架构比较	比较双塔 融合编码器 生成模型	按任务和资源选择模型
Day 14	理解系统	构建图片索引 检索 描述 问答 流水线	端到端 demo 可运行

第 3 周专项训练

天	主题	学习和实践	验收证据
Day 15	扩散基础	理解加噪 去噪和噪声预测 跑小型示例	能解释训练与采样的区别
Day 16	潜空间扩散	比较提示词 CFG 采样步数和随机种子	保存网格图和实验表
Day 17	视觉语言模型	理解视觉编码器 投影层 LLM 的连接	标注视觉 token 流向
Day 18	开源模型比较	选两个当前可用模型测 OCR 中文和图表	固定测试集并记录版本
Day 19	视频理解	均匀采帧 聚合 CLIP 特征 比较单帧	说明时间信息何时丢失
Day 20	音频与视觉	绘制梅尔频谱 体验音频文本对齐	完成零样本音频分类
Day 21	生成体系回顾	完成文生图再图生文闭环或视频摘要	写一页综合实验报告

第 4 周专项训练

天	主题	学习和实践	验收证据
Day 22	多模态 Agent	拆解感知 规划 行动和工具调用	设计可回放的任务轨迹
Day 23	评估	建立小型固定评测集和错误标签	结果含均值 方差 失败样例
Day 24	幻觉	测试物体 属性 关系 OCR 幻觉	计算简单幻觉率并提出缓解
Day 25	高效微调	理解 LoRA QLoRA 量化和显存估计	资源允许时训练小适配器
Day 26	多模态 RAG	用图文向量建立检索后回答流程	回答带来源和检索证据
Day 27	综合项目	实现核心功能 建立基线与评测	仓库可一键运行
Day 28	总结与展示	补 README 结果表 失败分析和演示	完成可公开项目页

专项阶段必须完成的两个作品

作品一 迷你 CLIP

- 数据：使用公开图文数据的小规模子集，明确许可证、划分方式和样本数。
- 模型：冻结图像与文本编码器，只训练投影层。先保证单个 batch 过拟合，再跑完整训练。
- 评估：报告图搜文和文搜图的 Recall at 1 5 10，同时展示成功与失败案例。
- 消融：至少比较两个温度系数或两个 batch size，其他条件保持一致。
- 复现：固定随机种子，保存配置、依赖、日志、检查点和运行命令。

作品二 多模态检索与问答工具

- 建立图片与文字索引，支持自然语言查询返回相关图片。
- 对结果图片生成描述或回答问题，并显示检索到的证据。
- 加入至少一个可靠性机制，例如拒答、来源展示、置信度阈值或错误提示。
- 建立二十至五十条固定测试集，覆盖正常、模糊、OCR、计数和关系问题。
- README 写明安装、运行、数据来源、已知限制和演示截图。

经典论文最小阅读清单

主题	论文	阅读任务
视觉	Vision Transformer	看方法图 写出 patch 到输出的形状变化
对齐	CLIP	理解双塔 对称对比损失和零样本分类
融合	ALBEF 或同类工作	比较对齐后融合与纯双塔
理解与生成	BLIP	解释 ITC ITM LM 和数据过滤
桥接冻结模型	BLIP 2	画出 Q Former 的查询流程
视觉指令微调	LLaVA	拆解两阶段训练和数据构造
生成	DDPM 与 Latent Diffusion	区分训练目标 潜空间和采样
高效微调	LoRA 与 QLoRA	估算可训练参数和显存变化
可靠性	POPE 或多模态幻觉评估	说明任务定义 指标和局限

第三部分 复现与科研训练

完成教程只说明你能跟着做。科研能力要求你在信息不完整时仍能形成假设、设计对照、解释结果，并让别人复现。第 11 至 16 周训练复现，第 17 至 24 周完成一个小而完整的研究闭环。

周次	主题	本周工作	交付物
第 11 周	复现准备	选一篇代码公开 数据可得 计算量适中的论文 锁定 commit 和环境	复现卡
第 12 周	基线复现	先跑官方配置 再缩小数据集 跑通训练与评估	基线结果
第 13 周	差异定位	比较论文和本地结果 检查数据切分 预处理 随机种子 指标	差异说明
第 14 周	消融实验	一次只改一个因素 至少三次随机种子 报均值和标准差	消融表
第 15 周	错误分析	按感知 对齐 推理 知识 数据质量标注失败样例	错误图谱
第 16 周	技术报告	写背景 方法 实验 局限并整理仓库	四至六页报告
第 17 周	方向扫描	阅读近两年目标会议论文 建立问题矩阵	二十篇论文表
第 18 周	问题定义	明确输入 输出 数据 评价指标和可证伪假设	一页研究提案
第 19 周	强基线	建立简单可靠基线 记录训练成本和吞吐	基线表
第 20 周	方法实现	只加入一个核心改动 保持其余条件一致	方法代码
第 21 周	完整实验	主结果 消融 泛化 效率 统计不确定性	结果包
第 22 周	论证与图表	把每个主张映射到证据 删除没有实验支持的结论	主张证据表

周次	主题	本周工作	交付物
第 23 周	论文初稿	完成摘要 引言 相关工作 方法 实验 局限	完整初稿
第 24 周	复核与投稿	复现最终数字 检查匿名 引用 许可和伦理要求	投稿包

怎样选择第一篇复现论文

优先选择问题清楚、代码公开、数据可得、单卡或小规模可运行、评价指标明确的论文。首篇复现的价值在于学会实验纪律，不在于模型规模。

维度	适合首篇复现	暂时避开
代码	官方仓库 有明确命令和配置	只有零散片段或依赖已失效
数据	公开 小规模 有清晰许可证	需要私有数据或复杂申请
算力	冻结主干 小模型 子集可验证	从零预训练大模型
指标	可本地自动计算	主要依赖昂贵人工评审
改进空间	论文明确报告局限和失败案例	只追求扩大参数和数据

论文阅读四遍法

12. 十分钟定位问题。只看标题、摘要、图一、结论，写下输入、输出、核心主张和评价指标。
13. 三十分钟理解方法。沿方法总图追踪张量，标出哪些模块冻结、哪些参数训练、每个损失约束什么。
14. 四十分钟审查实验。找数据划分、基线公平性、随机种子、消融、效率、失败样例和统计显著性。
15. 一小时对照代码。找到数据入口、模型入口、损失、训练配置、评估脚本和复现实验命令。
- 每篇论文只保留一张阅读卡 研究问题 基线 缺口 方法 核心证据 局限 可复现信息 你的一个疑问。

从兴趣到研究问题

研究问题要能被实验否定。不要写提升多模态能力，要写在固定视觉编码器和数据量下，引入基于 OCR 置信度的路由是否能降低文档问答的幻觉率，同时不显著增加延迟。

方向	可做的小问题	最小实验
幻觉	检索证据或对比解码是否减少对象幻觉	固定二百张图 比较准确率 幻觉率 延迟
高分辨率	不同切块策略对文档 OCR 问答的影响	控制 token 数 比较精度和吞吐
数据质量	小规模高质量指令数据能否优于等量噪声数据	同模型同轮数 两套数据对照
图文检索	难负样本采样是否改善细粒度检索	比较随机负样本与难负样本
参数高效微调	LoRA 位置和 rank 对效果与显存的影响	固定预算做小型网格消融
多模态 RAG	图文联合检索是否优于只检索文本	相同知识库和生成模型

实验设计底线

- 先写假设再跑实验，防止看到结果后改故事。
- 强基线优先。新方法若只胜过弱基线，论文结论不成立。
- 一次只改变一个核心因素。数据、训练步数、模型大小和评估提示应尽量一致。
- 报告随机性。小实验至少三次随机种子，给出均值和标准差；资源实在不足时明确说明。
- 同时报告效果和成本，包括参数量、显存峰值、训练时长、推理延迟或吞吐。
- 保留失败样例与负结果。它们用于限定结论，也常决定下一步改进。
- 区分开发集与测试集。不要根据测试集反复调参。
- 记录数据许可证、隐私、偏见和可能误用。医学、教育、安防等高风险场景尤其需要说明。

实验目录模板

```
project/  
  README.md  
  requirements.txt  
  configs/  
  src/  
  scripts/  
  tests/  
  data/README.md  
  outputs/experiment_name/  
    config.yaml  
    train.log  
    metrics.json  
    predictions.jsonl  
    figures/  
  paper/
```

每次实验至少记录 Git commit 数据版本 随机种子 模型版本 超参数 硬件 开始结束时间 主指标 失败状态和备注。

论文写作结构

部分	必须回答的问题	常见错误
摘要	问题 方法 关键结果 限制是什么	只写宏大背景 不给数字
引言	为什么重要 现有方法缺什么 贡献是什么	贡献和实验脱节
相关工作	你的方法与最接近工作差在哪	按论文逐篇罗列
方法	别人能否按描述实现	缺少形状 损失和训练细节
实验	设置是否公平 结果是否支持主张	只报最好结果 不报波动
局限	在哪些条件下可能失效	写成无关紧要的套话
结论	证据支持了什么范围的结论	扩大到实验未覆盖的场景

投稿前检查

- 所有表格数字都能从保存的评估文件重新生成。
- 摘要和结论中的每个结果与正文表格一致。
- 基线使用相同数据划分、预算和评价协议；无法相同时明确说明。
- 图轴、单位、图例、误差范围和样本数完整，颜色在灰度下仍可区分。
- 引用来自原始论文或官方文档，未引用二手博客替代原始证据。
- 代码移除本地绝对路径、密钥、账号和大文件，README 在干净环境可运行。
- 检查会议当年的模板、匿名要求、页数、补充材料、伦理和生成式 AI 政策。

第四部分 资源和工具

硬件分支

条件	推荐做法	避免
无独立显卡	CPU 学 Python 与小网络 使用在线 Notebook 或 API 体验大模型	本地下载多组大权重
八至十二 GB 显存	小模型 推理 预计算特征 冻结编码器 低分辨率	七十亿参数全量微调

条件	推荐做法	避免
二十四 GB 显存	量化推理 LoRA 小批量 梯度累积	未经估算直接跑官方大配置
云端 GPU	先在小数据跑通 再按小时扩展 设置预算和自动保存	长时间交互式空闲

推荐工具栈

类别	工具	用途
环境	Python venv 或 Conda	隔离依赖并冻结版本
深度学习	PyTorch	张量 自动微分 模型训练
预训练模型	Hugging Face 生态	加载模型 数据集和参数高效微调
视觉	torchvision timm Pillow OpenCV	图像模型与处理
检索	FAISS 或等价向量索引	小规模向量检索
实验	TensorBoard MLflow 或 Weights and Biases	记录曲线 参数和对比
工程	Git pytest 配置文件	版本控制 测试与复现
写作	Word LaTeX 或 Typst 与 Zotero	论文排版和文献管理

资源使用顺序

- 先看官方入门教程并运行最小示例。
- 再读高质量课程解释概念，自己写一遍代码。
- 接着读原始论文的方法图和实验表。
- 最后对照官方仓库，确认论文描述怎样落到配置和代码。

优先来源 Python 官方教程 PyTorch 官方教程 Hugging Face Course Dive into Deep Learning 论文原文 Papers with Code 和模型官方仓库。网页地址与模型版本会变化，使用时以官方页面和仓库发布说明为准。

常用术语速查

术语	一句话解释
模态	信息的形式 如文字 图像 音频 视频
编码器	把原始输入转换为可计算的向量表示
对齐	让不同模态中语义对应的表示靠近或可交互
融合	把多个模态的信息组合起来完成任务
Embedding	表示对象语义的向量
Token	模型处理的基本单元
预训练	在大规模数据上先学通用能力
微调	针对具体数据或任务调整模型
基线	新方法必须公平比较的参考方法
消融	删除或替换组件 判断它是否真正有效
Benchmark	固定数据和协议构成的评测基准
幻觉	模型生成了输入或证据不支持的内容

阶段毕业门槛

关卡	通过条件
基础关	独立写训练循环 解释梯度 注意力和过拟合
多模态关	实现迷你 CLIP 完成检索 问答或生成应用
复现关	论文主要趋势可复现 差异原因有记录
研究关	问题可证伪 基线公平 消融完整 结论有边界
写作关	他人按文档能复现实验 主张均有表图支持

现实预期 二十四周可以让你从零基础走到能完成第一轮科研闭环，但论文能否发表还取决于问题的新颖性、实验质量、写作、导师反馈和投稿竞争。不要把发表作为唯一指标。先让工作可复现、结论可信、问题明确，再逐轮提高创新性。

第五部分 零基础核心自学讲义

这一部分不是知识目录，而是后面 168 天任务的最小教材。先连续读完对应章节，再完成每日实践。概念学习一律采用四步：先说清对象和输入输出，再追踪形状或数据流，然后运行最小例子，最后改变一个条件验证理解。

先补够用的数学直觉

标量 向量 矩阵和张量

标量是一个数，例如损失 0.42。向量是一列有顺序的数，例如一个样本的 768 维表示。矩阵有行和列，例如 32 个样本、每个 768 维的 batch 可写成 32 乘 768。三维及以上的数组常统称张量，例如图像 batch 的形状通常是 batch 乘通道乘高度乘宽度。

形状不是排版信息，而是数据语义。若 x 的形状为 B 乘 D ，权重 W 为 D 乘 C ，则 xW 的形状为 B 乘 C 。中间的 D 必须相同，因为每个输出都要把 D 个输入特征与 D 个权重配对相乘再相加。调试模型时，先在每个关键节点写出字母形状，再代入具体数字。

广播表示较小数组在缺失维度上被逻辑扩展。例如 B 乘 D 的特征减去长度 D 的均值向量，相当于每个样本都减同一组均值。广播方便，但也可能让错误形状悄悄运行，因此要确认扩展的维度正是你想共享的维度。

```
# 形状推导示例
# x: [B, D] = [8, 10]
# W: [D, C] = [10, 3]
# logits = x @ W: [B, C] = [8, 3]
import torch
x = torch.randn(8, 10)
W = torch.randn(10, 3)
print((x @ W).shape)
```

概率 对数和损失

分类模型通常输出 logits，它们是尚未归一化的分数。softmax 把一组 logits 转成和为一的概率。某个正确类别的 logit 相对其他类别越大，它的概率越高。实际训练中把原始 logits 直接交给 CrossEntropyLoss，因为该函数会用数值更稳定的方式完成 $\log \text{softmax}$ 和负对数似然。

对数把乘法变成加法，也把很小的概率转成可计算的数。负对数损失对自信但错误的预测惩罚很大：正确类别概率接近一时损失接近零，概率接近零时损失迅速增大。不要把低损失直接解释为模型理解了世界，它只说明模型在当前数据和目标下更符合标签。

均值和标准差用于描述多次运行。均值反映中心水平，标准差反映不同随机种子之间的波动。只有一次运行时不能报告训练波动，也不应把一次偶然最高值当成稳定提升。

导数 梯度和链式法则

导数描述输入发生很小变化时输出怎样变化。多参数模型中，每个参数都有一个偏导数，所有偏导数组成梯度。梯度指向损失上升最快的局部方向，所以梯度下降用新参数等于旧参数减去学习率乘梯度。

链式法则处理连续函数。若参数 w 影响预测 y ，预测 y 又影响损失 L ，那么 w 对 L 的影响等于 L 对 y 的影响乘 y 对 w 的影响。反向传播从最终损失开始，沿计算图反向复用这些局部导数。

一个数值例子： x 等于 2， w 等于 3， b 等于 1，目标 t 等于 9。预测 y 等于 $w x$ 加 b ，得到 7；平方损失 L 等于 $(y-t)$ 的平方，得到 4。 dL/dy 等于 $2(y-t)$ ，为负 4； dy/dw 等于 x ，为 2，因此 dL/dw 等于负 8； dL/db 等于负 4。学习率为 0.1 时， w 更新为 3.8， b 更新为 1.4，新预测正好为 9。这个例子展示了梯度符号和更新方向。

```
import torch
x = torch.tensor(2.0)
t = torch.tensor(9.0)
w = torch.tensor(3.0, requires_grad=True)
b = torch.tensor(1.0, requires_grad=True)
y = w * x + b
loss = (y - t) ** 2
loss.backward()
print(y.item(), loss.item(), w.grad.item(), b.grad.item())
```


Python 从读懂到独立写

程序 数据和控制流

一段可靠脚本应有明确入口、输入、处理和输出。变量名表达含义，类型决定允许的操作。字符串用于文本，整数和浮点数用于数值，布尔值用于条件。input 返回字符串，即使用户输入看起来像数字，也要用 int 或 float 显式转换。

列表适合有顺序的一组同类对象，字典适合通过名字访问字段。研究代码常用列表保存多次运行结果，用字典保存配置和单条样本。索引从零开始，切片右端不包含；访问前先确认长度，避免越界。

if 选择分支，for 遍历序列，while 在条件成立时重复。边界条件必须单独测试，例如零个样本、一个样本、缺失值和恰好等于阈值。不要用复杂的一行表达式替代可读的多步逻辑。

```
runs = [
    {'seed': 1, 'accuracy': 0.82},
    {'seed': 2, 'accuracy': 0.85},
    {'seed': 3, 'accuracy': 0.83},
]
best = max(runs, key=lambda item: item['accuracy'])
print(best)
```

函数 模块 文件和异常

函数把一项任务封装为输入到输出的映射。函数名描述动作，参数提供依赖，return 返回结果。打印只是显示信息，不能替代 return。函数尽量只做一件事，这样数据加载、训练、评价和保存可以分别测试。

模块是 Python 文件，包是含多个模块的目录。把可复用功能放入模块，把运行顺序放入 main。导入时报错时，依次检查虚拟环境、当前目录、文件名、包是否安装以及名称是否拼错。

with open 会在结束时关闭文件。文本与 JSON 适合配置和记录，CSV 适合规则表格。异常处理只捕获能明确处理的错误，并把文件名、行号、参数和原始错误保留下来。空的 except 会把真正问题隐藏。

```
from pathlib import Path
import json

def save_metrics(path, metrics):
    path = Path(path)
    path.parent.mkdir(parents=True, exist_ok=True)
    with path.open('w', encoding='utf-8') as file:
        json.dump(metrics, file, ensure_ascii=False, indent=2)

if __name__ == '__main__':
    save_metrics('outputs/run01/metrics.json', {'accuracy': 0.85})
```

环境 Git 和报错处理

虚拟环境隔离项目依赖。激活后用 python -m pip 安装，能减少 pip 与 python 指向不同环境的问题。保存 requirements 或环境文件，同时记录 Python、PyTorch、CUDA 和显卡信息。CUDA 安装命令必须按 PyTorch 官方当前选择器生成。

Git 的最小循环是 status 查看变化，diff 检查内容，add 选择文件，commit 保存一个可运行里程碑。不要提交密钥、账号、原始敏感数据、大模型权重和临时缓存。

报错处理顺序是：保留完整错误；读取最后一行的类型和信息；定位最早由自己代码触发的位置；打印输入类型、shape、device 和路径；缩小到一个样本；确认修复后删除临时打印并写入日志。

```
python -m venv .venv
.\.venv\Scripts\Activate.ps1
python -m pip install --upgrade pip
python -m pip install numpy pandas matplotlib scikit-learn jupyter
python -m pip freeze > requirements-lock.txt
git status
git diff
```

PyTorch 和反向传播

Tensor 自动微分和设备

Tensor 的四项基本属性是 shape、dtype、device 和 requires_grad。图像通常用 float32，分类标签通常用 int64。参与同一运算的张量应位于同一设备；从 NumPy 转换后还要检查类型和内存共享。

requires_grad 为真时，PyTorch 记录产生张量的运算。最终损失应是标量，调用 backward 后，叶子参数的 grad 保存梯度。梯度默认累积，因此每次更新前要清零。验证和推理用 no_grad 或 inference_mode，减少内存并避免构建计算图。

不要随意使用 detach、item 或把张量转成 NumPy，它们会切断计算图。若某个参数没有梯度，先检查它是否在 model.parameters 中、是否被 no_grad 包围、是否经过不可导或断图操作。

```
for name, parameter in model.named_parameters():
    print(name, parameter.shape, parameter.requires_grad)

loss.backward()
for name, parameter in model.named_parameters():
    print(name, None if parameter.grad is None else parameter.grad.norm().item())
```

模型 数据和训练循环

nn.Module 的 forward 描述输入怎样变为输出。训练参数必须作为 Parameter 或子模块属性注册。先用一个小 batch 跑前向，逐层打印形状；再算损失和反向，检查梯度；最后让一个 batch 过拟合。这三步通过后才跑完整数据。

Dataset 的 __len__ 返回样本数，__getitem__ 返回一条输入与标签。DataLoader 负责 batch、打乱和并行。随机增强只用于训练，验证和测试必须确定；划分应按样本主体而不是单张图，避免同一主体泄漏到不同集合。

model.train 控制 Dropout 和 BatchNorm 的训练行为，model.eval 切换为评价行为。二者不会自动开关梯度。每轮训练后在验证集计算损失和主指标，根据预先定义的规则保存最佳模型。

```
def train_one_epoch(model, loader, optimizer, criterion, device):
    model.train()
    total_loss = 0.0
    for x, y in loader:
        x, y = x.to(device), y.to(device)
        optimizer.zero_grad(set_to_none=True)
        logits = model(x)
        loss = criterion(logits, y)
        loss.backward()
        optimizer.step()
        total_loss += loss.item() * x.size(0)
    return total_loss / len(loader.dataset)
```

过拟合 指标和调试

欠拟合表现为训练和验证都差，常见原因是模型能力不足、训练不够或特征无效。过拟合表现为训练继续变好而验证变差，常见处理是更多有效数据、正则化、数据增强、早停或降低模型容量。

准确率适合类别大致平衡的分类。类别不平衡时还看 precision、recall、F1、每类结果和混淆矩阵。评价单位必须与研究问题一致，例如病人级任务不能把同一病人的多张图当独立样本。

遇到 NaN 时先检查输入和标签范围、学习率、损失定义、除零和 log 输入，再查看梯度范数。显存不足时先降低 batch、分辨率和序列长度，再考虑梯度累积、混合精度、冻结模块和 checkpointing。

Transformer 和大语言模型

Token Embedding 位置和注意力

token 是模型处理的单元。文本 tokenizer 把字符串拆为词片并映射到编号，Embedding 再把编号映射为 D 维向量。图像可切成 P 乘 P 的 patch，每个 patch 展平并线性投影成视觉 token。位置编码告诉模型 token 的顺序或二维位置。

自注意力先由输入 X 乘三组权重得到 Q、K、V。若 X 形状为 B 乘 L 乘 D，单头 Q、K、V 可为 B 乘 L 乘 d。Q 与 K 转置相乘得到 B 乘 L 乘 L 的分数矩阵，除以根号 d 后在最后一维 softmax，再乘 V 得到 B 乘 L 乘 d。

交叉注意力的 Q 与 K、V 来自不同序列。例如文本问题产生 Q，图像 patch 产生 K 和 V，于是每个文本位置读取相关视觉信息。注意力权重显示模型怎样分配读取比例，但不能自动等同于可靠因果解释。

```
def attention(q, k, v, mask=None):
    scores = q @ k.transpose(-2, -1) / (q.size(-1) ** 0.5)
    if mask is not None:
        scores = scores.masked_fill(mask == 0, float('-inf'))
    weights = scores.softmax(dim=-1)
    return weights @ v, weights
```

多头 掩码 残差和前馈网络

多头注意力把 D 维拆成 H 个 d 维子空间，通常 D 等于 H 乘 d。计算后把各头拼接回 D 维，再经过输出投影。形状变化通常是 B,L,D 到 B,H,L,d，再回到 B,L,D。

因果掩码阻止位置读取未来 token，padding 掩码忽略补齐位置。两种掩码的方向和广播维度不同，必须用小矩阵检查被屏蔽位置是否真的得到近零权重。

一个 Transformer 块还包含残差、LayerNorm 和逐位置前馈网络。pre norm 在子层前归一化，常见写法是 x 加 Attention(LN(x))，再加 FFN(LN(x))。残差使信息和梯度更容易跨层传播。

从预训练到生成 RAG 和 LoRA

自回归预训练对序列中的下一个 token 做交叉熵。模型得到的是条件概率分布，不是数据库。贪心解码稳定但可能重复；采样带来多样性但增加波动。比较解码策略时固定提示、模型版本和随机种子。

指令微调让输入输出形式更符合任务，偏好对齐调整行为倾向。LoRA 把权重更新近似为两个低秩矩阵，只训练少量参数；rank 越大可训练容量和资源通常越高。量化减少每个权重的位数，可能改变速度、兼容性和精度。

RAG 的流程是切分资料、建立索引、检索、组织上下文、生成和引用。答案错误可能来自没有召回、召回错误、上下文太长、模型没有使用证据或资料本身错误，因此评估要分别测检索与生成。

方法	改变参数	引入外部资料	主要作用
提示设计	否	可选	改变任务表达和上下文
LoRA	是 少量	否	低成本适配任务
量化	通常否	否	降低存储和显存
RAG	否	是	提供可更新且可引用的证据

多模态模型核心路线

表示 对齐 融合和生成

表示是把图像、文本、音频或视频变成向量。对齐让语义对应的跨模态表示更接近或更容易匹配。融合让一种模态直接读取另一种模态的细节。生成则根据融合后的条件逐 token 或逐步去噪产生输出。

双塔模型分别编码两种模态，向量可预先计算，检索快，但细粒度交互有限。融合模型把 token 放入同一 Transformer 或使用交叉注意力，表达能力更强但计算量随序列长度增长。很多系统先用双塔召回，再用融合模型重排或回答。

多模态数据不只是图文文件，还包括配对关系、采集来源、许可证、隐私、语言、质量和偏差。错误配对会直接破坏对齐目标；训练前应抽样查看并统计重复、缺失和文本长度。

路线	典型结构	适合任务	主要代价
双塔对齐	图像编码器与文本编码器	检索 零样本分类	细粒度交互有限
对齐后融合	双塔加交叉注意力	匹配 VQA 描述	推理成本更高
视觉接语言模型	视觉编码器加连接模块加 LLM	对话 推理 生成	数据与幻觉问题突出
条件扩散	文本编码器加去噪网络	图像或视频生成	采样慢且评价困难

CLIP 对比学习和检索

CLIP 把一批匹配图文编码为归一化向量，计算所有图像与所有文本的相似度矩阵。对角线是正配对，非对角线通常作为批内负配对。图到文和文到图分别做交叉熵再取平均。

零样本分类把每个类别写成文本提示并编码，图像与所有类别文本比较相似度。提示模板是实验变量，同义词、语言和上下文都可能改变结果。

检索常用 Recall at K：对每个查询，正确配对是否出现在前 K 个结果中。还可报告 median rank 和 mAP。指标之外必须展示成功、近似但错误和完全错误的案例。

```
image_features = torch.nn.functional.normalize(image_features, dim=-1)
text_features = torch.nn.functional.normalize(text_features, dim=-1)
logits = image_features @ text_features.T / temperature
labels = torch.arange(logits.size(0), device=logits.device)
loss_i = torch.nn.functional.cross_entropy(logits, labels)
loss_t = torch.nn.functional.cross_entropy(logits.T, labels)
loss = (loss_i + loss_t) / 2
```

BLIP BLIP 2 和视觉指令模型

BLIP 把图文对比、图文匹配和语言生成结合。对比目标学习全局对齐，匹配目标使用更深交互判断配对，语言目标学习描述。数据过滤或生成伪描述可以提高有效配对质量，但也会引入模型偏差。

BLIP 2 冻结视觉编码器和语言模型，用 Q-Former 的少量可学习查询从视觉特征中提取与语言相关的信息。第一阶段学习图文表示，第二阶段把视觉查询输出连接到语言模型。资源较小时，冻结主干和训练连接模块是常用路线。

LLaVA 类视觉指令模型通常先对齐视觉特征和语言嵌入，再用图文指令数据训练对话。能力可能来自视觉编码器、语言模型、连接结构或数据质量，实验设计要用基线和消融区分。

扩散 视频 音频 Agent 和多模态 RAG

扩散模型的前向过程逐步加噪，训练模型预测噪声或等价目标；反向采样从噪声逐步恢复样本。潜空间扩散在压缩表示中完成去噪，文本条件通过交叉注意力影响生成。比较模型时同时看提示一致性、质量、多样性、失败和推理成本。

视频可以抽帧后编码，再用平均、循环网络或时序 Transformer 聚合。音频可从波形或频谱表示开始。时间顺序是关键信息，随机打乱帧或片段可作为检查模型是否真正利用时序的对照。

Agent 使用语言模型选择检索、OCR、计算或数据库等工具。工具调用必须有结构化参数、超时、重试、权限和日志。多模态 RAG 则把图、文、表格和元数据建立可检索证据链，先评价召回，再评价有证据生成。

从复现到论文

论文阅读和复现

论文第一遍只回答问题、输入输出、核心主张和主指标；第二遍沿方法图追踪张量与训练模块；第三遍审查数据、基线、消融、效率和限制；第四遍对照代码入口、配置和评价脚本。每篇论文最终形成一张阅读卡。

复现前冻结成功标准，例如在相同或明确缩小的设置下恢复主表趋势，而不是必须逐位一致。记录不能一致的条件，包括数据版本、权重、硬件和训练预算。

复现顺序是环境检查、数据抽样、官方最小推理、一个 batch 过拟合、小子集训练、完整评价。每一级失败都先解决，不跨级继续。

检查层	必须回答	常见证据
数据	版本 划分 预处理是否一致	样本清单 哈希 统计 抽样图
模型	结构 权重 冻结模块是否一致	配置 参数量 中间形状
训练	预算 优化器 学习率 种子是否一致	日志 曲线 检查点
评价	规范化 指标 聚合是否一致	预测文件 评价脚本 单元测试

问题 假设 基线和实验

研究方向要缩成可回答的问题。合格问题写清场景、输入、唯一改动、比较对象、结果变量和资源约束。例如：在固定视觉 token 预算下，OCR 置信度路由是否提高中文文档问答准确率，且中位延迟增加不超过百分之十。

假设必须允许失败，同时写零假设。变量表列出自变量、因变量和控制变量。强基线至少包括简单可靠方法和最接近公开方法；数据、训练预算和评价应尽量一致。

主结果回答方法是否有效，消融回答哪个组件起作用，敏感性回答超参数是否稳定，泛化和鲁棒性限定适用范围，效率说明代价。每项实验在运行前写入矩阵，并定义停止和失败规则。

主张	最低证据	不足时怎样改写
方法提高准确率	公平强基线 多种子 均值与波动	在当前设置观察到提升
组件 A 有效	关闭或替换 A 的消融	A 与整体改动共同相关
方法更高效	同硬件同输入的延迟 吞吐 显存	仅参数量更少

主张	最低证据	不足时怎样改写
方法更稳健	预定义扰动或分布外测试	在所测扰动下下降较少

统计 错误分析和可复现性

多个随机种子用于描述训练不确定性，样本级置信区间用于描述测试样本不确定性，两者不能混为一谈。报告运行数、均值、标准差和原始结果。比较同一测试样本上的两种方法时优先考虑配对分析。

错误分析先写标签规范，再盲法标注。错误类别应能指导实验，例如 OCR 失败、对象幻觉、计数、关系、检索未召回和证据未使用。先统计分布，再选代表样例，避免只挑戏剧性案例。

可复现包至少包含 README、环境、数据说明、配置、代码版本、运行命令、日志、指标、预测和图表生成入口。论文中的表图应从冻结结果自动生成。

论文写作和投稿

实验章节先写，确保每个数字有来源。方法章节从问题定义到数据流、模块、损失和训练推理流程。相关工作按问题或方法类别组织。引言说明具体缺口、方法差异、证据和贡献。

摘要给出问题、方法、关键设置、主要数字和限制。结论只陈述实验支持的范围。局限应写明失效条件、未覆盖数据、资源代价、偏见和潜在误用。

投稿前按当年作者指南检查模板、页数、匿名、补充材料、引用、许可、伦理和生成式 AI 政策。任何规则都可能变化，本指南不能替代目标会议当年的正式要求。

零基础学习失败时怎样调整

现象	先做什么	不要做什么
代码完全看不懂	逐行写输入 变量变化和输出，只保留十行以内	同时看多个长项目
训练不下降	一个 batch 过拟合 检查标签 损失 梯度 学习率	立刻换更大模型
显存不足	缩 batch 分辨率 长度 冻结模块 使用累积或混合精度	反复强行重跑同配置
论文看不懂	先看摘要 图一 主表和结论，再补对应概念	从公式第一行硬啃到最后
实验很多但无结论	回到研究问题和主张证据表，删掉无关运行	继续无目的调参

现象	先做什么	不要做什么
进度落后	只补未通过的验收项，保留复盘日	压缩睡眠或跳过验证

第六部分 168 天每日学习指南

前五部分提供知识地图、科研方法和零基础核心讲义，本部分把路线拆成 168 天。你不需要自行设计进度。每天依次完成原理拆解、例子、执行步骤、必交产出、自测和验收；任何一项缺失都不把当天标为完成。

只看文档的默认用法

不打开网页也可以完成主线。每天只需打开本页对应的 Day，先读“今天要懂”和“原理拆解”，再照着“今日执行卡”建立文件夹、运行最小例子、完成一个控制变量对照，最后按“今日验收”保存证据。网页只用于勾选进度和保存笔记，不承载本指南之外的必修知识。

时间	固定动作	必须留下的证据
0 至 15 分钟	复述昨天三点收获，读今天目标，写下预期结果	day README、预期 shape 或指标
15 至 60 分钟	阅读讲义并画输入到输出的数据流	五条自己的解释、一张数据流图
60 至 130 分钟	运行最小实践，只改变一个变量并保存原始输出	代码、命令、日志、一个对照结果
130 至 160 分钟	整理产出，保存图表、预测或实验表	当天必交文件，文件名含 Day 编号
160 至 180 分钟	闭卷自测、记录报错和明天第一步	验收勾选、失败记录、下一步

阅读纪律 遇到不会的名词先读本手册术语表和对应周讲义，再查官方材料；不要同时打开十个教程。当天实践没有运行成功时，记为“进行中”，第二天先补齐，不用进度百分比掩盖未完成。

开始前设置

文件夹 在固定位置建立 mm-research-learning。每周一个 week 文件夹，每天一个 day 文件夹，代码、数据说明、结果和日志放在同一天目录。不要把数据大文件提交到 Git。

```
mm-research-learning/  
  README.md  
  learning_log.md  
  week01_python/  
    day001_environment/  
    day002_types/  
  experiments/  
  papers/  
  figures/
```

Python 环境 Windows 建议先安装 Python 3.11，再用 PowerShell 创建虚拟环境。PyTorch 的 CUDA 安装命令必须从 PyTorch 官方安装选择器获取，因为命令取决于显卡驱动与 CUDA 版本。

```
py -3.11 -m venv .venv
.\.venv\Scripts\Activate.ps1
python -m pip install --upgrade pip
pip install jupyter numpy pandas matplotlib scikit-learn
python --version
pip freeze > requirements-lock.txt
```

每日记录 记录日期、投入时间、今天完成的文件、最重要结论、完整报错、修复方法和明天第一步。只写学了很多无法帮助复现。

阶段	天数	毕业作品	不能跳过的门槛
第一阶段 基础准备	Day 1 至 42	能够独立完成数据处理、训练循环、注意力实现和小模型评测。	口述 代码 重跑 产出全部通过
第二阶段 多模态专项	Day 43 至 70	能够实现迷你 CLIP，并完成带固定评测的多模态检索问答应用。	口述 代码 重跑 产出全部通过
第三阶段 复现与工程	Day 71 至 112	能够复现一篇公开论文的主要趋势，完成消融、错误分析和复现报告。	口述 代码 重跑 产出全部通过
第四阶段 科研与论文	Day 113 至 168	能够定义可证伪问题，完成公平实验、论文初稿和投稿检查。	口述 代码 重跑 产出全部通过

资源原则 本指南给出学习内容和操作路径。软件命令、模型版本、数据下载和会议政策可能变化，真正执行时以官方文档、原始论文和官方仓库当前说明为准。不要从不明网盘下载模型或数据。

第 1 周 Python 环境与基本语法

本周目标 从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。

首选材料 Python 官方教程第 1 至 4 章，Python 3.11 或项目明确支持的版本，Visual Studio Code Python 扩展。

本周小白讲义

Python 程序可以理解为一份精确的操作说明。解释器从上到下执行语句，变量保存当前状态，函数封装可重复步骤，文件把结果保存到程序之外。初学时不要追求语法数量，先反复练习输入、处理、输出和报错定位。

运行代码前先说出预期结果，运行后再比较实际结果。若失败，从报错最后一行开始读：错误类型说明属于语法、名称、类型、索引还是文件问题，行号指出最先检查的位置。保留错误和修复过程，它们是科研日志的一部分。

本周小项目以成绩数据为例，是为了练习真实研究都会用到的数据流。以后把成绩替换为样本、标签和模型指标，程序结构仍然是读取、校验、计算、保存。

本周知识地图

天	主题	核心知识	产出
Day 1	安装环境与认识文件	程序是按顺序执行的明确指令。Python 文件以 py 结尾，终端负责启动程序，编辑器负责修改程序；虚拟环境把本项目的依赖与其他项目隔离。	环境截图或版本记录、hello.py、第一条学习日志
Day 2	变量 类型与表达式	变量是指向值的名字。整数、浮点数、字符串和布尔值支持不同操作；type 可以检查类型，int 和 float 可以做显式转换。	types.py 和一张类型操作对照表
Day 3	列表 字典与索引	列表保存有顺序的一组值，字典用键查找值。索引从零开始，切片左闭右开；科研配置和单条样本经常用字典表示。	containers.py 和一个实验配置字典

天	主题	核心知识	产出
Day 4	条件判断与循环	if 根据真假选择分支，for 逐个处理序列。循环要有明确对象，条件要覆盖边界，例如分数正好等于六十分。	grade_report.py 和边界测试结果
Day 5	函数 模块与作用域	函数把输入映射为输出，参数是函数的输入，return 才是可供其他代码使用的结果。模块让功能按职责拆开，局部变量不会自动泄漏到函数外。	main.py、stats.py 和函数输入输出说明
Day 6	文件读写与错误处理	文件读写让结果在程序结束后仍被保存。with 会自动关闭文件；异常处理只捕获你知道如何处理的错误，不能用空 except 隐藏问题。	可处理坏数据的成绩统计小项目
Day 7	复盘验收	闭卷解释 独立重跑 整理失败补齐记录	周总结和通过记录

本周代码或记录模板

```
name = input('请输入姓名: ')
scores = [72, 85, 91]
average = sum(scores) / len(scores)
print(f'{name} 的平均分是 {average:.1f}')
```

本周常见错误

- 把代码全部写在一个长文件中，导致无法调试。
- 看到报错就重装软件，没有先读报错最后一行。
- 混淆字符串和数字，例如把输入的 3 当成数值直接相加。

Day 1 安装环境与认识文件

今天要懂 程序是按顺序执行的明确指令。Python 文件以 py 结尾，终端负责启动程序，编辑器负责修改程序；虚拟环境把本项目的依赖与其他项目隔离。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“安装环境与认识文件”，合格解释必须包含这一因果关系：程序是按顺序执行的明确指令。Python 文件以 py 结尾，终端负责启动程序，编辑器负责修改程序；虚拟环境把本项目的依赖与其他项目

隔离。。然后把它与本周目标连接起来：从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“安装环境与认识文件”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：新建 mm-research-learning 文件夹，用 `py -3.11 -m venv .venv` 创建环境，激活后运行 `python --version`，并创建 `hello.py` 输出一句文字。。运行后不要只保存截图，还要把关键结果写进“环境截图或版本记录、`hello.py`、第一条学习日志”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“环境截图或版本记录、`hello.py`、第一条学习日志”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Python 官方教程第 1 至 4 章，Python 3.11 或项目明确支持的版本，Visual Studio Code Python 扩展。

今天先准备 建立 week01/day001 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 <code>day001_main.py</code> ；先用最短代码跑通“安装环境与认识文件”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“环境截图或版本记录、 <code>hello.py</code> 、第一条学习日志”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 通读本阶段目标，建立今天的学习文件夹。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“程序是按顺序执行的明确指令。Python 文件以 py 结尾，终端负责启动程序，编辑器负责修改程序；虚拟环境把本项目的依赖与其他项目隔离。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 新建 mm-research-learning 文件夹，用 `py -3.11 -m venv .venv` 创建环境，激活后运行 `python --version`，并创建 `hello.py` 输出一句文字。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：环境截图或版本记录、`hello.py`、第一条学习日志。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“安装环境与认识文件”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“环境截图或版本记录、`hello.py`、第一条学习日志”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“安装环境与认识文件”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“环境截图或版本记录、`hello.py`、第一条学习日志”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：程序是按顺序执行的明确指令。Python 文件以 py 结尾，终端负责启动程序，编辑器负责修改程序；虚拟环境把本项目的依赖与其他项目隔离。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“环境截图或版本记录、hello.py、第一条学习日志”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把代码全部写在一个长文件中，导致无法调试。；看到报错就重装软件，没有先读报错最后一行。；混淆字符串和数字，例如把输入的 3 当成数值直接相加。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 2 变量 类型与表达式

今天要懂 变量是指向值的名字。整数、浮点数、字符串和布尔值支持不同操作；type 可以检查类型，int 和 float 可以做显式转换。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“变量 类型与表达式”，合格解释必须包含这一因果关系：变量是指向值的名字。整数、浮点数、字符串和布尔值支持不同操作；type 可以检查类型，int 和 float 可以做显式转换。。然后把它与本周目标连接起来：从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“变量 类型与表达式”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：分别创建四种类型，完成加减乘除、字符串拼接和比较；故意把字符串与整数相加，阅读并解释 TypeError。。运行后不要只保存截图，还要把关键结果写进“types.py 和一张类型操作对照表”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“types.py 和一张类型操作对照表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Python 官方教程第 1 至 4 章，Python 3.11 或项目明确支持的版本，Visual Studio Code Python 扩展。

今天先准备 建立 week01/day002 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day002_main.py；先用最短代码跑通“变量 类型与表达式”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“types.py 和一张类型操作对照表”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“变量是指向值的名字。整数、浮点数、字符串和布尔值支持不同操作；type 可以检查类型，int 和 float 可以做显式转换。”写不少于五条自己的话，并画出输入、处理、输出。

- 3 六十至一百三十分钟** 分别创建四种类型，完成加减乘除、字符串拼接和比较；故意把字符串与整数相加，阅读并解释 `TypeError`。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：`types.py` 和一张类型操作对照表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“变量 类型与表达式”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“`types.py` 和一张类型操作对照表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“变量 类型与表达式”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“`types.py` 和一张类型操作对照表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：变量是指向值的名字。整数、浮点数、字符串和布尔值支持不同操作；`type` 可以检查类型，`int` 和 `float` 可以做显式转换。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、`shape`、`device`、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“`types.py` 和一张类型操作对照表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把代码全部写在一个长文件中，导致无法调试。；看到报错就重装软件，没有先读报错最后一行。；混淆字符串和数字，例如把输入的 3 当成数值直接相加。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、`shape`、`device` 和中间输出，再记录完整报错。

Day 3 列表 字典与索引

今天要懂 列表保存有顺序的一组值，字典用键查找值。索引从零开始，切片左闭右开；科研配置和单条样本经常用字典表示。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“列表 字典与索引”，合格解释必须包含这一因果关系：列表保存有顺序的一组值，字典用键查找值。索引从零开始，切片左闭右开；科研配置和单条样本经常用字典表示。。然后把它与本周目标连接起来：从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“列表 字典与索引”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：用列表保存五次实验准确率，用字典保存模型名、学习率和批大小；计算最好结果并打印对应配置。。运行后不要只保存截图，还要把关键结果写进“containers.py 和一个实验配置字典”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“containers.py 和一个实验配置字典”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Python 官方教程第 1 至 4 章，Python 3.11 或项目明确支持的版本，Visual Studio Code Python 扩展。

今天先准备 建立 week01/day003 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day003_main.py；先用最短代码跑通“列表 字典与索引”，不要复制整套项目。

阶段	今天具体做什么
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入 “containers.py 和一个实验配置字典”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“列表保存有顺序的一组值，字典用键查找值。索引从零开始，切片左闭右开；科研配置和单条样本经常用字典表示。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用列表保存五次实验准确率，用字典保存模型名、学习率和批大小；计算最好结果并打印对应配置。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：containers.py 和一个实验配置字典。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“列表 字典与索引”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“containers.py 和一个实验配置字典”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“列表 字典与索引”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“containers.py 和一个实验配置字典”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：列表保存有顺序的一组值，字典用键查找值。索引从零开始，切片左闭右开；科研配置和单条样本经常用字典表示。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“containers.py 和一个实验配置字典”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把代码全部写在一个长文件中，导致无法调试。；看到报错就重装软件，没有先读报错最后一行。；混淆字符串和数字，例如把输入的 3 当成数值直接相加。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 4 条件判断与循环

今天要懂 if 根据真假选择分支，for 逐个处理序列。循环要有明确对象，条件要覆盖边界，例如分数正好等于六十分。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“条件判断与循环”，合格解释必须包含这一因果关系：if 根据真假选择分支，for 逐个处理序列。循环要有明确对象，条件要覆盖边界，例如分数正好等于六十分。。然后把它与本周目标连接起来：从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“条件判断与循环”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：遍历成绩列表，按区间给出等级并统计每个等级人数；测试 59、60、89、90 四个边界值。。运行后不要只保存截图，还要把关键结果写进“grade_report.py 和边界测试结果”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“grade_report.py 和边界测试结果”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Python 官方教程第 1 至 4 章，Python 3.11 或项目明确支持的版本，Visual Studio Code Python 扩展。

今天先准备 建立 week01/day004 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day004_main.py；先用最短代码跑通“条件判断与循环”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“grade_report.py 和边界测试结果”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。

- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“if 根据真假选择分支，for 逐个处理序列。循环要有明确对象，条件要覆盖边界，例如分数正好等于六十分。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 遍历成绩列表，按区间给出等级并统计每个等级人数；测试 59、60、89、90 四个边界值。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：grade_report.py 和边界测试结果。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“条件判断与循环”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“grade_report.py 和边界测试结果”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“条件判断与循环”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“grade_report.py 和边界测试结果”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：if 根据真假选择分支，for 逐个处理序列。循环要有明确对象，条件要覆盖边界，例如分数正好等于六十分。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“grade_report.py 和边界测试结果”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把代码全部写在一个长文件中，导致无法调试。；看到报错就重装软件，没有先读报错最后一行。；混淆字符串和数字，例如把输入的 3 当成数值直接相加。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 5 函数 模块与作用域

今天要懂 函数把输入映射为输出，参数是函数的输入，return 才是可供其他代码使用的结果。模块让功能按职责拆开，局部变量不会自动泄漏到函数外。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“函数 模块与作用域”，合格解释必须包含这一因果关系：函数把输入映射为输出，参数是函数的输入，return 才是可供其他代码使用的结果。模块让功能按职责拆开，局部变量不会自动泄漏到函数外。。然后把它与本周目标连接起来：从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“函数 模块与作用域”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：把成绩统计拆成 load_scores、summarize、save_report 三个函数，再从 main.py 导入调用。。运行后不要只保存截图，还要把关键结果写进“main.py、stats.py 和函数输入输出说明”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“main.py、stats.py 和函数输入输出说明”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Python 官方教程第 1 至 4 章，Python 3.11 或项目明确支持的版本，Visual Studio Code Python 扩展。

今天先准备 建立 week01/day005 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day005_main.py；先用最短代码跑通“函数 模块与作用域”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“main.py、stats.py 和函数输入输出说明”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“函数把输入映射为输出，参数是函数的输入，return 才是可供其他代码使用的结果。模块让功能按职责拆开，局部变量不会自动泄漏到函数外。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 把成绩统计拆成 load_scores、summarize、save_report 三个函数，再从 main.py 导入调用。

- 4 一百三十至一百六十分钟** 整理并保存今日产出：main.py、stats.py 和函数输入输出说明。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“函数 模块与作用域”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“main.py、stats.py 和函数输入输出说明”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“函数 模块与作用域”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“main.py、stats.py 和函数输入输出说明”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：函数把输入映射为输出，参数是函数的输入，return 才是可供其他代码使用的结果。模块让功能按职责拆开，局部变量不会自动泄漏到函数外。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“main.py、stats.py 和函数输入输出说明”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把代码全部写在一个长文件中，导致无法调试。；看到报错就重装软件，没有先读报错最后一行。；混淆字符串和数字，例如把输入的 3 当成数值直接相加。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 6 文件读写与错误处理

今天要懂 文件读写让结果在程序结束后仍被保存。with 会自动关闭文件；异常处理只捕获你知道如何处理的错误，不能用空 except 隐藏问题。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“文件读写与错误处理”，合格解释必须包含这一因果关系：文件读写让结果在程序结束后仍被保存。with 会自动关闭文件；异常处理只捕获你知道如何处理的错误，不能用空 except 隐藏问题。。然后把它与本周目标连接起来：从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“文件读写与错误处理”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：读取一个每行一项成绩的文本文件，跳过空行，遇到非法内容时显示行号，最后把统计结果写入 report.txt。。运行后不要只保存截图，还要把关键结果写进“可处理坏数据的成绩统计小项目”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“可处理坏数据的成绩统计小项目”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Python 官方教程第 1 至 4 章，Python 3.11 或项目明确支持的版本，Visual Studio Code Python 扩展。

今天先准备 建立 week01/day006 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day006_main.py；先用最短代码跑通“文件读写与错误处理”，不要复制整套项目。

阶段	今天具体做什么
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“可处理坏数据的成绩统计小项目”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“文件读写让结果在程序结束后仍被保存。with 会自动关闭文件；异常处理只捕获你知道如何处理的错误，不能用空 except 隐藏问题。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 读取一个每行一项成绩的文本文件，跳过空行，遇到非法内容时显示行号，最后把统计结果写入 report.txt。
- 4 一百三十至一百六十分** 整理并保存今日产出：可处理坏数据的成绩统计小项目。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“文件读写与错误处理”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“可处理坏数据的成绩统计小项目”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“文件读写与错误处理”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“可处理坏数据的成绩统计小项目”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：文件读写让结果在程序结束后仍被保存。with 会自动关闭文件；异常处理只捕获你知道如何处理的错误，不能用空 except 隐藏问题。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“可处理坏数据的成绩统计小项目”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把代码全部写在一个长文件中，导致无法调试。；看到报错就重装软件，没有先读报错最后一行。；混淆字符串和数字，例如把输入的 3 当成数值直接相加。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 7 第 1 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。

阶段	今天具体做什么
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：安装环境与认识文件、变量 类型与表达式、列表 字典与索引、条件判断与循环、函数 模块与作用域、文件读写与错误处理。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“可处理坏数据的成绩统计小项目”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：从完全不会编程开始，建立可重复使用的 Python 环境，并能独立完成输入、处理、输出三段式脚本。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 2 周 NumPy Pandas 可视化与 Git

本周目标 能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。

首选材料 NumPy Quickstart, Pandas Getting Started, Matplotlib Tutorials, Pro Git 第 1 至 2 章。

本周小白讲义

NumPy 的核心不是函数数量，而是形状。数组的每一维都代表一个业务含义，例如样本数、特征数、通道、高度和宽度。做任何 reshape、广播或矩阵乘法前，都要在纸上写出旧形状、新形状以及每一维的含义。

Pandas 负责带标签的表格数据。任何清洗都可能改变样本集合，所以先建立数据字典，再逐步记录缺失值、重复值、异常范围和处理前后行数。图表用于回答问题，而不是装饰结果。

Git 保存的是可追溯的代码状态。一次提交只表达一个清晰变化，并让提交后的版本能够运行。数据原文件、模型权重和密钥通常不应直接提交。

本周知识地图

天	主题	核心知识	产出
Day 8	NumPy 数组与形状	NumPy 数组是固定类型的多维数字表。shape 描述每一维长度，ndim 是维数，dtype 是数值类型；模型错误常常首先表现为形状不匹配。	numpy_shapes.py 和形状推导记录
Day 9	索引 广播与矩阵乘法	广播让不同形状的数组按规则扩展，末尾维度必须相等或其中一个为一。矩阵乘法要求前者最后一维等于后者倒数第二维。	broadcast_matmul.py 和手写形状计算
Day 10	Pandas 读取与体检	DataFrame 是带行列标签的表格。读取后先看 head、shape、columns、dtypes、isna 和 duplicated，确认每一行代表什么。	data_audit.md 和 audit.py
Day 11	清洗 分组与合并	清洗规则必须可以解释且可重复。dropna、fillna、groupby 和 merge 都可能改变样本数量，因此每步要记录处理前后行数。	clean_data.csv 和清洗日志

天	主题	核心知识	产出
Day 12	科研图表	折线图适合过程，柱状图适合离散比较，散点图适合两个变量关系。图形只能支持它实际展示的结论，相关性不自动等于因果。	三张论文风格图及每图一句结论
Day 13	Notebook 脚本与 Git	Notebook 适合探索，脚本适合复现。Git 保存代码变化，不替代数据备份；一次提交只包含一个清晰目的。	可从空环境运行的数据分析仓库
Day 14	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('data.csv')
print(df.info())
df.groupby('label')['score'].mean().plot(kind='bar')
plt.tight_layout()
plt.savefig('mean_score.png', dpi=200)
```

本周常见错误

- 不看 shape 就做矩阵运算。
- 在原始数据上直接覆盖修改，无法追溯。
- 图表没有标题、坐标单位、图例或样本数。

Day 8 NumPy 数组与形状

今天要懂 NumPy 数组是固定类型的多维数字表。shape 描述每一维长度，ndim 是维数，dtype 是数值类型；模型错误常常首先表现为形状不匹配。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“NumPy 数组与形状”，合格解释必须包含这一因果关系：NumPy 数组是固定类型的多维数字表。shape 描述每一维长度，ndim 是维数，dtype 是数值类型；模型错误常常首先表现为形状不匹配。。然后把它与本周目标连接起来：能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“NumPy 数组与形状”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：创建形状为 3 乘 4 的数组，完成转置、reshape、按行求均值和按列求最大值，每一步先写预计 shape 再运行。。运行后不要只保存截图，还要把关键结果写进“numpy_shapes.py 和形状推导记录”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“numpy_shapes.py 和形状推导记录”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 NumPy Quickstart, Pandas Getting Started, Matplotlib Tutorials, Pro Git 第 1 至 2 章。

今天先准备 建立 week02/day008 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day008_main.py；先用最短代码跑通“NumPy 数组与形状”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“numpy_shapes.py 和形状推导记录”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“NumPy 数组是固定类型的多维数字表。shape 描述每一维长度，ndim 是维数，dtype 是数值类型；模型错误常常首先表现为形状不匹配。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 创建形状为 3 乘 4 的数组，完成转置、reshape、按行求均值和按列求最大值，每一步先写预计 shape 再运行。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：numpy_shapes.py 和形状推导记录。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“NumPy 数组与形状”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“numpy_shapes.py 和形状推导记录”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“NumPy 数组与形状”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“numpy_shapes.py 和形状推导记录”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：NumPy 数组是固定类型的多维数字表。shape 描述每一维长度，ndim 是维数，dtype 是数值类型；模型错误常常首先表现为形状不匹配。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“numpy_shapes.py 和形状推导记录”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：不看 shape 就做矩阵运算。；在原始数据上直接覆盖修改，无法追溯。；图表没有标题、坐标单位、图例或样本数。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 9 索引 广播与矩阵乘法

今天要懂 广播让不同形状的数组按规则扩展，末尾维度必须相等或其中一个为一。矩阵乘法要求前者最后一维等于后者倒数第二维。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“索引 广播与矩阵乘法”，合格解释必须包含这一因果关系：广播让不同形状的数组按规则扩展，末尾维度必须相等或其中一个为一。矩阵乘法要求前者最后一维等于后者倒数第二维。。然后把它与本周目标连接起来：能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“索引 广播与矩阵乘法”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：对一批 5 乘 3 特征减去长度为 3 的均值向量；再与 3 乘 2 权重相乘，解释结果为什么是 5 乘 2。。运行后不要只保存截图，还要把关键结果写进“broadcast_matmul.py 和手写形状计算”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“broadcast_matmul.py 和手写形状计算”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 NumPy Quickstart, Pandas Getting Started, Matplotlib Tutorials, Pro Git 第 1 至 2 章。

今天先准备 建立 week02/day009 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day009_main.py；先用最短代码跑通“索引广播与矩阵乘法”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“broadcast_matmul.py 和手写形状计算”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“广播让不同形状的数组按规则扩展，末尾维度必须相等或其中一个为一。矩阵乘法要求前者最后一维等于后者倒数第二维。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 对一批 5 乘 3 特征减去长度为 3 的均值向量；再与 3 乘 2 权重相乘，解释结果为什么是 5 乘 2。

4 一百三十至一百六十分钟 整理并保存今日产出：broadcast_matmul.py 和手写形状计算。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“索引广播与矩阵乘法”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“broadcast_matmul.py 和手写形状计算”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“索引广播与矩阵乘法”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“broadcast_matmul.py 和手写形状计算”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：广播让不同形状的数组按规则扩展，末尾维度必须相等或其中一个为一。矩阵乘法要求前者最后一维等于后者倒数第二维。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“broadcast_matmul.py 和手写形状计算”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：不看 shape 就做矩阵运算。；在原始数据上直接覆盖修改，无法追溯。；图表没有标题、坐标单位、图例或样本数。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 10 Pandas 读取与体检

今天要懂 DataFrame 是带行列标签的表格。读取后先看 head、shape、columns、dtypes、isna 和 duplicated，确认每一行代表什么。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“Pandas 读取与体检”，合格解释必须包含这一因果关系：DataFrame 是带行列标签的表格。读取后先看 head、shape、columns、dtypes、isna 和 duplicated，确认每一行代表什么。。然后把它与本周目标连接起来：能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“Pandas 读取与体检”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：读取一个 CSV，写出数据字典，统计缺失值、重复行、类别数量和数值范围，不做任何清洗前先保存体检报告。。运行后不要只保存截图，还要把关键结果写进“data_audit.md 和 audit.py”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“data_audit.md 和 audit.py”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 NumPy Quickstart, Pandas Getting Started, Matplotlib Tutorials, Pro Git 第 1 至 2 章。

今天先准备 建立 week02/day010 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day010_main.py；先用最短代码跑通“Pandas 读取与体检”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。

阶段	今天具体做什么
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入 “data_audit.md 和 audit.py”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕 “DataFrame 是带行列标签的表格。读取后先看 head、shape、columns、dtypes、isna 和 duplicated，确认每一行代表什么。” 写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 读取一个 CSV，写出数据字典，统计缺失值、重复行、类别数量和数值范围，不做任何清洗前先保存体检报告。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：data_audit.md 和 audit.py。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释 “Pandas 读取与体检”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查 “data_audit.md 和 audit.py”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释 “Pandas 读取与体检”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。

3 假设 “data_audit.md 和 audit.py”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。

4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

1 第一题至少应覆盖：DataFrame 是带行列标签的表格。读取后先看 head、shape、columns、dtypes、isna 和 duplicated，确认每一行代表什么。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和 “data_audit.md 和 audit.py”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：不看 shape 就做矩阵运算。；在原始数据上直接覆盖修改，无法追溯。；图表没有标题、坐标单位、图例或样本数。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 11 清洗 分组与合并

今天要懂 清洗规则必须可以解释且可重复。dropna、fillna、groupby 和 merge 都可能改变样本数量，因此每步要记录处理前后行数。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“清洗 分组与合并”，合格解释必须包含这一因果关系：清洗规则必须可以解释且可重复。dropna、fillna、groupby 和 merge 都可能改变样本数量，因此每步要记录处理前后行数。。然后把它与本周目标连接起来：能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“清洗 分组与合并”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：复制原始表到处理变量，处理缺失与重复，按类别计算均值和标准差，再把一张标签

说明表合并进来。。运行后不要只保存截图，还要把关键结果写进 “clean_data.csv 和清洗日志” ，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到 “clean_data.csv 和清洗日志” 。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 NumPy Quickstart, Pandas Getting Started, Matplotlib Tutorials, Pro Git 第 1 至 2 章。

今天先准备 建立 week02/day011 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day011_main.py；先用最短代码跑通 “清洗 分组与合并” ，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入 “clean_data.csv 和清洗日志” ；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“清洗规则必须可以解释且可重复。dropna、fillna、groupby 和 merge 都可能改变样本数量，因此每步要记录处理前后行数。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 复制原始表到处理变量，处理缺失与重复，按类别计算均值和标准差，再把一张标签说明表合并进来。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：clean_data.csv 和清洗日志。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“清洗 分组与合并”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“clean_data.csv 和清洗日志”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“清洗 分组与合并”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“clean_data.csv 和清洗日志”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：清洗规则必须可以解释且可重复。dropna、fillna、groupby 和 merge 都可能改变样本数量，因此每步要记录处理前后行数。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“clean_data.csv 和清洗日志”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：不看 shape 就做矩阵运算。；在原始数据上直接覆盖修改，无法追溯。；图表没有标题、坐标单位、图例或样本数。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 12 科研图表

今天要懂 折线图适合过程，柱状图适合离散比较，散点图适合两个变量关系。图形只能支持它实际展示的结论，相关性不自动等于因果。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“科研图表”，合格解释必须包含这一因果关系：折线图适合过程，柱状图适合离散比较，散点图适合两个变量关系。图形只能支持它实际展示的结论，相关性不自动等于因果。。然后把它与本周目标连接起来：能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“科研图表”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：从同一份数据制作分组柱状图、训练趋势折线图和误差样例散点图，补齐标题、单位、图例、样本数和图片分辨率。。运行后不要只保存截图，还要把关键结果写进“三张论文风格图及每图一句结论”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“三张论文风格图及每图一句结论”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 NumPy Quickstart, Pandas Getting Started, Matplotlib Tutorials, Pro Git 第 1 至 2 章。

今天先准备 建立 week02/day012 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day012_main.py；先用最短代码跑通“科研图表”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“三张论文风格图及每图一句结论”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“折线图适合过程，柱状图适合离散比较，散点图适合两个变量关系。图形只能支持它实际展示的结论，相关性不自动等于因果。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 从同一份数据制作分组柱状图、训练趋势折线图和误差样例散点图，补齐标题、单位、图例、样本数和图片分辨率。
- 4 一百三十至一百六十分** 整理并保存今日产出：三张论文风格图及每图一句结论。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“科研图表”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“三张论文风格图及每图一句结论”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“科研图表”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“三张论文风格图及每图一句结论”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：折线图适合过程，柱状图适合离散比较，散点图适合两个变量关系。图形只能支持它实际展示的结论，相关性不自动等于因果。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“三张论文风格图及每图一句结论”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：不看 shape 就做矩阵运算。；在原始数据上直接覆盖修改，无法追溯。；图表没有标题、坐标单位、图例或样本数。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 13 Notebook 脚本与 Git

今天要懂 Notebook 适合探索，脚本适合复现。Git 保存代码变化，不替代数据备份；一次提交只包含一个清晰目的。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“Notebook 脚本与 Git”，合格解释必须包含这一因果关系：Notebook 适合探索，脚本适合复现。Git 保存代码变化，不替代数据备份；一次提交只包含一个清晰目的。。然后把它与本周目标连接起来：能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把 “Notebook 脚本与 Git”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：把探索 Notebook 整理为 prepare.py、analyze.py、plot.py，创建 README，依次提交数据体检、清洗和画图三个版本。。运行后不要只保存截图，还要把关键结果写进 “可从空环境运行的数据分析仓库”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到 “可从空环境运行的数据分析仓库”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 NumPy Quickstart, Pandas Getting Started, Matplotlib Tutorials, Pro Git 第 1 至 2 章。

今天先准备 建立 week02/day013 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day013_main.py；先用最短代码跑通 “Notebook 脚本与 Git”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入 “可从空环境运行的数据分析仓库”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“Notebook 适合探索，脚本适合复现。Git 保存代码变化，不替代数据备份；一次提交只包含一个清晰目的。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 把探索 Notebook 整理为 prepare.py、analyze.py、plot.py，创建 README，依次提交数据体检、清洗和画图三个版本。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：可从空环境运行的数据分析仓库。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“Notebook 脚本与 Git”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“可从空环境运行的数据分析仓库”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“Notebook 脚本与 Git”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“可从空环境运行的数据分析仓库”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：Notebook 适合探索，脚本适合复现。Git 保存代码变化，不替代数据备份；一次提交只包含一个清晰目的。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“可从空环境运行的数据分析仓库”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：不看 shape 就做矩阵运算。；在原始数据上直接覆盖修改，无法追溯。；图表没有标题、坐标单位、图例或样本数。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 14 第 2 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：NumPy 数组与形状、索引 广播与矩阵乘法、Pandas 读取与体检、清洗 分组与合并、科研图表、Notebook 脚本与 Git。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“可从空环境运行的数据分析仓库”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：能读取真实表格，检查数据质量，完成统计和可解释图表，并用 Git 保存每个可运行版本。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 3 周 PyTorch 张量与模型

本周目标 理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。

首选材料 PyTorch Learn the Basics，官方 Tensors、Datasets and DataLoaders、Build Model 三节。

本周小白讲义

Tensor 是带有形状、数据类型和设备信息的多维数组。PyTorch 额外提供自动微分和 GPU 计算。进入模型前检查 shape、dtype、device，往往比盯着网络结构更快找到错误。

nn.Module 负责注册参数和定义前向过程。只要参数没有注册到模块中，优化器就看不到它。Dataset 定义单个样本，DataLoader 把样本组成 batch；先检查一个样本，再检查一个 batch，最后才训练。

线性回归是最小训练系统测试。若模型连已知的 y 等于 $3x$ 加 2 都学不会，先修复数据、损失、梯度和更新流程，不要立即换复杂模型。

本周知识地图

天	主题	核心知识	产出
Day 15	Tensor 创建与属性	Tensor 与 NumPy 数组相似，但可在 GPU 上计算并记录梯度。shape、dtype、device 和 requires_grad 是每次调试首先检查的四项。	tensor_basics.py 和属性检查函数
Day 16	变形 拼接与批次维	模型通常接收 batch 乘特征的张量。view 或 reshape 改形状，cat 沿已有维拼接，stack 新增一维；错误使用会改变样本含义。	tensor_ops.py 和形状注释
Day 17	自动微分入门	开启 requires_grad 后，PyTorch 建立计算图。backward 从标量结果反推每个叶子参数的梯度，grad 保存导数；默认梯度会累积。	autograd_demo.py 和手算过程
Day 18	模型层与 Module	nn.Module 管理参数和前向过程。Linear 做仿射变换，激活函数加入非线性；参数只有注册在模块内才会被优化器发现。	两种模型实现和参数量统计

天	主题	核心知识	产出
Day 19	Dataset 与 DataLoader	Dataset 定义如何取得一个样本，DataLoader 负责分批、打乱和并行加载。训练集通常 shuffle，验证集不需要打乱。	dataset.py 和 batch 可视化
Day 20	线性回归小项目	线性回归用权重和偏置拟合连续值。训练前先用已知关系生成数据，确认模型能够恢复近似参数，这是后续复杂模型的 sanity check。	线性回归代码、损失曲线和参数表
Day 21	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
import torch
from torch import nn

model = nn.Sequential(
    nn.Linear(10, 32), nn.ReLU(), nn.Linear(32, 2)
)
x = torch.randn(8, 10)
print(model(x).shape) # torch.Size([8, 2])
```

本周常见错误

- 忘记检查 dtype 和 device。
- 把 batch 维和特征维混淆。
- 只复制网络代码，却说不出每层输入输出形状。

Day 15 Tensor 创建与属性

今天要懂 Tensor 与 NumPy 数组相似，但可在 GPU 上计算并记录梯度。shape、dtype、device 和 requires_grad 是每次调试首先检查的四项。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“Tensor 创建与属性”，合格解释必须包含这一因果关系：Tensor 与 NumPy 数组相似，但可在 GPU 上计算并记录梯度。shape、dtype、device 和 requires_grad 是每次调试首先检查的四项。。然后把它与本周目标连接起来：理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“Tensor 创建与属性”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：用 zeros、ones、randn 和 tensor 创建张量，转换 dtype，打印四项属性，并测试 CPU 环境下的 device。。运行后不要只保存截图，还要把关键结果写进“tensor_basics.py 和属性检查函数”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“tensor_basics.py 和属性检查函数”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 PyTorch Learn the Basics，官方 Tensors、Datasets and DataLoaders、Build Model 三节。

今天先准备 建立 week03/day015 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day015_main.py；先用最短代码跑通“Tensor 创建与属性”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“tensor_basics.py 和属性检查函数”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“Tensor 与 NumPy 数组相似，但可在 GPU 上计算并记录梯度。shape、dtype、device 和 requires_grad 是每次调试首先检查的四项。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用 zeros、ones、randn 和 tensor 创建张量，转换 dtype，打印四项属性，并测试 CPU 环境下的 device。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：tensor_basics.py 和属性检查函数。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“Tensor 创建与属性”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“tensor_basics.py 和属性检查函数”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“Tensor 创建与属性”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“tensor_basics.py 和属性检查函数”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：Tensor 与 NumPy 数组相似，但可在 GPU 上计算并记录梯度。shape、dtype、device 和 requires_grad 是每次调试首先检查的四项。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“tensor_basics.py 和属性检查函数”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：忘记检查 dtype 和 device。；把 batch 维和特征维混淆。；只复制网络代码，却说不出每层输入输出形状。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 16 变形 拼接与批次维

今天要懂 模型通常接收 batch 乘特征的张量。view 或 reshape 改形状，cat 沿已有维拼接，stack 新增一维；错误使用会改变样本含义。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“变形 拼接与批次维”，合格解释必须包含这一因果关系：模型通常接收 batch 乘特征的张量。view 或 reshape 改形状，cat 沿已有维拼接，stack 新增一维；错误使用会改变样本含义。。然后把它与本周目标连接起来：理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“变形 拼接与批次维”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：构造 8 张 3 乘 32 乘 32 图像，展平为 8 乘 3072，再恢复；比较 cat 与 stack 的输出形状。。运行后不要只保存截图，还要把关键结果写进“tensor_ops.py 和形状注释”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“tensor_ops.py 和形状注释”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 PyTorch Learn the Basics，官方 Tensors、Datasets and DataLoaders、Build Model 三节。

今天先准备 建立 week03/day016 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day016_main.py；先用最短代码跑通“变形 拼接与批次维”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“tensor_ops.py 和形状注释”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“模型通常接收 batch 乘特征的张量。view 或 reshape 改形状，cat 沿已有维拼接，stack 新增一维；错误使用会改变样本含义。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 构造 8 张 3 乘 32 乘 32 图像，展平为 8 乘 3072，再恢复；比较 cat 与 stack 的输出形状。
- 4 一百三十至一百六十分** 整理并保存今日产出：tensor_ops.py 和形状注释。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“变形 拼接与批次维”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“tensor_ops.py 和形状注释”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“变形 拼接与批次维”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“tensor_ops.py 和形状注释”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：模型通常接收 batch 乘特征的张量。view 或 reshape 改形状，cat 沿已有维拼接，stack 新增一维；错误使用会改变样本含义。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“tensor_ops.py 和形状注释”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：忘记检查 dtype 和 device。；把 batch 维和特征维混淆。；只复制网络代码，却说不出每层输入输出形状。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 17 自动微分入门

今天要懂 开启 requires_grad 后，PyTorch 建立计算图。backward 从标量结果反推每个叶子参数的梯度，grad 保存导数；默认梯度会累积。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“自动微分入门”，合格解释必须包含这一因果关系：开启 requires_grad 后，PyTorch 建立计算图。backward 从标量结果反推每个叶子参数的梯度，grad 保存导数；默认梯度会累积。。然后把它与本周目标连接起来：理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“自动微分入门”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：对 y 等于 x 平方加 $3x$ 的函数，在 x 等于 2 时比较手算导数和 PyTorch 梯度，并连续 backward 观察累积。。运行后不要只保存截图，还要把关键结果写进“autograd_demo.py 和手算过程”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“autograd_demo.py 和手算过程”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 PyTorch Learn the Basics，官方 Tensors、Datasets and DataLoaders、Build Model 三节。

今天先准备 建立 week03/day017 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day017_main.py；先用最短代码跑通“自动微分入门”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“autograd_demo.py 和手算过程”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“开启 `requires_grad` 后，PyTorch 建立计算图。`backward` 从标量结果反推每个叶子参数的梯度，`grad` 保存导数；默认梯度会累积。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 对 y 等于 x 平方加 $3x$ 的函数，在 x 等于 2 时比较手算导数和 PyTorch 梯度，并连续 `backward` 观察累积。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：`autograd_demo.py` 和手算过程。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“自动微分入门”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“`autograd_demo.py` 和手算过程”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“自动微分入门”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“`autograd_demo.py` 和手算过程”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：开启 `requires_grad` 后，PyTorch 建立计算图。`backward` 从标量结果反推每个叶子参数的梯度，`grad` 保存导数；默认梯度会累积。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、`shape`、`device`、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和 “autograd_demo.py 和手算过程” 。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：忘记检查 dtype 和 device。；把 batch 维和特征维混淆。；只复制网络代码，却说不出每层输入输出形状。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 18 模型层与 Module

今天要懂 nn.Module 管理参数和前向过程。Linear 做仿射变换，激活函数加入非线性；参数只有注册在模块内才会被优化器发现。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于 “模型层与 Module”，合格解释必须包含这一因果关系：nn.Module 管理参数和前向过程。

Linear 做仿射变换，激活函数加入非线性；参数只有注册在模块内才会被优化器发现。。然后把它与本周目标连接起来：理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把 “模型层与 Module” 缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：用 Sequential 和自定义 Module 分别实现 10 到 32 到 2 的网络，打印 named_parameters 和每层输出形状。。运行后不要只保存截图，还要把关键结果写进 “两种模型实现和参数量统计”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到 “两种模型实现和参数量统计” 。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 PyTorch Learn the Basics，官方 Tensors、Datasets and DataLoaders、Build Model 三节。

今天先准备 建立 week03/day018 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day018_main.py；先用最短代码跑通“模型层与 Module”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“两种模型实现和参数量统计”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“nn.Module 管理参数和前向过程。Linear 做仿射变换，激活函数加入非线性；参数只有注册在模块内才会被优化器发现。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用 Sequential 和自定义 Module 分别实现 10 到 32 到 2 的网络，打印 named_parameters 和每层输出形状。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：两种模型实现和参数量统计。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“模型层与 Module”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“两种模型实现和参数量统计”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“模型层与 Module”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“两种模型实现和参数量统计”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：nn.Module 管理参数和前向过程。Linear 做仿射变换，激活函数加入非线性；参数只有注册在模块内才会被优化器发现。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“两种模型实现和参数量统计”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：忘记检查 dtype 和 device。；把 batch 维和特征维混淆。；只复制网络代码，却说不出每层输入输出形状。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 19 Dataset 与 DataLoader

今天要懂 Dataset 定义如何取得一个样本，DataLoader 负责分批、打乱和并行加载。训练集通常 shuffle，验证集不需要打乱。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“Dataset 与 DataLoader”，合格解释必须包含这一因果关系：Dataset 定义如何取得一个样本，DataLoader 负责分批、打乱和并行加载。训练集通常 shuffle，验证集不需要打乱。。然后把它与本周目标连接起来：理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把 “Dataset 与 DataLoader” 缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：为二维点分类数据实现自定义 Dataset，分别建立训练和验证 DataLoader，检查第一批输入标签是否对应。。运行后不要只保存截图，还要把关键结果写进 “dataset.py 和 batch 可视化”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到 “dataset.py 和 batch 可视化”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 PyTorch Learn the Basics，官方 Tensors、Datasets and DataLoaders、Build Model 三节。

今天先准备 建立 week03/day019 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day019_main.py；先用最短代码跑通 “Dataset 与 DataLoader”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入 “dataset.py 和 batch 可视化”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“Dataset 定义如何取得一个样本，DataLoader 负责分批、打乱和并行加载。训练集通常 shuffle，验证集不需要打乱。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 为二维点分类数据实现自定义 Dataset，分别建立训练和验证 DataLoader，检查第一批输入标签是否对应。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：dataset.py 和 batch 可视化。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“Dataset 与 DataLoader”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“dataset.py 和 batch 可视化”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“Dataset 与 DataLoader”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“dataset.py 和 batch 可视化”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：Dataset 定义如何取得一个样本，DataLoader 负责分批、打乱和并行加载。训练集通常 shuffle，验证集不需要打乱。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“dataset.py 和 batch 可视化”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：忘记检查 dtype 和 device。；把 batch 维和特征维混淆。；只复制网络代码，却说不出每层输入输出形状。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 20 线性回归小项目

今天要懂 线性回归用权重和偏置拟合连续值。训练前先用已知关系生成数据，确认模型能够恢复近似参数，这是后续复杂模型的 sanity check。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“线性回归小项目”，合格解释必须包含这一因果关系：线性回归用权重和偏置拟合连续值。训练前先用已知关系生成数据，确认模型能够恢复近似参数，这是后续复杂模型的 sanity check。。然后把它与本周目标连接起来：理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“线性回归小项目”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：生成 y 等于 $3x$ 加 2 再加小噪声的数据，训练 Linear 模型，画预测线和真实点，并比较学到的参数。。运行后不要只保存截图，还要把关键结果写进“线性回归代码、损失曲线和参数表”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“线性回归代码、损失曲线和参数表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 PyTorch Learn the Basics, 官方 Tensors、Datasets and DataLoaders、Build Model 三节。

今天先准备 建立 week03/day020 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day020_main.py；先用最短代码跑通“线性回归小项目”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“线性回归代码、损失曲线和参数表”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“线性回归用权重和偏置拟合连续值。训练前先用已知关系生成数据，确认模型能够恢复近似参数，这是后续复杂模型的 sanity check。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 生成 y 等于 $3x$ 加 2 再加小噪声的数据，训练 Linear 模型，画预测线和真实点，并比较学到的参数。
- 4 一百三十至一百六十分** 整理并保存今日产出：线性回归代码、损失曲线和参数表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“线性回归小项目”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。

- □ 打开并检查“线性回归代码、损失曲线和参数表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“线性回归小项目”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“线性回归代码、损失曲线和参数表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：线性回归用权重和偏置拟合连续值。训练前先用已知关系生成数据，确认模型能够恢复近似参数，这是后续复杂模型的 sanity check。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“线性回归代码、损失曲线和参数表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：忘记检查 dtype 和 device。；把 batch 维和特征维混淆。；只复制网络代码，却说不出每层输入输出形状。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 21 第 3 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：Tensor 创建与属性、变形 拼接与批次维、自动微分入门、模型层与 Module、Dataset 与 DataLoader、线性回归小项目。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“线性回归代码、损失曲线和参数表”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：理解 Tensor、自动微分、模型层和数据管道，能够在 CPU 上运行最小神经网络。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 4 周 训练循环与反向传播

本周目标 能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。

首选材料 PyTorch Automatic Differentiation 和 Optimization Tutorial, Dive into Deep Learning 多层感知机章节。

本周小白讲义

训练循环的本质是重复五件事：得到一批数据、前向得到预测、计算标量损失、反向得到梯度、更新参数。验证循环只测量，不更新。训练和验证必须使用明确分开的数据和模式。

反向传播只是链式法则的高效实现。梯度表示参数发生微小变化时损失怎样变化；学习率控制每次沿负梯度走多远。损失下降不等于泛化变好，因此必须同时看验证集和错误样例。

研究中的公平比较要求一次只改变一个主要因素，并保存数据划分、随机种子、配置和指标。正则化、早停和学习率都应在验证集选择，测试集只用于最后报告。

本周知识地图

天	主题	核心知识	产出
Day 22	分类输出与交叉熵	多分类模型输出每个类别的 logits，不需要先手动 softmax 再交给 CrossEntropyLoss。损失把真实类别位置的概率与其他类别一起考虑。	loss_demo.py 和损失变化表
Day 23	梯度下降与学习率	梯度指出损失上升最快的方向，优化器沿相反方向更新参数。学习率太大可能震荡，太小则收敛缓慢；它是训练最敏感的超参数之一。	学习率对比图和结论
Day 24	完整训练循环	标准循环包含训练模式、清梯度、前向、损失、反向和更新。日志至少记录 epoch、训练损失、验证损失和主指标。	train.py、evaluate.py 和 metrics.csv
Day 25	验证集与过拟合	训练集用于拟合，验证集用于选择超参数，测试集只用于最终报告。训练损失继续下降而验证损失上升通常表示过拟合。	数据划分说明和最佳模型

天	主题	核心知识	产出
Day 26	正则化与稳定训练	权重衰减限制参数过大，Dropout 随机屏蔽部分激活，早停根据验证集结束训练。它们不是越强越好，需要在验证集选择。	三项正则化消融表
Day 27	图像分类综合实验	完整实验除了准确率，还要有类别分布、混淆矩阵和错误样例。错误分析能发现数据问题、类别混淆和模型偏差。	可复现图像分类项目
Day 28	复盘验收	闭卷解释 独立重跑 整理失败补齐记录	周总结和通过记录

本周代码或记录模板

```
for x, y in train_loader:
    model.train()
    optimizer.zero_grad()
    logits = model(x)
    loss = criterion(logits, y)
    loss.backward()
    optimizer.step()
```

本周常见错误

- 忘记 `optimizer.zero_grad` 导致梯度累计。
- 验证阶段仍更新参数。
- 根据测试集反复调参，造成测试信息泄漏。

Day 22 分类输出与交叉熵

今天要懂 多分类模型输出每个类别的 logits，不需要先手动 softmax 再交给 `CrossEntropyLoss`。损失把真实类别位置的概率与其他类别一起考虑。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“分类输出与交叉熵”，合格解释必须包含这一因果关系：多分类模型输出每个类别的 logits，不需要先手动 softmax 再交给 `CrossEntropyLoss`。损失把真实类别位置的概率与其他类别一起考虑。。然后把它与本周目标连接起来：能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“分类输出与交叉熵”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：为三分类玩具样本构造 logits 和标签，计算交叉熵，改变正确类别 logit 并观察损失变化。。运行后不要只保存截图，还要把关键结果写进“loss_demo.py 和损失变化表”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“loss_demo.py 和损失变化表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 PyTorch Automatic Differentiation 和 Optimization Tutorial, Dive into Deep Learning 多层感知机章节。

今天先准备 建立 week04/day022 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day022_main.py；先用最短代码跑通“分类输出与交叉熵”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“loss_demo.py 和损失变化表”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“多分类模型输出每个类别的 logits，不需要先手动 softmax 再交给 CrossEntropyLoss。损失把真实类别位置的概率与其他类别一起考虑。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 为三分类玩具样本构造 logits 和标签，计算交叉熵，改变正确类别 logit 并观察损失变化。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：loss_demo.py 和损失变化表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“分类输出与交叉熵”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“loss_demo.py 和损失变化表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“分类输出与交叉熵”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“loss_demo.py 和损失变化表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：多分类模型输出每个类别的 logits，不需要先手动 softmax 再交给 CrossEntropyLoss。损失把真实类别位置的概率与其他类别一起考虑。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和 “loss_demo.py 和损失变化表” 。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：忘记 optimizer.zero_grad 导致梯度累计。；验证阶段仍更新参数。；根据测试集反复调参，造成测试信息泄漏。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 23 梯度下降与学习率

今天要懂 梯度指出损失上升最快的方向，优化器沿相反方向更新参数。学习率太大可能震荡，太小则收敛缓慢；它是训练最敏感的超参数之一。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“梯度下降与学习率”，合格解释必须包含这一因果关系：梯度指出损失上升最快的方向，优化器沿相反方向更新参数。学习率太大可能震荡，太小则收敛缓慢；它是训练最敏感的超参数之一。。然后把它与本周目标连接起来：能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“梯度下降与学习率”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：对同一模型比较 0.1、0.01、0.001 三个学习率，固定其他设置，画出前五十步损失。。运行后不要只保存截图，还要把关键结果写进“学习率对比图和结论”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“学习率对比图和结论”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 PyTorch Automatic Differentiation 和 Optimization Tutorial, Dive into Deep Learning 多层感知机章节。

今天先准备 建立 week04/day023 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day023_main.py；先用最短代码跑通“梯度下降与学习率”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“学习率对比图和结论”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“梯度指出损失上升最快的方向，优化器沿相反方向更新参数。学习率太大可能震荡，太小则收敛缓慢；它是训练最敏感的超参数之一。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 对同一模型比较 0.1、0.01、0.001 三个学习率，固定其他设置，画出前五十步损失。
- 4 一百三十至一百六十分** 整理并保存今日产出：学习率对比图和结论。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“梯度下降与学习率”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“学习率对比图和结论”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“梯度下降与学习率”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“学习率对比图和结论”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：梯度指出损失上升最快的方向，优化器沿相反方向更新参数。学习率太大可能震荡，太小则收敛缓慢；它是训练最敏感的超参数之一。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“学习率对比图和结论”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：忘记 `optimizer.zero_grad` 导致梯度累计。；验证阶段仍更新参数。；根据测试集反复调参，造成测试信息泄漏。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 24 完整训练循环

今天要懂 标准循环包含训练模式、清梯度、前向、损失、反向和更新。日志至少记录 epoch、训练损失、验证损失和主指标。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“完整训练循环”，合格解释必须包含这一因果关系：标准循环包含训练模式、清梯度、前向、损失、反向和更新。日志至少记录 epoch、训练损失、验证损失和主指标。。然后把它与本周目标连接起来：能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“完整训练循环”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：不复制示例，独立写一个 `train_one_epoch`，再写 `evaluate`；在每轮结束保存指标到 CSV。。运行后不要只保存截图，还要把关键结果写进“`train.py`、`evaluate.py` 和 `metrics.csv`”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“`train.py`、`evaluate.py` 和 `metrics.csv`”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 PyTorch Automatic Differentiation 和 Optimization Tutorial, Dive into Deep Learning 多层感知机章节。

今天先准备 建立 `week04/day024` 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 <code>day024_main.py</code> ；先用最短代码跑通“完整训练循环”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“ <code>train.py</code> 、 <code>evaluate.py</code> 和 <code>metrics.csv</code> ”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“标准循环包含训练模式、清梯度、前向、损失、反向和更新。日志至少记录 epoch、训练损失、验证损失和主指标。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 不复制示例，独立写一个 `train_one_epoch`，再写 `evaluate`；在每轮结束保存指标到 CSV。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：`train.py`、`evaluate.py` 和 `metrics.csv`。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“完整训练循环”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“`train.py`、`evaluate.py` 和 `metrics.csv`”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“完整训练循环”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“`train.py`、`evaluate.py` 和 `metrics.csv`”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：标准循环包含训练模式、清梯度、前向、损失、反向和更新。日志至少记录 epoch、训练损失、验证损失和主指标。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“train.py、evaluate.py 和 metrics.csv”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：忘记 optimizer.zero_grad 导致梯度累计。；验证阶段仍更新参数。；根据测试集反复调参，造成测试信息泄漏。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 25 验证集与过拟合

今天要懂 训练集用于拟合，验证集用于选择超参数，测试集只用于最终报告。训练损失继续下降而验证损失上升通常表示过拟合。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“验证集与过拟合”，合格解释必须包含这一因果关系：训练集用于拟合，验证集用于选择超参数，测试集只用于最终报告。训练损失继续下降而验证损失上升通常表示过拟合。。然后把它与本周目标连接起来：能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“验证集与过拟合”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：把数据拆成训练、验证、测试三部分，训练足够多轮，找出验证性能最佳的轮次并保存检查点。。运行后不要只保存截图，还要把关键结果写进“数据划分说明和最佳模型”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“数据划分说明和最佳模型”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 PyTorch Automatic Differentiation 和 Optimization Tutorial, Dive into Deep Learning 多层感知机章节。

今天先准备 建立 week04/day025 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一

个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day025_main.py；先用最短代码跑通“验证集与过拟合”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“数据划分说明和最佳模型”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“训练集用于拟合，验证集用于选择超参数，测试集只用于最终报告。训练损失继续下降而验证损失上升通常表示过拟合。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 把数据拆成训练、验证、测试三部分，训练足够多轮，找出验证性能最佳的轮次并保存检查点。
- 4 一百三十至一百六十分** 整理并保存今日产出：数据划分说明和最佳模型。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“验证集与过拟合”，并说明它在本周任务中的作用。

- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“数据划分说明和最佳模型”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“验证集与过拟合”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“数据划分说明和最佳模型”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：训练集用于拟合，验证集用于选择超参数，测试集只用于最终报告。训练损失继续下降而验证损失上升通常表示过拟合。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“数据划分说明和最佳模型”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：忘记 `optimizer.zero_grad` 导致梯度累计。；验证阶段仍更新参数。；根据测试集反复调参，造成测试信息泄漏。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 26 正则化与稳定训练

今天要懂 权重衰减限制参数过大，Dropout 随机屏蔽部分激活，早停根据验证集结束训练。它们不是越强越好，需要在验证集选择。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“正则化与稳定训练”，合格解释必须包含这一因果关系：权重衰减限制参数过大，Dropout 随机屏蔽部分激活，早停根据验证集结束训练。它们不是越强越好，需要在验证集选择。。然后把它与本

周目标连接起来：能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“正则化与稳定训练”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：在基线基础上分别加入 weight decay、Dropout 和早停，一次只改一项，记录验证结果。。运行后不要只保存截图，还要把关键结果写进“三项正则化消融表”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“三项正则化消融表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 PyTorch Automatic Differentiation 和 Optimization Tutorial, Dive into Deep Learning 多层感知机章节。

今天先准备 建立 week04/day026 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day026_main.py；先用最短代码跑通“正则化与稳定训练”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“三项正则化消融表”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“权重衰减限制参数过大，Dropout 随机屏蔽部分激活，早停根据验证集结束训练。它们不是越强越好，需要在验证集选择。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 在基线基础上分别加入 weight decay、Dropout 和早停，一次只改一项，记录验证结果。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：三项正则化消融表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“正则化与稳定训练”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“三项正则化消融表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“正则化与稳定训练”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“三项正则化消融表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：权重衰减限制参数过大，Dropout 随机屏蔽部分激活，早停根据验证集结束训练。它们不是越强越好，需要在验证集选择。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“三项正则化消融表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：忘记 `optimizer.zero_grad` 导致梯度累计。；验证阶段仍更新参数。；根据测试集反复调参，造成测试信息泄漏。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 27 图像分类综合实验

今天要懂 完整实验除了准确率，还要有类别分布、混淆矩阵和错误样例。错误分析能发现数据问题、类别混淆和模型偏差。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“图像分类综合实验”，合格解释必须包含这一因果关系：完整实验除了准确率，还要有类别分布、混淆矩阵和错误样例。错误分析能发现数据问题、类别混淆和模型偏差。。然后把它与本周目标连接起来：能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“图像分类综合实验”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：用 MNIST 或 FashionMNIST 训练 MLP，生成损失曲线、混淆矩阵和至少十个错误样例，给错误分类。。运行后不要只保存截图，还要把关键结果写进“可复现图像分类项目”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“可复现图像分类项目”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 PyTorch Automatic Differentiation 和 Optimization Tutorial, Dive into Deep Learning 多层感知机章节。

今天先准备 建立 week04/day027 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day027_main.py；先用最短代码跑通“图像分类综合实验”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“可复现图像分类项目”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“完整实验除了准确率，还要有类别分布、混淆矩阵和错误样例。错误分析能发现数据问题、类别混淆和模型偏差。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用 MNIST 或 FashionMNIST 训练 MLP，生成损失曲线、混淆矩阵和至少十个错误样例，给错误分类。

- 4 一百三十至一百六十分钟** 整理并保存今日产出：可复现图像分类项目。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“图像分类综合实验”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“可复现图像分类项目”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“图像分类综合实验”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“可复现图像分类项目”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：完整实验除了准确率，还要有类别分布、混淆矩阵和错误样例。错误分析能发现数据问题、类别混淆和模型偏差。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“可复现图像分类项目”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：忘记 `optimizer.zero_grad` 导致梯度累计。；验证阶段仍更新参数。；根据测试集反复调参，造成测试信息泄漏。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 28 第 4 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：分类输出与交叉熵、梯度下降与学习率、完整训练循环、验证集与过拟合、正则化与稳定训练、图像分类综合实验。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“可复现图像分类项目”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：能从头写出分类训练循环，解释每一步为何存在，并识别过拟合、欠拟合和数据泄漏。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。

- □ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- □ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 5 周 Transformer 与注意力

本周目标 理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。

首选材料 The Illustrated Transformer, The Annotated Transformer, Hugging Face Course Transformer Models。

本周小白讲义

Transformer 把输入变成 token 序列。每个 token 先得到向量表示和位置信息，然后通过注意力读取其他 token 的内容。Q 表示查询，K 表示可被匹配的索引，V 表示真正被汇总的信息。

注意力矩阵的最后两维通常是查询长度和键长度。softmax 在键长度上进行，使每个查询对所有键的权重和为一。多头注意力是在不同子空间重复这个过程，再拼接回隐藏维。

残差连接保留原表示，LayerNorm 稳定数值，前馈网络对每个位置独立变换。图像 Transformer 只是把 patch 当成 token，因此理解文本 Transformer 的形状后，多模态结构会清楚很多。

本周知识地图

天	主题	核心知识	产出
Day 29	Token 与 Embedding	token 是模型处理的基本单元，文本常用词片，图像常用 patch。embedding 表把离散编号映射成连续向量，形状常为 batch 乘序列长度乘隐藏维。	token_embedding.py 和标注形状的示意图
Day 30	位置编码	注意力只比较内容，本身不知道顺序。位置编码把序列位置加入表示；图像模型还需要表达二维 patch 的空间位置。	position_encoding.py 和热力图
Day 31	Q K V 与缩放点积注意力	Q 表示当前位置想查询什么，K 表示每个位置可被怎样匹配，V 是被读取的内容。QK 转置产生位置之间的相似度，缩放防止 softmax 过度饱和。	attention_single_head.py 和手算页
Day 32	多头注意力	多头把隐藏维分成多个子空间，各头独立计算注意力后拼接。它允许模型同时学习不同关系，但并不保证每个头都有容易解释的语义。	multi_head_shapes.py 和形状流程

天	主题	核心知识	产出
Day 33	编码器 解码器与掩码	编码器使用双向自注意力理解全部输入；自回归解码器用因果掩码禁止读取未来 token；交叉注意力让解码器读取编码器输出。	mask_demo.py 和结构对比图
Day 34	手写最小 Transformer 块	一个常见块由注意力、残差连接、LayerNorm、前馈网络和第二次残差组成。残差保留原信息，归一化帮助深层训练稳定。	mini_transformer.py 和单元测试
Day 35	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
scores = q @ k.transpose(-2, -1) / (q.size(-1) ** 0.5)
weights = scores.softmax(dim=-1)
output = weights @ v
```

本周常见错误

- 只背 Q K V 名称，不理解矩阵形状。
- 忘记注意力权重是在最后一维做 softmax。
- 把自注意力和交叉注意力视为同一输入。

Day 29 Token 与 Embedding

今天要懂 token 是模型处理的基本单元，文本常用词片，图像常用 patch。embedding 表把离散编号映射成连续向量，形状常为 batch 乘序列长度乘隐藏维。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于 “Token 与 Embedding”，合格解释必须包含这一因果关系：token 是模型处理的基本单元，文本常用词片，图像常用 patch。embedding 表把离散编号映射成连续向量，形状常为 batch 乘序列长度乘隐藏维。。然后把它与本周目标连接起来：理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把 “Token 与 Embedding”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：用一个 tokenizer 查看句子的 token、编号和特殊符号；建立小型 Embedding 层并检查输出形状。。运行后不要只保存截图，还要把关键结果写进 “token_embedding.py 和标注形状的示意图”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到 “token_embedding.py 和标注形状的示意图”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 The Illustrated Transformer, The Annotated Transformer, Hugging Face Course Transformer Models。

今天先准备 建立 week05/day029 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day029_main.py；先用最短代码跑通 “Token 与 Embedding”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入 “token_embedding.py 和标注形状的示意图”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“token 是模型处理的基本单元，文本常用词片，图像常用 patch。embedding 表把离散编号映射成连续向量，形状常为 batch 乘序列长度乘隐藏维。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用一个 tokenizer 查看句子的 token、编号和特殊符号；建立小型 Embedding 层并检查输出形状。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：token_embedding.py 和标注形状的示意图。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“Token 与 Embedding”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“token_embedding.py 和标注形状的示意图”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“Token 与 Embedding”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“token_embedding.py 和标注形状的示意图”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：token 是模型处理的基本单元，文本常用词片，图像常用 patch。embedding 表把离散编号映射成连续向量，形状常为 batch 乘序列长度乘隐藏维。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“token_embedding.py 和标注形状的示意图”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只背 Q K V 名称，不理解矩阵形状。；忘记注意力权重是在最后一维做 softmax。；把自注意力和交叉注意力视为同一输入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 30 位置编码

今天要懂 注意力只比较内容，本身不知道顺序。位置编码把序列位置加入表示；图像模型还需要表达二维 patch 的空间位置。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“位置编码”，合格解释必须包含这一因果关系：注意力只比较内容，本身不知道顺序。位置编码把序列位置加入表示；图像模型还需要表达二维 patch 的空间位置。。然后把它与本周目标连接起来：理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“位置编码”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：生成正弦位置编码，画出前十个位置的部分维度；打乱 token 后比较不含和包含位置编码的差异。。运行后不要只保存截图，还要把关键结果写进“position_encoding.py 和热力图”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“position_encoding.py 和热力图”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 The Illustrated Transformer, The Annotated Transformer, Hugging Face Course Transformer Models。

今天先准备 建立 week05/day030 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day030_main.py；先用最短代码跑通“位置编码”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“position_encoding.py 和热力图”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“注意力只比较内容，本身不知道顺序。位置编码把序列位置加入表示；图像模型还需要表达二维 patch 的空间位置。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 生成正弦位置编码，画出前十个位置的部分维度；打乱 token 后比较不含和包含位置编码的差异。

4 一百三十至一百六十分钟 整理并保存今日产出：position_encoding.py 和热力图。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“位置编码”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“position_encoding.py 和热力图”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“位置编码”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“position_encoding.py 和热力图”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：注意力只比较内容，本身不知道顺序。位置编码把序列位置加入表示；图像模型还需要表达二维 patch 的空间位置。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 第四题应让别人找到输入、环境、运行命令、输出目录和“position_encoding.py 和热力图”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只背 Q K V 名称，不理解矩阵形状。；忘记注意力权重是在最后一维做 softmax。；把自注意力和交叉注意力视为同一输入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 31 Q K V 与缩放点积注意力

今天要懂 Q 表示当前位置想查询什么，K 表示每个位置可被怎样匹配，V 是被读取的内容。QK 转置产生位置之间的相似度，缩放防止 softmax 过度饱和。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“Q K V 与缩放点积注意力”，合格解释必须包含这一因果关系：Q 表示当前位置想查询什么，K 表示每个位置可被怎样匹配，V 是被读取的内容。QK 转置产生位置之间的相似度，缩放防止 softmax 过度饱和。。然后把它与本周目标连接起来：理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“Q K V 与缩放点积注意力”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：用四个 token 的小张量手算一行注意力分数，再用 PyTorch 验证；打印 scores、weights 和 output。。运行后不要只保存截图，还要把关键结果写进“attention_single_head.py 和手算页”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“attention_single_head.py 和手算页”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 The Illustrated Transformer, The Annotated Transformer, Hugging Face Course Transformer Models。

今天先准备 建立 week05/day031 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day031_main.py；先用最短代码跑通“Q K V 与缩放点积注意力”，不要复制整套项目。

阶段	今天具体做什么
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入 “attention_single_head.py 和手算页”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“Q 表示当前位置想查询什么，K 表示每个位置可被怎样匹配，V 是被读取的内容。QK 转置产生位置之间的相似度，缩放防止 softmax 过度饱和。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用四个 token 的小张量手算一行注意力分数，再用 PyTorch 验证；打印 scores、weights 和 output。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：attention_single_head.py 和手算页。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“Q K V 与缩放点积注意力”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“attention_single_head.py 和手算页”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“Q K V 与缩放点积注意力”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“attention_single_head.py 和手算页”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：Q 表示当前位置想查询什么，K 表示每个位置可被怎样匹配，V 是被读取的内容。QK 转置产生位置之间的相似度，缩放防止 softmax 过度饱和。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“attention_single_head.py 和手算页”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只背 Q K V 名称，不理解矩阵形状。；忘记注意力权重是在最后一维做 softmax。；把自注意力和交叉注意力视为同一输入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 32 多头注意力

今天要懂 多头把隐藏维分成多个子空间，各头独立计算注意力后拼接。它允许模型同时学习不同关系，但并不保证每个头都有容易解释的语义。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“多头注意力”，合格解释必须包含这一因果关系：多头把隐藏维分成多个子空间，各头独立计算注意力后拼接。它允许模型同时学习不同关系，但并不保证每个头都有容易解释的语义。。然后把它与本周目标连接起来：理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“多头注意力”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：把 batch 2、长度 5、隐藏维 12 拆成 3 个头，写出 reshape、transpose、拼接的每一步形状。。运行后不要只保存截图，还要把关键结果写进 “multi_head_shapes.py 和形状流程”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到 “multi_head_shapes.py 和形状流程”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 The Illustrated Transformer, The Annotated Transformer, Hugging Face Course Transformer Models。

今天先准备 建立 week05/day032 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day032_main.py；先用最短代码跑通“多头注意力”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入 “multi_head_shapes.py 和形状流程”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。

- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“多头把隐藏维分成多个子空间，各头独立计算注意力后拼接。它允许模型同时学习不同关系，但并不保证每个头都有容易解释的语义。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 把 batch 2、长度 5、隐藏维 12 拆成 3 个头，写出 reshape、transpose、拼接的每一步形状。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：multi_head_shapes.py 和形状流程。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“多头注意力”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“multi_head_shapes.py 和形状流程”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“多头注意力”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“multi_head_shapes.py 和形状流程”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：多头把隐藏维分成多个子空间，各头独立计算注意力后拼接。它允许模型同时学习不同关系，但并不保证每个头都有容易解释的语义。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“multi_head_shapes.py 和形状流程”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只背 Q K V 名称，不理解矩阵形状。；忘记注意力权重是在最后一维做 softmax。；把自注意力和交叉注意力视为同一输入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 33 编码器 解码器与掩码

今天要懂 编码器使用双向自注意力理解全部输入；自回归解码器用因果掩码禁止读取未来 token；交叉注意力让解码器读取编码器输出。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“编码器 解码器与掩码”，合格解释必须包含这一因果关系：编码器使用双向自注意力理解全部输入；自回归解码器用因果掩码禁止读取未来 token；交叉注意力让解码器读取编码器输出。。然后把它与本周目标连接起来：理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“编码器 解码器与掩码”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：画出三种注意力的信息来源，构造上三角因果掩码，验证第一个 token 不能关注后续位置。。运行后不要只保存截图，还要把关键结果写进“mask_demo.py 和结构对比图”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“mask_demo.py 和结构对比图”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 The Illustrated Transformer, The Annotated Transformer, Hugging Face Course Transformer Models。

今天先准备 建立 week05/day033 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day033_main.py；先用最短代码跑通“编码器 解码器与掩码”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“mask_demo.py 和结构对比图”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“编码器使用双向自注意力理解全部输入；自回归解码器用因果掩码禁止读取未来 token；交叉注意力让解码器读取编码器输出。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 画出三种注意力的信息来源，构造上三角因果掩码，验证第一个 token 不能关注后续位置。

4 一百三十至一百六十分钟 整理并保存今日产出：mask_demo.py 和结构对比图。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“编码器 解码器与掩码”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“mask_demo.py 和结构对比图”，其中包含日期、环境或配置、结果和一句结论。

自测题

1 不用资料，用三句话解释“编码器 解码器与掩码”：第一句给定义，第二句说输入输出，第三句说一个失败条件。

2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。

3 假设“mask_demo.py 和结构对比图”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。

4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

1 第一题至少应覆盖：编码器使用双向自注意力理解全部输入；自回归解码器用因果掩码禁止读取未来 token；交叉注意力让解码器读取编码器输出。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“mask_demo.py 和结构对比图”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只背 Q K V 名称，不理解矩阵形状。；忘记注意力权重是在最后一维做 softmax。；把自注意力和交叉注意力视为同一输入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 34 手写最小 Transformer 块

今天要懂 一个常见块由注意力、残差连接、LayerNorm、前馈网络和第二次残差组成。残差保留原信息，归一化帮助深层训练稳定。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“手写最小 Transformer 块”，合格解释必须包含这一因果关系：一个常见块由注意力、残差连接、LayerNorm、前馈网络和第二次残差组成。残差保留原信息，归一化帮助深层训练稳定。。然后把它与本周目标连接起来：理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“手写最小 Transformer 块”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：用 PyTorch 模块实现单个 pre norm Transformer block，在随机输入上检查形状不变并完成反向传播。。运行后不要只保存截图，还要把关键结果写进“mini_transformer.py 和单元测试”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“mini_transformer.py 和单元测试”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 The Illustrated Transformer, The Annotated Transformer, Hugging Face Course Transformer Models。

今天先准备 建立 week05/day034 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day034_main.py；先用最短代码跑通“手写最小 Transformer 块”，不要复制整套项目。

阶段	今天具体做什么
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入 “mini_transformer.py 和单元测试”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“一个常见块由注意力、残差连接、LayerNorm、前馈网络和第二次残差组成。残差保留原信息，归一化帮助深层训练稳定。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用 PyTorch 模块实现单个 pre norm Transformer block，在随机输入上检查形状不变并完成反向传播。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：mini_transformer.py 和单元测试。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“手写最小 Transformer 块”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“mini_transformer.py 和单元测试”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“手写最小 Transformer 块”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“mini_transformer.py 和单元测试”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：一个常见块由注意力、残差连接、LayerNorm、前馈网络和第二次残差组成。残差保留原信息，归一化帮助深层训练稳定。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“mini_transformer.py 和单元测试”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只背 Q K V 名称，不理解矩阵形状。；忘记注意力权重是在最后一维做 softmax。；把自注意力和交叉注意力视为同一输入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 35 第 5 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。

阶段	今天具体做什么
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：Token 与 Embedding、位置编码、Q K V 与缩放点积注意力、多头注意力、编码器 解码器与掩码、手写最小 Transformer 块。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成 “mini_transformer.py 和单元测试”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：理解 token、embedding、位置、注意力、掩码和残差结构，能手写单头注意力并推导形状。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 6 周 大语言模型基础

本周目标 区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。

首选材料 Hugging Face Course Language Models，LoRA 与 QLoRA 原论文，模型官方 model card。

本周小白讲义

语言模型的预训练目标通常是预测下一个 token。生成时，解码策略从概率分布中选择 token；temperature 和 top p 改变选择方式，不会给模型增加新知识。

指令微调、偏好对齐、LoRA、量化和 RAG 解决不同问题。LoRA 改少量参数以适配任务，量化减少权重存储，RAG 在生成前引入可核对的外部材料。评价时必须把准确性、格式、引用、延迟和成本分开测量。

任何开放生成演示都要转化为固定测试集。先写预期要点和错误标签，再运行模型，避免看到答案后改变评分规则。

本周知识地图

天	主题	核心知识	产出
Day 36	预训练与下一个 Token	自回归语言模型根据前文预测下一个 token。预训练学到统计模式和部分知识，但训练目标不是事实核验，也不是自动遵循用户指令。	next_token 实验记录
Day 37	解码与生成参数	贪心解码每步选最大概率，采样允许多样性。temperature 调整分布尖锐度，top p 限制候选集合；比较时一次只改一个参数并固定随机种子。	生成参数对比表
Day 38	指令微调与对齐	指令微调用任务形式的数据训练模型按要求回答。偏好对齐改变输出行为和安全边界，但不能替代领域事实和外部证据。	三种提示的输出与评分规则
Day 39	LoRA 与量化	LoRA 用低秩矩阵表示权重更新，只训练少量参数；量化用更少位数存储权重。前者是适配方法，后者主要降低存储和显存。	参数量和显存估算表

天	主题	核心知识	产出
Day 40	RAG 基础	RAG 先检索资料，再把相关片段放入上下文生成回答。它不修改基础模型参数，质量取决于切分、检索、上下文组织和引用。	最小文本 RAG 和十条测试集
Day 41	模型评估与安全边界	评价必须对应真实任务。准确性、格式、引用、延迟和成本可分别度量；开放生成需要清晰规则或人工复核，单次漂亮回答不是证据。	小模型评测报告和模型卡摘要
Day 42	复盘验收	闭卷解释 独立重跑 整理失败补齐记录	周总结和通过记录

本周代码或记录模板

```
tests = ['解释过拟合', '给出一个反例', '总结这段文字']
for prompt in tests:
    output = generate(prompt, temperature=0)
    save(prompt, output)
```

本周常见错误

- 把模型说得流畅当成事实正确。
- 改变多个生成参数后无法解释结果差异。
- 忽略模型许可证、上下文长度和训练数据限制。

Day 36 预训练与下一个 Token

今天要懂 自回归语言模型根据前文预测下一个 token。预训练学到统计模式和部分知识，但训练目标不是事实核验，也不是自动遵循用户指令。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“预训练与下一个 Token”，合格解释必须包含这一因果关系：自回归语言模型根据前文预测下一个 token。预训练学到统计模式和部分知识，但训练目标不是事实核验，也不是自动遵循用户指令。。然后把它与本周目标连接起来：区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“预训练与下一个 Token”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：用小模型查看同一前缀的下一个 token 概率，改变前文观察分布变化，并记录模型版本。。运行后不要只保存截图，还要把关键结果写进“next_token 实验记录”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“next_token 实验记录”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Hugging Face Course Language Models, LoRA 与 QLoRA 原论文，模型官方 model card。

今天先准备 建立 week06/day036 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day036_main.py；先用最短代码跑通“预训练与下一个 Token”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“next_token 实验记录”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“自回归语言模型根据前文预测下一个 token。预训练学到统计模式和部分知识，但训练目标不是事实核验，也不是自动遵循用户指令。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用小模型查看同一前缀的下一个 token 概率，改变前文观察分布变化，并记录模型版本。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：next_token 实验记录。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“预训练与下一个 Token”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“next_token 实验记录”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“预训练与下一个 Token”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“next_token 实验记录”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：自回归语言模型根据前文预测下一个 token。预训练学到统计模式和部分知识，但训练目标不是事实核验，也不是自动遵循用户指令。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“next_token 实验记录”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把模型说得流畅当成事实正确。；改变多个生成参数后无法解释结果差异。；忽略模型许可证、上下文长度和训练数据限制。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 37 解码与生成参数

今天要懂 贪心解码每步选最大概率，采样允许多样性。temperature 调整分布尖锐度，top p 限制候选集合；比较时一次只改一个参数并固定随机种子。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“解码与生成参数”，合格解释必须包含这一因果关系：贪心解码每步选最大概率，采样允许多样性。temperature 调整分布尖锐度，top p 限制候选集合；比较时一次只改一个参数并固定随机种子。。然后把它与本周目标连接起来：区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“解码与生成参数”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：固定五个提示，比较 temperature 为 0、0.7、1.2 的输出，记录重复、事实错误和格式稳定性。。运行后不要只保存截图，还要把关键结果写进“生成参数对比表”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“生成参数对比表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Hugging Face Course Language Models, LoRA 与 QLoRA 原论文，模型官方 model card。

今天先准备 建立 week06/day037 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day037_main.py；先用最短代码跑通“解码与生成参数”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“生成参数对比表”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“贪心解码每步选最大概率，采样允许多样性。temperature 调整分布尖锐度，top p 限制候选集合；比较时一次只改一个参数并固定随机种子。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 固定五个提示，比较 temperature 为 0、0.7、1.2 的输出，记录重复、事实错误和格式稳定性。
- 4 一百三十至一百六十分** 整理并保存今日产出：生成参数对比表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“解码与生成参数”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“生成参数对比表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“解码与生成参数”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“生成参数对比表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：贪心解码每步选最大概率，采样允许多样性。temperature 调整分布尖锐度，top p 限制候选集合；比较时一次只改一个参数并固定随机种子。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
 - 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。”解释差异。
 - 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
 - 4 第四题应让别人找到输入、环境、运行命令、输出目录和“生成参数对比表”。缺少任一项都不能算可复现。
- 卡住时** 先检查本周常见错误：把模型说得流畅当成事实正确。；改变多个生成参数后无法解释结果差异。；忽略模型许可证、上下文长度和训练数据限制。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 38 指令微调与对齐

今天要懂 指令微调用任务形式的数据训练模型按要求回答。偏好对齐改变输出行为和安全边界，但不能替代领域事实和外部证据。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“指令微调与对齐”，合格解释必须包含这一因果关系：指令微调用任务形式的数据训练模型按要求回答。偏好对齐改变输出行为和安全边界，但不能替代领域事实和外部证据。。然后把它与本周目标连接起来：区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“指令微调与对齐”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：把同一任务写成零样本、带示例和带明确输出格式三种提示，比较遵循程度。。运行后不要只保存截图，还要把关键结果写进“三种提示的输出与评分规则”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“三种提示的输出与评分规则”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Hugging Face Course Language Models, LoRA 与 QLoRA 原论文，模型官方 model card。

今天先准备 建立 week06/day038 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day038_main.py；先用最短代码跑通“指令微调与对齐”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“三种提示的输出与评分规则”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“指令微调用任务形式的数据训练模型按要求回答。偏好对齐改变输出行为和安全边界，但不能替代领域事实和外部证据。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 把同一任务写成零样本、带示例和带明确输出格式三种提示，比较遵循程度。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：三种提示的输出与评分规则。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“指令微调与对齐”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“三种提示的输出与评分规则”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“指令微调与对齐”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“三种提示的输出与评分规则”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：指令微调用任务形式的数据训练模型按要求回答。偏好对齐改变输出行为和安全边界，但不能替代领域事实和外部证据。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“三种提示的输出与评分规则”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把模型说得流畅当成事实正确。；改变多个生成参数后无法解释结果差异。；忽略模型许可证、上下文长度和训练数据限制。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 39 LoRA 与量化

今天要懂 LoRA 用低秩矩阵表示权重更新，只训练少量参数；量化用更少位数存储权重。前者是适配方法，后者主要降低存储和显存。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“LoRA 与量化”，合格解释必须包含这一因果关系：LoRA 用低秩矩阵表示权重更新，只训练少量参数；量化用更少位数存储权重。前者是适配方法，后者主要降低存储和显存。。然后把它与本周目标连接起来：区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“LoRA 与量化”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：选择一个小模型，统计全参数与 LoRA 可训练参数；阅读量化配置并估算权重存储量。。运行后不要只保存截图，还要把关键结果写进“参数量和显存估算表”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“参数量和显存估算表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Hugging Face Course Language Models, LoRA 与 QLoRA 原论文，模型官方 model card。

今天先准备 建立 week06/day039 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day039_main.py；先用最短代码跑通“LoRA 与量化”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“参数量和显存估算表”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“LoRA 用低秩矩阵表示权重更新，只训练少量参数；量化用更少位数存储权重。前者是适配方法，后者主要降低存储和显存。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 选择一个小模型，统计全参数与 LoRA 可训练参数；阅读量化配置并估算权重存储量。
- 4 一百三十至一百六十分** 整理并保存今日产出：参数量和显存估算表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“LoRA 与量化”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“参数量和显存估算表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“LoRA 与量化”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“参数量和显存估算表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：LoRA 用低秩矩阵表示权重更新，只训练少量参数；量化用更少位数存储权重。前者是适配方法，后者主要降低存储和显存。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“参数量和显存估算表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把模型说得流畅当成事实正确。；改变多个生成参数后无法解释结果差异。；忽略模型许可证、上下文长度和训练数据限制。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 40 RAG 基础

今天要懂 RAG 先检索资料，再把相关片段放入上下文生成回答。它不修改基础模型参数，质量取决于切分、检索、上下文组织和引用。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“RAG 基础”，合格解释必须包含这一因果关系：RAG 先检索资料，再把相关片段放入上下文生成回答。它不修改基础模型参数，质量取决于切分、检索、上下文组织和引用。。然后把它与本周目标连接起来：区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“RAG 基础”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：用五段自建资料做关键词或向量检索，要求回答展示来源；设计一个资料不存在时的拒答。。运行后不要只保存截图，还要把关键结果写进“最小文本 RAG 和十条测试集”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“最小文本 RAG 和十条测试集”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Hugging Face Course Language Models，LoRA 与 QLoRA 原论文，模型官方 model card。

今天先准备 建立 week06/day040 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day040_main.py；先用最短代码跑通“RAG 基础”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“最小文本 RAG 和十条测试集”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“RAG 先检索资料，再把相关片段放入上下文生成回答。它不修改基础模型参数，质量取决于切分、检索、上下文组织和引用。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用五段自建资料做关键词或向量检索，要求回答展示来源；设计一个资料不存在时的拒答。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：最小文本 RAG 和十条测试集。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“RAG 基础”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“最小文本 RAG 和十条测试集”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“RAG 基础”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“最小文本 RAG 和十条测试集”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：RAG 先检索资料，再把相关片段放入上下文生成回答。它不修改基础模型参数，质量取决于切分、检索、上下文组织和引用。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“最小文本 RAG 和十条测试集”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把模型说得流畅当成事实正确。；改变多个生成参数后无法解释结果差异。；忽略模型许可证、上下文长度和训练数据限制。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 41 模型评估与安全边界

今天要懂 评价必须对应真实任务。准确性、格式、引用、延迟和成本可分别度量；开放生成需要清晰规则或人工复核，单次漂亮回答不是证据。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“模型评估与安全边界”，合格解释必须包含这一因果关系：评价必须对应真实任务。准确性、格式、引用、延迟和成本可分别度量；开放生成需要清晰规则或人工复核，单次漂亮回答不是证据。。然后把它与本周目标连接起来：区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。

科研连接 把今天的代码视为未来实验管线的一部分。你现在养成的形状检查、版本记录和独立重跑习惯，会直接决定以后能否复现论文结果。

跟着做一个例子

把“模型评估与安全边界”缩成一个可在十分钟内运行的例子。先在注释中写输入、预期输出和至少一个边界条件，再执行：建立二十条固定测试，定义每条的预期要点和错误标签，运行模型并计算通过率。。运行后不要只保存截图，还要把关键结果写进“小模型评测报告和模型卡摘要”，并用一句话说明结果是否符合预期。

完成标准 今天结束前必须得到“小模型评测报告和模型卡摘要”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Hugging Face Course Language Models, LoRA 与 QLoRA 原论文，模型官方 model card。

今天先准备 建立 week06/day041 文件夹，先写一个 README，记录 Python 版本、操作系统、开始时间和今天的预计输出。准备一个最小输入，不要一开始就用大数据：通常 5 至 32 条样本、一个 batch 或一个小张量就足够验证概念。运行前在注释中写出预期的类型、shape、数值范围或指标方向；运行后逐项对照，不用截图替代文字记录。

阶段	今天具体做什么
建立最小版本	创建一个只完成今天目标的文件，例如 day041_main.py；先用最短代码跑通“模型评估与安全边界”，不要复制整套项目。
逐项检查	在关键语句前后打印或记录输入、输出、类型、shape、device 和一个数值样例；发现不符时先停下来修复。
控制变量	只改变一个设置，例如 batch size、学习率、输入顺序或一个边界值，先写预测再运行，比较前后差异。
整理证据	把命令、关键输出、一个失败案例和结论写入“小模型评测报告和模型卡摘要”；别人应能从该文件复现你的结果。

通过标准

- 代码能从新的终端或新的 Notebook 内核运行，不依赖上一次运行残留的变量。
- 能解释关键输入输出、shape 或数据结构，并能复现一个故意制造的边界情况或报错。
- 产出包含运行命令、关键结果、一个失败与修复记录，而不是只有截图。

低算力和时间不足 没有独立显卡时只用 CPU 和 toy data；安装失败时先使用课程已有环境或 Google Colab，禁止把重装环境当作学习本身。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“评价必须对应真实任务。准确性、格式、引用、延迟和成本可分别度量；开放生成需要清晰规则或人工复核，单次漂亮回答不是证据。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 建立二十条固定测试，定义每条的预期要点和错误标签，运行模型并计算通过率。
- 4 一百三十至一百六十分** 整理并保存今日产出：小模型评测报告和模型卡摘要。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“模型评估与安全边界”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“小模型评测报告和模型卡摘要”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“模型评估与安全边界”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“小模型评测报告和模型卡摘要”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：评价必须对应真实任务。准确性、格式、引用、延迟和成本可分别度量；开放生成需要清晰规则或人工复核，单次漂亮回答不是证据。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“小模型评测报告和模型卡摘要”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把模型说得流畅当成事实正确。；改变多个生成参数后无法解释结果差异。；忽略模型许可证、上下文长度和训练数据限制。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 42 第 6 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。

阶段	今天具体做什么
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：预训练与下一个 Token、解码与生成参数、指令微调与对齐、LoRA 与量化、RAG 基础、模型评估与安全边界。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“小模型评测报告和模型卡摘要”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：区分预训练、生成、指令微调、LoRA、量化和 RAG，并能用固定测试集评价小模型。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 7 周 多模态任务 对齐与 CLIP

本周目标 理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。

首选材料 Multimodal Machine Learning Survey，Vision Transformer，CLIP 和 SigLIP 原论文及官方代码。

本周小白讲义

多模态学习的第一层问题是表示，第二层是对齐，第三层是融合。双塔模型分别编码图像和文本并比较相似度，适合检索和零样本分类；交叉注意力让两种 token 深度交互，适合理解细节但成本更高。

CLIP 使用同一批次中的匹配图文作为正样本，其他配对作为负样本。图搜文和文搜图是两个方向，损失也通常对称计算。温度系数决定相似度分布的尖锐程度。

零样本分类仍然依赖提示模板。类别名称、语言和模板都会影响文本向量，因此提示必须作为实验设置保存并做敏感性检查。

本周知识地图

天	主题	核心知识	产出
Day 43	多模态任务全景	多模态学习同时处理文字、图像、音频或视频。核心任务包括跨模态检索、生成、视觉问答和对话；共同难点是表示、对齐、融合、缺失模态和数据偏差。	多模态任务矩阵和系统图
Day 44	图像 Patch 与 ViT	ViT 把图像切成 patch，再把每个 patch 投影成 token。二维空间被转为序列，位置编码保存空间线索，CLS 或池化得到全局表示。	vit_features.py 和形状流程图
Day 45	文本编码器与句向量	文本编码器把 token 序列变成上下文表示。句向量可取特殊 token、平均池化或专门训练的句子表示，不同方法不应直接假设可比。	text_embeddings.py 和相似度矩阵
Day 46	对齐与融合	对齐让对应图文在表示空间靠近，适合高效检索；融合让图像 token 与文本 token 深度交互，适合细粒度推理但计算更贵。	双塔与融合架构对比表

天	主题	核心知识	产出
Day 47	InfoNCE 与温度系数	InfoNCE 把正确图文对视为正样本，批内其他组合视为负样本。温度控制相似度分布尖锐程度，配对索引错误会让训练目标完全失效。	infonce.py、损失曲线和温度消融
Day 48	CLIP 零样本分类	CLIP 把类别写成自然语言提示，与图像特征比较。模板集成能缓解单个措辞偏差，但类别设计和数据分布仍决定上限。	零样本分类报告与提示模板表
Day 49	复盘验收	闭卷解释 独立重跑 整理失败补齐记录	周总结和通过记录

本周代码或记录模板

```
logits = image_features @ text_features.T / temperature
labels = torch.arange(logits.size(0), device=logits.device)
loss = (ce(logits, labels) + ce(logits.T, labels)) / 2
```

本周常见错误

- 把对齐和融合混为一谈。
- 只看总体准确率，不检查提示模板和失败样例。
- 对比学习批次太小或正负样本对应错误。

Day 43 多模态任务全景

今天要懂 多模态学习同时处理文字、图像、音频或视频。核心任务包括跨模态检索、生成、视觉问答和对话；共同难点是表示、对齐、融合、缺失模态和数据偏差。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“多模态任务全景”，合格解释必须包含这一因果关系：多模态学习同时处理文字、图像、音频或视频。核心任务包括跨模态检索、生成、视觉问答和对话；共同难点是表示、对齐、融合、缺失模态和数据偏差。。然后把它与本周目标连接起来：理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：为四类任务各写出输入、输出、训练信号和评价指标，并画编码器、连接模块、解码器总图。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“多模态任务矩阵和系统图”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Multimodal Machine Learning Survey, Vision Transformer, CLIP 和 SigLIP 原论文及官方代码。

今天先准备 建立 week07/day043 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：为四类任务各写出输入、输出、训练信号和评价指标，并画编码器、连接模块、解码器总图。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“多模态任务矩阵和系统图”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“多模态学习同时处理文字、图像、音频或视频。核心任务包括跨模态检索、生成、视觉问答和对话；共同难点是表示、对齐、融合、缺失模态和数据偏差。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 为四类任务各写出输入、输出、训练信号和评价指标，并画编码器、连接模块、解码器总图。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：多模态任务矩阵和系统图。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“多模态任务全景”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“多模态任务矩阵和系统图”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“多模态任务全景”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“多模态任务矩阵和系统图”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：多模态学习同时处理文字、图像、音频或视频。核心任务包括跨模态检索、生成、视觉问答和对话；共同难点是表示、对齐、融合、缺失模态和数据偏差。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“多模态任务矩阵和系统图”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把对齐和融合混为一谈。；只看总体准确率，不检查提示模板和失败样例。；对比学习批次太小或正负样本对应错误。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 44 图像 Patch 与 ViT

今天要懂 ViT 把图像切成 patch，再把每个 patch 投影成 token。二维空间被转为序列，位置编码保存空间线索，CLS 或池化得到全局表示。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“图像 Patch 与 ViT”，合格解释必须包含这一因果关系：ViT 把图像切成 patch，再把每个 patch 投影成 token。二维空间被转为序列，位置编码保存空间线索，CLS 或池化得到全局表示。。然后把它与本周目标连接起来：理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：加载预训练 ViT，给一张 224 乘 224 图像，打印 patch 数、隐藏维和最终特征形状。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“vit_features.py 和形状流程图”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Multimodal Machine Learning Survey，Vision Transformer，CLIP 和 SigLIP 原论文及官方代码。

今天先准备 建立 week07/day044 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。

阶段	今天具体做什么
跑固定样例	对同一批编号样例执行：加载预训练 ViT，给一张 224 乘 224 图像，打印 patch 数、隐藏维和最终特征形状。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“vit_features.py 和形状流程图”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“ViT 把图像切成 patch，再把每个 patch 投影成 token。二维空间被转为序列，位置编码保存空间线索，CLS 或池化得到全局表示。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 加载预训练 ViT，给一张 224 乘 224 图像，打印 patch 数、隐藏维和最终特征形状。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：vit_features.py 和形状流程图。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“图像 Patch 与 ViT”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“vit_features.py 和形状流程图”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“图像 Patch 与 ViT”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“vit_features.py 和形状流程图”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：ViT 把图像切成 patch，再把每个 patch 投影成 token。二维空间被转为序列，位置编码保存空间线索，CLS 或池化得到全局表示。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“vit_features.py 和形状流程图”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把对齐和融合混为一谈。；只看总体准确率，不检查提示模板和失败样例。；对比学习批次太小或正负样本对应错误。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 45 文本编码器与句向量

今天要懂 文本编码器把 token 序列变成上下文表示。句向量可取特殊 token、平均池化或专门训练的句子表示，不同方法不应直接假设可比。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“文本编码器与句向量”，合格解释必须包含这一因果关系：文本编码器把 token 序列变成上下文表示。句向量可取特殊 token、平均池化或专门训练的句子表示，不同方法不应直接假设可比。。然后把它与本周目标连接起来：理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：对五组同义和非同义句提取 embedding，计算余弦相似度，检查语义相似句是否更近。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到 “text_embeddings.py 和相似度矩阵” 。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Multimodal Machine Learning Survey，Vision Transformer，CLIP 和 SigLIP 原论文及官方代码。

今天先准备 建立 week07/day045 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：对五组同义和非同义句提取 embedding，计算余弦相似度，检查语义相似句是否更近。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入 “text_embeddings.py 和相似度矩阵”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“文本编码器把 token 序列变成上下文表示。句向量可取特殊 token、平均池化或专门训练的句子表示，不同方法不应直接假设可比。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 对五组同义和非同义句提取 embedding，计算余弦相似度，检查语义相似句是否更近。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：text_embeddings.py 和相似度矩阵。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“文本编码器与句向量”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“text_embeddings.py 和相似度矩阵”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“文本编码器与句向量”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“text_embeddings.py 和相似度矩阵”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：文本编码器把 token 序列变成上下文表示。句向量可取特殊 token、平均池化或专门训练的句子表示，不同方法不应直接假设可比。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和 “text_embeddings.py 和相似度矩阵” 。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把对齐和融合混为一谈。；只看总体准确率，不检查提示模板和失败样例。；对比学习批次太小或正负样本对应错误。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 46 对齐与融合

今天要懂 对齐让对应图文在表示空间靠近，适合高效检索；融合让图像 token 与文本 token 深度交互，适合细粒度推理但计算更贵。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“对齐与融合”，合格解释必须包含这一因果关系：对齐让对应图文在表示空间靠近，适合高效检索；融合让图像 token 与文本 token 深度交互，适合细粒度推理但计算更贵。。然后把它与本周目标连接起来：理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：用同一图文样例画出双塔对齐和交叉注意力融合两条数据流，列出速度、缓存和精度差异。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“双塔与融合架构对比表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Multimodal Machine Learning Survey, Vision Transformer, CLIP 和 SigLIP 原论文及官方代码。

今天先准备 建立 week07/day046 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。

阶段	今天具体做什么
跑固定样例	对同一批编号样例执行：用同一图文样例画出双塔对齐和交叉注意力融合两条数据流，列出速度、缓存和精度差异。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“双塔与融合架构对比表”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“对齐让对应图文在表示空间靠近，适合高效检索；融合让图像 token 与文本 token 深度交互，适合细粒度推理但计算更贵。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用同一图文样例画出双塔对齐和交叉注意力融合两条数据流，列出速度、缓存和精度差异。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：双塔与融合架构对比表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“对齐与融合”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“双塔与融合架构对比表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“对齐与融合”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“双塔与融合架构对比表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：对齐让对应图文在表示空间靠近，适合高效检索；融合让图像 token 与文本 token 深度交互，适合细粒度推理但计算更贵。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“双塔与融合架构对比表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把对齐和融合混为一谈。；只看总体准确率，不检查提示模板和失败样例。；对比学习批次太小或正负样本对应错误。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 47 InfoNCE 与温度系数

今天要懂 InfoNCE 把正确图文对视为正样本，批内其他组合视为负样本。温度控制相似度分布尖锐程度，配对索引错误会让训练目标完全失效。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“InfoNCE 与温度系数”，合格解释必须包含这一因果关系：InfoNCE 把正确图文对视为正样本，批内其他组合视为负样本。温度控制相似度分布尖锐程度，配对索引错误会让训练目标完全失效。。然后把它与本周目标连接起来：理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：手写对称 InfoNCE，在随机向量上让正确配对逐渐靠近，比较三个温度并画损失。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到 “infonce.py、损失曲线和温度消融” 。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Multimodal Machine Learning Survey, Vision Transformer, CLIP 和 SigLIP 原论文及官方代码。

今天先准备 建立 week07/day047 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：手写对称 InfoNCE，在随机向量上让正确配对逐渐靠近，比较三个温度并画损失。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入 “infonce.py、损失曲线和温度消融”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“InfoNCE 把正确图文对视为正样本，批内其他组合视为负样本。温度控制相似度分布尖锐程度，配对索引错误会让训练目标完全失效。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 手写对称 InfoNCE，在随机向量上让正确配对逐渐靠近，比较三个温度并画损失。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：infonce.py、损失曲线和温度消融。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“InfoNCE 与温度系数”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“infonce.py、损失曲线和温度消融”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“InfoNCE 与温度系数”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“infonce.py、损失曲线和温度消融”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：InfoNCE 把正确图文对视为正样本，批内其他组合视为负样本。温度控制相似度分布尖锐程度，配对索引错误会让训练目标完全失效。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“infonce.py、损失曲线和温度消融”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把对齐和融合混为一谈。；只看总体准确率，不检查提示模板和失败样例。；对比学习批次太小或正负样本对应错误。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 48 CLIP 零样本分类

今天要懂 CLIP 把类别写成自然语言提示，与图像特征比较。模板集成能缓解单个措辞偏差，但类别设计和数据分布仍决定上限。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“CLIP 零样本分类”，合格解释必须包含这一因果关系：CLIP 把类别写成自然语言提示，与图像特征比较。模板集成能缓解单个措辞偏差，但类别设计和数据分布仍决定上限。。然后把它与本周目标连接起来：理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：选择十类小图片集，比较裸类别名、a photo of 和至少五个中文或英文模板，保存混淆矩阵和失败例。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“零样本分类报告与提示模板表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Multimodal Machine Learning Survey, Vision Transformer, CLIP 和 SigLIP 原论文及官方代码。

今天先准备 建立 week07/day048 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。

阶段	今天具体做什么
跑固定样例	对同一批编号样例执行：选择十类小图片集，比较裸类别名、a photo of 和至少五个中文或英文模板，保存混淆矩阵和失败例。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“零样本分类报告与提示模板表”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“CLIP 把类别写成自然语言提示，与图像特征比较。模板集成能缓解单个措辞偏差，但类别设计和数据分布仍决定上限。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 选择十类小图片集，比较裸类别名、a photo of 和至少五个中文或英文模板，保存混淆矩阵和失败例。
- 4 一百三十至一百六十分** 整理并保存今日产出：零样本分类报告与提示模板表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“CLIP 零样本分类”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“零样本分类报告与提示模板表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“CLIP 零样本分类”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“零样本分类报告与提示模板表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：CLIP 把类别写成自然语言提示，与图像特征比较。模板集成能缓解单个措辞偏差，但类别设计和数据分布仍决定上限。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“零样本分类报告与提示模板表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把对齐和融合混为一谈。；只看总体准确率，不检查提示模板和失败样例。；对比学习批次太小或正负样本对应错误。。连续四十分钟没有进展时，缩小为一个样本或一个batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 49 第 7 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。

阶段	今天具体做什么
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：多模态任务全景、图像 Patch 与 ViT、文本编码器与句向量、对齐与融合、InfoNCE 与温度系数、CLIP 零样本分类。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“零样本分类报告与提示模板表”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：理解图像和文本怎样被编码到联合空间，能运行零样本分类并实现迷你对比学习。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 8 周 BLIP 检索 描述与视觉问答

本周目标 掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。

首选材料 BLIP、BLIP 2、VQA 和 ALBEF 原论文，Hugging Face 对应模型文档。

本周小白讲义

BLIP 类模型把图文对齐、匹配和语言生成放进统一框架。对齐损失学习全局语义，匹配损失判断图文是否对应，语言建模损失负责生成文本。不同目标共享数据，但监督信号并不相同。

视觉问答不是只看答案是否流畅，而是检查答案是否由图像支持。数据划分、答案规范化和评价脚本必须固定；计数、OCR、关系和常识问题应分开分析。

BLIP 2 的 Q Former 用少量查询 token 从冻结视觉编码器提取信息，再连接冻结语言模型。理解哪些模块冻结、哪些模块训练，是估算资源和解释能力边界的关键。

本周知识地图

天	主题	核心知识	产出
Day 50	BLIP 的三个目标	ITC 学全局对齐，ITM 判断图文是否匹配，LM 根据图像生成文字。三个目标解决的层次不同，组合后兼顾检索、细粒度交互和生成。	BLIP 目标图和二十条描述
Day 51	CapFilt 与数据质量	网络图文对常含噪声。CapFilt 用描述器生成候选文本，再用过滤器判断匹配，说明数据质量本身可以成为模型改进点。	简化数据过滤实验
Day 52	Q Former	Q Former 使用少量可学习 query 通过交叉注意力读取冻结视觉编码器特征，再映射给冻结语言模型。它把长视觉序列压缩为固定数量的语义查询。	Q Former 数据流与形状表
Day 53	图文检索与 Recall at K	图搜文和文搜图方向不同，Recall at K 表示正确目标是否进入前 K。数据存在多对多标注时，评价代码必须接受多个正确答案。	双向检索指标表和失败页面

天	主题	核心知识	产出
Day 54	视觉问答错误类型	VQA 可能依赖物体识别、计数、空间、OCR、知识或多步推理。总体分数无法告诉你瓶颈，错误类型比例才指导改进。	四十题 VQA 测试集与错误分布
Day 55	图像描述评估	BLEU 重表面 n gram, CIDEr 强调与参考描述一致的信息, CLIPScore 衡量图文语义。它们都可能漏掉事实错误，因此需要人工事实核查。	图像描述评测报告
Day 56	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
similarity = text_emb @ image_emb.T
topk = similarity.topk(k=5, dim=-1).indices
# 对每个 query 检查正确目标是否出现在 top k
```

本周常见错误

- 用 BLEU 一项指标判断生成质量。
- 检索数据中一张图多条描述却按单标签处理。
- 视觉问答只看正确率，不区分感知、OCR、推理和知识错误。

Day 50 BLIP 的三个目标

今天要懂 ITC 学全局对齐，ITM 判断图文是否匹配，LM 根据图像生成文字。三个目标解决的层次不同，组合后兼顾检索、细粒度交互和生成。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“BLIP 的三个目标”，合格解释必须包含这一因果关系：ITC 学全局对齐，ITM 判断图文是否匹配，LM 根据图像生成文字。三个目标解决的层次不同，组合后兼顾检索、细粒度交互和生成。。然后把它与本周目标连接起来：掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：阅读 BLIP 方法图，为三个目标分别标注输入、负样本、输出和损失，运行图像描述示例。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“BLIP 目标图和二十条描述”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 BLIP、BLIP 2、VQA 和 ALBEF 原论文，Hugging Face 对应模型文档。

今天先准备 建立 week08/day050 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：阅读 BLIP 方法图，为三个目标分别标注输入、负样本、输出和损失，运行图像描述示例。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“BLIP 目标图和二十条描述”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“ITC 学全局对齐，ITM 判断图文是否匹配，LM 根据图像生成文字。三个目标解决的层次不同，组合后兼顾检索、细粒度交互和生成。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 阅读 BLIP 方法图，为三个目标分别标注输入、负样本、输出和损失，运行图像描述示例。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：BLIP 目标图和二十条描述。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“BLIP 的三个目标”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“BLIP 目标图和二十条描述”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“BLIP 的三个目标”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“BLIP 目标图和二十条描述”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：ITC 学全局对齐，ITM 判断图文是否匹配，LM 根据图像生成文字。三个目标解决的层次不同，组合后兼顾检索、细粒度交互和生成。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“BLIP 目标图和二十条描述”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：用 BLEU 一项指标判断生成质量。；检索数据中一张图多条描述却按单标签处理。；视觉问答只看正确率，不区分感知、OCR、推理和知识错误。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 51 CapFilt 与数据质量

今天要懂 网络图文对常含噪声。CapFilt 用描述器生成候选文本，再用过滤器判断匹配，说明数据质量本身可以成为模型改进点。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“CapFilt 与数据质量”，合格解释必须包含这一因果关系：网络图文对常含噪声。CapFilt 用描述器生成候选文本，再用过滤器判断匹配，说明数据质量本身可以成为模型改进点。。然后把它与本周目标连接起来：掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：构造二十个正确和错误图文对，用相似度或 ITM 打分过滤，比较过滤前后的样例质量。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“简化数据过滤实验”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 BLIP、BLIP 2、VQA 和 ALBEF 原论文，Hugging Face 对应模型文档。

今天先准备 建立 week08/day051 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。

阶段	今天具体做什么
跑固定样例	对同一批编号样例执行：构造二十个正确和错误图文对，用相似度或 ITM 打分过滤，比较过滤前后的样例质量。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“简化数据过滤实验”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“网络图文对常含噪声。CapFilt 用描述器生成候选文本，再用过滤器判断匹配，说明数据质量本身可以成为模型改进点。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 构造二十个正确和错误图文对，用相似度或 ITM 打分过滤，比较过滤前后的样例质量。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：简化数据过滤实验。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“CapFilt 与数据质量”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“简化数据过滤实验”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“CapFilt 与数据质量”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“简化数据过滤实验”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：网络图文对常含噪声。CapFilt 用描述器生成候选文本，再用过滤器判断匹配，说明数据质量本身可以成为模型改进点。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“简化数据过滤实验”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：用 BLEU 一项指标判断生成质量。；检索数据中一张图多条描述却按单标签处理。；视觉问答只看正确率，不区分感知、OCR、推理和知识错误。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 52 Q Former

今天要懂 Q Former 使用少量可学习 query 通过交叉注意力读取冻结视觉编码器特征，再映射给冻结语言模型。它把长视觉序列压缩为固定数量的语义查询。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“Q Former”，合格解释必须包含这一因果关系：Q Former 使用少量可学习 query 通过交叉注意力读取冻结视觉编码器特征，再映射给冻结语言模型。它把长视觉序列压缩为固定数量的语义查询。。然后把它与本周目标连接起来：掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：沿 BLIP 2 架构图写出图像特征、query、Q Former 输出和 LLM 输入的 shape，运行一个问答示例。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到 “Q Former 数据流与形状表” 。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 BLIP、BLIP 2、VQA 和 ALBEF 原论文，Hugging Face 对应模型文档。

今天先准备 建立 week08/day052 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：沿 BLIP 2 架构图写出图像特征、query、Q Former 输出和 LLM 输入的 shape，运行一个问答示例。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入 “Q Former 数据流与形状表”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“Q Former 使用少量可学习 query 通过交叉注意力读取冻结视觉编码器特征，再映射给冻结语言模型。它把长视觉序列压缩为固定数量的语义查询。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 沿 BLIP 2 架构图写出图像特征、query、Q Former 输出和 LLM 输入的形狀，运行一个问答示例。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：Q Former 数据流与形状表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“Q Former”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“Q Former 数据流与形状表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“Q Former”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“Q Former 数据流与形状表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：Q Former 使用少量可学习 query 通过交叉注意力读取冻结视觉编码器特征，再映射给冻结语言模型。它把长视觉序列压缩为固定数量的语义查询。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“Q Former 数据流与形状表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：用 BLEU 一项指标判断生成质量。；检索数据中一张图多条描述却按单标签处理。；视觉问答只看正确率，不区分感知、OCR、推理和知识错误。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 53 图文检索与 Recall at K

今天要懂 图搜文和文搜图方向不同，Recall at K 表示正确目标是否进入前 K。数据存在多对多标注时，评价代码必须接受多个正确答案。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“图文检索与 Recall at K”，合格解释必须包含这一因果关系：图搜文和文搜图方向不同，Recall at K 表示正确目标是否进入前 K。数据存在多对多标注时，评价代码必须接受多个正确答案。。然后把它与本周目标连接起来：掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：在小数据集计算双向 R at 1、5、10，逐条检查 ground truth 映射并展示五个检索失败案例。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“双向检索指标表和失败页面”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 BLIP、BLIP 2、VQA 和 ALBEF 原论文，Hugging Face 对应模型文档。

今天先准备 建立 week08/day053 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。

阶段	今天具体做什么
跑固定样例	对同一批编号样例执行：在小数据集计算双向 R at 1、5、10，逐条检查 ground truth 映射并展示五个检索失败案例。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“双向检索指标表和失败页面”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“图搜文和文搜图方向不同，Recall at K 表示正确目标是否进入前 K。数据存在多对多标注时，评价代码必须接受多个正确答案。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 在小数据集计算双向 R at 1、5、10，逐条检查 ground truth 映射并展示五个检索失败案例。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：双向检索指标表和失败页面。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“图文检索与 Recall at K”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“双向检索指标表和失败页面”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“图文检索与 Recall at K”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“双向检索指标表和失败页面”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：图搜文和文搜图方向不同，Recall at K 表示正确目标是否进入前 K。数据存在多对多标注时，评价代码必须接受多个正确答案。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“双向检索指标表和失败页面”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：用 BLEU 一项指标判断生成质量。；检索数据中一张图多条描述却按单标签处理。；视觉问答只看正确率，不区分感知、OCR、推理和知识错误。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 54 视觉问答错误类型

今天要懂 VQA 可能依赖物体识别、计数、空间、OCR、知识或多步推理。总体分数无法告诉你瓶颈，错误类型比例才指导改进。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“视觉问答错误类型”，合格解释必须包含这一因果关系：VQA 可能依赖物体识别、计数、空间、OCR、知识或多步推理。总体分数无法告诉你瓶颈，错误类型比例才指导改进。。然后把它与本周目标连接起来：掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：准备至少四类各十题，固定提示运行模型，人工标注错误类型和是否真正看图。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“四十题 VQA 测试集与错误分布”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 BLIP、BLIP 2、VQA 和 ALBEF 原论文，Hugging Face 对应模型文档。

今天先准备 建立 week08/day054 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：准备至少四类各十题，固定提示运行模型，人工标注错误类型和是否真正看图。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“四十题 VQA 测试集与错误分布”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“VQA 可能依赖物体识别、计数、空间、OCR、知识或多步推理。总体分数无法告诉你瓶颈，错误类型比例才指导改进。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 准备至少四类各十题，固定提示运行模型，人工标注错误类型和是否真正看图。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：四十题 VQA 测试集与错误分布。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“视觉问答错误类型”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“四十题 VQA 测试集与错误分布”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“视觉问答错误类型”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“四十题 VQA 测试集与错误分布”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：VQA 可能依赖物体识别、计数、空间、OCR、知识或多步推理。总体分数无法告诉你瓶颈，错误类型比例才指导改进。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“四十题 VQA 测试集与错误分布”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：用 BLEU 一项指标判断生成质量。；检索数据中一张图多条描述却按单标签处理。；视觉问答只看正确率，不区分感知、OCR、推理和知识错误。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 55 图像描述评估

今天要懂 BLEU 重表面 n gram，CIDEr 强调与参考描述一致的信息，CLIPScore 衡量图文语义。它们都可能漏掉事实错误，因此需要人工事实核查。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“图像描述评估”，合格解释必须包含这一因果关系：BLEU 重表面 n gram，CIDEr 强调与参考描述一致的信息，CLIPScore 衡量图文语义。它们都可能漏掉事实错误，因此需要人工事实核查。。然后把它与本周目标连接起来：掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：生成五十条描述，计算可用自动指标，抽查对象、属性和关系幻觉，比较自动分数与人工判断。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“图像描述评测报告”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 BLIP、BLIP 2、VQA 和 ALBEF 原论文，Hugging Face 对应模型文档。

今天先准备 建立 week08/day055 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：生成五十条描述，计算可用自动指标，抽查对象、属性和关系幻觉，比较自动分数与人工判断。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“图像描述评测报告”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“BLEU 重表面 n gram，CIDEr 强调与参考描述一致的信息，CLIPScore 衡量图文语义。它们都可能漏掉事实错误，因此需要人工事实核查。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 生成五十条描述，计算可用自动指标，抽查对象、属性和关系幻觉，比较自动分数与人工判断。
- 4 一百三十至一百六十分** 整理并保存今日产出：图像描述评测报告。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“图像描述评估”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。

- 打开并检查“图像描述评测报告”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“图像描述评估”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“图像描述评测报告”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：BLEU 重表面 n gram，CIDEr 强调与参考描述一致的信息，CLIPScore 衡量图文语义。它们都可能漏掉事实错误，因此需要人工事实核查。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“图像描述评测报告”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：用 BLEU 一项指标判断生成质量。；检索数据中一张图多条描述却按单标签处理。；视觉问答只看正确率，不区分感知、OCR、推理和知识错误。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 56 第 8 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。

阶段	今天具体做什么
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：BLIP 的三个目标、CapFilt 与数据质量、Q Former、图文检索与 Recall at K、视觉问答错误类型、图像描述评估。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“图像描述评测报告”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：掌握图文理解与生成的主要目标、检索指标和错误分析方法，完成一个图文理解流水线。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 9 周 扩散模型与视觉语言模型

本周目标 理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。

首选材料 DDPM、Latent Diffusion、Classifier Free Guidance、LLaVA 和视觉指令微调原论文。

本周小白讲义

扩散模型训练时向数据逐步加噪，再学习预测噪声或等价目标；采样时从噪声开始逐步去噪。潜空间扩散先把图像压缩到更小的潜变量中，显著降低计算量。

文本条件通常通过交叉注意力进入去噪网络。classifier free guidance 调整条件与无条件预测的差异，较大引导可能增强提示一致性，也可能降低多样性或带来伪影。

视觉语言模型把视觉特征通过投影层或查询模块送入语言模型，再用图文指令数据训练回答。研究时要区分视觉编码器、连接模块、语言模型和训练数据分别贡献了什么。

本周知识地图

天	主题	核心知识	产出
Day 57	扩散模型直觉	前向过程逐步加入高斯噪声，反向模型学习从带噪样本预测噪声或相关目标。训练能随机抽时间步并行进行，采样却通常要从噪声多步迭代。	扩散时间轴图和一页解释
Day 58	前向公式与训练目标	任意时间步的带噪样本可由原图和噪声直接构造，避免逐步加噪。模型常以均方误差预测噪声，时间步和文本条件共同影响预测。	forward_diffusion.py 和单元测试
Day 59	UNet 去噪器	UNet 通过下采样获得全局信息，再上采样恢复空间细节，并用跳跃连接保留高分辨率特征。时间 embedding 告诉网络当前噪声级别。	UNet 形状图和采样记录
Day 60	潜空间扩散	像素空间成本高，潜空间扩散先用 VAE 把图像压缩，再在 latent 中去噪，最后解码。压缩提高效率，但小文字和细节可能损失。	VAE 重建对比图

天	主题	核心知识	产出
Day 61	CFG 采样器与可重复比较	classifier free guidance 组合有条件和无条件预测，增大引导通常更贴提示但可能过饱和。采样器、步数和种子都影响结果。	CFG 与步数控制变量实验
Day 62	视觉 Token 接入 LLM	视觉语言模型用视觉编码器产生 patch 特征，再经投影层或 Q Former 转成 LLM 可接收的 embedding。指令微调让模型把视觉信息用于对话。	视觉语言模型固定评测与错误表
Day 63	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
noise = torch.randn_like(x0)
xt = alpha_bar.sqrt() * x0 + (1 - alpha_bar).sqrt() * noise
noise_pred = model(xt, timestep, text_condition)
loss = ((noise_pred - noise) ** 2).mean()
```

本周常见错误

- 把训练时噪声预测与推理时逐步采样混为一谈。
- 比较生成结果时没有固定随机种子。
- 只测试漂亮自然图，不测试 OCR、图表和模糊输入。

Day 57 扩散模型直觉

今天要懂 前向过程逐步加入高斯噪声，反向模型学习从带噪样本预测噪声或相关目标。训练能随机抽时间步并行进行，采样却通常要从噪声多步迭代。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“扩散模型直觉”，合格解释必须包含这一因果关系：前向过程逐步加入高斯噪声，反向模型学习从带噪样本预测噪声或相关目标。训练能随机抽时间步并行进行，采样却通常要从噪声多步迭代。。然后把它与本周目标连接起来：理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：给一张图在多个噪声强度下加噪并排展示，解释信号何时开始难以辨认。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“扩散时间轴图和一页解释”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 DDPM、Latent Diffusion、Classifier Free Guidance、LLaVA 和视觉指令微调原论文。

今天先准备 建立 week09/day057 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：给一张图在多个噪声强度下加噪并排展示，解释信号何时开始难以辨认。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“扩散时间轴图和一页解释”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“前向过程逐步加入高斯噪声，反向模型学习从带噪样本预测噪声或相关目标。训练能随机抽时间步并行进行，采样却通常要从噪声多步迭代。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 给一张图在多个噪声强度下加噪并排展示，解释信号何时开始难以辨认。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：扩散时间轴图和一页解释。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“扩散模型直觉”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“扩散时间轴图和一页解释”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“扩散模型直觉”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“扩散时间轴图和一页解释”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：前向过程逐步加入高斯噪声，反向模型学习从带噪样本预测噪声或相关目标。训练能随机抽时间步并行进行，采样却通常要从噪声多步迭代。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“扩散时间轴图和一页解释”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把训练时噪声预测与推理时逐步采样混为一谈。；比较生成结果时没有固定随机种子。；只测试漂亮自然图，不测试 OCR、图表和模糊输入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 58 前向公式与训练目标

今天要懂 任意时间步的带噪样本可由原图和噪声直接构造，避免逐步加噪。模型常以均方误差预测噪声，时间步和文本条件共同影响预测。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“前向公式与训练目标”，合格解释必须包含这一因果关系：任意时间步的带噪样本可由原图和噪声直接构造，避免逐步加噪。模型常以均方误差预测噪声，时间步和文本条件共同影响预测。。然后把它与本周目标连接起来：理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：实现 q_sample，抽取不同 t 检查形状和数值范围，再用假模型验证损失计算。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“forward_diffusion.py 和单元测试”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 DDPM、Latent Diffusion、Classifier Free Guidance、LLaVA 和视觉指令微调原论文。

今天先准备 建立 week09/day058 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。

阶段	今天具体做什么
跑固定样例	对同一批编号样例执行：实现 <code>q_sample</code> ，抽取不同 <code>t</code> 检查形状和数值范围，再用假模型验证损失计算。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“ <code>forward_diffusion.py</code> 和单元测试”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“任意时间步的带噪样本可由原图和噪声直接构造，避免逐步加噪。模型常以均方误差预测噪声，时间步和文本条件共同影响预测。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 实现 `q_sample`，抽取不同 `t` 检查形状和数值范围，再用假模型验证损失计算。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：`forward_diffusion.py` 和单元测试。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“前向公式与训练目标”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“`forward_diffusion.py` 和单元测试”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“前向公式与训练目标”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“forward_diffusion.py 和单元测试”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：任意时间步的带噪样本可由原图和噪声直接构造，避免逐步加噪。模型常以均方误差预测噪声，时间步和文本条件共同影响预测。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“forward_diffusion.py 和单元测试”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把训练时噪声预测与推理时逐步采样混为一谈。；比较生成结果时没有固定随机种子。；只测试漂亮自然图，不测试 OCR、图表和模糊输入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 59 UNet 去噪器

今天要懂 UNet 通过下采样获得全局信息，再上采样恢复空间细节，并用跳跃连接保留高分辨率特征。时间 embedding 告诉网络当前噪声级别。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“UNet 去噪器”，合格解释必须包含这一因果关系：UNet 通过下采样获得全局信息，再上采样恢复空间细节，并用跳跃连接保留高分辨率特征。时间 embedding 告诉网络当前噪声级别。。然后把它与本周目标连接起来：理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：画出输入、下采样、瓶颈、上采样和跳连形状；运行一个小型预训练扩散示例并记录采样时间。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“UNet 形状图和采样记录”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 DDPM、Latent Diffusion、Classifier Free Guidance、LLaVA 和视觉指令微调原论文。

今天先准备 建立 week09/day059 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：画出输入、下采样、瓶颈、上采样和跳连形状；运行一个小型预训练扩散示例并记录采样时间。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“UNet 形状图和采样记录”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“UNet 通过下采样获得全局信息，再上采样恢复空间细节，并用跳跃连接保留高分辨率特征。时间 embedding 告诉网络当前噪声级别。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 画出输入、下采样、瓶颈、上采样和跳连形状；运行一个小型预训练扩散示例并记录采样时间。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：UNet 形状图和采样记录。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“UNet 去噪器”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“UNet 形状图和采样记录”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“UNet 去噪器”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“UNet 形状图和采样记录”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：UNet 通过下采样获得全局信息，再上采样恢复空间细节，并用跳跃连接保留高分辨率特征。时间 embedding 告诉网络当前噪声级别。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“UNet 形状图和采样记录”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把训练时噪声预测与推理时逐步采样混为一谈。；比较生成结果时没有固定随机种子。；只测试漂亮自然图，不测试 OCR、图表和模糊输入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 60 潜空间扩散

今天要懂 像素空间成本高，潜空间扩散先用 VAE 把图像压缩，再在 latent 中去噪，最后解码。压缩提高效率，但小文字和细节可能损失。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“潜空间扩散”，合格解释必须包含这一因果关系：像素空间成本高，潜空间扩散先用 VAE 把图像压缩，再在 latent 中去噪，最后解码。压缩提高效率，但小文字和细节可能损失。。然后把它与本周目标连接起来：理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：对同一图片执行 VAE 编码和解码，比较原图与重建，观察尺寸、范围和细节差异。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“VAE 重建对比图”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 DDPM、Latent Diffusion、Classifier Free Guidance、LLaVA 和视觉指令微调原论文。

今天先准备 建立 week09/day060 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：对同一图片执行 VAE 编码和解码，比较原图与重建，观察尺寸、范围和细节差异。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“VAE 重建对比图”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“像素空间成本高，潜空间扩散先用 VAE 把图像压缩，再在 latent 中去噪，最后解码。压缩提高效率，但小文字和细节可能损失。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 对同一图片执行 VAE 编码和解码，比较原图与重建，观察尺寸、范围和细节差异。
- 4 一百三十至一百六十分** 整理并保存今日产出：VAE 重建对比图。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“潜空间扩散”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。

- □ 打开并检查“VAE 重建对比图”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“潜空间扩散”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“VAE 重建对比图”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：像素空间成本高，潜空间扩散先用 VAE 把图像压缩，再在 latent 中去噪，最后解码。压缩提高效率，但小文字和细节可能损失。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“VAE 重建对比图”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把训练时噪声预测与推理时逐步采样混为一谈。；比较生成结果时没有固定随机种子。；只测试漂亮自然图，不测试 OCR、图表和模糊输入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 61 CFG 采样器与可重复比较

今天要懂 classifier free guidance 组合有条件和无条件预测，增大引导通常更贴提示但可能过饱和。采样器、步数和种子都影响结果。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“CFG 采样器与可重复比较”，合格解释必须包含这一因果关系：classifier free guidance 组合有条件和无条件预测，增大引导通常更贴提示但可能过饱和。采样器、步数和种子都影响结果。。然后把它与本周目标连接起来：理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：固定提示和种子，一次只改变 CFG 或步数，生成网格并记录耗时，不凭单张图下结论。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“CFG 与步数控制变量实验”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 DDPM、Latent Diffusion、Classifier Free Guidance、LLaVA 和视觉指令微调原论文。

今天先准备 建立 week09/day061 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：固定提示和种子，一次只改变 CFG 或步数，生成网格并记录耗时，不凭单张图下结论。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“CFG 与步数控制变量实验”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“classifier free guidance 组合有条件和无条件预测，增大引导通常更贴提示但可能过饱和。采样器、步数和种子都影响结果。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 固定提示和种子，一次只改变 CFG 或步数，生成网格并记录耗时，不凭单张图下结论。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：CFG 与步数控制变量实验。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“CFG 采样器与可重复比较”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“CFG 与步数控制变量实验”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“CFG 采样器与可重复比较”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“CFG 与步数控制变量实验”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：classifier free guidance 组合有条件和无条件预测，增大引导通常更贴提示但可能过饱和。采样器、步数和种子都影响结果。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“CFG 与步数控制变量实验”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把训练时噪声预测与推理时逐步采样混为一谈。；比较生成结果时没有固定随机种子。；只测试漂亮自然图，不测试 OCR、图表和模糊输入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 62 视觉 Token 接入 LLM

今天要懂 视觉语言模型用视觉编码器产生 patch 特征，再经投影层或 Q Former 转成 LLM 可接收的 embedding。指令微调让模型把视觉信息用于对话。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“视觉 Token 接入 LLM”，合格解释必须包含这一因果关系：视觉语言模型用视觉编码器产生 patch 特征，再经投影层或 Q Former 转成 LLM 可接收的 embedding。指令微调让模型把视觉信息用于对话。。然后把它与本周目标连接起来：理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：画出 LLaVA 类模型的数据流，选择自然图、截图、图表、模糊图各三张，测试描述、问答和推理。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“视觉语言模型固定评测与错误表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 DDPM、Latent Diffusion、Classifier Free Guidance、LLaVA 和视觉指令微调原论文。

今天先准备 建立 week09/day062 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：画出 LLaVA 类模型的数据流，选择自然图、截图、图表、模糊图各三张，测试描述、问答和推理。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“视觉语言模型固定评测与错误表”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“视觉语言模型用视觉编码器产生 patch 特征，再经投影层或 Q Former 转成 LLM 可接收的 embedding。指令微调让模型把视觉信息用于对话。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 画出 LLaVA 类模型的数据流，选择自然图、截图、图表、模糊图各三张，测试描述、问答和推理。
- 4 一百三十至一百六十分** 整理并保存今日产出：视觉语言模型固定评测与错误表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“视觉 Token 接入 LLM”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。

- □ 打开并检查“视觉语言模型固定评测与错误表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“视觉 Token 接入 LLM”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“视觉语言模型固定评测与错误表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：视觉语言模型用视觉编码器产生 patch 特征，再经投影层或 Q Former 转成 LLM 可接收的 embedding。指令微调让模型把视觉信息用于对话。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“视觉语言模型固定评测与错误表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：把训练时噪声预测与推理时逐步采样混为一谈。；比较生成结果时没有固定随机种子。；只测试漂亮自然图，不测试 OCR、图表和模糊输入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 63 第 9 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：扩散模型直觉、前向公式与训练目标、UNet 去噪器、潜空间扩散、CFG 采样器与可重复比较、视觉 Token 接入 LLM。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“视觉语言模型固定评测与错误表”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：理解文生图的加噪去噪过程和视觉语言模型的视觉 token 连接方式，能够设计固定比较实验。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 10 周 视频 音频 Agent 评估与 RAG

本周目标 从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。

首选材料 ViViT、VideoCLIP、CLAP、ReAct、POPE 和多模态 RAG 代表工作及官方仓库。

本周小白讲义

视频增加时间维，音频增加采样率、频谱和时序结构。简单方案可以先抽帧或分片编码，再做时序聚合；评价必须覆盖顺序、事件边界和跨片段依赖，而不只是静态物体识别。

多模态 RAG 需要同时考虑文本、图片及其元数据的切分、嵌入、索引、召回和证据展示。Agent 只是让模型在规则下选择工具，可靠性来自受控状态、参数校验、超时、重试和可追踪日志。

多模态评价要拆成感知、推理、知识、幻觉、鲁棒性和效率。单一总分会掩盖失败类型，固定测试集和错误标签比一次漂亮演示更接近科研证据。

本周知识地图

天	主题	核心知识	产出
Day 64	开源视觉语言模型比较	模型比较必须固定图片、提示、解码参数和评价规则。模型大小、视觉分辨率、语言能力和训练数据不同，不能用几个演示样例宣称全面优劣。	公平模型比较表
Day 65	视频采样与时序	视频比图像多出时间维。均匀采帧能覆盖全片但可能错过短事件；特征平均会丢失顺序，时序模型保留帧间关系但成本更高。	视频采帧代码和对比结果
Day 66	音频频谱与 CLAP	声音波形随时间变化，梅尔频谱把频率能量变成二维表示。CLAP 用类似 CLIP 的对比学习对齐音频与文字。	频谱图和音频分类表
Day 67	多模态 Agent	Agent 把感知结果用于计划和行动，并根据环境反馈继续。评价时要记录观察、动作、结果和失败恢复，而不是只看最终是否完成。	可回放的 Agent 任务轨迹

天	主题	核心知识	产出
Day 68	多模态评估与幻觉	视觉幻觉包括不存在的物体、错误属性、错误关系和虚构文字。评价集要覆盖能力与风险，并防止提示泄漏和训练污染。	幻觉评测集与错误分析
Day 69	多模态 RAG 综合应用	多模态 RAG 可检索文字、图片或图文对，再让生成模型基于证据回答。有效系统必须显示来源、验证支持关系并在无证据时拒答。	可运行的多模态检索问答应用
Day 70	复盘验收	闭卷解释 独立重跑 整理失败补齐记录	周总结和通过记录

本周代码或记录模板

```
record = {
    'question': question,
    'retrieved_ids': retrieved_ids,
    'answer': answer,
    'supported': supported,
    'latency_ms': latency_ms,
}
```

本周常见错误

- 随机抽帧导致不同运行不可比。
- 把模型自述的理由当成真实内部推理。
- RAG 展示了来源，却没有检查来源是否真的支持答案。

Day 64 开源视觉语言模型比较

今天要懂 模型比较必须固定图片、提示、解码参数和评价规则。模型大小、视觉分辨率、语言能力和训练数据不同，不能用几个演示样例宣称全面优劣。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“开源视觉语言模型比较”，合格解释必须包含这一因果关系：模型比较必须固定图片、提示、解码参数和评价规则。模型大小、视觉分辨率、语言能力和训练数据不同，不能用几个演示样例宣称全面优劣。。然后把它与本周目标连接起来：从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：选择两个当前可用模型，在同一三十题集上测中文、OCR、图表和空间关系，记录版本、显存和延迟。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“公平模型比较表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 ViViT、VideoCLIP、CLAP、ReAct、POPE 和多模态 RAG 代表工作及官方仓库。

今天先准备 建立 week10/day064 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：选择两个当前可用模型，在同一三十题集上测中文、OCR、图表和空间关系，记录版本、显存和延迟。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“公平模型比较表”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“模型比较必须固定图片、提示、解码参数和评价规则。模型大小、视觉分辨率、语言能力和训练数据不同，不能用几个演示样例宣称全面优劣。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 选择两个当前可用模型，在同一三十题集上测中文、OCR、图表和空间关系，记录版本、显存和延迟。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：公平模型比较表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“开源视觉语言模型比较”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“公平模型比较表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“开源视觉语言模型比较”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“公平模型比较表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：模型比较必须固定图片、提示、解码参数和评价规则。模型大小、视觉分辨率、语言能力和训练数据不同，不能用几个演示样例宣称全面优劣。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“公平模型比较表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：随机抽帧导致不同运行不可比。；把模型自述的理由当成真实内部推理。；RAG 展示了来源，却没有检查来源是否真的支持答案。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 65 视频采样与时序

今天要懂 视频比图像多出时间维。均匀采帧能覆盖全片但可能错过短事件；特征平均会丢失顺序，时序模型保留帧间关系但成本更高。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“视频采样与时序”，合格解释必须包含这一因果关系：视频比图像多出时间维。均匀采帧能覆盖全片但可能错过短事件；特征平均会丢失顺序，时序模型保留帧间关系但成本更高。。然后把它与本周目标连接起来：从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：从短视频均匀取 8 帧，比较单帧、帧特征平均和按顺序输入模型的回答。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“视频采帧代码和对比结果”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 ViViT、VideoCLIP、CLAP、ReAct、POPE 和多模态 RAG 代表工作及官方仓库。

今天先准备 建立 week10/day065 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：从短视频均匀取 8 帧，比较单帧、帧特征平均和按顺序输入模型的回答。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“视频采帧代码和对比结果”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“视频比图像多出时间维。均匀采帧能覆盖全片但可能错过短事件；特征平均会丢失顺序，时序模型保留帧间关系但成本更高。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 从短视频均匀取 8 帧，比较单帧、帧特征平均和按顺序输入模型的回答。
- 4 一百三十至一百六十分** 整理并保存今日产出：视频采帧代码和对比结果。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“视频采样与时序”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“视频采帧代码和对比结果”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“视频采样与时序”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“视频采帧代码和对比结果”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：视频比图像多出时间维。均匀采帧能覆盖全片但可能错过短事件；特征平均会丢失顺序，时序模型保留帧间关系但成本更高。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“视频采帧代码和对比结果”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：随机抽帧导致不同运行不可比。；把模型自述的理由当成真实内部推理。；RAG 展示了来源，却没有检查来源是否真的支持答案。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 66 音频频谱与 CLAP

今天要懂 声音波形随时间变化，梅尔频谱把频率能量变成二维表示。CLAP 用类似 CLIP 的对比学习对齐音频与文字。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“音频频谱与 CLAP”，合格解释必须包含这一因果关系：声音波形随时间变化，梅尔频谱把频率能量变成二维表示。CLAP 用类似 CLIP 的对比学习对齐音频与文字。。然后把它与本周目标连接起来：从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：加载三类短音频，绘制梅尔频谱，使用可用模型完成零样本分类并分析背景噪声错误。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“频谱图和音频分类表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 ViViT、VideoCLIP、CLAP、ReAct、POPE 和多模态 RAG 代表工作及官方仓库。

今天先准备 建立 week10/day066 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：加载三类短音频，绘制梅尔频谱，使用可用模型完成零样本分类并分析背景噪声错误。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“频谱图和音频分类表”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“声音波形随时间变化，梅尔频谱把频率能量变成二维表示。CLAP 用类似 CLIP 的对比学习对齐音频与文字。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 加载三类短音频，绘制梅尔频谱，使用可用模型完成零样本分类并分析背景噪声错误。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：频谱图和音频分类表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“音频频谱与 CLAP”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“频谱图和音频分类表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“音频频谱与 CLAP”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“频谱图和音频分类表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：声音波形随时间变化，梅尔频谱把频率能量变成二维表示。CLAP 用类似 CLIP 的对比学习对齐音频与文字。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“频谱图和音频分类表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：随机抽帧导致不同运行不可比。；把模型自述的理由当成真实内部推理。；RAG 展示了来源，却没有检查来源是否真的支持答案。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 67 多模态 Agent

今天要懂 Agent 把感知结果用于计划和行动，并根据环境反馈继续。评价时要记录观察、动作、结果和失败恢复，而不是只看最终是否完成。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“多模态 Agent”，合格解释必须包含这一因果关系：Agent 把感知结果用于计划和行动，并根据环境反馈继续。评价时要记录观察、动作、结果和失败恢复，而不是只看最终是否完成。。然后把它与本周目标连接起来：从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：设计一个网页截图任务，人工写出每步观察、可选动作、选择理由和成功条件，不执行危险或不可逆操作。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“可回放的 Agent 任务轨迹”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 ViViT、VideoCLIP、CLAP、ReAct、POPE 和多模态 RAG 代表工作及官方仓库。

今天先准备 建立 week10/day067 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。

阶段	今天具体做什么
跑固定样例	对同一批编号样例执行：设计一个网页截图任务，人工写出每步观察、可选动作、选择理由和成功条件，不执行危险或不可逆操作。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“可回放的 Agent 任务轨迹”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“Agent 把感知结果用于计划和行动，并根据环境反馈继续。评价时要记录观察、动作、结果和失败恢复，而不是只看最终是否完成。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 设计一个网页截图任务，人工写出每步观察、可选动作、选择理由和成功条件，不执行危险或不可逆操作。
- 4 一百三十至一百六十分** 整理并保存今日产出：可回放的 Agent 任务轨迹。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“多模态 Agent”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“可回放的 Agent 任务轨迹”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“多模态 Agent”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“可回放的 Agent 任务轨迹”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：Agent 把感知结果用于计划和行动，并根据环境反馈继续。评价时要记录观察、动作、结果和失败恢复，而不是只看最终是否完成。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“可回放的 Agent 任务轨迹”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：随机抽帧导致不同运行不可比。；把模型自述的理由当成真实内部推理。；RAG 展示了来源，却没有检查来源是否真的支持答案。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 68 多模态评估与幻觉

今天要懂 视觉幻觉包括不存在的物体、错误属性、错误关系和虚构文字。评价集要覆盖能力与风险，并防止提示泄漏和训练污染。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“多模态评估与幻觉”，合格解释必须包含这一因果关系：视觉幻觉包括不存在的物体、错误属性、错误关系和虚构文字。评价集要覆盖能力与风险，并防止提示泄漏和训练污染。。然后把它与本周目标连接起来：从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：用二十张图建立 yes no 对象测试和开放描述测试，统计准确率、幻觉率并保存证据图片。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“幻觉评测集与错误分析”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 ViViT、VideoCLIP、CLAP、ReAct、POPE 和多模态 RAG 代表工作及官方仓库。

今天先准备 建立 week10/day068 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。
跑固定样例	对同一批编号样例执行：用二十张图建立 yes no 对象测试和开放描述测试，统计准确率、幻觉率并保存证据图片。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“幻觉评测集与错误分析”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“视觉幻觉包括不存在的物体、错误属性、错误关系和虚构文字。评价集要覆盖能力与风险，并防止提示泄漏和训练污染。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用二十张图建立 yes no 对象测试和开放描述测试，统计准确率、幻觉率并保存证据图片。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：幻觉评测集与错误分析。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“多模态评估与幻觉”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“幻觉评测集与错误分析”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“多模态评估与幻觉”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“幻觉评测集与错误分析”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：视觉幻觉包括不存在的物体、错误属性、错误关系和虚构文字。评价集要覆盖能力与风险，并防止提示泄漏和训练污染。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“幻觉评测集与错误分析”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：随机抽帧导致不同运行不可比。；把模型自述的理由当成真实内部推理。；RAG 展示了来源，却没有检查来源是否真的支持答案。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 69 多模态 RAG 综合应用

今天要懂 多模态 RAG 可检索文字、图片或图文对，再让生成模型基于证据回答。有效系统必须显示来源、验证支持关系并在无证据时拒答。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“多模态 RAG 综合应用”，合格解释必须包含这一因果关系：多模态 RAG 可检索文字、图片或图文对，再让生成模型基于证据回答。有效系统必须显示来源、验证支持关系并在无证据时拒答。。然后把它与本周目标连接起来：从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。

科研连接 把今天的概念放回统一框架：每个模态怎样编码，在哪里对齐或融合，训练信号是什么，评价指标能证明什么。回答不出其中一项时，不继续堆模型。

跟着做一个例子

先画出今天系统的模态输入、编码器、交互位置、输出与指标，再完成实践：建立至少一百条图文知识库，实现查询、top k 检索、回答与来源展示，加入二十至五十条固定测试。。对至少三个正常样例和两个失败样例逐项保存输入、输出和原因，不用单个成功演示代表整体能力。

完成标准 今天结束前必须得到“可运行的多模态检索问答应用”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 ViViT、VideoCLIP、CLAP、ReAct、POPE 和多模态 RAG 代表工作及官方仓库。

今天先准备 建立 week10/day069 文件夹，保存输入样例、模型名称、权重版本、提示词和随机种子。先选 5 个正常样例和 2 个容易失败的样例，给每个样例编号；没有失败样例就无法判断系统边界。先跑通冻结模型或预计算特征的最小版本，再考虑训练、提高分辨率或增加数据。

阶段	今天具体做什么
画数据流	先画出模态输入、编码器、投影或交互位置、输出和指标，给每个箭头标注张量形状或文本格式。

阶段	今天具体做什么
跑固定样例	对同一批编号样例执行：建立至少一百条图文知识库，实现查询、top k 检索、回答与来源展示，加入二十至五十条固定测试。保存原始输入、输出、耗时和模型配置。
做一个对照	只改提示模板、采样帧数、温度、检索数量或一个训练因素，其他设置不变，并保留至少两个失败样例。
形成判断	将成功与失败按错误类型归类，写入“可运行的多模态检索问答应用”，结论只覆盖你真正测试过的样例和指标。

通过标准

- 至少 5 个样例有完整输入、输出和配置，另有 2 个失败样例并标注失败类型。
- 至少有一个固定对照或消融，指标和定性分析结论方向一致或解释不一致的原因。
- 能说清模型究竟使用了哪种模态证据，以及无证据时应如何拒答或标记不确定。

低算力和时间不足 显存不足时冻结编码器、使用预计算特征、降低分辨率或只取 32 至 256 个样本；目标是验证数据流和评价，不是追求大模型规模。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“多模态 RAG 可检索文字、图片或图文对，再让生成模型基于证据回答。有效系统必须显示来源、验证支持关系并在无证据时拒答。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 建立至少一百条图文知识库，实现查询、top k 检索、回答与来源展示，加入二十至五十条固定测试。
- 4 一百三十至一百六十分** 整理并保存今日产出：可运行的多模态检索问答应用。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“多模态 RAG 综合应用”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“可运行的多模态检索问答应用”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“多模态 RAG 综合应用”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“可运行的多模态检索问答应用”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：多模态 RAG 可检索文字、图片或图文对，再让生成模型基于证据回答。有效系统必须显示来源、验证支持关系并在无证据时拒答。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“可运行的多模态检索问答应用”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：随机抽帧导致不同运行不可比。；把模型自述的理由当成真实内部推理。；RAG 展示了来源，却没有检查来源是否真的支持答案。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 70 第 10 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。

阶段	今天具体做什么
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：开源视觉语言模型比较、视频采样与时序、音频频谱与 CLAP、多模态 Agent、多模态评估与幻觉、多模态 RAG 综合应用。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“可运行的多模态检索问答应用”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：从静态图文扩展到视频、音频和工具使用，并完成第一个带固定评测的多模态应用。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 11 周 选择可复现论文

本周目标 把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。

首选材料 Papers with Code，目标论文与补充材料，官方仓库 README、issues、license 和 model card。

本周小白讲义

首篇复现论文的目标是训练实验纪律，不是挑战最大模型。优先选择代码公开、数据可得、评价明确、单卡或子集可运行的工作，并在开始前写下成功标准和退出条件。

读论文时把文字主张映射到表格、图和代码入口。无法找到证据的主张先标问号，不凭作者语气判断可信度。

资源估算至少包含数据体积、权重体积、训练显存、GPU 小时和人工标注时间。预算不够时缩小问题或冻结模块，而不是删掉必要基线。

本周知识地图

天	主题	核心知识	产出
Day 71	确定方向与约束	首篇复现的目标是掌握科研流程，不是训练最大模型。先写可用显存、预算、期限、数据权限和你能接受的失败范围。	个人复现约束表
Day 72	搜索候选论文	检索式应包含任务、模态、方法和年份。先看标题摘要和图一，再确认论文是否报告了公开数据、评价协议和主要基线。	十篇候选论文表
Day 73	审查代码仓库	可复现仓库应有明确依赖、配置、训练和评估入口。README 成功案例不等于当前仍可运行，issues 和 release 能暴露依赖与数据问题。	仓库可运行性评分
Day 74	审查数据与许可证	数据可下载不代表可自由研究或再分发。要确认用途限制、标注协议、划分、下载体积和预处理时间。	数据合规与可得性清单

天	主题	核心知识	产出
Day 75	估算算力与时间	训练成本取决于参数、序列长度、分辨率、批次和步数。先跑短基准测显存与每步时间，再外推总时长并留出失败余量。	算力预算与降级方案
Day 76	完成复现卡	复现卡把论文主张、环境、数据、命令、目标数字和风险固定下来。没有复现卡就开跑，后续很难区分计划变化与实验问题。	一页复现卡和风险清单
Day 77	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
paper_card = {
  'claim': '', 'dataset': '', 'metric': '',
  'official_repo': '', 'commit': '', 'gpu_hours': 0,
  'risks': []
}
```

本周常见错误

- 因为论文热门就选择，不估算算力。
- 只看仓库星数，不验证关键训练代码。
- 忽略数据和权重许可证。

Day 71 确定方向与约束

今天要懂 首篇复现的目标是掌握科研流程，不是训练最大模型。先写可用显存、预算、期限、数据权限和你能接受的失败范围。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“确定方向与约束”，合格解释必须包含这一因果关系：首篇复现的目标是掌握科研流程，不是训练最大模型。先写可用显存、预算、期限、数据权限和你能接受的失败范围。。然后把它与本周目标连接起来：把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：列出一个主方向和两个备选，填写硬件、每周时间、最高费用、数据限制和预期交付。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“个人复现约束表”，失败运行同样保留。

完成标准 今天结束前必须得到“个人复现约束表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Papers with Code，目标论文与补充材料，官方仓库 README、issues、license 和 model card。

今天先准备 建立 week11/day071 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：列出一个主方向和两个备选，填写硬件、每周时间、最高费用、数据限制和预期交付。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“个人复现约束表”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“首篇复现的目标是掌握科研流程，不是训练最大模型。先写可用显存、预算、期限、数据权限和你能接受的失败范围。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 列出一个主方向和两个备选，填写硬件、每周时间、最高费用、数据限制和预期交付。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：个人复现约束表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“确定方向与约束”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“个人复现约束表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“确定方向与约束”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“个人复现约束表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：首篇复现的目标是掌握科研流程，不是训练最大模型。先写可用显存、预算、期限、数据权限和你能接受的失败范围。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 第四题应让别人找到输入、环境、运行命令、输出目录和“个人复现约束表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：因为论文热门就选择，不估算算力。；只看仓库星数，不验证关键训练代码。；忽略数据和权重许可证。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 72 搜索候选论文

今天要懂 检索式应包含任务、模态、方法和年份。先看标题摘要和图一，再确认论文是否报告了公开数据、评价协议和主要基线。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“搜索候选论文”，合格解释必须包含这一因果关系：检索式应包含任务、模态、方法和年份。先看标题摘要和图一，再确认论文是否报告了公开数据、评价协议和主要基线。。然后把它与本周目标连接起来：把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：在目标会议或 arXiv 找十篇候选，记录问题、年份、代码、数据、主指标和模型规模。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“十篇候选论文表”，失败运行同样保留。

完成标准 今天结束前必须得到“十篇候选论文表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Papers with Code，目标论文与补充材料，官方仓库 README、issues、license 和 model card。

今天先准备 建立 week11/day072 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。

阶段	今天具体做什么
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：在目标会议或 arXiv 找十篇候选，记录问题、年份、代码、数据、主指标和模型规模。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“十篇候选论文表”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“检索式应包含任务、模态、方法和年份。先看标题摘要和图一，再确认论文是否报告了公开数据、评价协议和主要基线。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 在目标会议或 arXiv 找十篇候选，记录问题、年份、代码、数据、主指标和模型规模。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：十篇候选论文表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“搜索候选论文”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“十篇候选论文表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“搜索候选论文”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“十篇候选论文表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：检索式应包含任务、模态、方法和年份。先看标题摘要和图一，再确认论文是否报告了公开数据、评价协议和主要基线。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“十篇候选论文表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：因为论文热门就选择，不估算算力。；只看仓库星数，不验证关键训练代码。；忽略数据和权重许可证。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 73 审查代码仓库

今天要懂 可复现仓库应有明确依赖、配置、训练和评估入口。README 成功案例不等于当前仍可运行，issues 和 release 能暴露依赖与数据问题。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“审查代码仓库”，合格解释必须包含这一因果关系：可复现仓库应有明确依赖、配置、训练和评估入口。README 成功案例不等于当前仍可运行，issues 和 release 能暴露依赖与数据问题。。然后把它与本周目标连接起来：把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：对五个仓库检查安装命令、预训练权重、配置、评估脚本、最近维护和已知问题。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“仓库可运行性评分”，失败运行同样保留。

完成标准 今天结束前必须得到“仓库可运行性评分”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Papers with Code，目标论文与补充材料，官方仓库 README、issues、license 和 model card。

今天先准备 建立 week11/day073 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：对五个仓库检查安装命令、预训练权重、配置、评估脚本、最近维护和已知问题。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“仓库可运行性评分”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“可复现仓库应有明确依赖、配置、训练和评估入口。README 成功案例不等于当前仍可运行，issues 和 release 能暴露依赖与数据问题。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 对五个仓库检查安装命令、预训练权重、配置、评估脚本、最近维护和已知问题。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：仓库可运行性评分。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“审查代码仓库”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“仓库可运行性评分”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“审查代码仓库”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“仓库可运行性评分”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：可复现仓库应有明确依赖、配置、训练和评估入口。README 成功案例不等于当前仍可运行，issues 和 release 能暴露依赖与数据问题。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“仓库可运行性评分”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：因为论文热门就选择，不估算算力。；只看仓库星数，不验证关键训练代码。；忽略数据和权重许可证。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 74 审查数据与许可证

今天要懂 数据可下载不代表可自由研究或再分发。要确认用途限制、标注协议、划分、下载体积和预处理时间。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“审查数据与许可证”，合格解释必须包含这一因果关系：数据可下载不代表可自由研究或再分发。要确认用途限制、标注协议、划分、下载体积和预处理时间。。然后把它与本周目标连接起来：把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：为前三篇候选核对数据许可证、样本规模、划分文件、敏感信息和获取步骤。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“数据合规与可得性清单”，失败运行同样保留。

完成标准 今天结束前必须得到“数据合规与可得性清单”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Papers with Code，目标论文与补充材料，官方仓库 README、issues、license 和 model card。

今天先准备 建立 week11/day074 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。

阶段	今天具体做什么
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：为前三篇候选核对数据许可证、样本规模、划分文件、敏感信息和获取步骤。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“数据合规与可得性清单”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“数据可下载不代表可自由研究或再分发。要确认用途限制、标注协议、划分、下载体积和预处理时间。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 为前三篇候选核对数据许可证、样本规模、划分文件、敏感信息和获取步骤。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：数据合规与可得性清单。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“审查数据与许可证”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“数据合规与可得性清单”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“审查数据与许可证”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“数据合规与可得性清单”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：数据可下载不代表可自由研究或再分发。要确认用途限制、标注协议、划分、下载体积和预处理时间。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“数据合规与可得性清单”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：因为论文热门就选择，不估算算力。；只看仓库星数，不验证关键训练代码。；忽略数据和权重许可证。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 75 估算算力与时间

今天要懂 训练成本取决于参数、序列长度、分辨率、批次和步数。先跑短基准测显存与每步时间，再外推总时长并留出失败余量。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“估算算力与时间”，合格解释必须包含这一因果关系：训练成本取决于参数、序列长度、分辨率、批次和步数。先跑短基准测显存与每步时间，再外推总时长并留出失败余量。。然后把它与本周目标连接起来：把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：根据论文配置估算 GPU 小时，设计子集或冻结主干的降级路径，给每篇论文打可行性分。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“算力预算与降级方案”，失败运行同样保留。

完成标准 今天结束前必须得到“算力预算与降级方案”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Papers with Code，目标论文与补充材料，官方仓库 README、issues、license 和 model card。

今天先准备 建立 week11/day075 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：根据论文配置估算 GPU 小时，设计子集或冻结主干的降级路径，给每篇论文打可行性分。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“算力预算与降级方案”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“训练成本取决于参数、序列长度、分辨率、批次和步数。先跑短基准测显存与每步时间，再外推总时长并留出失败余量。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 根据论文配置估算 GPU 小时，设计子集或冻结主干的降级路径，给每篇论文打可行性分。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：算力预算与降级方案。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“估算算力与时间”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“算力预算与降级方案”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“估算算力与时间”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“算力预算与降级方案”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：训练成本取决于参数、序列长度、分辨率、批次和步数。先跑短基准测显存与每步时间，再外推总时长并留出失败余量。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 第四题应让别人找到输入、环境、运行命令、输出目录和“算力预算与降级方案”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：因为论文热门就选择，不估算算力。；只看仓库星数，不验证关键训练代码。；忽略数据和权重许可证。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 76 完成复现卡

今天要懂 复现卡把论文主张、环境、数据、命令、目标数字和风险固定下来。没有复现卡就开跑，后续很难区分计划变化与实验问题。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“完成复现卡”，合格解释必须包含这一因果关系：复现卡把论文主张、环境、数据、命令、目标数字和风险固定下来。没有复现卡就开跑，后续很难区分计划变化与实验问题。。然后把它与本周目标连接起来：把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：选择最终论文，锁定论文版本、仓库 commit、数据版本、目标表格行和两周里程碑。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“一页复现卡和风险清单”，失败运行同样保留。

完成标准 今天结束前必须得到“一页复现卡和风险清单”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 Papers with Code，目标论文与补充材料，官方仓库 README、issues、license 和 model card。

今天先准备 建立 week11/day076 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变变量；终端命令、commit、数据路径和输出目录写入 run.md。

阶段	今天具体做什么
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：选择最终论文，锁定论文版本、仓库 commit、数据版本、目标表格行和两周里程碑。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“一页复现卡和风险清单”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“复现卡把论文主张、环境、数据、命令、目标数字和风险固定下来。没有复现卡就开跑，后续很难区分计划变化与实验问题。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 选择最终论文，锁定论文版本、仓库 commit、数据版本、目标表格行和两周里程碑。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：一页复现卡和风险清单。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“完成复现卡”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“一页复现卡和风险清单”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“完成复现卡”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“一页复现卡和风险清单”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：复现卡把论文主张、环境、数据、命令、目标数字和风险固定下来。没有复现卡就开跑，后续很难区分计划变化与实验问题。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“一页复现卡和风险清单”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：因为论文热门就选择，不估算算力。；只看仓库星数，不验证关键训练代码。；忽略数据和权重许可证。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 77 第 11 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。

阶段	今天具体做什么
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：确定方向与约束、搜索候选论文、审查代码仓库、审查数据与许可证、估算算力与时间、完成复现卡。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“一页复现卡和风险清单”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：把兴趣收敛成一篇在数据、代码、算力和时间上可完成的首篇复现论文。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再执行本页验收。

第 12 周 跑通官方基线

本周目标 从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。

首选材料 所选论文官方 README、environment 文件、训练配置、评估脚本和补充材料。

本周小白讲义

跑通官方基线时不要一开始改代码。先锁定仓库版本，按官方最小命令验证数据加载、前向和评价，再逐步替换为本地数据与配置。

先让一个 batch 过拟合，再运行小子集，最后扩展完整设置。这三层检查分别验证模型可训练、管线完整和规模可承受。

每次运行保存配置、命令、日志、指标、预测和环境。只有最终分数而没有过程文件，不能支持差异分析。

本周知识地图

天	主题	核心知识	产出
Day 78	冻结环境与版本	复现首先要固定 Python、CUDA、框架和关键依赖。能安装不代表版本正确，环境文件与实际导出清单都要保留。	environment.txt 和 system_info.txt
Day 79	跑通官方推理	推理路径比训练短，先验证权重、预处理、模型和输出接口能连接。应使用官方样例与一个自选样例，并保存原始输出。	inference_log.md 和预测文件
Day 80	核对数据管道	数据加载错误会让训练看似运行却学不到东西。必须抽查原始样本、预处理后张量、标签映射和数据划分。	data_sanity_report.md
Day 81	过拟合单个 Batch	模型若不能在一个很小批次上把损失降得很低，通常存在标签、损失、梯度或模型连接错误。它是训练前最有效的体检。	single_batch_curve.png 和诊断记录

天	主题	核心知识	产出
Day 82	运行子集训练	子集训练用于验证完整训练链和趋势，不用于宣称最终性能。配置必须与正式实验分开，避免误把调试参数带入完整训练。	debug 配置和可恢复检查点
Day 83	运行官方评估	评价代码是复现的一部分。要确认输入预测格式、多个正确答案、无效样本处理和单位，不能只相信脚本最后打印的一个数字。	第一版基线结果与一键运行命令
Day 84	复盘验收	闭卷解释 独立重跑 整理失败补齐记录	周总结和通过记录

本周代码或记录模板

```
python train.py --config configs/baseline.yaml --seed 42
python evaluate.py --checkpoint outputs/baseline/best.pt
# 把完整命令、commit 和环境一并写入 run.md
```

本周常见错误

- 一开始直接跑完整训练，错误成本过高。
- 修改官方代码却没有保存 diff。
- 只保存终端截图，没有配置、日志和命令。

Day 78 冻结环境与版本

今天要懂 复现首先要固定 Python、CUDA、框架和关键依赖。能安装不代表版本正确，环境文件与实际导出清单都要保留。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“冻结环境与版本”，合格解释必须包含这一因果关系：复现首先要固定 Python、CUDA、框架和关键依赖。能安装不代表版本正确，环境文件与实际导出清单都要保留。。然后把它与本周目标连接起来：从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：建立新环境，按官方说明安装，运行最小 import，导出依赖和 GPU 信息，记录仓库 commit。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到 “environment.txt 和 system_info.txt”，失败运行同样保留。

完成标准 今天结束前必须得到 “environment.txt 和 system_info.txt”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 所选论文官方 README、environment 文件、训练配置、评估脚本和补充材料。

今天先准备 建立 week12/day078 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：建立新环境，按官方说明安装，运行最小 import，导出依赖和 GPU 信息，记录仓库 commit。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到 “environment.txt 和 system_info.txt”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“复现首先要固定 Python、CUDA、框架和关键依赖。能安装不代表版本正确，环境文件与实际导出清单都要保留。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 建立新环境，按官方说明安装，运行最小 import，导出依赖和 GPU 信息，记录仓库 commit。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：environment.txt 和 system_info.txt。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“冻结环境与版本”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“environment.txt 和 system_info.txt”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“冻结环境与版本”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“environment.txt 和 system_info.txt”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：复现首先要固定 Python、CUDA、框架和关键依赖。能安装不代表版本正确，环境文件与实际导出清单都要保留。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和 “environment.txt 和 system_info.txt”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：一开始直接跑完整训练，错误成本过高。；修改官方代码却没有保存 diff。；只保存终端截图，没有配置、日志和命令。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 79 跑通官方推理

今天要懂 推理路径比训练短，先验证权重、预处理、模型和输出接口能连接。应使用官方样例与一个自选样例，并保存原始输出。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“跑通官方推理”，合格解释必须包含这一因果关系：推理路径比训练短，先验证权重、预处理、模型和输出接口能连接。应使用官方样例与一个自选样例，并保存原始输出。。然后把它与本周目标连接起来：从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：下载或定位官方权重，运行推理，核对输出格式和示例趋势，不修改模型结构。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到 “inference_log.md 和预测文件”，失败运行同样保留。

完成标准 今天结束前必须得到 “inference_log.md 和预测文件”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 所选论文官方 README、environment 文件、训练配置、评估脚本和补充材料。

今天先准备 建立 week12/day079 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。

阶段	今天具体做什么
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：下载或定位官方权重，运行推理，核对输出格式和示例趋势，不修改模型结构。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到 “inference_log.md 和预测文件”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“推理路径比训练短，先验证权重、预处理、模型和输出接口能连接。应使用官方样例与一个自选样例，并保存原始输出。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 下载或定位官方权重，运行推理，核对输出格式和示例趋势，不修改模型结构。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：inference_log.md 和预测文件。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“跑通官方推理”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查 “inference_log.md 和预测文件”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“跑通官方推理”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“inference_log.md 和预测文件”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：推理路径比训练短，先验证权重、预处理、模型和输出接口能连接。应使用官方样例与一个自选样例，并保存原始输出。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“inference_log.md 和预测文件”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：一开始直接跑完整训练，错误成本过高。；修改官方代码却没有保存 diff。；只保存终端截图，没有配置、日志和命令。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 80 核对数据管道

今天要懂 数据加载错误会让训练看似运行却学不到东西。必须抽查原始样本、预处理后张量、标签映射和数据划分。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“核对数据管道”，合格解释必须包含这一因果关系：数据加载错误会让训练看似运行却学不到东西。必须抽查原始样本、预处理后张量、标签映射和数据划分。。然后把它与本周目标连接起来：从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：可视化至少二十个样本，打印变换前后尺寸、范围和 token 长度，检查训练验证是否重叠。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到 “data_sanity_report.md”，失败运行同样保留。

完成标准 今天结束前必须得到 “data_sanity_report.md”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 所选论文官方 README、environment 文件、训练配置、评估脚本和补充材料。

今天先准备 建立 week12/day080 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：可视化至少二十个样本，打印变换前后尺寸、范围和 token 长度，检查训练验证是否重叠。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到 “data_sanity_report.md”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“数据加载错误会让训练看似运行却学不到东西。必须抽查原始样本、预处理后张量、标签映射和数据划分。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 可视化至少二十个样本，打印变换前后尺寸、范围和 token 长度，检查训练验证是否重叠。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：data_sanity_report.md。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“核对数据管道”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“data_sanity_report.md”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“核对数据管道”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“data_sanity_report.md”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：数据加载错误会让训练看似运行却学不到东西。必须抽查原始样本、预处理后张量、标签映射和数据划分。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 第四题应让别人找到输入、环境、运行命令、输出目录和“data_sanity_report.md”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：一开始直接跑完整训练，错误成本过高。；修改官方代码却没有保存 diff。；只保存终端截图，没有配置、日志和命令。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 81 过拟合单个 Batch

今天要懂 模型若不能在一个很小批次上把损失降得很低，通常存在标签、损失、梯度或模型连接错误。它是训练前最有效的体检。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“过拟合单个 Batch”，合格解释必须包含这一因果关系：模型若不能在一个很小批次上把损失降得很低，通常存在标签、损失、梯度或模型连接错误。它是训练前最有效的体检。。然后把它与本周目标连接起来：从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：关闭强数据增强，用一个固定 batch 训练数百步，观察损失和预测；逐项排查无法过拟合的原因。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“single_batch_curve.png 和诊断记录”，失败运行同样保留。

完成标准 今天结束前必须得到“single_batch_curve.png 和诊断记录”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 所选论文官方 README、environment 文件、训练配置、评估脚本和补充材料。

今天先准备 建立 week12/day081 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。

阶段	今天具体做什么
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：关闭强数据增强，用一个固定 batch 训练数百步，观察损失和预测；逐项排查无法过拟合的原因。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到 “single_batch_curve.png 和诊断记录”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“模型若不能在一个很小批次上把损失降得很低，通常存在标签、损失、梯度或模型连接错误。它是训练前最有效的体检。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 关闭强数据增强，用一个固定 batch 训练数百步，观察损失和预测；逐项排查无法过拟合的原因。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：single_batch_curve.png 和诊断记录。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“过拟合单个 Batch”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。

- □ 打开并检查 “single_batch_curve.png 和诊断记录” ， 其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释 “过拟合单个 Batch”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设 “single_batch_curve.png 和诊断记录” 中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：模型若不能在一个很小批次上把损失降得很低，通常存在标签、损失、梯度或模型连接错误。它是训练前最有效的体检。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标 “从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。” 解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和 “single_batch_curve.png 和诊断记录” 。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：一开始直接跑完整训练，错误成本过高。；修改官方代码却没有保存 diff。；只保存终端截图，没有配置、日志和命令。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 82 运行子集训练

今天要懂 子集训练用于验证完整训练链和趋势，不用于宣称最终性能。配置必须与正式实验分开，避免误把调试参数带入完整训练。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于 “运行子集训练” ，合格解释必须包含这一因果关系：子集训练用于验证完整训练链和趋势，不用于宣称最终性能。配置必须与正式实验分开，避免误把调试参数带入完整训练。。然后把它与本周目标连接起来：从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：使用百分之一到百分之十数据运行短训练，确认保存、恢复、日志、评估和中断重启都正常。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“debug 配置和可恢复检查点”，失败运行同样保留。

完成标准 今天结束前必须得到“debug 配置和可恢复检查点”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 所选论文官方 README、environment 文件、训练配置、评估脚本和补充材料。

今天先准备 建立 week12/day082 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：使用百分之一到百分之十数据运行短训练，确认保存、恢复、日志、评估和中断重启都正常。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“debug 配置和可恢复检查点”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“子集训练用于验证完整训练链和趋势，不用于宣称最终性能。配置必须与正式实验分开，避免误把调试参数带入完整训练。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 使用百分之一到百分之十数据运行短训练，确认保存、恢复、日志、评估和中断重启都正常。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：debug 配置和可恢复检查点。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“运行子集训练”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“debug 配置和可恢复检查点”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“运行子集训练”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“debug 配置和可恢复检查点”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：子集训练用于验证完整训练链和趋势，不用于宣称最终性能。配置必须与正式实验分开，避免误把调试参数带入完整训练。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“debug 配置和可恢复检查点”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：一开始直接跑完整训练，错误成本过高。；修改官方代码却没有保存 diff。；只保存终端截图，没有配置、日志和命令。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 83 运行官方评估

今天要懂 评价代码是复现的一部分。要确认输入预测格式、多个正确答案、无效样本处理和单位，不能只相信脚本最后打印的一个数字。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“运行官方评估”，合格解释必须包含这一因果关系：评价代码是复现的一部分。要确认输入预测格式、多个正确答案、无效样本处理和单位，不能只相信脚本最后打印的一个数字。。然后把它与本周目标连接起来：从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：对官方或子集预测运行评估，手算少量样本核对指标，保存 metrics.json。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“第一版基线结果与一键运行命令”，失败运行同样保留。

完成标准 今天结束前必须得到“第一版基线结果与一键运行命令”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 所选论文官方 README、environment 文件、训练配置、评估脚本和补充材料。

今天先准备 建立 week12/day083 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
----	---------

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“评价代码是复现的一部分。要确认输入预测格式、多个正确答案、无效样本处理和单位，不能只相信脚本最后打印的一个数字。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 对官方或子集预测运行评估，手算少量样本核对指标，保存 metrics.json。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：第一版基线结果与一键运行命令。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“运行官方评估”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“第一版基线结果与一键运行命令”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“运行官方评估”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“第一版基线结果与一键运行命令”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：评价代码是复现的一部分。要确认输入预测格式、多个正确答案、无效样本处理和单位，不能只相信脚本最后打印的一个数字。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“第一版基线结果与一键运行命令”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：一开始直接跑完整训练，错误成本过高。；修改官方代码却没有保存diff。；只保存终端截图，没有配置、日志和命令。。连续四十分钟没有进展时，缩小为一个样本或一个batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 84 第 12 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。

- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：冻结环境与版本、跑通官方推理、核对数据管道、过拟合单个 Batch、运行子集训练、运行官方评估。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“第一版基线结果与一键运行命令”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：从推理样例到小数据训练再到官方评估，形成第一条可重复执行的基线。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再执行本页验收。

第 13 周 定位复现差异

本周目标 系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。

首选材料 论文实验章节和补充材料，仓库配置历史、issues、数据集官方评测说明。

本周小白讲义

复现差异不应立即归因于随机性。按数据、预处理、模型、训练、评价和硬件六层逐项比较，并用一批固定样本检查中间张量。

先区分实现错误与合理差异。若评价脚本、数据版本或预训练权重不同，结果不完全一致是预期现象，但必须量化并说明。

调试时一次只改一项并保留前后日志。没有记录的多项同时修改会让任何改善都无法归因。

本周知识地图

天	主题	核心知识	产出
Day 85	建立数字对照	先确定你要复现论文哪张表的哪一行，并写清模型、数据、训练规模和指标。不同设置的数字不能直接比较。	结果差异总表
Day 86	核对数据划分	同名数据集可能有不同版本、过滤规则和 train val test 划分。重复样本、缺失文件和类别映射也会改变结果。	数据划分核对报告
Day 87	核对预处理	图像尺寸、裁剪、归一化、tokenizer、最大长度和文本清洗都会改变输入。训练和评估预处理还可能不同。	预处理逐项对照表
Day 88	核对指标协议	指标名称相同也可能实现不同，例如平均方式、大小写、标点、多答案处理和无效样本规则。评价协议必须以数据集官方说明为准。	指标单元测试和协议说明
Day 89	随机性与数值差异	随机种子不能消除所有非确定性，GPU 内核、混合精度和数据加载顺序都可能造成波动。一次运行不足以判定差异。	多种子基线统计

天	主题	核心知识	产出
Day 90	撰写差异说明	差异说明应把已验证原因、合理怀疑和未知项分开。不能确认的地方要保留不确定性，不能为了完整故事而编造原因。	复现差异说明文档
Day 91	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
comparison = {
    'paper_value': None, 'local_value': None,
    'data_split': '', 'preprocess': '', 'metric_impl': '',
    'seed': 42, 'known_difference': ''
}
```

本周常见错误

- 看到数字不同就立即改模型。
- 同时修改数据、学习率和评价脚本。
- 为了接近论文数字而在测试集上调参。

Day 85 建立数字对照

今天要懂 先确定你要复现论文哪张表的哪一行，并写清模型、数据、训练规模和指标。不同设置的数字不能直接比较。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“建立数字对照”，合格解释必须包含这一因果关系：先确定你要复现论文哪张表的哪一行，并写清模型、数据、训练规模和指标。不同设置的数字不能直接比较。。然后把它与本周目标连接起来：系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：把论文目标值、官方发布值和本地值放入同一表，逐项附上配置与来源。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“结果差异总表”，失败运行同样保留。

完成标准 今天结束前必须得到“结果差异总表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 论文实验章节和补充材料，仓库配置历史、issues、数据集官方评测说明。

今天先准备 建立 week13/day085 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：把论文目标值、官方发布值和本地值放入同一表，逐项附上配置与来源。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“结果差异总表”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“先确定你要复现论文哪张表的哪一行，并写清模型、数据、训练规模和指标。不同设置的数字不能直接比较。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 把论文目标值、官方发布值和本地值放入同一表，逐项附上配置与来源。

4 一百三十至一百六十分钟 整理并保存今日产出：结果差异总表。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“建立数字对照”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“结果差异总表”，其中包含日期、环境或配置、结果和一句结论。

自测题

1 不用资料，用三句话解释“建立数字对照”：第一句给定义，第二句说输入输出，第三句说一个失败条件。

2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。

3 假设“结果差异总表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。

4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

1 第一题至少应覆盖：先确定你要复现论文哪张表的哪一行，并写清模型、数据、训练规模和指标。不同设置的数字不能直接比较。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“结果差异总表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：看到数字不同就立即改模型。；同时修改数据、学习率和评价脚本。；为了接近论文数字而在测试集上调参。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 86 核对数据划分

今天要懂 同名数据集可能有不同版本、过滤规则和 train val test 划分。重复样本、缺失文件和类别映射也会改变结果。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“核对数据划分”，合格解释必须包含这一因果关系：同名数据集可能有不同版本、过滤规则和 train val test 划分。重复样本、缺失文件和类别映射也会改变结果。。然后把它与本周目标连接起来：系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：比较样本数量、文件哈希或 ID 列表，检查各划分交集和缺失，并保存统计。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“数据划分核对报告”，失败运行同样保留。

完成标准 今天结束前必须得到“数据划分核对报告”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 论文实验章节和补充材料，仓库配置历史、issues、数据集官方评测说明。

今天先准备 建立 week13/day086 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：比较样本数量、文件哈希或 ID 列表，检查各划分交集和缺失，并保存统计。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“数据划分核对报告”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“同名数据集可能有不同版本、过滤规则和 train val test 划分。重复样本、缺失文件和类别映射也会改变结果。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 比较样本数量、文件哈希或 ID 列表，检查各划分交集和缺失，并保存统计。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：数据划分核对报告。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“核对数据划分”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“数据划分核对报告”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“核对数据划分”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“数据划分核对报告”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：同名数据集可能有不同版本、过滤规则和 train val test 划分。重复样本、缺失文件和类别映射也会改变结果。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“数据划分核对报告”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：看到数字不同就立即改模型。；同时修改数据、学习率和评价脚本。；为了接近论文数字而在测试集上调参。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 87 核对预处理

今天要懂 图像尺寸、裁剪、归一化、tokenizer、最大长度和文本清洗都会改变输入。训练和评估预处理还可能不同。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“核对预处理”，合格解释必须包含这一因果关系：图像尺寸、裁剪、归一化、tokenizer、最大长度和文本清洗都会改变输入。训练和评估预处理还可能不同。。然后把它与本周目标连接起来：系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：从论文、配置和代码分别提取预处理，选同一原始样本逐步比较中间输出。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“预处理逐项对照表”，失败运行同样保留。

完成标准 今天结束前必须得到“预处理逐项对照表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 论文实验章节和补充材料，仓库配置历史、issues、数据集官方评测说明。

今天先准备 建立 week13/day087 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：从论文、配置和代码分别提取预处理，选同一原始样本逐步比较中间输出。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“预处理逐项对照表”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“图像尺寸、裁剪、归一化、tokenizer、最大长度和文本清洗都会改变输入。训练和评估预处理还可能不同。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 从论文、配置和代码分别提取预处理，选同一原始样本逐步比较中间输出。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：预处理逐项对照表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“核对预处理”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“预处理逐项对照表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“核对预处理”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“预处理逐项对照表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：图像尺寸、裁剪、归一化、tokenizer、最大长度和文本清洗都会改变输入。训练和评估预处理还可能不同。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“预处理逐项对照表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：看到数字不同就立即改模型。；同时修改数据、学习率和评价脚本。；为了接近论文数字而在测试集上调参。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 88 核对指标协议

今天要懂 指标名称相同也可能实现不同，例如平均方式、大小写、标点、多答案处理和无效样本规则。评价协议必须以数据集官方说明为准。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“核对指标协议”，合格解释必须包含这一因果关系：指标名称相同也可能实现不同，例如平均方式、大小写、标点、多答案处理和无效样本规则。评价协议必须以数据集官方说明为准。。然后把

它与本周目标连接起来：系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：手算十个样本，对比官方评估和本地实现；记录版本、参数和任何后处理。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“指标单元测试和协议说明”，失败运行同样保留。

完成标准 今天结束前必须得到“指标单元测试和协议说明”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 论文实验章节和补充材料，仓库配置历史、issues、数据集官方评测说明。

今天先准备 建立 week13/day088 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：手算十个样本，对比官方评估和本地实现；记录版本、参数和任何后处理。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“指标单元测试和协议说明”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“指标名称相同也可能实现不同，例如平均方式、大小写、标点、多答案处理和无效样本规则。评价协议必须以数据集官方说明为准。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 手算十个样本，对比官方评估和本地实现；记录版本、参数和任何后处理。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：指标单元测试和协议说明。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“核对指标协议”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“指标单元测试和协议说明”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“核对指标协议”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“指标单元测试和协议说明”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：指标名称相同也可能实现不同，例如平均方式、大小写、标点、多答案处理和无效样本规则。评价协议必须以数据集官方说明为准。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“指标单元测试和协议说明”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：看到数字不同就立即改模型。；同时修改数据、学习率和评价脚本。；为了接近论文数字而在测试集上调参。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 89 随机性与数值差异

今天要懂 随机种子不能消除所有非确定性，GPU 内核、混合精度和数据加载顺序都可能造成波动。一次运行不足以判定差异。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“随机性与数值差异”，合格解释必须包含这一因果关系：随机种子不能消除所有非确定性，GPU 内核、混合精度和数据加载顺序都可能造成波动。一次运行不足以判定差异。。然后把它与本周目标连接起来：系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：在相同配置下运行至少三个种子，报告均值、标准差和范围；记录确定性设置。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“多种子基线统计”，失败运行同样保留。

完成标准 今天结束前必须得到“多种子基线统计”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 论文实验章节和补充材料，仓库配置历史、issues、数据集官方评测说明。

今天先准备 建立 week13/day089 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：在相同配置下运行至少三个种子，报告均值、标准差和范围；记录确定性设置。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“多种子基线统计”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“随机种子不能消除所有非确定性，GPU 内核、混合精度和数据加载顺序都可能造成波动。一次运行不足以判定差异。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 在相同配置下运行至少三个种子，报告均值、标准差和范围；记录确定性设置。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：多种子基线统计。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“随机性与数值差异”，并说明它在本周任务中的作用。

- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“多种子基线统计”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“随机性与数值差异”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“多种子基线统计”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：随机种子不能消除所有非确定性，GPU 内核、混合精度和数据加载顺序都可能造成波动。一次运行不足以判定差异。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“多种子基线统计”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：看到数字不同就立即改模型。；同时修改数据、学习率和评价脚本。；为了接近论文数字而在测试集上调参。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 90 撰写差异说明

今天要懂 差异说明应把已验证原因、合理怀疑和未知项分开。不能确认的地方要保留不确定性，不能为了完整故事而编造原因。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“撰写差异说明”，合格解释必须包含这一因果关系：差异说明应把已验证原因、合理怀疑和未知项分开。不能确认的地方要保留不确定性，不能为了完整故事而编造原因。。然后把它与本周目标连接起来：系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：按数据、实现、训练、指标、随机性五类整理证据，写出下一轮只改变一项的行动。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“复现差异说明文档”，失败运行同样保留。

完成标准 今天结束前必须得到“复现差异说明文档”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 论文实验章节和补充材料，仓库配置历史、issues、数据集官方评测说明。

今天先准备 建立 week13/day090 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：按数据、实现、训练、指标、随机性五类整理证据，写出下一轮只改变一项的行动。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“复现差异说明文档”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“差异说明应把已验证原因、合理怀疑和未知项分开。不能确认的地方要保留不确定性，不能为了完整故事而编造原因。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 按数据、实现、训练、指标、随机性五类整理证据，写出下一轮只改变一项的行动。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：复现差异说明文档。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“撰写差异说明”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“复现差异说明文档”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“撰写差异说明”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“复现差异说明文档”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：差异说明应把已验证原因、合理怀疑和未知项分开。不能确认的地方要保留不确定性，不能为了完整故事而编造原因。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“复现差异说明文档”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：看到数字不同就立即改模型。；同时修改数据、学习率和评价脚本。；为了接近论文数字而在测试集上调参。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 91 第 13 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：建立数字对照、核对数据划分、核对预处理、核对指标协议、随机性与数值差异、撰写差异说明。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“复现差异说明文档”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：系统核对论文结果与本地结果的差距，区分实现错误、协议差异、随机性和无法确认的信息。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 14 周 消融实验与统计

本周目标 用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。

首选材料 目标论文消融章节，统计均值与标准差基础，实验跟踪工具官方文档。

本周小白讲义

消融实验回答组件是否必要，敏感性实验回答结论是否依赖超参数。两者都需要与主结果相同的数据、训练预算和评价协议。

多个随机种子用于估计训练波动。报告均值、标准差和运行数；小提升若落在波动范围内，应降低结论强度。

统计检验不是显著性印章。先确认比较是否配对、样本是否独立以及指标分布，再选择方法并报告效应大小。

本周知识地图

天	主题	核心知识	产出
Day 92	提出因果问题	消融不是随便关模块，而是回答某个组件在其他条件相同时是否贡献效果。先写因果问题，再决定比较设置。	一条可检验消融问题
Day 93	定义变量与控制项	自变量是你主动改变的因素，因变量是观察指标，控制变量是保持一致的条件。算力预算也应作为控制或报告项。	消融控制变量表
Day 94	设计随机种子	多种子用于估计训练随机性。种子数量应在预算内预先决定，失败运行也要记录，不能看到结果后任意增加。	种子计划与异常规则
Day 95	执行实验矩阵	实验矩阵要有唯一名称、完整配置和状态。一次只改目标因素，运行顺序可随机化以降低机器状态带来的偏差。	完整实验目录
Day 96	汇总均值与标准差	平均值表示中心表现，标准差表示波动。提升若小于波动，需要谨慎解释；不要只报告最优种子。	统计汇总脚本和结果表

天	主题	核心知识	产出
Day 97	制作消融表	消融表应明确组件、指标、误差和成本。加粗最好值不能替代解释，结论要说明稳定性和适用条件。	论文级消融表和分析
Day 98	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
import numpy as np
values = np.array([0.731, 0.724, 0.736])
print(values.mean(), values.std(ddof=1))
```

本周常见错误

- 只运行一次就比较小数点后的提升。
- 删除模块时同时改变参数量和训练步数。
- 从大量实验中只挑最好结果。

Day 92 提出因果问题

今天要懂 消融不是随便关模块，而是回答某个组件在其他条件相同时是否贡献效果。先写因果问题，再决定比较设置。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“提出因果问题”，合格解释必须包含这一因果关系：消融不是随便关模块，而是回答某个组件在其他条件相同时是否贡献效果。先写因果问题，再决定比较设置。。然后把它与本周目标连接起来：用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：从论文方法中选一个核心组件，写出有组件和无组件的预期差异及可能混杂因素。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“一条可检验消融问题”，失败运行同样保留。

完成标准 今天结束前必须得到“一条可检验消融问题”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标论文消融章节，统计均值与标准差基础，实验跟踪工具官方文档。

今天先准备 建立 week14/day092 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：从论文方法中选一个核心组件，写出有组件和无组件的预期差异及可能混杂因素。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“一条可检验消融问题”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“消融不是随便关模块，而是回答某个组件在其他条件相同时是否贡献效果。先写因果问题，再决定比较设置。”写不少于五条自己的话，并画出输入、处理、输出。

- 3 六十至一百三十分钟** 从论文方法中选一个核心组件，写出有组件和无组件的预期差异及可能混杂因素。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：一条可检验消融问题。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“提出因果问题”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“一条可检验消融问题”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“提出因果问题”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“一条可检验消融问题”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：消融不是随便关模块，而是回答某个组件在其他条件相同时是否贡献效果。先写因果问题，再决定比较设置。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“一条可检验消融问题”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只运行一次就比较小数点后的提升。；删除模块时同时改变参数量和训练步数。；从大量实验中只挑最好结果。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 93 定义变量与控制项

今天要懂 自变量是你主动改变的因素，因变量是观察指标，控制变量是保持一致的条件。算力预算也应作为控制或报告项。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“定义变量与控制项”，合格解释必须包含这一因果关系：自变量是你主动改变的因素，因变量是观察指标，控制变量是保持一致的条件。算力预算也应作为控制或报告项。。然后把它与本周目标连接起来：用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：建立实验变量表，列出数据、初始化、训练步数、优化器、评估和硬件是否一致。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“消融控制变量表”，失败运行同样保留。

完成标准 今天结束前必须得到“消融控制变量表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标论文消融章节，统计均值与标准差基础，实验跟踪工具官方文档。

今天先准备 建立 week14/day093 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：建立实验变量表，列出数据、初始化、训练步数、优化器、评估和硬件是否一致。运行是否完整、日志是否连续、检查点是否可恢复。

阶段	今天具体做什么
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“消融控制变量表”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“自变量是你主动改变的因素，因变量是观察指标，控制变量是保持一致的条件。算力预算也应作为控制或报告项。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 建立实验变量表，列出数据、初始化、训练步数、优化器、评估和硬件是否一致。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：消融控制变量表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“定义变量与控制项”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“消融控制变量表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“定义变量与控制项”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。

3 假设“消融控制变量表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。

4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

1 第一题至少应覆盖：自变量是你主动改变的因素，因变量是观察指标，控制变量是保持一致的条件。算力预算也应作为控制或报告项。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“消融控制变量表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只运行一次就比较小数点后的提升。；删除模块时同时改变参数量和训练步数。；从大量实验中只挑最好结果。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 94 设计随机种子

今天要懂 多种子用于估计训练随机性。种子数量应在预算内预先决定，失败运行也要记录，不能看到结果后任意增加。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“设计随机种子”，合格解释必须包含这一因果关系：多种子用于估计训练随机性。种子数量应在预算内预先决定，失败运行也要记录，不能看到结果后任意增加。。然后把它与本周目标连接起来：用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：选择至少三个种子，先跑短实验确认波动和成本，定义异常运行判定规则。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“种子计划与异常规则”，失败运行同样保留。

完成标准 今天结束前必须得到“种子计划与异常规则”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标论文消融章节，统计均值与标准差基础，实验跟踪工具官方文档。

今天先准备 建立 week14/day094 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：选择至少三个种子，先跑短实验确认波动和成本，定义异常运行判定规则。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“种子计划与异常规则”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“多种子用于估计训练随机性。种子数量应在预算内预先决定，失败运行也要记录，不能看到结果后任意增加。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 选择至少三个种子，先跑短实验确认波动和成本，定义异常运行判定规则。

4 一百三十至一百六十分钟 整理并保存今日产出：种子计划与异常规则。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“设计随机种子”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“种子计划与异常规则”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“设计随机种子”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“种子计划与异常规则”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：多种子用于估计训练随机性。种子数量应在预算内预先决定，失败运行也要记录，不能看到结果后任意增加。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“种子计划与异常规则”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只运行一次就比较小数点后的提升。；删除模块时同时改变参数量和训练步数。；从大量实验中只挑最好结果。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 95 执行实验矩阵

今天要懂 实验矩阵要有唯一名称、完整配置和状态。一次只改目标因素，运行顺序可随机化以降低机器状态带来的偏差。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“执行实验矩阵”，合格解释必须包含这一因果关系：实验矩阵要有唯一名称、完整配置和状态。一次只改目标因素，运行顺序可随机化以降低机器状态带来的偏差。。然后把它与本周目标连接起来：用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：运行基线和三个设置，自动保存配置、日志、指标和预测；失败时保留日志。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“完整实验目录”，失败运行同样保留。

完成标准 今天结束前必须得到“完整实验目录”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标论文消融章节，统计均值与标准差基础，实验跟踪工具官方文档。

今天先准备 建立 week14/day095 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：运行基线和三个设置，自动保存配置、日志、指标和预测；失败时保留日志。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“完整实验目录”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“实验矩阵要有唯一名称、完整配置和状态。一次只改目标因素，运行顺序可随机化以降低机器状态带来的偏差。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 运行基线和三个设置，自动保存配置、日志、指标和预测；失败时保留日志。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：完整实验目录。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“执行实验矩阵”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“完整实验目录”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“执行实验矩阵”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“完整实验目录”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：实验矩阵要有唯一名称、完整配置和状态。一次只改目标因素，运行顺序可随机化以降低机器状态带来的偏差。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“完整实验目录”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只运行一次就比较小数点后的提升。；删除模块时同时改变参数量和训练步数。；从大量实验中只挑最好结果。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 96 汇总均值与标准差

今天要懂 平均值表示中心表现，标准差表示波动。提升若小于波动，需要谨慎解释；不要只报告最优种子。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“汇总均值与标准差”，合格解释必须包含这一因果关系：平均值表示中心表现，标准差表示波动。提升若小于波动，需要谨慎解释；不要只报告最优种子。。然后把它与本周目标连接起来：用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：从 metrics 文件自动汇总均值、样本标准差、运行数和成本，检查单位与小数位。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“统计汇总脚本和结果表”，失败运行同样保留。

完成标准 今天结束前必须得到“统计汇总脚本和结果表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标论文消融章节，统计均值与标准差基础，实验跟踪工具官方文档。

今天先准备 建立 week14/day096 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：从 metrics 文件自动汇总均值、样本标准差、运行数和成本，检查单位与小数位。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“统计汇总脚本和结果表”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“平均值表示中心表现，标准差表示波动。提升若小于波动，需要谨慎解释；不要只报告最优种子。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 从 metrics 文件自动汇总均值、样本标准差、运行数和成本，检查单位与小数位。
- 4 一百三十至一百六十分** 整理并保存今日产出：统计汇总脚本和结果表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“汇总均值与标准差”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“统计汇总脚本和结果表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“汇总均值与标准差”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“统计汇总脚本和结果表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：平均值表示中心表现，标准差表示波动。提升若小于波动，需要谨慎解释；不要只报告最优种子。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“统计汇总脚本和结果表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只运行一次就比较小数点后的提升。；删除模块时同时改变参数量和训练步数。；从大量实验中只挑最好结果。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 97 制作消融表

今天要懂 消融表应明确组件、指标、误差和成本。加粗最好值不能替代解释，结论要说明稳定性和适用条件。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“制作消融表”，合格解释必须包含这一因果关系：消融表应明确组件、指标、误差和成本。加粗最好值不能替代解释，结论要说明稳定性和适用条件。。然后把它与本周目标连接起来：用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：生成论文表格，写三段解释：主要发现、意外结果、结论边界。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“论文级消融表和分析”，失败运行同样保留。

完成标准 今天结束前必须得到“论文级消融表和分析”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标论文消融章节，统计均值与标准差基础，实验跟踪工具官方文档。

今天先准备 建立 week14/day097 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：生成论文表格，写三段解释：主要发现、意外结果、结论边界。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“论文级消融表和分析”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“消融表应明确组件、指标、误差和成本。加粗最好值不能替代解释，结论要说明稳定性和适用条件。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 生成论文表格，写三段解释：主要发现、意外结果、结论边界。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：论文级消融表和分析。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“制作消融表”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“论文级消融表和分析”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“制作消融表”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“论文级消融表和分析”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：消融表应明确组件、指标、误差和成本。加粗最好值不能替代解释，结论要说明稳定性和适用条件。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 第四题应让别人找到输入、环境、运行命令、输出目录和“论文级消融表和分析”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只运行一次就比较小数点后的提升。；删除模块时同时改变参数量和训练步数。；从大量实验中只挑最好结果。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 98 第 14 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：提出因果问题、定义变量与控制项、设计随机种子、执行实验矩阵、汇总均值与标准差、制作消融表。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“论文级消融表和分析”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。

- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：用控制变量和多随机种子判断组件是否真正有效，形成可以放入论文的消融表。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 15 周 错误分析与效率

本周目标 建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。

首选材料 数据集错误分析惯例，混淆矩阵、校准与效率测量教程，相关论文失败案例。

本周小白讲义

错误分析先建立互斥或可多标签的错误规范，再在不知道模型设置的情况下标注。典型类别包括感知、OCR、计数、关系、知识、推理、检索和生成幻觉。

效率测量要固定硬件、精度、输入长度和 batch，先预热再计时。报告中位延迟、P95、吞吐和峰值显存，避免只报最有利数字。

错误图谱应同时包含比例、代表样例、证据和可能原因。原因若未被实验验证，必须明确写成假设。

本周知识地图

天	主题	核心知识	产出
Day 99	抽样与保存预测	错误分析需要可追溯预测，包括样本 ID、输入、真实答案、模型输出、置信度和配置。抽样要覆盖类别与难度。	prediction_analysis.jsonl
Day 100	定义错误分类	错误标签要互斥或明确允许多标签，并有可执行定义。多模态常见类别包括感知、OCR、计数、关系、知识、推理和数据问题。	错误标注规范
Day 101	盲法标注与复核	先不知道实验设置地标注能减少确认偏差。至少隔一段时间复核一部分样本，有同伴时可计算一致性。	错误标注表和一致性记录
Day 102	混淆与校准	混淆矩阵显示类别之间的系统混淆，校准检查置信度是否对应真实正确率。高置信错误比低置信错误风险更大。	混淆矩阵与可靠性图
Day 103	测量显存 延迟与吞吐	效率测量要固定 batch、输入尺寸、精度和硬件，先预热再计时。延迟是单次时间，吞吐是单位时间样本数，两者场景不同。	效率测量脚本和表

天	主题	核心知识	产出
Day 104	形成错误图谱	高质量分析把错误比例、代表案例和可能原因联系起来，并区分证据与猜测。每个改进方向都应指向可测问题。	错误图谱与效率结论
Day 105	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
errors = predictions[predictions['correct'] == 0]
summary = errors.groupby('error_type').size().sort_values(ascending=False)
print(summary)
```

本周常见错误

- 只展示最有趣的三个错误。
- 先看方法再定义错误标签，造成偏向。
- 测延迟时把模型加载和首次预热混入。

Day 99 抽样与保存预测

今天要懂 错误分析需要可追溯预测，包括样本 ID、输入、真实答案、模型输出、置信度和配置。抽样要覆盖类别与难度。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“抽样与保存预测”，合格解释必须包含这一因果关系：错误分析需要可追溯预测，包括样本 ID、输入、真实答案、模型输出、置信度和配置。抽样要覆盖类别与难度。。然后把它与本周目标连接起来：建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：导出验证集预测，按类别分层抽取至少五十个错误和二十个正确样本。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“prediction_analysis.jsonl”，失败运行同样保留。

完成标准 今天结束前必须得到“prediction_analysis.jsonl”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 数据集错误分析惯例，混淆矩阵、校准与效率测量教程，相关论文失败案例。

今天先准备 建立 week15/day099 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：导出验证集预测，按类别分层抽取至少五十个错误和二十个正确样本。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到 “prediction_analysis.jsonl”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“错误分析需要可追溯预测，包括样本 ID、输入、真实答案、模型输出、置信度和配置。抽样要覆盖类别与难度。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 导出验证集预测，按类别分层抽取至少五十个错误和二十个正确样本。

- 4 一百三十至一百六十分钟** 整理并保存今日产出：prediction_analysis.jsonl。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“抽样与保存预测”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“prediction_analysis.jsonl”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“抽样与保存预测”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“prediction_analysis.jsonl”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：错误分析需要可追溯预测，包括样本 ID、输入、真实答案、模型输出、置信度和配置。抽样要覆盖类别与难度。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 第四题应让别人找到输入、环境、运行命令、输出目录和“prediction_analysis.jsonl”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只展示最有趣的三个错误。；先看方法再定义错误标签，造成偏向。；测试延迟时把模型加载和首次预热混入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 100 定义错误分类

今天要懂 错误标签要互斥或明确允许多标签，并有可执行定义。多模态常见类别包括感知、OCR、计数、关系、知识、推理和数据问题。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“定义错误分类”，合格解释必须包含这一因果关系：错误标签要互斥或明确允许多标签，并有可执行定义。多模态常见类别包括感知、OCR、计数、关系、知识、推理和数据问题。。然后把它与本周目标连接起来：建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：在看完少量样本后写错误标签手册，为每类给出正例、反例和边界情况。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“错误标注规范”，失败运行同样保留。

完成标准 今天结束前必须得到“错误标注规范”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 数据集错误分析惯例，混淆矩阵、校准与效率测量教程，相关论文失败案例。

今天先准备 建立 week15/day100 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：在看完少量样本后写错误标签手册，为每类给出正例、反例和边界情况。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“错误标注规范”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“错误标签要互斥或明确允许多标签，并有可执行定义。多模态常见类别包括感知、OCR、计数、关系、知识、推理和数据问题。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 在看完少量样本后写错误标签手册，为每类给出正例、反例和边界情况。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：错误标注规范。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“定义错误分类”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“错误标注规范”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“定义错误分类”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“错误标注规范”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：错误标签要互斥或明确允许多标签，并有可执行定义。多模态常见类别包括感知、OCR、计数、关系、知识、推理和数据问题。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“错误标注规范”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只展示最有趣的三个错误。；先看方法再定义错误标签，造成偏向。；测试延迟时把模型加载和首次预热混入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 101 盲法标注与复核

今天要懂 先不知道实验设置地标注能减少确认偏差。至少隔一段时间复核一部分样本，有同伴时可计算一致性。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“盲法标注与复核”，合格解释必须包含这一因果关系：先不知道实验设置地标注能减少确认偏差。至少隔一段时间复核一部分样本，有同伴时可计算一致性。。然后把它与本周目标连接起来：建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：隐藏模型设置标注全部样本，第二次复核百分之二十，记录分歧及最终规则。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“错误标注表和一致性记录”，失败运行同样保留。

完成标准 今天结束前必须得到“错误标注表和一致性记录”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 数据集错误分析惯例，混淆矩阵、校准与效率测量教程，相关论文失败案例。

今天先准备 建立 week15/day101 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：隐藏模型设置标注全部样本，第二次复核百分之二十，记录分歧及最终规则。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“错误标注表和一致性记录”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“先不知道实验设置地标注能减少确认偏差。至少隔一段时间复核一部分样本，有同伴时可计算一致性。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 隐藏模型设置标注全部样本，第二次复核百分之二十，记录分歧及最终规则。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：错误标注表和一致性记录。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“盲法标注与复核”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“错误标注表和一致性记录”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“盲法标注与复核”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“错误标注表和一致性记录”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：先不知道实验设置地标注能减少确认偏差。至少隔一段时间复核一部分样本，有同伴时可计算一致性。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“错误标注表和一致性记录”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只展示最有趣的三个错误。；先看方法再定义错误标签，造成偏向。；测试延迟时把模型加载和首次预热混入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 102 混淆与校准

今天要懂 混淆矩阵显示类别之间的系统混淆，校准检查置信度是否对应真实正确率。高置信错误比低置信错误风险更大。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“混淆与校准”，合格解释必须包含这一因果关系：混淆矩阵显示类别之间的系统混淆，校准

检查置信度是否对应真实正确率。高置信错误比低置信错误风险更大。。然后把它与本周目标连接起来：建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：生成归一化混淆矩阵，把预测按置信度分箱，比较每箱平均置信度和准确率。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“混淆矩阵与可靠性图”，失败运行同样保留。

完成标准 今天结束前必须得到“混淆矩阵与可靠性图”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 数据集错误分析惯例，混淆矩阵、校准与效率测量教程，相关论文失败案例。

今天先准备 建立 week15/day102 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：生成归一化混淆矩阵，把预测按置信度分箱，比较每箱平均置信度和准确率。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“混淆矩阵与可靠性图”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“混淆矩阵显示类别之间的系统混淆，校准检查置信度是否对应真实正确率。高置信错误比低置信错误风险更大。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 生成归一化混淆矩阵，把预测按置信度分箱，比较每箱平均置信度和准确率。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：混淆矩阵与可靠性图。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“混淆与校准”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“混淆矩阵与可靠性图”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“混淆与校准”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“混淆矩阵与可靠性图”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：混淆矩阵显示类别之间的系统混淆，校准检查置信度是否对应真实正确率。高置信错误比低置信错误风险更大。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“混淆矩阵与可靠性图”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只展示最有趣的三个错误。；先看方法再定义错误标签，造成偏向。；测试延迟时把模型加载和首次预热混入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 103 测量显存 延迟与吞吐

今天要懂 效率测量要固定 batch、输入尺寸、精度和硬件，先预热再计时。延迟是单次时间，吞吐是单位时间样本数，两者场景不同。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“测量显存 延迟与吞吐”，合格解释必须包含这一因果关系：效率测量要固定 batch、输入尺寸、精度和硬件，先预热再计时。延迟是单次时间，吞吐是单位时间样本数，两者场景不同。。然后把它与本周目标连接起来：建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：预热十次，正式测量五十次，报告中位延迟、P95、吞吐和峰值显存。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“效率测量脚本和表”，失败运行同样保留。

完成标准 今天结束前必须得到“效率测量脚本和表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 数据集错误分析惯例，混淆矩阵、校准与效率测量教程，相关论文失败案例。

今天先准备 建立 week15/day103 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。

阶段	今天具体做什么
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：预热十次，正式测量五十次，报告中位延迟、P95、吞吐和峰值显存。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“效率测量脚本和表”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“效率测量要固定 batch、输入尺寸、精度和硬件，先预热再计时。延迟是单次时间，吞吐是单位时间样本数，两者场景不同。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 预热十次，正式测量五十次，报告中位延迟、P95、吞吐和峰值显存。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：效率测量脚本和表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“测量显存 延迟与吞吐”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“效率测量脚本和表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“测量显存 延迟与吞吐”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“效率测量脚本和表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：效率测量要固定 batch、输入尺寸、精度和硬件，先预热再计时。延迟是单次时间，吞吐是单位时间样本数，两者场景不同。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“效率测量脚本和表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只展示最有趣的三个错误。；先看方法再定义错误标签，造成偏向。；测试延迟时把模型加载和首次预热混入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 104 形成错误图谱

今天要懂 高质量分析把错误比例、代表案例和可能原因联系起来，并区分证据与猜测。每个改进方向都应指向可测问题。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“形成错误图谱”，合格解释必须包含这一因果关系：高质量分析把错误比例、代表案例和可能原因联系起来，并区分证据与猜测。每个改进方向都应指向可测问题。。然后把它与本周目标连接起来：建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：制作成功与失败对照图，写出三项主要瓶颈和对应下一步实验。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“错误图谱与效率结论”，失败运行同样保留。

完成标准 今天结束前必须得到“错误图谱与效率结论”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 数据集错误分析惯例，混淆矩阵、校准与效率测量教程，相关论文失败案例。

今天先准备 建立 week15/day104 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：制作成功与失败对照图，写出三项主要瓶颈和对应下一步实验。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“错误图谱与效率结论”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“高质量分析把错误比例、代表案例和可能原因联系起来，并区分证据与猜测。每个改进方向都应指向可测问题。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 制作成功与失败对照图，写出三项主要瓶颈和对应下一步实验。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：错误图谱与效率结论。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“形成错误图谱”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“错误图谱与效率结论”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“形成错误图谱”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“错误图谱与效率结论”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：高质量分析把错误比例、代表案例和可能原因联系起来，并区分证据与猜测。每个改进方向都应指向可测问题。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“错误图谱与效率结论”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只展示最有趣的三个错误。；先看方法再定义错误标签，造成偏向。；测试延迟时把模型加载和首次预热混入。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 105 第 15 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：抽样与保存预测、定义错误分类、盲法标注与复核、混淆与校准、测量显存延迟与吞吐、形成错误图谱。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“错误图谱与效率结论”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- □ 能独立解释本周目标：建立稳定的错误类型，解释模型在哪些样本失败，并量化显存、延迟和吞吐。
- □ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- □ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- □ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 16 周 复现报告与仓库

本周目标 把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。

首选材料 目标论文写作结构，README 最佳实践，软件可复现性清单。

本周小白讲义

复现报告按目标、方法、设置、结果、差异和限制组织。它不是流水账，而是回答是否复现、在哪些条件下复现以及差异从何而来。

README 是复现接口，必须包含环境、数据、命令、预计资源、输出位置和已知限制。干净环境测试比作者本人再次运行更有价值。

发布版本应固定代码、配置和结果包。后续修改用新版本，不覆盖已经用于报告的文件。

本周知识地图

天	主题	核心知识	产出
Day 106	确定报告主线	复现报告的主线是目标、方法、设置、结果、差异和限制。它不是学习日记，也不需要重复原论文所有内容。	报告大纲和主张清单
Day 107	重写方法与形状	方法部分应让读者理解输入如何变成输出，哪些模块训练、哪些冻结、损失如何组合。不要只复制论文原句。	方法章节和流程图
Day 108	写实验设置与结果	设置必须包含数据版本、划分、预处理、指标、硬件和训练配置。结果表要区分论文值、官方值和本地值。	实验章节初稿
Day 109	写错误分析与限制	限制不是礼貌性结尾，而是指出结论不能扩展到哪些条件。错误分析用于说明失败模式，不应只挑模型看起来聪明的例子。	分析与限制章节
Day 110	整理 README 和入口	README 应让陌生读者知道做什么、需要什么、如何运行、会得到什么。命令中的路径、配置和检查点必须真实存在。	可执行 README 和入口脚本

天	主题	核心知识	产出
Day 111	他人复现与定稿	真正的复现测试是让不了解你代码的人运行。对方的问题优先反映文档和接口不足，不应归咎于对方。	四至六页复现报告和发布版本
Day 112	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
# README 最小结构
## 复现目标
## 环境
## 数据
## 训练
## 评估
## 结果
## 已知差异与限制
```

本周常见错误

- README 只有安装命令，没有预期输出。
- 报告结果无法追溯到具体配置。
- 清理仓库时误删失败日志和负结果。

Day 106 确定报告主线

今天要懂 复现报告的主线是目标、方法、设置、结果、差异和限制。它不是学习日记，也不需要重复原论文所有内容。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“确定报告主线”，合格解释必须包含这一因果关系：复现报告的主线是目标、方法、设置、结果、差异和限制。它不是学习日记，也不需要重复原论文所有内容。。然后把它与本周目标连接起来：把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：把已有材料按六个问题排序，给每节写一句主旨和证据来源。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“报告大纲和主张清单”，失败运行同样保留。

完成标准 今天结束前必须得到“报告大纲和主张清单”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标论文写作结构，README 最佳实践，软件可复现性清单。

今天先准备 建立 week16/day106 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：把已有材料按六个问题排序，给每节写一句主旨和证据来源。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“报告大纲和主张清单”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“复现报告的主线是目标、方法、设置、结果、差异和限制。它不是学习日记，也不需要重复原论文所有内容。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 把已有材料按六个问题排序，给每节写一句主旨和证据来源。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：报告大纲和主张清单。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“确定报告主线”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“报告大纲和主张清单”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“确定报告主线”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“报告大纲和主张清单”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：复现报告的主线是目标、方法、设置、结果、差异和限制。它不是学习日记，也不需要重复原论文所有内容。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“报告大纲和主张清单”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：README 只有安装命令，没有预期输出。；报告结果无法追溯到具体配置。；清理仓库时误删失败日志和负结果。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 107 重写方法与形状

今天要懂 方法部分应让读者理解输入如何变成输出，哪些模块训练、哪些冻结、损失如何组合。不要只复制论文原句。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“重写方法与形状”，合格解释必须包含这一因果关系：方法部分应让读者理解输入如何变成输出，哪些模块训练、哪些冻结、损失如何组合。不要只复制论文原句。。然后把它与本周目标连接起来：把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：依据代码重画简化方法图，写出核心张量形状、训练参数和损失。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“方法章节和流程图”，失败运行同样保留。

完成标准 今天结束前必须得到“方法章节和流程图”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标论文写作结构，README 最佳实践，软件可复现性清单。

今天先准备 建立 week16/day107 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：依据代码重画简化方法图，写出核心张量形状、训练参数和损失。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。

阶段	今天具体做什么
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“方法章节和流程图”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“方法部分应让读者理解输入如何变成输出，哪些模块训练、哪些冻结、损失如何组合。不要只复制论文原句。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 依据代码重画简化方法图，写出核心张量形状、训练参数和损失。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：方法章节和流程图。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“重写方法与形状”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“方法章节和流程图”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“重写方法与形状”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“方法章节和流程图”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：方法部分应让读者理解输入如何变成输出，哪些模块训练、哪些冻结、损失如何组合。不要只复制论文原句。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“方法章节和流程图”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：README 只有安装命令，没有预期输出。；报告结果无法追溯到具体配置。；清理仓库时误删失败日志和负结果。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 108 写实验设置与结果

今天要懂 设置必须包含数据版本、划分、预处理、指标、硬件和训练配置。结果表要区分论文值、官方值和本地值。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“写实验设置与结果”，合格解释必须包含这一因果关系：设置必须包含数据版本、划分、预处理、指标、硬件和训练配置。结果表要区分论文值、官方值和本地值。。然后把它与本周目标连接起来：把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：生成设置表、主结果表和消融表，确保数字直接来自冻结指标文件。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“实验章节初稿”，失败运行同样保留。

完成标准 今天结束前必须得到“实验章节初稿”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标论文写作结构，README 最佳实践，软件可复现性清单。

今天先准备 建立 week16/day108 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：生成设置表、主结果表和消融表，确保数字直接来自冻结指标文件。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“实验章节初稿”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“设置必须包含数据版本、划分、预处理、指标、硬件和训练配置。结果表要区分论文值、官方值和本地值。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 生成设置表、主结果表和消融表，确保数字直接来自冻结指标文件。
- 4 一百三十至一百六十分** 整理并保存今日产出：实验章节初稿。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“写实验设置与结果”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“实验章节初稿”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“写实验设置与结果”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“实验章节初稿”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：设置必须包含数据版本、划分、预处理、指标、硬件和训练配置。结果表要区分论文值、官方值和本地值。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“实验章节初稿”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：README 只有安装命令，没有预期输出。；报告结果无法追溯到具体配置。；清理仓库时误删失败日志和负结果。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 109 写错误分析与限制

今天要懂 限制不是礼貌性结尾，而是指出结论不能扩展到哪些条件。错误分析用于说明失败模式，不应只挑模型看起来聪明的例子。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“写错误分析与限制”，合格解释必须包含这一因果关系：限制不是礼貌性结尾，而是指出结

论不能扩展到哪些条件。错误分析用于说明失败模式，不应只挑模型看起来聪明的例子。。然后把它与本周目标连接起来：把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：加入错误分布、成功失败样例、效率成本和至少三条具体限制。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“分析与限制章节”，失败运行同样保留。

完成标准 今天结束前必须得到“分析与限制章节”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标论文写作结构，README 最佳实践，软件可复现性清单。

今天先准备 建立 week16/day109 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：加入错误分布、成功失败样例、效率成本和至少三条具体限制。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“分析与限制章节”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“限制不是礼貌性结尾，而是指出结论不能扩展到哪些条件。错误分析用于说明失败模式，不应只挑模型看起来聪明的例子。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 加入错误分布、成功失败样例、效率成本和至少三条具体限制。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：分析与限制章节。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“写错误分析与限制”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“分析与限制章节”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“写错误分析与限制”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“分析与限制章节”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：限制不是礼貌性结尾，而是指出结论不能扩展到哪些条件。错误分析用于说明失败模式，不应只挑模型看起来聪明的例子。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“分析与限制章节”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：README 只有安装命令，没有预期输出。；报告结果无法追溯到具体配置。；清理仓库时误删失败日志和负结果。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 110 整理 README 和入口

今天要懂 README 应让陌生读者知道做什么、需要什么、如何运行、会得到什么。命令中的路径、配置和检查点必须真实存在。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“整理 README 和入口”，合格解释必须包含这一因果关系：README 应让陌生读者知道做什么、需要什么、如何运行、会得到什么。命令中的路径、配置和检查点必须真实存在。。然后把它与本周目标连接起来：把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：从新目录按 README 安装或让另一环境试跑，修复每个缺失步骤。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“可执行 README 和入口脚本”，失败运行同样保留。

完成标准 今天结束前必须得到“可执行 README 和入口脚本”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标论文写作结构，README 最佳实践，软件可复现性清单。

今天先准备 建立 week16/day110 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。

阶段	今天具体做什么
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：从新目录按 README 安装或让另一环境试跑，修复每个缺失步骤。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“可执行 README 和入口脚本”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“README 应让陌生读者知道做什么、需要什么、如何运行、会得到什么。命令中的路径、配置和检查点必须真实存在。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 从新目录按 README 安装或让另一环境试跑，修复每个缺失步骤。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：可执行 README 和入口脚本。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“整理 README 和入口”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“可执行 README 和入口脚本”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“整理 README 和入口”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“可执行 README 和入口脚本”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：README 应让陌生读者知道做什么、需要什么、如何运行、会得到什么。命令中的路径、配置和检查点必须真实存在。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“可执行 README 和入口脚本”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：README 只有安装命令，没有预期输出。；报告结果无法追溯到具体配置。；清理仓库时误删失败日志和负结果。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 111 他人复现与定稿

今天要懂 真正的复现测试是让不了解你代码的人运行。对方的问题优先反映文档和接口不足，不应归咎于对方。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“他人复现与定稿”，合格解释必须包含这一因果关系：真正的复现测试是让不了解你代码的人运行。对方的问题优先反映文档和接口不足，不应归咎于对方。。然后把它与本周目标连接起来：把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。

科研连接 今天的工作必须留下可审计证据。结果文件应能回到代码版本、数据版本、配置和命令，否则它不能进入复现报告或论文。

跟着做一个例子

将今天的实践视为一次正式复现运行：请同学或使用干净环境执行核心命令，记录失败点，修正文档后冻结版本。。运行前登记假设、代码版本、数据版本、配置和命令；运行后把日志、指标、预测和结论汇总到“四至六页复现报告和发布版本”，失败运行同样保留。

完成标准 今天结束前必须得到“四至六页复现报告和发布版本”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标论文写作结构，README 最佳实践，软件可复现性清单。

今天先准备 建立 week16/day111 文件夹，复制上一日的配置并生成新的 run id，不覆盖旧结果。运行前登记 Git commit、数据版本、模型权重、随机种子、硬件、预算和本日只改变的一个因素。先用 1% 至 10% 数据或 50 至 200 个 step 做短跑；短跑通过后再安排完整运行，并保留失败日志。

阶段	今天具体做什么
锁定运行	从配置文件启动，不在 Notebook 里临时改变量；终端命令、commit、数据路径和输出目录写入 run.md。
先做短跑	先执行一个可在十至三十分钟内结束的版本，检查：请同学或使用干净环境执行核心命令，记录失败点，修正文档后冻结版本。运行是否完整、日志是否连续、检查点是否可恢复。
核对结果	手算或抽查至少 5 个样本，确认预测格式、标签映射、评估脚本和单位没有错误；不能只相信最后一行数字。
保存差异	把 metrics、预测、图表、资源消耗和失败原因归档到“四至六页复现报告和发布版本”；失败运行也要保留。

通过标准

- 运行可由配置和命令重启，Git commit、数据版本、权重和随机种子完整。
- 指标可从保存的预测或日志重新计算，且与论文设置、数据划分和评价协议逐项对照。
- 报告效果之外的成本和失败状态，不能用删掉失败运行的方式提高平均结果。

低算力和时间不足 预算不足时先跑官方推理、单 batch 过拟合和 1% 子集；只有短跑能复现趋势，才申请云端 GPU 或扩大训练。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“真正的复现测试是让不了解你代码的人运行。对方的问题优先反映文档和接口不足，不应归咎于对方。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 请同学或使用干净环境执行核心命令，记录失败点，修正文档后冻结版本。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：四至六页复现报告和发布版本。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“他人复现与定稿”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“四至六页复现报告和发布版本”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“他人复现与定稿”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“四至六页复现报告和发布版本”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：真正的复现测试是让不了解你代码的人运行。对方的问题优先反映文档和接口不足，不应归咎于对方。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“四至六页复现报告和发布版本”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：README 只有安装命令，没有预期输出。；报告结果无法追溯到具体配置。；清理仓库时误删失败日志和负结果。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 112 第 16 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：确定报告主线、重写方法与形状、写实验设置与结果、写错误分析与限制、整理 README 和入口、他人复现与定稿。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“四至六页复现报告和发布版本”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。

- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- □ 能独立解释本周目标：把前五周成果整理成陌生读者可检查、可运行、可评价的复现报告和代码仓库。
- □ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- □ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- □ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 17 周 方向扫描与文献矩阵

本周目标 围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。

首选材料 arXiv、Semantic Scholar、Google Scholar 或学校数据库，目标会议近两年论文集。

本周小白讲义

文献扫描从明确方向边界开始。任务、模态、场景和约束共同定义范围；检索式同时包含同义词、方法词和风险词。

快速筛选用于排除不相关论文，精读用于建立可比较证据。文献矩阵统一记录问题、数据、方法、基线、指标、成本、限制和可复现性。

研究空白不是没人用过某个模块，而是一个重要问题在现有证据下仍未被充分回答，并且你能设计实验检验。

本周知识地图

天	主题	核心知识	产出
Day 113	锁定研究方向	方向应由任务、模态、场景和约束共同定义，例如低资源中文文档视觉问答，而不是笼统的多模态。	方向边界说明
Day 114	设计检索式	高质量检索式包含任务词、方法词、风险词和同义词。引用链可以从一篇核心论文向前找基础、向后找改进。	检索式和论文候选库
Day 115	快速筛选二十篇	第一遍只判断是否相关、是否有方法贡献、数据和实验是否可读。不要在不相关论文上花一小时。	二十篇初筛记录
Day 116	精读五篇核心论文	精读要追踪问题、假设、方法、证据和限制。重点看最接近你计划的工作，而不是只看引用最多的工作。	五张论文阅读卡
Day 117	建立文献矩阵	矩阵让不同论文在同一维度比较，能显示哪些数据、指标和设置反复出现，哪些结论缺少直接证据。	完整文献矩阵

天	主题	核心知识	产出
Day 118	综合研究空白	研究空白必须同时重要、未充分解决、可验证且在资源内。单纯换模型或换数据集通常不足以构成问题。	三个候选研究问题
Day 119	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
columns = ['paper', 'problem', 'data', 'method', 'baseline',  
           'metric', 'cost', 'limitation', 'idea']
```

本周常见错误

- 只用宽泛关键词，得到无关论文。
- 把没有做过视为有价值空白。
- 只记论文摘要，没有记录数据、基线和限制。

Day 113 锁定研究方向

今天要懂 方向应由任务、模态、场景和约束共同定义，例如低资源中文文档视觉问答，而不是笼统的多模态。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“锁定研究方向”，合格解释必须包含这一因果关系：方向应由任务、模态、场景和约束共同定义，例如低资源中文文档视觉问答，而不是笼统的多模态。。然后把它与本周目标连接起来：围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：写下你的专业资源、可获得数据、可用算力和兴趣，形成一个主方向与明确排除项。。检查“方向边界说明”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“方向边界说明”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 arXiv、Semantic Scholar、Google Scholar 或学校数据库，目标会议近两年论文集。

今天先准备 建立 week17/day113 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：写下你的专业资源、可获得数据、可用算力和兴趣，形成一个主方向与明确排除项。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“方向边界说明”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“方向应由任务、模态、场景和约束共同定义，例如低资源中文文档视觉问答，而不是笼统的多模态。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 写下你的专业资源、可获得数据、可用算力和兴趣，形成一个主方向与明确排除项。
- 4 一百三十至一百六十分** 整理并保存今日产出：方向边界说明。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“锁定研究方向”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“方向边界说明”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“锁定研究方向”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“方向边界说明”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：方向应由任务、模态、场景和约束共同定义，例如低资源中文文档视觉问答，而不是笼统的多模态。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“方向边界说明”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只用宽泛关键词，得到无关论文。；把没有做过视为有价值空白。；只记论文摘要，没有记录数据、基线和限制。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 114 设计检索式

今天要懂 高质量检索式包含任务词、方法词、风险词和同义词。引用链可以从一篇核心论文向前找基础、向后找改进。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“设计检索式”，合格解释必须包含这一因果关系：高质量检索式包含任务词、方法词、风险

词和同义词。引用链可以从一篇核心论文向前找基础、向后找改进。。然后把它与本周目标连接起来：围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：建立中英文关键词表，在两个数据库检索并保存查询日期与筛选条件。。检查“检索式和论文候选库”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“检索式和论文候选库”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 arXiv、Semantic Scholar、Google Scholar 或学校数据库，目标会议近两年论文集。

今天先准备 建立 week17/day114 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：建立中英文关键词表，在两个数据库检索并保存查询日期与筛选条件。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“检索式和论文候选库”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“高质量检索式包含任务词、方法词、风险词和同义词。引用链可以从一篇核心论文向前找基础、向后找改进。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 建立中英文关键词表，在两个数据库检索并保存查询日期与筛选条件。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：检索式和论文候选库。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“设计检索式”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“检索式和论文候选库”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“设计检索式”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“检索式和论文候选库”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：高质量检索式包含任务词、方法词、风险词和同义词。引用链可以从一篇核心论文向前找基础、向后找改进。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“检索式和论文候选库”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只用宽泛关键词，得到无关论文。；把没有做过视为有价值空白。；只记论文摘要，没有记录数据、基线和限制。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 115 快速筛选二十篇

今天要懂 第一遍只判断是否相关、是否有方法贡献、数据和实验是否可读。不要在不相关论文上花一小时。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“快速筛选二十篇”，合格解释必须包含这一因果关系：第一遍只判断是否相关、是否有方法贡献、数据和实验是否可读。不要在不相关论文上花一小时。。然后把它与本周目标连接起来：围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：每篇用十分钟看标题、摘要、图一、主结果和结论，记录保留或排除理由。。检查“二十篇初筛记录”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“二十篇初筛记录”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 arXiv、Semantic Scholar、Google Scholar 或学校数据库，目标会议近两年论文集。

今天先准备 建立 week17/day115 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。

阶段	今天具体做什么
完成低成本检验	优先执行：每篇用十分钟看标题、摘要、图一、主结果和结论，记录保留或排除理由。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“二十篇初筛记录”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“第一遍只判断是否相关、是否有方法贡献、数据和实验是否可读。不要在不相关论文上花一小时。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 每篇用十分钟看标题、摘要、图一、主结果和结论，记录保留或排除理由。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：二十篇初筛记录。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“快速筛选二十篇”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“二十篇初筛记录”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“快速筛选二十篇”：第一句给定义，第二句说输入输出，第三句说一个失败条件。

- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“二十篇初筛记录”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：第一遍只判断是否相关、是否有方法贡献、数据和实验是否可读。不要在不相关论文上花一小时。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“二十篇初筛记录”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只用宽泛关键词，得到无关论文。；把没有做过视为有价值空白。；只记论文摘要，没有记录数据、基线和限制。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 116 精读五篇核心论文

今天要懂 精读要追踪问题、假设、方法、证据和限制。重点看最接近你计划的工作，而不是只看引用最多的工作。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“精读五篇核心论文”，合格解释必须包含这一因果关系：精读要追踪问题、假设、方法、证据和限制。重点看最接近你计划的工作，而不是只看引用最多的工作。。然后把它与本周目标连接起来：围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：对五篇论文完成四遍阅读，画方法图，核对表格，并定位代码入口。。检查“五张论文阅读卡”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“五张论文阅读卡”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 arXiv、Semantic Scholar、Google Scholar 或学校数据库，目标会议近两年论文集。

今天先准备 建立 week17/day116 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：对五篇论文完成四遍阅读，画方法图，核对表格，并定位代码入口。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“五张论文阅读卡”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“精读要追踪问题、假设、方法、证据和限制。重点看最接近你计划的工作，而不是只看引用最多的工作。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 对五篇论文完成四遍阅读，画方法图，核对表格，并定位代码入口。

- 4 一百三十至一百六十分钟** 整理并保存今日产出：五张论文阅读卡。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“精读五篇核心论文”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“五张论文阅读卡”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“精读五篇核心论文”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“五张论文阅读卡”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：精读要追踪问题、假设、方法、证据和限制。重点看最接近你计划的工作，而不是只看引用最多的工作。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“五张论文阅读卡”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只用宽泛关键词，得到无关论文。；把没有做过视为有价值空白。；只记论文摘要，没有记录数据、基线和限制。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 117 建立文献矩阵

今天要懂 矩阵让不同论文在同一维度比较，能显示哪些数据、指标和设置反复出现，哪些结论缺少直接证据。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“建立文献矩阵”，合格解释必须包含这一因果关系：矩阵让不同论文在同一维度比较，能显示哪些数据、指标和设置反复出现，哪些结论缺少直接证据。。然后把它与本周目标连接起来：围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：填写问题、数据、方法、基线、指标、成本、局限和可复现性，统一术语。。检查“完整文献矩阵”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“完整文献矩阵”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 arXiv、Semantic Scholar、Google Scholar 或学校数据库，目标会议近两年论文集。

今天先准备 建立 week17/day117 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：填写问题、数据、方法、基线、指标、成本、局限和可复现性，统一术语。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“完整文献矩阵”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“矩阵让不同论文在同一维度比较，能显示哪些数据、指标和设置反复出现，哪些结论缺少直接证据。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 填写问题、数据、方法、基线、指标、成本、局限和可复现性，统一术语。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：完整文献矩阵。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“建立文献矩阵”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“完整文献矩阵”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“建立文献矩阵”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“完整文献矩阵”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：矩阵让不同论文在同一维度比较，能显示哪些数据、指标和设置反复出现，哪些结论缺少直接证据。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“完整文献矩阵”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只用宽泛关键词，得到无关论文。；把没有做过视为有价值空白。；只记论文摘要，没有记录数据、基线和限制。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 118 综合研究空白

今天要懂 研究空白必须同时重要、未充分解决、可验证且在资源内。单纯换模型或换数据集通常不足以构成问题。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“综合研究空白”，合格解释必须包含这一因果关系：研究空白必须同时重要、未充分解决、可验证且在资源内。单纯换模型或换数据集通常不足以构成问题。。然后把它与本周目标连接起来：围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：从矩阵提出三个候选空白，为每个写证据、最小实验、风险和预期负结果。。检查“三个候选研究问题”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“三个候选研究问题”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 arXiv、Semantic Scholar、Google Scholar 或学校数据库，目标会议近两年论文集。

今天先准备 建立 week17/day118 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：从矩阵提出三个候选空白，为每个写证据、最小实验、风险和预期负结果。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“三个候选研究问题”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“研究空白必须同时重要、未充分解决、可验证且在资源内。单纯换模型或换数据集通常不足以构成问题。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 从矩阵提出三个候选空白，为每个写证据、最小实验、风险和预期负结果。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：三个候选研究问题。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“综合研究空白”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“三个候选研究问题”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“综合研究空白”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“三个候选研究问题”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：研究空白必须同时重要、未充分解决、可验证且在资源内。单纯换模型或换数据集通常不足以构成问题。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“三个候选研究问题”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：只用宽泛关键词，得到无关论文。；把没有做过视为有价值空白。；只记论文摘要，没有记录数据、基线和限制。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 119 第 17 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：锁定研究方向、设计检索式、快速筛选二十篇、精读五篇核心论文、建立文献矩阵、综合研究空白。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“三个候选研究问题”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- □ 能独立解释本周目标：围绕一个具体能力或应用建立文献地图，从重复工作中识别可验证的空白。
- □ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- □ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- □ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再执行本页验收。

第 18 周 研究问题与提案

本周目标 把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。

首选材料 目标领域近作，研究设计基础，数据与模型许可证，目标会议伦理指南。

本周小白讲义

可研究的问题必须可被证伪。问题句中写清场景、输入、唯一改动、比较对象、主指标和资源约束，避免使用全面提升等无法检验的词。

假设写出期望方向，也写出零假设和失败条件。变量表明确自变量、因变量和控制变量，评价协议在看到方法结果前冻结。

一页提案用于做可行性决策。若导师无法从一页内容判断价值、证据、预算和风险，说明问题仍然过宽。

本周知识地图

天	主题	核心知识	产出
Day 120	写可测研究问题	问题要包含场景、输入、改动、比较对象和结果变量。读者应能仅凭一句话判断需要什么实验回答。	三个规范研究问题
Day 121	写可证伪假设	假设必须允许失败，例如方法 M 在相同视觉 token 预算下将 OCR 问答准确率提高，同时延迟增加不超过某范围。	假设与反证条件表
Day 122	定义变量和基线	明确自变量、因变量和控制变量可防止结果解释漂移。强基线包括简单可靠方法和最接近工作。	变量与基线设计
Day 123	选择数据和评价协议	数据应覆盖目标场景且许可明确。开发集用于选择，测试集用于最终报告；评价脚本在方法实验前冻结。	冻结评价协议
Day 124	分析风险与伦理	风险包括算力不足、标签噪声、指标失真、隐私、群体偏差和潜在误用。预先计划缓解措施能减少后期被动修改。	风险与伦理清单

天	主题	核心知识	产出
Day 125	完成一页提案	一页提案应让导师快速判断问题价值、证据基础、方法差异和可行性。它不是长综述，而是执行合同。	一页研究提案
Day 126	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
proposal = {
  'question': '在条件 C 下 方法 M 是否比基线 B 改善指标 Y',
  'hypothesis': '', 'main_metric': '', 'controls': [],
  'risks': [], 'stop_condition': ''
}
```

本周常见错误

- 研究问题使用提升能力等不可测措辞。
- 假设只写会更好，没有比较对象和指标。
- 低估数据清洗、评估和失败重跑时间。

Day 120 写可测研究问题

今天要懂 问题要包含场景、输入、改动、比较对象和结果变量。读者应能仅凭一句话判断需要什么实验回答。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“写可测研究问题”，合格解释必须包含这一因果关系：问题要包含场景、输入、改动、比较对象和结果变量。读者应能仅凭一句话判断需要什么实验回答。。然后把它与本周目标连接起来：把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：把三个候选各改写三次，删除提高性能等模糊词，补上固定条件和主指标。。检查“三个规范研究问题”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“三个规范研究问题”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标领域近作，研究设计基础，数据与模型许可证，目标会议伦理指南。

今天先准备 建立 week18/day120 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：把三个候选各改写三次，删除提高性能等模糊词，补上固定条件和主指标。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“三个规范研究问题”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“问题要包含场景、输入、改动、比较对象和结果变量。读者应能仅凭一句话判断需要什么实验回答。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 把三个候选各改写三次，删除提高性能等模糊词，补上固定条件和主指标。

4 一百三十至一百六十分钟 整理并保存今日产出：三个规范研究问题。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“写可测研究问题”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“三个规范研究问题”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“写可测研究问题”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“三个规范研究问题”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：问题要包含场景、输入、改动、比较对象和结果变量。读者应能仅凭一句话判断需要什么实验回答。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“三个规范研究问题”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：研究问题使用提升能力等不可测措辞。；假设只写会更好，没有比较对象和指标。；低估数据清洗、评估和失败重跑时间。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 121 写可证伪假设

今天要懂 假设必须允许失败，例如方法 M 在相同视觉 token 预算下将 OCR 问答准确率提高，同时延迟增加不超过某范围。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“写可证伪假设”，合格解释必须包含这一因果关系：假设必须允许失败，例如方法 M 在相同视觉 token 预算下将 OCR 问答准确率提高，同时延迟增加不超过某范围。。然后把它与本周目标连接起来：把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：为每个问题写方向性假设、零假设和可能出现的反结果。。检查“假设与反证条件表”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“假设与反证条件表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标领域近作，研究设计基础，数据与模型许可证，目标会议伦理指南。

今天先准备 建立 week18/day121 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：为每个问题写方向性假设、零假设和可能出现的反结果。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“假设与反证条件表”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“假设必须允许失败，例如方法 M 在相同视觉 token 预算下将 OCR 问答准确率提高，同时延迟增加不超过某范围。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 为每个问题写方向性假设、零假设和可能出现的反结果。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：假设与反证条件表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“写可证伪假设”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“假设与反证条件表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“写可证伪假设”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“假设与反证条件表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：假设必须允许失败，例如方法 M 在相同视觉 token 预算下将 OCR 问答准确率提高，同时延迟增加不超过某范围。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“假设与反证条件表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：研究问题使用提升能力等不可测措辞。；假设只写会更好，没有比较对象和指标。；低估数据清洗、评估和失败重跑时间。。连续四十分钟没有进展时，缩小为一个样本或一个batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 122 定义变量和基线

今天要懂 明确自变量、因变量和控制变量可防止结果解释漂移。强基线包括简单可靠方法和最接近工作。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“定义变量和基线”，合格解释必须包含这一因果关系：明确自变量、因变量和控制变量可防止结果解释漂移。强基线包括简单可靠方法和最接近工作。。然后把它与本周目标连接起来：把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：列出新方法唯一变化、主副指标、所有控制项和至少两个基线。。检查“变量与基线设计”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“变量与基线设计”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标领域近作，研究设计基础，数据与模型许可证，目标会议伦理指南。

今天先准备 建立 week18/day122 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：列出新方法唯一变化、主副指标、所有控制项和至少两个基线。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“变量与基线设计”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“明确自变量、因变量和控制变量可防止结果解释漂
移。强基线包括简单可靠方法和最接近工作。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 列出新方法唯一变化、主副指标、所有控制项和至少两个基线。
- 4 一百三十至一百六十分** 整理并保存今日产出：变量与基线设计。文件名写明 day 编号，不覆盖昨天
结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一
步。

今日验收

- □ 不看资料，用不超过两分钟解释“定义变量和基线”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。

- □ 打开并检查“变量与基线设计”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“定义变量和基线”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“变量与基线设计”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：明确自变量、因变量和控制变量可防止结果解释漂移。强基线包括简单可靠方法和最接近工作。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“变量与基线设计”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：研究问题使用提升能力等不可测措辞。；假设只写会更好，没有比较对象和指标。；低估数据清洗、评估和失败重跑时间。。连续四十分钟没有进展时，缩小为一个样本或一个batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 123 选择数据和评价协议

今天要懂 数据应覆盖目标场景且许可明确。开发集用于选择，测试集用于最终报告；评价脚本在方法实验前冻结。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“选择数据和评价协议”，合格解释必须包含这一因果关系：数据应覆盖目标场景且许可明确。开发集用于选择，测试集用于最终报告；评价脚本在方法实验前冻结。。然后把它与本周目标连接起来：把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：确定数据版本、样本数、划分、污染检查、主指标、副指标和人工评审规则。。检查“冻结评价协议”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“冻结评价协议”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标领域近作，研究设计基础，数据与模型许可证，目标会议伦理指南。

今天先准备 建立 week18/day123 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：确定数据版本、样本数、划分、污染检查、主指标、副指标和人工评审规则。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“冻结评价协议”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要一次不稳定运行冒充科研证据。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“数据应覆盖目标场景且许可明确。开发集用于选择，测试集用于最终报告；评价脚本在方法实验前冻结。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 确定数据版本、样本数、划分、污染检查、主指标、副指标和人工评审规则。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：冻结评价协议。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“选择数据和评价协议”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“冻结评价协议”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“选择数据和评价协议”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“冻结评价协议”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：数据应覆盖目标场景且许可明确。开发集用于选择，测试集用于最终报告；评价脚本在方法实验前冻结。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“冻结评价协议”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：研究问题使用提升能力等不可测措辞。；假设只写会更好，没有比较对象和指标。；低估数据清洗、评估和失败重跑时间。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 124 分析风险与伦理

今天要懂 风险包括算力不足、标签噪声、指标失真、隐私、群体偏差和潜在误用。预先计划缓解措施能减少后期被动修改。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“分析风险与伦理”，合格解释必须包含这一因果关系：风险包括算力不足、标签噪声、指标失真、隐私、群体偏差和潜在误用。预先计划缓解措施能减少后期被动修改。。然后把它与本周目标连接起来：把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：建立风险表，给出概率、影响、预警信号和应对；检查数据和模型许可证。。检查“风险与伦理清单”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“风险与伦理清单”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标领域近作，研究设计基础，数据与模型许可证，目标会议伦理指南。

今天先准备 建立 week18/day124 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：建立风险表，给出概率、影响、预警信号和应对；检查数据和模型许可证。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。

阶段	今天具体做什么
更新研究记录	把证据、反例、结论边界和下一步写入“风险与伦理清单”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“风险包括算力不足、标签噪声、指标失真、隐私、群体偏差和潜在误用。预先计划缓解措施能减少后期被动修改。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 建立风险表，给出概率、影响、预警信号和应对；检查数据和模型许可证。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：风险与伦理清单。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“分析风险与伦理”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“风险与伦理清单”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“分析风险与伦理”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“风险与伦理清单”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：风险包括算力不足、标签噪声、指标失真、隐私、群体偏差和潜在误用。预先计划缓解措施能减少后期被动修改。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“风险与伦理清单”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：研究问题使用提升能力等不可测措辞。；假设只写会更好，没有比较对象和指标。；低估数据清洗、评估和失败重跑时间。。连续四十分钟没有进展时，缩小为一个样本或一个batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 125 完成一页提案

今天要懂 一页提案应让导师快速判断问题价值、证据基础、方法差异和可行性。它不是长综述，而是执行合同。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“完成一页提案”，合格解释必须包含这一因果关系：一页提案应让导师快速判断问题价值、证据基础、方法差异和可行性。它不是长综述，而是执行合同。。然后把它与本周目标连接起来：把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：写背景两段、问题一句、假设、方法草图、实验矩阵、预算、风险和四周里程碑。。检查“一页研究提案”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“一页研究提案”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标领域近作，研究设计基础，数据与模型许可证，目标会议伦理指南。

今天先准备 建立 week18/day125 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：写背景两段、问题一句、假设、方法草图、实验矩阵、预算、风险和四周里程碑。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“一页研究提案”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“一页提案应让导师快速判断问题价值、证据基础、方法差异和可行性。它不是长综述，而是执行合同。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 写背景两段、问题一句、假设、方法草图、实验矩阵、预算、风险和四周里程碑。
- 4 一百三十至一百六十分** 整理并保存今日产出：一页研究提案。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“完成一页提案”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“一页研究提案”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“完成一页提案”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“一页研究提案”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：一页提案应让导师快速判断问题价值、证据基础、方法差异和可行性。它不是长综述，而是执行合同。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“一页研究提案”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：研究问题使用提升能力等不可测措辞。；假设只写会更好，没有比较对象和指标。；低估数据清洗、评估和失败重跑时间。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 126 第 18 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：写可测研究问题、写可证伪假设、定义变量和基线、选择数据和评价协议、分析风险与伦理、完成一页提案。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“一页研究提案”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：把候选方向写成可证伪假设、清晰变量和在预算内可以完成的一页研究提案。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 19 周 建立强基线

本周目标 在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。

首选材料 数据集官方说明、最接近工作代码、评价脚本和实验跟踪规范。

本周小白讲义

强基线决定新方法的可信度。至少包含一个简单基线和最接近公开方法，并尽量统一数据、预算、指标和提示。无法统一的差异必须单独列出。

数据泄漏可能来自重复样本、文件名、元数据、近重复图像或测试信息进入提示。高得异常的结果首先做泄漏审计。

冻结基线意味着后续新方法通过清晰开关接入，关闭开关时应恢复同样预测和指标。

本周知识地图

天	主题	核心知识	产出
Day 127	冻结实验协议	协议先于方法结果冻结，包含数据、预处理、预算、指标和停止规则。后续若修改必须记录原因并重跑所有可比方法。	实验协议第一版
Day 128	实现简单基线	简单基线提供任务下限并暴露数据问题，例如多数类、线性分类器或纯文本模型。复杂方法胜过它不等于创新成立。	简单基线结果
Day 129	复现强基线	强基线应是公开、相关且在相同条件可运行的方法。无法完全相同时要报告差异，不能把非公平结果放在同一列比较。	强基线结果与差异说明
Day 130	检查数据泄漏	泄漏可能来自重复图、近重复文本、文件名、元数据或预训练污染。它会让结果看似提高却无法泛化。	数据泄漏审计
Day 131	测量基线成本	科研比较除了效果还需成本。新方法若提高很小却显著增加视觉 token、显存和延迟，价值需要重新判断。	基线效果成本表

天	主题	核心知识	产出
Day 132	冻结基线版本	基线冻结后，新方法只通过配置开关接入。保存预测可让后续做配对分析和错误比较。	可追溯基线结果包
Day 133	复盘验收	闭卷解释 独立重跑 整理失败补齐记录	周总结和通过记录

本周代码或记录模板

```
split_ids = {k: set(load_ids(k)) for k in ['train', 'val', 'test']}
assert split_ids['train'].isdisjoint(split_ids['test'])
assert split_ids['val'].isdisjoint(split_ids['test'])
```

本周常见错误

- 为了让新方法胜出而弱化基线。
- 不同方法使用不同数据增强或训练步数。
- 把测试集用于选择超参数。

Day 127 冻结实验协议

今天要懂 协议先于方法结果冻结，包含数据、预处理、预算、指标和停止规则。后续若修改必须记录原因并重跑所有可比方法。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“冻结实验协议”，合格解释必须包含这一因果关系：协议先于方法结果冻结，包含数据、预处理、预算、指标和停止规则。后续若修改必须记录原因并重跑所有可比方法。。然后把它与本周目标连接起来：在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：把提案转成 config 和 protocol.md，给配置计算哈希或版本号。。检查“实验协议第一版”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“实验协议第一版”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 数据集官方说明、最接近工作代码、评价脚本和实验跟踪规范。

今天先准备 建立 week19/day127 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：把提案转成 config 和 protocol.md，给配置计算哈希或版本号。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“实验协议第一版”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“协议先于方法结果冻结，包含数据、预处理、预算、指标和停止规则。后续若修改必须记录原因并重跑所有可比方法。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 把提案转成 config 和 protocol.md，给配置计算哈希或版本号。
- 4 一百三十至一百六十分** 整理并保存今日产出：实验协议第一版。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“冻结实验协议”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“实验协议第一版”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“冻结实验协议”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“实验协议第一版”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：协议先于方法结果冻结，包含数据、预处理、预算、指标和停止规则。后续若修改必须记录原因并重跑所有可比方法。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“实验协议第一版”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：为了让新方法胜出而弱化基线。；不同方法使用不同数据增强或训练步数。；把测试集用于选择超参数。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 128 实现简单基线

今天要懂 简单基线提供任务下限并暴露数据问题，例如多数类、线性分类器或纯文本模型。复杂方法胜过它不等于创新成立。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“实现简单基线”，合格解释必须包含这一因果关系：简单基线提供任务下限并暴露数据问题，例如多数类、线性分类器或纯文本模型。复杂方法胜过它不等于创新成立。。然后把它与本周目标连接起来：在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：实现一个无需新组件的简单基线，检查预测分布、类别不平衡和输入泄漏。。检查“简单基线结果”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“简单基线结果”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 数据集官方说明、最接近工作代码、评价脚本和实验跟踪规范。

今天先准备 建立 week19/day128 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：实现一个无需新组件的简单基线，检查预测分布、类别不平衡和输入泄漏。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“简单基线结果”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“简单基线提供任务下限并暴露数据问题，例如多数类、线性分类器或纯文本模型。复杂方法胜过它不等于创新成立。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 实现一个无需新组件的简单基线，检查预测分布、类别不平衡和输入泄漏。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：简单基线结果。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“实现简单基线”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“简单基线结果”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“实现简单基线”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“简单基线结果”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：简单基线提供任务下限并暴露数据问题，例如多数类、线性分类器或纯文本模型。复杂方法胜过它不等于创新成立。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 第四题应让别人找到输入、环境、运行命令、输出目录和“简单基线结果”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：为了让新方法胜出而弱化基线。；不同方法使用不同数据增强或训练步数。；把测试集用于选择超参数。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 129 复现强基线

今天要懂 强基线应是公开、相关且在相同条件可运行的方法。无法完全相同时要报告差异，不能把不公平结果放在同一列比较。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“复现强基线”，合格解释必须包含这一因果关系：强基线应是公开、相关且在相同条件可运行的方法。无法完全相同时要报告差异，不能把不公平结果放在同一列比较。。然后把它与本周目标连接起来：在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：运行最接近工作，统一数据与指标，记录不能统一的模型规模或预训练差异。。检查“强基线结果与差异说明”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“强基线结果与差异说明”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 数据集官方说明、最接近工作代码、评价脚本和实验跟踪规范。

今天先准备 建立 week19/day129 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。

阶段	今天具体做什么
完成低成本检验	优先执行：运行最接近工作，统一数据与指标，记录不能统一的模型规模或预训练差异。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“强基线结果与差异说明”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“强基线应是公开、相关且在相同条件可运行的方法。
无法完全相同时要报告差异，不能把非公平结果放在同一列比较。”写不少于五条自己的话，并画出
输入、处理、输出。
- 3 六十至一百三十分钟** 运行最接近工作，统一数据与指标，记录不能统一的模型规模或预训练差异。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：强基线结果与差异说明。文件名写明 day 编号，不覆
盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一
步。

今日验收

- □ 不看资料，用不超过两分钟解释“复现强基线”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“强基线结果与差异说明”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“复现强基线”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“强基线结果与差异说明”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：强基线应是公开、相关且在相同条件可运行的方法。无法完全相同时要报告差异，不能把不公平结果放在同一列比较。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“强基线结果与差异说明”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：为了让新方法胜出而弱化基线。；不同方法使用不同数据增强或训练步数。；把测试集用于选择超参数。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 130 检查数据泄漏

今天要懂 泄漏可能来自重复图、近重复文本、文件名、元数据或预训练污染。它会让结果看似提高却无法泛化。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“检查数据泄漏”，合格解释必须包含这一因果关系：泄漏可能来自重复图、近重复文本、文件名、元数据或预训练污染。它会让结果看似提高却无法泛化。。然后把它与本周目标连接起来：在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：检查划分 ID 交集、文本重复、图像感知哈希和标签字段，抽查高置信异常样本。。检查“数据泄漏审计”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“数据泄漏审计”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 数据集官方说明、最接近工作代码、评价脚本和实验跟踪规范。

今天先准备 建立 week19/day130 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：检查划分 ID 交集、文本重复、图像感知哈希和标签字段，抽查高置信异常样本。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“数据泄漏审计”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要一次不稳定运行冒充科研证据。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“泄漏可能来自重复图、近重复文本、文件名、元数据或预训练污染。它会让结果看似提高却无法泛化。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 检查划分 ID 交集、文本重复、图像感知哈希和标签字段，抽查高置信异常样本。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：数据泄漏审计。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“检查数据泄漏”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“数据泄漏审计”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“检查数据泄漏”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“数据泄漏审计”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：泄漏可能来自重复图、近重复文本、文件名、元数据或预训练污染。它会让结果看似提高却无法泛化。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“数据泄漏审计”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：为了让新方法胜出而弱化基线。；不同方法使用不同数据增强或训练步数。；把测试集用于选择超参数。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 131 测量基线成本

今天要懂 科研比较除了效果还需成本。新方法若提高很小却显著增加视觉 token、显存和延迟，价值需要重新判断。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“测量基线成本”，合格解释必须包含这一因果关系：科研比较除了效果还需成本。新方法若提高很小却显著增加视觉 token、显存和延迟，价值需要重新判断。。然后把它与本周目标连接起来：在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：按统一脚本测参数量、可训练参数、显存、训练时间、延迟和吞吐。。检查“基线效果成本表”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“基线效果成本表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 数据集官方说明、最接近工作代码、评价脚本和实验跟踪规范。

今天先准备 建立 week19/day131 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。

阶段	今天具体做什么
完成低成本检验	优先执行：按统一脚本测参数量、可训练参数、显存、训练时间、延迟和吞吐。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“基线效果成本表”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“科研比较除了效果还需成本。新方法若提高很小却显著增加视觉 token、显存和延迟，价值需要重新判断。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 按统一脚本测参数量、可训练参数、显存、训练时间、延迟和吞吐。
- 4 一百三十至一百六十分** 整理并保存今日产出：基线效果成本表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“测量基线成本”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“基线效果成本表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“测量基线成本”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“基线效果成本表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：科研比较除了效果还需成本。新方法若提高很小却显著增加视觉 token、显存和延迟，价值需要重新判断。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“基线效果成本表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：为了让新方法胜出而弱化基线。；不同方法使用不同数据增强或训练步数。；把测试集用于选择超参数。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 132 冻结基线版本

今天要懂 基线冻结后，新方法只通过配置开关接入。保存预测可让后续做配对分析和错误比较。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“冻结基线版本”，合格解释必须包含这一因果关系：基线冻结后，新方法只通过配置开关接入。保存预测可让后续做配对分析和错误比较。。然后把它与本周目标连接起来：在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：重跑主要基线，归档配置、指标、预测、日志和环境，标记只读发布版本。。检查“可追溯基线结果包”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“可追溯基线结果包”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 数据集官方说明、最接近工作代码、评价脚本和实验跟踪规范。

今天先准备 建立 week19/day132 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：重跑主要基线，归档配置、指标、预测、日志和环境，标记只读发布版本。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“可追溯基线结果包”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要一次不稳定运行冒充科研证据。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“基线冻结后，新方法只通过配置开关接入。保存预测可让后续做配对分析和错误比较。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 重跑主要基线，归档配置、指标、预测、日志和环境，标记只读发布版本。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：可追溯基线结果包。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“冻结基线版本”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“可追溯基线结果包”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“冻结基线版本”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“可追溯基线结果包”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：基线冻结后，新方法只通过配置开关接入。保存预测可让后续做配对分析和错误比较。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“可追溯基线结果包”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：为了让新方法胜出而弱化基线。；不同方法使用不同数据增强或训练步数。；把测试集用于选择超参数。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 133 第 19 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：冻结实验协议、实现简单基线、复现强基线、检查数据泄漏、测量基线成本、冻结基线版本。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“可追溯基线结果包”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- □ 能独立解释本周目标：在实现新方法前冻结数据和评价协议，建立简单基线与最接近工作的可靠结果。
- □ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- □ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- □ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 20 周 实现核心方法

本周目标 把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。

首选材料 项目代码规范，PyTorch 单元测试，最接近工作方法细节与接口。

本周小白讲义

核心方法先写成十行左右伪代码和数据流，再实现模块。接口要写明 shape、dtype、device、边界条件和可训练参数。

最小实现只验证核心假设。先检查输出有限、损失可计算、梯度到达新参数和参数确实更新，再做速度优化。

关闭新模块后的回归测试必须通过，否则主结果无法归因于新方法。小规模趋势失败时先检查假设与实现，不直接扩大算力。

本周知识地图

天	主题	核心知识	产出
Day 134	写伪代码和数据流	先用伪代码说明新方法读取什么、改变什么、输出什么。若无法用十行左右说清，方法通常还没有收敛。	方法伪代码与数据流
Day 135	定义接口与形状	清晰接口隔离新旧代码，减少无关改动。输入输出 dtype、device、shape 和异常条件都应明确。	接口说明和 shape 测试
Day 136	完成最小实现	最小实现只覆盖核心假设，不先优化速度或加入多个技巧。先确保前向结果有限、损失可算、梯度能到达新参数。	核心模块初版
Day 137	编写单元测试	单元测试验证局部性质，例如输出形状、掩码、边界值、确定性和关闭开关。它不能替代端到端实验，但能快速阻止回归。	通过的单元测试
Day 138	基线回归检查	关闭新方法时，数据、模型输出和指标应与冻结基线在允许误差内一致，否则无法归因新方法。	回归检查报告

天	主题	核心知识	产出
Day 139	小规模趋势实验	小实验只判断方向和稳定性，不用于最终结论。若趋势失败，应先检查假设和实现，不立即扩大算力。	趋势实验与是否继续的决定
Day 140	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
if config.use_new_method:
    features = new_module(features, evidence)
else:
    features = features
# 关闭开关时必须与基线一致
```

本周常见错误

- 一次加入多个新模块。
- 新方法关闭后仍改变数据或训练过程。
- 只在完整训练中调试，反馈周期过长。

Day 134 写伪代码和数据流

今天要懂 先用伪代码说明新方法读取什么、改变什么、输出什么。若无法用十行左右说清，方法通常还没有收敛。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“写伪代码和数据流”，合格解释必须包含这一因果关系：先用伪代码说明新方法读取什么、改变什么、输出什么。若无法用十行左右说清，方法通常还没有收敛。。然后把它与本周目标连接起来：把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：画新模块插入位置，写输入输出形状、可训练参数、计算复杂度和损失变化。。检查“方法伪代码与数据流”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“方法伪代码与数据流”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 项目代码规范，PyTorch 单元测试，最接近工作方法细节与接口。

今天先准备 建立 week20/day134 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：画新模块插入位置，写输入输出形状、可训练参数、计算复杂度和损失变化。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“方法伪代码与数据流”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“先用伪代码说明新方法读取什么、改变什么、输出什么。若无法用十行左右说清，方法通常还没有收敛。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 画新模块插入位置，写输入输出形状、可训练参数、计算复杂度和损失变化。

4 一百三十至一百六十分钟 整理并保存今日产出：方法伪代码与数据流。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“写伪代码和数据流”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“方法伪代码与数据流”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“写伪代码和数据流”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“方法伪代码与数据流”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：先用伪代码说明新方法读取什么、改变什么、输出什么。若无法用十行左右说清，方法通常还没有收敛。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“方法伪代码与数据流”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：一次加入多个新模块。；新方法关闭后仍改变数据或训练过程。；只在完整训练中调试，反馈周期过长。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 135 定义接口与形状

今天要懂 清晰接口隔离新旧代码，减少无关改动。输入输出 dtype、device、shape 和异常条件都应明确。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“定义接口与形状”，合格解释必须包含这一因果关系：清晰接口隔离新旧代码，减少无关改动。输入输出 dtype、device、shape 和异常条件都应明确。。然后把它与本周目标连接起来：把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：为新模块写函数签名和形状断言，用随机小张量验证正常与错误输入。。检查“接口说明和 shape 测试”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“接口说明和 shape 测试”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 项目代码规范，PyTorch 单元测试，最接近工作方法细节与接口。

今天先准备 建立 week20/day135 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：为新模块写函数签名和形状断言，用随机小张量验证正常与错误输入。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“接口说明和 shape 测试”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“清晰接口隔离新旧代码，减少无关改动。输入输出 dtype、device、shape 和异常条件都应明确。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 为新模块写函数签名和形状断言，用随机小张量验证正常与错误输入。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：接口说明和 shape 测试。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“定义接口与形状”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“接口说明和 shape 测试”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“定义接口与形状”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“接口说明和 shape 测试”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：清晰接口隔离新旧代码，减少无关改动。输入输出 dtype、device、shape 和异常条件都应明确。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“接口说明和 shape 测试”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：一次加入多个新模块。；新方法关闭后仍改变数据或训练过程。；只在完整训练中调试，反馈周期过长。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 136 完成最小实现

今天要懂 最小实现只覆盖核心假设，不先优化速度或加入多个技巧。先确保前向结果有限、损失可算、梯度能到达新参数。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“完成最小实现”，合格解释必须包含这一因果关系：最小实现只覆盖核心假设，不先优化速度或加入多个技巧。先确保前向结果有限、损失可算、梯度能到达新参数。。然后把它与本周目标连接起来：把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：实现模块，在玩具输入上检查 NaN、梯度和参数更新，保存最小运行脚本。。检查“核心模块初版”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“核心模块初版”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 项目代码规范，PyTorch 单元测试，最接近工作方法细节与接口。

今天先准备 建立 week20/day136 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
----	---------

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“最小实现只覆盖核心假设，不先优化速度或加入多个技巧。先确保前向结果有限、损失可算、梯度能到达新参数。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 实现模块，在玩具输入上检查 NaN、梯度和参数更新，保存最小运行脚本。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：核心模块初版。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“完成最小实现”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“核心模块初版”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“完成最小实现”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“核心模块初版”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：最小实现只覆盖核心假设，不先优化速度或加入多个技巧。先确保前向结果有限、损失可算、梯度能到达新参数。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“核心模块初版”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：一次加入多个新模块。；新方法关闭后仍改变数据或训练过程。；只在完整训练中调试，反馈周期过长。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 137 编写单元测试

今天要懂 单元测试验证局部性质，例如输出形状、掩码、边界值、确定性和关闭开关。它不能替代端到端实验，但能快速阻止回归。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“编写单元测试”，合格解释必须包含这一因果关系：单元测试验证局部性质，例如输出形状、掩码、边界值、确定性和关闭开关。它不能替代端到端实验，但能快速阻止回归。。然后把它与本周目标连接起来：把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：为关键函数写至少五个测试，包含空输入、极端长度或无检索证据等边界。。检查“通过的单元测试”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“通过的单元测试”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 项目代码规范，PyTorch 单元测试，最接近工作方法细节与接口。

今天先准备 建立 week20/day137 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：为关键函数写至少五个测试，包含空输入、极端长度或无检索证据等边界。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“通过的单元测试”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“单元测试验证局部性质，例如输出形状、掩码、边界值、确定性和关闭开关。它不能替代端到端实验，但能快速阻止回归。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 为关键函数写至少五个测试，包含空输入、极端长度或无检索证据等边界。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：通过的单元测试。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“编写单元测试”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“通过的单元测试”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“编写单元测试”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“通过的单元测试”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：单元测试验证局部性质，例如输出形状、掩码、边界值、确定性和关闭开关。它不能替代端到端实验，但能快速阻止回归。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“通过的单元测试”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：一次加入多个新模块。；新方法关闭后仍改变数据或训练过程。；只在完整训练中调试，反馈周期过长。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 138 基线回归检查

今天要懂 关闭新方法时，数据、模型输出和指标应与冻结基线在允许误差内一致，否则无法归因新方法。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“基线回归检查”，合格解释必须包含这一因果关系：关闭新方法时，数据、模型输出和指标

应与冻结基线在允许误差内一致，否则无法归因新方法。。然后把它与本周目标连接起来：把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：加载同一权重与输入，对比关闭开关前后的中间张量、预测和指标。。检查“回归检查报告”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“回归检查报告”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 项目代码规范，PyTorch 单元测试，最接近工作方法细节与接口。

今天先准备 建立 week20/day138 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：加载同一权重与输入，对比关闭开关前后的中间张量、预测和指标。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“回归检查报告”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“关闭新方法时，数据、模型输出和指标应与冻结基线
在允许误差内一致，否则无法归因新方法。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 加载同一权重与输入，对比关闭开关前后的中间张量、预测和指标。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：回归检查报告。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“基线回归检查”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“回归检查报告”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“基线回归检查”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“回归检查报告”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：关闭新方法时，数据、模型输出和指标应与冻结基线在允许误差内一致，否则无法归因新方法。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 第四题应让别人找到输入、环境、运行命令、输出目录和“回归检查报告”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：一次加入多个新模块。；新方法关闭后仍改变数据或训练过程。；只在完整训练中调试，反馈周期过长。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 139 小规模趋势实验

今天要懂 小实验只判断方向和稳定性，不用于最终结论。若趋势失败，应先检查假设和实现，不立即扩大算力。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“小规模趋势实验”，合格解释必须包含这一因果关系：小实验只判断方向和稳定性，不用于最终结论。若趋势失败，应先检查假设和实现，不立即扩大算力。。然后把它与本周目标连接起来：把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：在小子集比较基线与新方法，运行两个种子，分析差异和主要错误。。检查“趋势实验与是否继续的决定”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“趋势实验与是否继续的决定”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 项目代码规范，PyTorch 单元测试，最接近工作方法细节与接口。

今天先准备 建立 week20/day139 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。

阶段	今天具体做什么
完成低成本检验	优先执行：在小子集比较基线与新方法，运行两个种子，分析差异和主要错误。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“趋势实验与是否继续的决定”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“小实验只判断方向和稳定性，不用于最终结论。若趋势失败，应先检查假设和实现，不立即扩大算力。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 在小子集比较基线与新方法，运行两个种子，分析差异和主要错误。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：趋势实验与是否继续的决定。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“小规模趋势实验”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“趋势实验与是否继续的决定”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“小规模趋势实验”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“趋势实验与是否继续的决定”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：小实验只判断方向和稳定性，不用于最终结论。若趋势失败，应先检查假设和实现，不立即扩大算力。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“趋势实验与是否继续的决定”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：一次加入多个新模块。；新方法关闭后仍改变数据或训练过程。；只在完整训练中调试，反馈周期过长。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 140 第 20 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。

阶段	今天具体做什么
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：写伪代码和数据流、定义接口与形状、完成最小实现、编写单元测试、基线回归检查、小规模趋势实验。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“趋势实验与是否继续的决定”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：把论文想法缩成一个可关闭、可测试的最小改动，并证明关闭后回到原基线。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 21 周 完整实验与结果冻结

本周目标 按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。

首选材料 研究提案、冻结协议、实验配置、统计与评估脚本。

本周小白讲义

完整实验矩阵按证据价值排序：主结果和关键消融是必做，泛化、鲁棒性和效率用于限定价值，可选实验只有在核心证据完成后运行。

运行期间只修执行故障，不根据中途结果更改指标和停止规则。每个结果都要能回溯到配置、代码版本和预测文件。

结果冻结后，论文只引用该版本。若发现错误，生成新版本并列出所有受影响数字。

本周知识地图

天	主题	核心知识	产出
Day 141	排定实验矩阵	实验矩阵应区分必做、支持性和可选实验，并按证据价值与成本排序。主结果和关键消融优先。	带优先级的实验矩阵
Day 142	运行主结果	主结果比较你的方法与强基线，使用完全相同的协议。运行期间只处理执行故障，不根据中途结果改变评价标准。	主结果原始文件
Day 143	运行关键消融	关键消融对应论文贡献中的每个可分组件。若贡献无法被消融验证，应降低主张强度。	关键消融原始文件
Day 144	测试泛化与鲁棒性	泛化测试检查不同数据、类别或分布，鲁棒性测试检查噪声、缺失模态和提示扰动。测试应与论文目标相关。	泛化与鲁棒性结果
Day 145	汇总统计与成本	把效果、波动和成本一起汇总。对配对预测可做更有力的样本级比较，但统计方法要与数据条件相符。	自动汇总脚本和结果草表

天	主题	核心知识	产出
Day 146	冻结结果包	冻结意味着论文写作只引用这一批可追溯结果。若后续修复错误，要生成新版本并说明全部受影响数字。	results v1 冻结包
Day 147	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
run_id = f'{method}_{dataset}_seed{seed}'
save_json({
    'run_id': run_id, 'commit': commit, 'config': config,
    'metrics': metrics, 'status': status
}, f'outputs/{run_id}/manifest.json')
```

本周常见错误

- 主结果不理想就不断改测试协议。
- 只保存聚合数字，不保存预测。
- 实验失败后用新名称重跑，丢失失败记录。

Day 141 排定实验矩阵

今天要懂 实验矩阵应区分必做、支持性和可选实验，并按证据价值与成本排序。主结果和关键消融优先。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“排定实验矩阵”，合格解释必须包含这一因果关系：实验矩阵应区分必做、支持性和可选实验，并按证据价值与成本排序。主结果和关键消融优先。。然后把它与本周目标连接起来：按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：列出所有运行、依赖、预计 GPU 小时和停止条件，预留至少百分之二十失败预算。。检查“带优先级的实验矩阵”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“带优先级的实验矩阵”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 研究提案、冻结协议、实验配置、统计与评估脚本。

今天先准备 建立 week21/day141 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：列出所有运行、依赖、预计 GPU 小时和停止条件，预留至少百分之二十失败预算。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“带优先级的实验矩阵”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“实验矩阵应区分必做、支持性和可选实验，并按证据价值与成本排序。主结果和关键消融优先。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 列出所有运行、依赖、预计 GPU 小时和停止条件，预留至少百分之二十失败预算。

4 一百三十至一百六十分钟 整理并保存今日产出：带优先级的实验矩阵。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“排定实验矩阵”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“带优先级的实验矩阵”，其中包含日期、环境或配置、结果和一句结论。

自测题

1 不用资料，用三句话解释“排定实验矩阵”：第一句给定义，第二句说输入输出，第三句说一个失败条件。

2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。

3 假设“带优先级的实验矩阵”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。

4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

1 第一题至少应覆盖：实验矩阵应区分必做、支持性和可选实验，并按证据价值与成本排序。主结果和关键消融优先。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“带优先级的实验矩阵”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：主结果不理想就不断改测试协议。；只保存聚合数字，不保存预测。；实验失败后用新名称重跑，丢失失败记录。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 142 运行主结果

今天要懂 主结果比较你的方法与强基线，使用完全相同的协议。运行期间只处理执行故障，不根据中途结果改变评价标准。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“运行主结果”，合格解释必须包含这一因果关系：主结果比较你的方法与强基线，使用完全相同的协议。运行期间只处理执行故障，不根据中途结果改变评价标准。。然后把它与本周目标连接起来：按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：完成所有主结果种子，监控损失和资源，保存最佳与最终检查点以及预测。。检查“主结果原始文件”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“主结果原始文件”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 研究提案、冻结协议、实验配置、统计与评估脚本。

今天先准备 建立 week21/day142 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：完成所有主结果种子，监控损失和资源，保存最佳与最终检查点以及预测。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“主结果原始文件”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“主结果比较你的方法与强基线，使用完全相同的协议。运行期间只处理执行故障，不根据中途结果改变评价标准。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 完成所有主结果种子，监控损失和资源，保存最佳与最终检查点以及预测。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：主结果原始文件。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“运行主结果”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“主结果原始文件”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“运行主结果”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“主结果原始文件”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：主结果比较你的方法与强基线，使用完全相同的协议。运行期间只处理执行故障，不根据中途结果改变评价标准。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“主结果原始文件”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：主结果不理想就不断改测试协议。；只保存聚合数字，不保存预测。；实验失败后用新名称重跑，丢失失败记录。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 143 运行关键消融

今天要懂 关键消融对应论文贡献中的每个可分组件。若贡献无法被消融验证，应降低主张强度。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“运行关键消融”，合格解释必须包含这一因果关系：关键消融对应论文贡献中的每个可分组件。若贡献无法被消融验证，应降低主张强度。。然后把它与本周目标连接起来：按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：关闭、替换或改变核心组件，保持训练预算一致，完成预定种子。。检查“关键消融原始文件”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“关键消融原始文件”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 研究提案、冻结协议、实验配置、统计与评估脚本。

今天先准备 建立 week21/day143 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：关闭、替换或改变核心组件，保持训练预算一致，完成预定种子。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“关键消融原始文件”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“关键消融对应论文贡献中的每个可分组件。若贡献无法被消融验证，应降低主张强度。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 关闭、替换或改变核心组件，保持训练预算一致，完成预定种子。
- 4 一百三十至一百六十分** 整理并保存今日产出：关键消融原始文件。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“运行关键消融”，并说明它在本周任务中的作用。

- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“关键消融原始文件”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“运行关键消融”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“关键消融原始文件”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：关键消融对应论文贡献中的每个可分组件。若贡献无法被消融验证，应降低主张强度。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“关键消融原始文件”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：主结果不理想就不断改测试协议。；只保存聚合数字，不保存预测。；实验失败后用新名称重跑，丢失失败记录。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 144 测试泛化与鲁棒性

今天要懂 泛化测试检查不同数据、类别或分布，鲁棒性测试检查噪声、缺失模态和提示扰动。测试应与论文目标相关。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“测试泛化与鲁棒性”，合格解释必须包含这一因果关系：泛化测试检查不同数据、类别或分布，鲁棒性测试检查噪声、缺失模态和提示扰动。测试应与论文目标相关。。然后把它与本周目标连接起来：按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：选择一个分布外数据或两个合理扰动，比较性能下降比例而非只报绝对分数。。检查“泛化与鲁棒性结果”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“泛化与鲁棒性结果”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 研究提案、冻结协议、实验配置、统计与评估脚本。

今天先准备 建立 week21/day144 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：选择一个分布外数据或两个合理扰动，比较性能下降比例而非只报绝对分数。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“泛化与鲁棒性结果”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“泛化测试检查不同数据、类别或分布，鲁棒性测试检查噪声、缺失模态和提示扰动。测试应与论文目标相关。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 选择一个分布外数据或两个合理扰动，比较性能下降比例而非只报绝对分数。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：泛化与鲁棒性结果。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“测试泛化与鲁棒性”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“泛化与鲁棒性结果”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“测试泛化与鲁棒性”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“泛化与鲁棒性结果”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：泛化测试检查不同数据、类别或分布，鲁棒性测试检查噪声、缺失模态和提示扰动。测试应与论文目标相关。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 第四题应让别人找到输入、环境、运行命令、输出目录和“泛化与鲁棒性结果”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：主结果不理想就不断改测试协议。；只保存聚合数字，不保存预测。；实验失败后用新名称重跑，丢失失败记录。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 145 汇总统计与成本

今天要懂 把效果、波动和成本一起汇总。对配对预测可做更有力的样本级比较，但统计方法要与数据条件相符。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“汇总统计与成本”，合格解释必须包含这一因果关系：把效果、波动和成本一起汇总。对配对预测可做更有力的样本级比较，但统计方法要与数据条件相符。。然后把它与本周目标连接起来：按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：自动读取 manifest，生成均值、标准差、运行数、显存和时间表，检查缺失运行。。检查“自动汇总脚本和结果草表”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“自动汇总脚本和结果草表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 研究提案、冻结协议、实验配置、统计与评估脚本。

今天先准备 建立 week21/day145 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。

阶段	今天具体做什么
完成低成本检验	优先执行：自动读取 manifest，生成均值、标准差、运行数、显存和时间表，检查缺失运行。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“自动汇总脚本和结果草表”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“把效果、波动和成本一起汇总。对配对预测可做更有力的样本级比较，但统计方法要与数据条件相符。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 自动读取 manifest，生成均值、标准差、运行数、显存和时间表，检查缺失运行。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：自动汇总脚本和结果草表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“汇总统计与成本”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“自动汇总脚本和结果草表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“汇总统计与成本”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“自动汇总脚本和结果草表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：把效果、波动和成本一起汇总。对配对预测可做更有力的样本级比较，但统计方法要与数据条件相符。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“自动汇总脚本和结果草表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：主结果不理想就不断改测试协议。；只保存聚合数字，不保存预测。；实验失败后用新名称重跑，丢失失败记录。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 146 冻结结果包

今天要懂 冻结意味着论文写作只引用这一批可追溯结果。若后续修复错误，要生成新版本并说明全部受影响数字。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“冻结结果包”，合格解释必须包含这一因果关系：冻结意味着论文写作只引用这一批可追溯结果。若后续修复错误，要生成新版本并说明全部受影响数字。。然后把它与本周目标连接起来：按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：归档配置、代码 commit、环境、指标、预测和图表输入，写 checksum 与变更说明。。检查 “results v1 冻结包” 中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到 “results v1 冻结包” 。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 研究提案、冻结协议、实验配置、统计与评估脚本。

今天先准备 建立 week21/day146 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：归档配置、代码 commit、环境、指标、预测和图表输入，写 checksum 与变更说明。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入 “results v1 冻结包”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要一次不稳定运行冒充科研证据。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“冻结意味着论文写作只引用这一批可追溯结果。若后续修复错误，要生成新版本并说明全部受影响数字。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 归档配置、代码 commit、环境、指标、预测和图表输入，写 checksum 与变更说明。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：results v1 冻结包。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“冻结结果包”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“results v1 冻结包”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“冻结结果包”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“results v1 冻结包”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：冻结意味着论文写作只引用这一批可追溯结果。若后续修复错误，要生成新版本并说明全部受影响数字。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“results v1 冻结包”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：主结果不理想就不断改测试协议。；只保存聚合数字，不保存预测。；实验失败后用新名称重跑，丢失失败记录。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 147 第 21 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：排定实验矩阵、运行主结果、运行关键消融、测试泛化与鲁棒性、汇总统计与成本、冻结结果包。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“results v1 冻结包”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。

- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：按预先设计的矩阵完成主结果、消融、泛化、鲁棒性和效率实验，并冻结可追溯结果。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第 22 周 主张 证据与论文图表

本周目标 让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。

首选材料 目标会议图表规范，Matplotlib 可访问性指南，相关论文主结果和消融写法。

本周小白讲义

主张证据表把每句结论与表格、图、统计或错误分析连接起来。没有证据的主张删除，有反证的主张缩小范围。

主结果表回答研究问题，消融表回答组件作用，效率图回答代价，失败案例说明边界。图表由脚本从冻结结果生成，不手工抄数字。

误差条必须说明含义和运行数。统计显著不等于实际重要，还要结合效应大小、成本和应用场景解释。

本周知识地图

天	主题	核心知识	产出
Day 148	建立主张证据表	主张必须包含范围，例如在哪个数据、预算和模型上成立。每条主张要有支持证据、反证或限制。	主张证据矩阵
Day 149	制作主结果表	主结果表应优先回答研究问题，方法顺序和分组要清晰，指标注明越高或越低越好，最好值要结合误差解释。	主结果表
Day 150	制作消融与效率图	图表适合展示趋势、权衡和分布。颜色不能是唯一区分方式，轴和单位必须完整，图注要能独立理解。	两张论文级图
Day 151	误差范围与统计说明	误差条可能表示标准差、标准误差或置信区间，必须明确。样本数和随机种子数决定不确定性解释。	统计说明段落
Day 152	选择失败案例	失败案例要代表已统计的主要错误，而不是只挑戏剧性样本。输入、预测、真值和证据应同时展示。	定性结果图

天	主题	核心知识	产出
Day 153	收紧结论范围	实验只支持特定条件下的结论。若泛化或效率结果不支持原主张，应缩小范围或明确权衡。	冻结的论文主张清单
Day 154	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
claim_map = [  
    {'claim': '方法改善 OCR 问答', 'evidence': '表 1 与图 3',  
     'scope': '固定 token 预算的数据集 A', 'counterevidence': ''}  
]
```

本周常见错误

- 先写强结论，再挑支持它的实验。
- 图轴截断却未说明。
- 没有误差、样本数或评价方向。

Day 148 建立主张证据表

今天要懂 主张必须包含范围，例如在哪个数据、预算和模型上成立。每条主张要有支持证据、反证或限制。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“建立主张证据表”，合格解释必须包含这一因果关系：主张必须包含范围，例如在哪个数据、预算和模型上成立。每条主张要有支持证据、反证或限制。。然后把它与本周目标连接起来：让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：从提案和结果写出五至八条候选主张，链接到表格、图或错误分析。。检查“主张证据矩阵”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“主张证据矩阵”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议图表规范，Matplotlib 可访问性指南，相关论文主结果和消融写法。

今天先准备 建立 week22/day148 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：从提案和结果写出五至八条候选主张，链接到表格、图或错误分析。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“主张证据矩阵”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要使用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“主张必须包含范围，例如在哪个数据、预算和模型上成立。每条主张要有支持证据、反证或限制。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 从提案和结果写出五至八条候选主张，链接到表格、图或错误分析。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：主张证据矩阵。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“建立主张证据表”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“主张证据矩阵”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“建立主张证据表”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“主张证据矩阵”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：主张必须包含范围，例如在哪个数据、预算和模型上成立。每条主张要有支持证据、反证或限制。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“主张证据矩阵”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：先写强结论，再挑支持它的实验。；图轴截断却未说明。；没有误差、样本数或评价方向。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 149 制作主结果表

今天要懂 主结果表应优先回答研究问题，方法顺序和分组要清晰，指标注明越高或越低越好，最好值要结合误差解释。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“制作主结果表”，合格解释必须包含这一因果关系：主结果表应优先回答研究问题，方法顺

序和分组要清晰，指标注明越高或越低越好，最好值要结合误差解释。。然后把它与本周目标连接起来：让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：用脚本从冻结结果生成表格，加入均值、标准差、成本和必要脚注。。检查“主结果表”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“主结果表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议图表规范，Matplotlib 可访问性指南，相关论文主结果和消融写法。

今天先准备 建立 week22/day149 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：用脚本从冻结结果生成表格，加入均值、标准差、成本和必要脚注。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“主结果表”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“主结果表应优先回答研究问题，方法顺序和分组要清晰，指标注明越高或越低越好，最好值要结合误差解释。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用脚本从冻结结果生成表格，加入均值、标准差、成本和必要脚注。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：主结果表。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“制作主结果表”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“主结果表”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“制作主结果表”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“主结果表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：主结果表应优先回答研究问题，方法顺序和分组要清晰，指标注明越高或越低越好，最好值要结合误差解释。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 第四题应让别人找到输入、环境、运行命令、输出目录和“主结果表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：先写强结论，再挑支持它的实验。；图轴截断却未说明。；没有误差、样本数或评价方向。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 150 制作消融与效率图

今天要懂 图表适合展示趋势、权衡和分布。颜色不能是唯一区分方式，轴和单位必须完整，图注要能独立理解。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“制作消融与效率图”，合格解释必须包含这一因果关系：图表适合展示趋势、权衡和分布。颜色不能是唯一区分方式，轴和单位必须完整，图注要能独立理解。。然后把它与本周目标连接起来：让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：绘制消融趋势或效果成本散点图，使用不同标记和可辨颜色，导出 PDF 与高分辨率 PNG。。检查“两张论文级图”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“两张论文级图”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议图表规范，Matplotlib 可访问性指南，相关论文主结果和消融写法。

今天先准备 建立 week22/day150 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。

阶段	今天具体做什么
完成低成本检验	优先执行：绘制消融趋势或效果成本散点图，使用不同标记和可辨颜色，导出 PDF 与高分辨率 PNG。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“两张论文级图”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“图表适合展示趋势、权衡和分布。颜色不能是唯一区分方式，轴和单位必须完整，图注要能独立理解。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 绘制消融趋势或效果成本散点图，使用不同标记和可辨颜色，导出 PDF 与高分辨率 PNG。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：两张论文级图。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“制作消融与效率图”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“两张论文级图”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“制作消融与效率图”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“两张论文级图”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：图表适合展示趋势、权衡和分布。颜色不能是唯一区分方式，轴和单位必须完整，图注要能独立理解。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“两张论文级图”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：先写强结论，再挑支持它的实验。；图轴截断却未说明。；没有误差、样本数或评价方向。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 151 误差范围与统计说明

今天要懂 误差条可能表示标准差、标准误或置信区间，必须明确。样本数和随机种子数决定不确定性解释。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“误差范围与统计说明”，合格解释必须包含这一因果关系：误差条可能表示标准差、标准误或置信区间，必须明确。样本数和随机种子数决定不确定性解释。。然后把它与本周目标连接起来：让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：为所有误差条补定义，核对运行数，写一段说明小提升是否超过运行波动。。检查“统计说明段落”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“统计说明段落”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议图表规范，Matplotlib 可访问性指南，相关论文主结果和消融写法。

今天先准备 建立 week22/day151 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：为所有误差条补定义，核对运行数，写一段说明小提升是否超过运行波动。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“统计说明段落”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要一次不稳定运行冒充科研证据。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“误差条可能表示标准差、标准误或置信区间，必须明确。样本数和随机种子数决定不确定性解释。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 为所有误差条补定义，核对运行数，写一段说明小提升是否超过运行波动。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：统计说明段落。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“误差范围与统计说明”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“统计说明段落”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“误差范围与统计说明”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“统计说明段落”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：误差条可能表示标准差、标准误或置信区间，必须明确。样本数和随机种子数决定不确定性解释。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“统计说明段落”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：先写强结论，再挑支持它的实验。；图轴截断却未说明。；没有误差、样本数或评价方向。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 152 选择失败案例

今天要懂 失败案例要代表已统计的主要错误，而不是只挑戏剧性样本。输入、预测、真值和证据应同时展示。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“选择失败案例”，合格解释必须包含这一因果关系：失败案例要代表已统计的主要错误，而不是只挑戏剧性样本。输入、预测、真值和证据应同时展示。。然后把它与本周目标连接起来：让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：从错误标签中按比例选择成功与失败对照，匿名化敏感内容并写事实性图注。。检查“定性结果图”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“定性结果图”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议图表规范，Matplotlib 可访问性指南，相关论文主结果和消融写法。

今天先准备 建立 week22/day152 文件夹，先打开研究问题、强基线和上一轮结果。 把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：从错误标签中按比例选择成功与失败对照，匿名化敏感内容并写事实性图注。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。

阶段	今天具体做什么
更新研究记录	把证据、反例、结论边界和下一步写入“定性结果图”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“失败案例要代表已统计的主要错误，而不是只挑戏剧性样本。输入、预测、真值和证据应同时展示。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 从错误标签中按比例选择成功与失败对照，匿名化敏感内容并写事实性图注。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：定性结果图。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“选择失败案例”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“定性结果图”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“选择失败案例”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“定性结果图”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：失败案例要代表已统计的主要错误，而不是只挑戏剧性样本。输入、预测、真值和证据应同时展示。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“定性结果图”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：先写强结论，再挑支持它的实验。；图轴截断却未说明。；没有误差、样本数或评价方向。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 153 收紧结论范围

今天要懂 实验只支持特定条件下的结论。若泛化或效率结果不支持原主张，应缩小范围或明确权衡。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“收紧结论范围”，合格解释必须包含这一因果关系：实验只支持特定条件下的结论。若泛化或效率结果不支持原主张，应缩小范围或明确权衡。。然后把它与本周目标连接起来：让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：逐条审查主张证据表，删除无证据项，改写过强因果词，记录剩余未知。。检查“冻结的论文主张清单”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“冻结的论文主张清单”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议图表规范，Matplotlib 可访问性指南，相关论文主结果和消融写法。

今天先准备 建立 week22/day153 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：逐条审查主张证据表，删除无证据项，改写过强因果词，记录剩余未知。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“冻结的论文主张清单”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“实验只支持特定条件下的结论。若泛化或效率结果不支持原主张，应缩小范围或明确权衡。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 逐条审查主张证据表，删除无证据项，改写过强因果词，记录剩余未知。
- 4 一百三十至一百六十分** 整理并保存今日产出：冻结的论文主张清单。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“收紧结论范围”，并说明它在本周任务中的作用。

- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“冻结的论文主张清单”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“收紧结论范围”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“冻结的论文主张清单”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：实验只支持特定条件下的结论。若泛化或效率结果不支持原主张，应缩小范围或明确权衡。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 第四题应让别人找到输入、环境、运行命令、输出目录和“冻结的论文主张清单”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：先写强结论，再挑支持它的实验。；图轴截断却未说明。；没有误差、样本数或评价方向。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 154 第 22 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。

阶段	今天具体做什么
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：建立主张证据表、制作主结果表、制作消融与效率图、误差范围与统计说明、选择失败案例、收紧结论范围。每个主题说明定义、输入输出和一个常见错误。
- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“冻结的论文主张清单”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：让论文每个主要主张都能指向具体实验，并制作信息完整、不误导的表格和图。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再执行本页验收。

第 23 周 论文初稿

本周目标 完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。

首选材料 目标会议论文模板，五篇最接近论文，Zotero 或其他文献管理工具。

本周小白讲义

先写实验章节，因为它最接近已冻结证据，也最容易暴露缺失信息。方法章节沿数据流写，相关工作按问题类别组织，引言最后收束问题、缺口和贡献。

摘要必须给出具体任务、方法差异、关键设置、主要数字和限制。贡献数量由真实证据决定，不需要凑固定条数。

系统修订先查逻辑与证据，再查表达。逐段确认中心句、术语、数字来源和图表引用。

本周知识地图

天	主题	核心知识	产出
Day 155	写实验章节	实验章节最接近已完成证据，先写能暴露缺失信息。设置、基线、指标、主结果、消融、分析和效率都要对应冻结文件。	完整实验章节
Day 156	写方法章节	方法从问题定义开始，再给总图、组件、目标函数和训练推理流程。符号在首次出现时定义，文字与代码保持一致。	可由他人实现的方法章节
Day 157	写相关工作	相关工作用于定位差异，按研究问题或方法类别组织。每段先概括一类工作，再说明与你方法最相关的差别。	相关工作章节
Day 158	写引言与贡献	引言应说明具体问题、现有方法的证据性缺口、你的做法和主要结果。贡献数量由真实工作决定，不需要凑三条。	引言初稿
Day 159	写摘要 结论与局限	摘要包含问题、方法、关键设置、主要数字和限制。结论只总结已证实范围；局限说明失效条件和未覆盖群体。	摘要 结论 局限章节

天	主题	核心知识	产出
Day 160	完成第一次系统修订	修订先检查逻辑和证据，再处理句子。每段应有一个中心点，术语一致，所有表图先在正文被引用。	完整论文初稿 v1
Day 161	复盘验收	闭卷解释 独立重跑 整理失败补齐记录	周总结和通过记录

本周代码或记录模板

论文写作顺序

- 1 实验设置和结果
- 2 方法
- 3 相关工作
- 4 引言
- 5 摘要与结论
- 6 局限和伦理说明

本周常见错误

- 从摘要开始反复润色，实验章节仍不完整。
- 相关工作按论文逐篇罗列。
- 方法缺少训练细节和张量定义。

Day 155 写实验章节

今天要懂 实验章节最接近已完成证据，先写能暴露缺失信息。设置、基线、指标、主结果、消融、分析和效率都要对应冻结文件。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“写实验章节”，合格解释必须包含这一因果关系：实验章节最接近已完成证据，先写能暴露缺失信息。设置、基线、指标、主结果、消融、分析和效率都要对应冻结文件。。然后把它与本周目标连接起来：完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：按问题而非运行顺序组织实验，插入表图并写出每个结果支持的有限结论。。检查“完整实验章节”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“完整实验章节”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议论文模板，五篇最接近论文，Zotero 或其他文献管理工具。

今天先准备 建立 week23/day155 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：按问题而非运行顺序组织实验，插入表图并写出每个结果支持的有限结论。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“完整实验章节”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要一次不稳定运行冒充科研证据。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“实验章节最接近已完成证据，先写能暴露缺失信息。设置、基线、指标、主结果、消融、分析和效率都要对应冻结文件。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 按问题而非运行顺序组织实验，插入表图并写出每个结果支持的有限结论。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：完整实验章节。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“写实验章节”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“完整实验章节”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“写实验章节”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“完整实验章节”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：实验章节最接近已完成证据，先写能暴露缺失信息。设置、基线、指标、主结果、消融、分析和效率都要对应冻结文件。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“完整实验章节”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：从摘要开始反复润色，实验章节仍不完整。；相关工作按论文逐篇罗列。；方法缺少训练细节和张量定义。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 156 写方法章节

今天要懂 方法从问题定义开始，再给总图、组件、目标函数和训练推理流程。符号在首次出现时定义，文字与代码保持一致。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“写方法章节”，合格解释必须包含这一因果关系：方法从问题定义开始，再给总图、组件、目标函数和训练推理流程。符号在首次出现时定义，文字与代码保持一致。。然后把它与本周目标连接起来：完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：按数据流写方法，逐项加入形状、冻结参数、损失权重、复杂度和伪代码。。检查“可由他人实现的方法章节”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“可由他人实现的方法章节”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议论文模板，五篇最接近论文，Zotero 或其他文献管理工具。

今天先准备 建立 week23/day156 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。

阶段	今天具体做什么
完成低成本检验	优先执行：按数据流写方法，逐项加入形状、冻结参数、损失权重、复杂度和伪代码。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“可由他人实现的方法章节”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“方法从问题定义开始，再给总图、组件、目标函数和训练推理流程。符号在首次出现时定义，文字与代码保持一致。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 按数据流写方法，逐项加入形状、冻结参数、损失权重、复杂度和伪代码。
- 4 一百三十至一百六十分** 整理并保存今日产出：可由他人实现的方法章节。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“写方法章节”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“可由他人实现的方法章节”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“写方法章节”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“可由他人实现的方法章节”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：方法从问题定义开始，再给总图、组件、目标函数和训练推理流程。符号在首次出现时定义，文字与代码保持一致。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“可由他人实现的方法章节”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：从摘要开始反复润色，实验章节仍不完整。；相关工作按论文逐篇罗列。；方法缺少训练细节和张量定义。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 157 写相关工作

今天要懂 相关工作用于定位差异，按研究问题或方法类别组织。每段先概括一类工作，再说明与你方法最相关的差别。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“写相关工作”，合格解释必须包含这一因果关系：相关工作用于定位差异，按研究问题或方法类别组织。每段先概括一类工作，再说明与你方法最相关的差别。。然后把它与本周目标连接起来：完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：把文献矩阵聚成三至四类，引用原始论文，避免用博客替代证据。。检查“相关工作章节”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“相关工作章节”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议论文模板，五篇最接近论文，Zotero 或其他文献管理工具。

今天先准备 建立 week23/day157 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：把文献矩阵聚成三至四类，引用原始论文，避免用博客替代证据。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“相关工作章节”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要一次不稳定运行冒充科研证据。

执行步骤

I 零至十五分钟 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。

- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“相关工作用于定位差异，按研究问题或方法类别组织。每段先概括一类工作，再说明与你方法最相关的差别。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 把文献矩阵聚成三至四类，引用原始论文，避免用博客替代证据。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：相关工作章节。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“写相关工作”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“相关工作章节”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“写相关工作”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“相关工作章节”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：相关工作用于定位差异，按研究问题或方法类别组织。每段先概括一类工作，再说明与你方法最相关的差别。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2** 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。”解释差异。
- 3** 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4** 第四题应让别人找到输入、环境、运行命令、输出目录和“相关工作章节”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：从摘要开始反复润色，实验章节仍不完整。；相关工作按论文逐篇罗列。；方法缺少训练细节和张量定义。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 158 写引言与贡献

今天要懂 引言应说明具体问题、现有方法的证据性缺口、你的做法和主要结果。贡献数量由真实工作决定，不需要凑三条。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“写引言与贡献”，合格解释必须包含这一因果关系：引言应说明具体问题、现有方法的证据性缺口、你的做法和主要结果。贡献数量由真实工作决定，不需要凑三条。。然后把它与本周目标连接起来：完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：用五段结构写引言，并逐条核对贡献是否有实验或公开产出支持。。检查“引言初稿”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“引言初稿”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议论文模板，五篇最接近论文，Zotero 或其他文献管理工具。

今天先准备 建立 week23/day158 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：用五段结构写引言，并逐条核对贡献是否有实验或公开产出支持。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。

阶段	今天具体做什么
更新研究记录	把证据、反例、结论边界和下一步写入“引言初稿”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要使用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“引言应说明具体问题、现有方法的证据性缺口、你的做法和主要结果。贡献数量由真实工作决定，不需要凑三条。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 用五段结构写引言，并逐条核对贡献是否有实验或公开产出支持。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：引言初稿。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“写引言与贡献”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“引言初稿”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“写引言与贡献”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“引言初稿”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：引言应说明具体问题、现有方法的证据性缺口、你的做法和主要结果。贡献数量由真实工作决定，不需要凑三条。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“引言初稿”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：从摘要开始反复润色，实验章节仍不完整。；相关工作按论文逐篇罗列。；方法缺少训练细节和张量定义。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 159 写摘要 结论与局限

今天要懂 摘要包含问题、方法、关键设置、主要数字和限制。结论只总结已证实范围；局限说明失效条件和未覆盖群体。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“写摘要 结论与局限”，合格解释必须包含这一因果关系：摘要包含问题、方法、关键设置、主要数字和限制。结论只总结已证实范围；局限说明失效条件和未覆盖群体。。然后把它与本周目标连接起来：完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：从主张证据表压缩摘要，写具体局限和可能误用，不使用宏大空话。。检查“摘要 结论 局限章节”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“摘要 结论 局限章节”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议论文模板，五篇最接近论文，Zotero 或其他文献管理工具。

今天先准备 建立 week23/day159 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：从主张证据表压缩摘要，写具体局限和可能误用，不使用宏大空话。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“摘要 结论 局限章节”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“摘要包含问题、方法、关键设置、主要数字和限制。结论只总结已证实范围；局限说明失效条件和未覆盖群体。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 从主张证据表压缩摘要，写具体局限和可能误用，不使用宏大空话。
- 4 一百三十至一百六十分** 整理并保存今日产出：摘要 结论 局限章节。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“写摘要 结论与局限”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“摘要 结论 局限章节”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“写摘要 结论与局限”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“摘要 结论 局限章节”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：摘要包含问题、方法、关键设置、主要数字和限制。结论只总结已证实范围；局限说明失效条件和未覆盖群体。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“摘要 结论 局限章节”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：从摘要开始反复润色，实验章节仍不完整。；相关工作按论文逐篇罗列。；方法缺少训练细节和张量定义。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 160 完成第一次系统修订

今天要懂 修订先检查逻辑和证据，再处理句子。每段应有一个中心点，术语一致，所有表图先在正文被引用。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“完成第一次系统修订”，合格解释必须包含这一因果关系：修订先检查逻辑和证据，再处理

句子。每段应有一个中心点，术语一致，所有表图先在正文被引用。。然后把它与本周目标连接起来：完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：打印或导出全文，逐节检查主张、证据、数字、图注和引用，请一名读者写三条最难理解之处。。检查“完整论文初稿 v1”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“完整论文初稿 v1”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议论文模板，五篇最接近论文，Zotero 或其他文献管理工具。

今天先准备 建立 week23/day160 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：打印或导出全文，逐节检查主张、证据、数字、图注和引用，请一名读者写三条最难理解之处。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“完整论文初稿 v1”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“修订先检查逻辑和证据，再处理句子。每段应有一个中心点，术语一致，所有表图先在正文被引用。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 打印或导出全文，逐节检查主张、证据、数字、图注和引用，请一名读者写三条最难理解之处。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：完整论文初稿 v1。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“完成第一次系统修订”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“完整论文初稿 v1”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“完成第一次系统修订”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“完整论文初稿 v1”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：修订先检查逻辑和证据，再处理句子。每段应有一个中心点，术语一致，所有表图先在正文被引用。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“完整论文初稿 v1”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：从摘要开始反复润色，实验章节仍不完整。；相关工作按论文逐篇罗列。；方法缺少训练细节和张量定义。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 161 第 23 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：写实验章节、写方法章节、写相关工作、写引言与贡献、写摘要 结论与局限、完成第一次系统修订。每个主题说明定义、输入输出和一个常见错误。

- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“完整论文初稿 v1”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- ☐ 能独立解释本周目标：完成一篇方法可实现、实验可复核、主张不超过证据范围的论文初稿。
- ☐ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- ☐ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- ☐ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再执行本页验收。

第 24 周 复核 投稿与科研习惯

本周目标 完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。

首选材料 目标会议当年作者指南、模板、匿名要求、补充材料、伦理和生成式 AI 政策。

本周小白讲义

投稿前从空目录重生成表图并做干净环境复现。摘要、正文、表格和补充材料中的同一数字必须一致。

模板、匿名、页数、引用、数据许可、伦理和生成式 AI 政策都以目标会议当年规则为准。匿名还包括元数据、仓库和自引用措辞。

提交包保留论文、补充材料、代码快照、结果清单和检查记录。录用不由单次努力保证，但可复现和证据完整是可以主动控制的质量底线。

本周知识地图

天	主题	核心知识	产出
Day 162	重生成全部数字	最终数字必须从冻结结果由脚本生成，不能依赖手工复制。摘要、正文、表格和补充材料中的同一指标应完全一致。	最终数字审计表
Day 163	干净环境复现	提交前要在干净环境验证核心命令。缓存、本地绝对路径和未提交代码可能让你的机器能跑而别人失败。	干净复现日志
Day 164	格式与匿名检查	会议模板、页数、字体、图表、双盲和补充材料规则每年可能变化。匿名不只删除作者名，还包括仓库、致谢和元数据。	格式与匿名清单
Day 165	引用 许可与素材	每个事实和比较应引用原始来源。数据、权重、代码、字体和图片可能有不同许可证，公开产出必须遵守。	引用与许可审计
Day 166	伦理 局限与 AI 政策	伦理说明要对应数据、应用、偏差、隐私和误用。若使用生成式 AI 辅助写作或代码，必须遵守目标会议当年披露政策。	伦理和政策合规说明

天	主题	核心知识	产出
Day 167	形成投稿包与下一轮计划	论文提交不是研究结束。无论结果如何，都要保存可复现版本、审稿反馈和下一步优先级，避免下轮重新寻找上下文。	完整投稿包和后续迭代计划
Day 168	复盘验收	闭卷解释 独立重跑 整理失败 补齐记录	周总结和通过记录

本周代码或记录模板

```
final_checks = [
    'numbers_regenerated', 'citations_verified', 'anonymous',
    'licenses_checked', 'ethics_written', 'clean_reproduction_passed'
]
assert all(status[item] for item in final_checks)
```

本周常见错误

- 沿用往年投稿规则。
- 手工复制数字造成正文与表格不一致。
- 匿名稿、代码或文件元数据泄露身份。

Day 162 重生成全部数字

今天要懂 最终数字必须从冻结结果由脚本生成，不能依赖手工复制。摘要、正文、表格和补充材料中的同一指标应完全一致。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“重生成全部数字”，合格解释必须包含这一因果关系：最终数字必须从冻结结果由脚本生成，不能依赖手工复制。摘要、正文、表格和补充材料中的同一指标应完全一致。。然后把它与本周目标连接起来：完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：在新输出目录运行全部表图脚本，逐项比较论文数字并记录来源文件。。检查“最终数字审计表”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“最终数字审计表”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议当年作者指南、模板、匿名要求、补充材料、伦理和生成式 AI 政策。

今天先准备 建立 week24/day162 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：在新输出目录运行全部表图脚本，逐项比较论文数字并记录来源文件。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“最终数字审计表”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“最终数字必须从冻结结果由脚本生成，不能依赖手工复制。摘要、正文、表格和补充材料中的同一指标应完全一致。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 在新输出目录运行全部表图脚本，逐项比较论文数字并记录来源文件。

4 一百三十至一百六十分钟 整理并保存今日产出：最终数字审计表。文件名写明 day 编号，不覆盖昨天结果。

5 一百六十至一百八十分钟 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“重生成全部数字”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“最终数字审计表”，其中包含日期、环境或配置、结果和一句结论。

自测题

1 不用资料，用三句话解释“重生成全部数字”：第一句给定义，第二句说输入输出，第三句说一个失败条件。

2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。

3 假设“最终数字审计表”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。

4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

1 第一题至少应覆盖：最终数字必须从冻结结果由脚本生成，不能依赖手工复制。摘要、正文、表格和补充材料中的同一指标应完全一致。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“最终数字审计表”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：沿用往年投稿规则。；手工复制数字造成正文与表格不一致。；匿名稿、代码或文件元数据泄露身份。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 163 干净环境复现

今天要懂 提交前要在干净环境验证核心命令。缓存、本地绝对路径和未提交代码可能让你的机器能跑而别人失败。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“干净环境复现”，合格解释必须包含这一因果关系：提交前要在干净环境验证核心命令。缓存、本地绝对路径和未提交代码可能让你的机器能跑而别人失败。。然后把它与本周目标连接起来：完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：从新目录获取冻结代码，按 README 安装，运行最小训练或推理和最终评估。。检查“干净复现日志”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“干净复现日志”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议当年作者指南、模板、匿名要求、补充材料、伦理和生成式 AI 政策。

今天先准备 建立 week24/day163 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：从新目录获取冻结代码，按 README 安装，运行最小训练或推理和最终评估。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“干净复现日志”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“提交前要在干净环境验证核心命令。缓存、本地绝对路径和未提交代码可能让你的机器能跑而别人失败。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 从新目录获取冻结代码，按 README 安装，运行最小训练或推理和最终评估。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：干净复现日志。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- ☐ 不看资料，用不超过两分钟解释“干净环境复现”，并说明它在本周任务中的作用。
- ☐ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- ☐ 打开并检查“干净复现日志”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1** 不用资料，用三句话解释“干净环境复现”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2** 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3** 假设“干净复现日志”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4** 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1** 第一题至少应覆盖：提交前要在干净环境验证核心命令。缓存、本地绝对路径和未提交代码可能让你的机器能跑而别人失败。。只背名词、没有输入输出或失败条件，说明理解仍不完整。

2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。”解释差异。

3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“干净复现日志”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：沿用往年投稿规则。；手工复制数字造成正文与表格不一致。；匿名稿、代码或文件元数据泄露身份。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 164 格式与匿名检查

今天要懂 会议模板、页数、字体、图表、双盲和补充材料规则每年可能变化。匿名不只删除作者名，还包括仓库、致谢和元数据。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“格式与匿名检查”，合格解释必须包含这一因果关系：会议模板、页数、字体、图表、双盲和补充材料规则每年可能变化。匿名不只删除作者名，还包括仓库、致谢和元数据。。然后把它与本周目标连接起来：完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：逐条核对当年指南，检查 PDF 属性、文件名、自引用措辞、代码链接和图片中的身份信息。。检查“格式与匿名清单”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“格式与匿名清单”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议当年作者指南、模板、匿名要求、补充材料、伦理和生成式 AI 政策。

今天先准备 建立 week24/day164 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：逐条核对当年指南，检查 PDF 属性、文件名、自引用措辞、代码链接和图片中的身份信息。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“格式与匿名清单”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“会议模板、页数、字体、图表、双盲和补充材料规则
每年可能变化。匿名不只删除作者名，还包括仓库、致谢和元数据。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分** 逐条核对当年指南，检查 PDF 属性、文件名、自引用措辞、代码链接和图片中的身份信息。
- 4 一百三十至一百六十分** 整理并保存今日产出：格式与匿名清单。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“格式与匿名检查”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- □ 打开并检查“格式与匿名清单”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“格式与匿名检查”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“格式与匿名清单”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：会议模板、页数、字体、图表、双盲和补充材料规则每年可能变化。匿名不只删除作者名，还包括仓库、致谢和元数据。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“格式与匿名清单”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：沿用往年投稿规则。；手工复制数字造成正文与表格不一致。；匿名稿、代码或文件元数据泄露身份。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 165 引用 许可与素材

今天要懂 每个事实和比较应引用原始来源。数据、权重、代码、字体和图片可能有不同许可证，公开产出必须遵守。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“引用 许可与素材”，合格解释必须包含这一因果关系：每个事实和比较应引用原始来源。数

据、权重、代码、字体和图片可能有不同许可证，公开产出必须遵守。。然后把它与本周目标连接起来：完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：核对每条引用、URL 或 DOI，建立素材许可表，删除无法确认权限的内容。。检查“引用与许可审计”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“引用与许可审计”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议当年作者指南、模板、匿名要求、补充材料、伦理和生成式 AI 政策。

今天先准备 建立 week24/day165 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：核对每条引用、URL 或 DOI，建立素材许可表，删除无法确认权限的内容。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“引用与许可审计”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“每个事实和比较应引用原始来源。数据、权重、代码、字体和图片可能有不同许可证，公开产出必须遵守。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 核对每条引用、URL 或 DOI，建立素材许可表，删除无法确认权限的内容。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：引用与许可审计。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“引用 许可与素材”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“引用与许可审计”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“引用 许可与素材”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“引用与许可审计”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：每个事实和比较应引用原始来源。数据、权重、代码、字体和图片可能有不同许可证，公开产出必须遵守。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“引用与许可审计”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：沿用往年投稿规则。；手工复制数字造成正文与表格不一致。；匿名稿、代码或文件元数据泄露身份。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 166 伦理 局限与 AI 政策

今天要懂 伦理说明要对应数据、应用、偏差、隐私和误用。若使用生成式 AI 辅助写作或代码，必须遵守目标会议当年披露政策。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“伦理 局限与 AI 政策”，合格解释必须包含这一因果关系：伦理说明要对应数据、应用、偏差、隐私和误用。若使用生成式 AI 辅助写作或代码，必须遵守目标会议当年披露政策。。然后把它与本周目标连接起来：完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：检查风险表是否在论文体现，核对人工标注、隐私处理、模型使用与 AI 辅助披露要求。。检查“伦理和政策合规说明”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“伦理和政策合规说明”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议当年作者指南、模板、匿名要求、补充材料、伦理和生成式 AI 政策。

今天先准备 建立 week24/day166 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：检查风险表是否在论文体现，核对人工标注、隐私处理、模型使用与 AI 辅助披露要求。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“伦理和政策合规说明”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“伦理说明要对应数据、应用、偏差、隐私和误用。若使用生成式 AI 辅助写作或代码，必须遵守目标会议当年披露政策。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 检查风险表是否在论文体现，核对人工标注、隐私处理、模型使用与 AI 辅助披露要求。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：伦理和政策合规说明。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- □ 不看资料，用不超过两分钟解释“伦理 局限与 AI 政策”，并说明它在本周任务中的作用。
- □ 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。

- □ 打开并检查“伦理和政策合规说明”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 1 不用资料，用三句话解释“伦理 局限与 AI 政策”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 2 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 3 假设“伦理和政策合规说明”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 4 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 1 第一题至少应覆盖：伦理说明要对应数据、应用、偏差、隐私和误用。若使用生成式 AI 辅助写作或代码，必须遵守目标会议当年披露政策。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 2 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。”解释差异。
- 3 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。
- 4 第四题应让别人找到输入、环境、运行命令、输出目录和“伦理和政策合规说明”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：沿用往年投稿规则。；手工复制数字造成正文与表格不一致。；匿名稿、代码或文件元数据泄露身份。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 167 形成投稿包与下一轮计划

今天要懂 论文提交不是研究结束。无论结果如何，都要保存可复现版本、审稿反馈和下一步优先级，避免下轮重新寻找上下文。

原理拆解

先回答三个问题：今天处理的对象是什么；它的输入怎样变成输出；结果错误时最可能从哪里开始排查。对于“形成投稿包与下一轮计划”，合格解释必须包含这一因果关系：论文提交不是研究结束。无论结果如何，都要保存可复现版本、审稿反馈和下一步优先级，避免下轮重新寻找上下文。。然后把它与本周目标连接起来：完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。

科研连接 今天的结论要写清适用范围和反证条件。研究不是证明自己一定正确，而是让问题能够被公平实验回答。

跟着做一个例子

先把今天的任务改写为可被反驳的问题，再执行：打包匿名论文、补充材料、代码快照和检查清单，建立提交后错误处理与审稿回复模板。。检查“完整投稿包和后续迭代计划”中的每项判断是否能链接到表格、图、日志、预测或原始论文，证据不足时缩小结论。

完成标准 今天结束前必须得到“完整投稿包和后续迭代计划”。只看完材料不算完成。

建议投入 二点五至三小时。若当日只有九十分钟，先完成知识讲解和最小实践，第二天开始前补齐产出与验收。

今日执行卡

首选材料 目标会议当年作者指南、模板、匿名要求、补充材料、伦理和生成式 AI 政策。

今天先准备 建立 week24/day167 文件夹，先打开研究问题、强基线和上一轮结果。把今天的工作写成一个可被反驳的判断：改变什么、相对谁、看哪个主指标、预期方向是什么。先做最低成本的对照或错误分析，再决定是否扩大数据、模型或实验次数；不要直接追逐最好数字。

阶段	今天具体做什么
写下假设	在运行前写出 H0、H1、主指标、最小样本量或种子数，以及什么结果会让你放弃当前解释。
完成低成本检验	优先执行：打包匿名论文、补充材料、代码快照和检查清单，建立提交后错误处理与审稿回复模板。；先验证数据、基线和评价协议，再扩大规模。
拆开结果	分别报告效果、波动、显存或延迟，并保存最好、最差和典型案例；不要只挑一个漂亮输出。
更新研究记录	把证据、反例、结论边界和下一步写入“完整投稿包和后续迭代计划”，并注明哪些判断仍然未知。

通过标准

- 假设、基线、唯一改变因素、主指标和结论边界都写在同一份研究记录中。
- 主结果至少有一个强基线、一个反例或失败分析，并报告随机性或明确说明无法估计的原因。
- 别人能沿着记录找到数据、命令、配置、输出和图表；未验证的部分明确标为未知。

低算力和时间不足 资源不足时优先做错误分析、协议核对和小规模消融；宁可缩小问题和结论，也不要
用一次不稳定运行冒充科研证据。

执行步骤

- 1 零至十五分钟** 关闭资料，用五分钟复述昨天的三点收获，再查看未解决问题。
- 2 十五至六十分钟** 阅读今天的知识讲解，围绕“论文提交不是研究结束。无论结果如何，都要保存可复现版本、审稿反馈和下一步优先级，避免下轮重新寻找上下文。”写不少于五条自己的话，并画出输入、处理、输出。
- 3 六十至一百三十分钟** 打包匿名论文、补充材料、代码快照和检查清单，建立提交后错误处理与审稿回复模板。
- 4 一百三十至一百六十分钟** 整理并保存今日产出：完整投稿包和后续迭代计划。文件名写明 day 编号，不覆盖昨天结果。
- 5 一百六十至一百八十分钟** 完成今日验收，提交 Git 记录或备份，写下一个仍不确定的问题和明天第一步。

今日验收

- 不看资料，用不超过两分钟解释“形成投稿包与下一轮计划”，并说明它在本周任务中的作用。
- 独立重新执行今日实践，能够指出输入、关键处理、输出和至少一个失败条件。
- 打开并检查“完整投稿包和后续迭代计划”，其中包含日期、环境或配置、结果和一句结论。

自测题

- 不用资料，用三句话解释“形成投稿包与下一轮计划”：第一句给定义，第二句说输入输出，第三句说一个失败条件。
- 改变今天实践中的一个关键值，先预测结果怎样变化，再运行验证。不得同时改变两个主要因素。
- 假设“完整投稿包和后续迭代计划”中的结论与预期相反，列出数据、实现、评价三个层面的可能原因。
- 向未来的自己写一条复现说明：从空环境开始，需要哪条命令、哪个输入和哪个配置。

参考答案和评分点

- 第一题至少应覆盖：论文提交不是研究结束。无论结果如何，都要保存可复现版本、审稿反馈和下一步优先级，避免下轮重新寻找上下文。。只背名词、没有输入输出或失败条件，说明理解仍不完整。
- 第二题没有唯一数值答案。合格记录必须先写预测，再只改变一个值，保存前后结果，并用本周目标“完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。”解释差异。
- 第三题至少分别检查一项：数据是否对应且无泄漏；代码的类型、shape、device、参数和随机种子是否正确；指标、划分和后处理是否与任务一致。

4 第四题应让别人找到输入、环境、运行命令、输出目录和“完整投稿包和后续迭代计划”。缺少任一项都不能算可复现。

卡住时 先检查本周常见错误：沿用往年投稿规则。；手工复制数字造成正文与表格不一致。；匿名稿、代码或文件元数据泄露身份。。连续四十分钟没有进展时，缩小为一个样本或一个 batch，打印类型、shape、device 和中间输出，再记录完整报错。

Day 168 第 24 周复盘验收

今天要做 停止学习新概念，检验本周知识能否在没有教程陪伴的情况下被解释、运行和复现。预计两至三小时。

今日执行卡

先准备 打开本周六个 Day 的产出、代码和日志，另建一个复盘文件夹；不要只凭记忆写总结。

阶段	今天具体做什么
闭卷回忆	不看资料写出本周六个主题的定义、输入输出、一个失败条件和它们之间的关系。
干净重跑	在新的终端或 Notebook 内核中重做本周最重要实践，确认没有依赖残留变量或手工改动。
证据审计	抽查两个代码文件、一个图表和一个指标，核对数据来源、配置、命令、单位和结论是否对应。
补课与归档	把最弱知识点和最有价值的报错写进周总结，补齐缺失文件，明确下周第一条命令。

通过标准

- 六个主题都能在不看资料的情况下说清定义、输入输出和一个失败条件。
- 至少一个关键实践能从新会话重跑，输出与原记录一致或差异有解释。
- 周总结包含本周作品、最弱知识点、累计投入、失败修复、下周风险和第一步。

低算力和时间不足 本周只完成了部分任务时，优先重跑最能代表本周目标的一个实践，再补最弱知识点和周总结；不要用浏览材料或勾选网页替代复盘证据。

执行步骤

- 1 闭卷口述本周六个主题：重生成全部数字、干净环境复现、格式与匿名检查、引用许可与素材、伦理局限与 AI 政策、形成投稿包与下一轮计划。每个主题说明定义、输入输出和一个常见错误。

- 2 从空白终端或新 Notebook 重做本周最重要实践，并生成“完整投稿包和后续迭代计划”。不要使用仍驻留在内存中的变量。
- 3 随机抽取本周两个代码文件，为关键张量或数据补充 shape、dtype、device 和数据来源说明。
- 4 整理本周失败记录，选一个最有价值的错误，写清现象、原因、修复和如何避免。
- 5 填写周总结：本周作品、最弱知识点、累计投入、下周风险和第一项任务。提交或备份全部文件。

周验收门槛

- □ 能独立解释本周目标：完成数字、代码、格式、匿名、许可和伦理检查，形成可提交且能继续迭代的投稿包。
- □ 六天的必交产出全部存在，文件可打开，结果不是空白或占位符。
- □ 关键实践可以从新会话重新运行，README 或日志足以恢复步骤。
- □ 自测至少达到八成。未通过时只补相关天，不通过验收前不进入下一周。

补课规则 解释不清回看对应知识讲解；代码不能重跑回到独立实践日；结果不可追溯回到产出记录日；指标不理解回到评估日。补课后再次执行本页验收。

第七部分 科研模板与毕业验收

每日学习日志模板

日期与 Day
投入时间
今天完成的文件
我能用自己的话解释
运行命令与环境
结果与一句结论
完整报错与修复
仍不确定的问题
明天第一步

论文阅读卡模板

字段	填写要求
论文信息	标题 作者 年份 会议或版本 代码链接 阅读日期
问题	输入 输出 使用场景和现有方法缺口
方法	总数据流 核心改动 训练参数 损失和复杂度
证据	数据划分 强基线 主指标 消融和效率
限制	失效条件 数据偏差 算力和未验证主张
复现	仓库 commit 环境 命令 预计成本和风险
我的问题	一个反例 一个可能改进 一个需要核对的细节

实验运行记录模板

字段	示例或要求
run id	method dataset seed timestamp
版本	Git commit 数据版本 模型权重版本
环境	Python PyTorch CUDA GPU 与关键依赖
变量	本次唯一改变的因素

字段	示例或要求
控制项	数据 预算 优化器 步数 指标和提示
结果	主指标 副指标 均值标准差 成本
状态	完成 失败 中断及完整原因
结论	结果支持什么 不支持什么 下一步

论文毕业门槛

- □ 基础 能独立写训练循环，解释反向传播、注意力、预训练、LoRA 和 RAG。
- □ 多模态 能解释对齐与融合，完成迷你 CLIP 和带固定评测的应用。
- □ 复现 目标论文主要趋势可复现，差异、失败和运行成本有记录。
- □ 研究 问题可证伪，强基线公平，主结果、消融、泛化和效率形成证据链。
- □ 写作 方法可实现，所有数字可重新生成，主张范围不超过实验。
- □ 投稿 模板、匿名、引用、许可、伦理和生成式 AI 政策按当年要求检查。

最终说明 完成 168 天意味着你具备执行第一轮多模态科研闭环的基本能力，不代表论文必然录用。论文结果还取决于问题价值、创新性、实验质量、导师反馈和投稿竞争。把可复现和可信结论放在追求漂亮数字之前。