机器学习系统 2026春

第十章 大模型训练

张燕咏 讲席教授 yanyongz@ustc.edu.cn
张午阳 特任教授 wuyangz@ustc.edu.cn

中国科学技术大学
University of Science and Technology of China

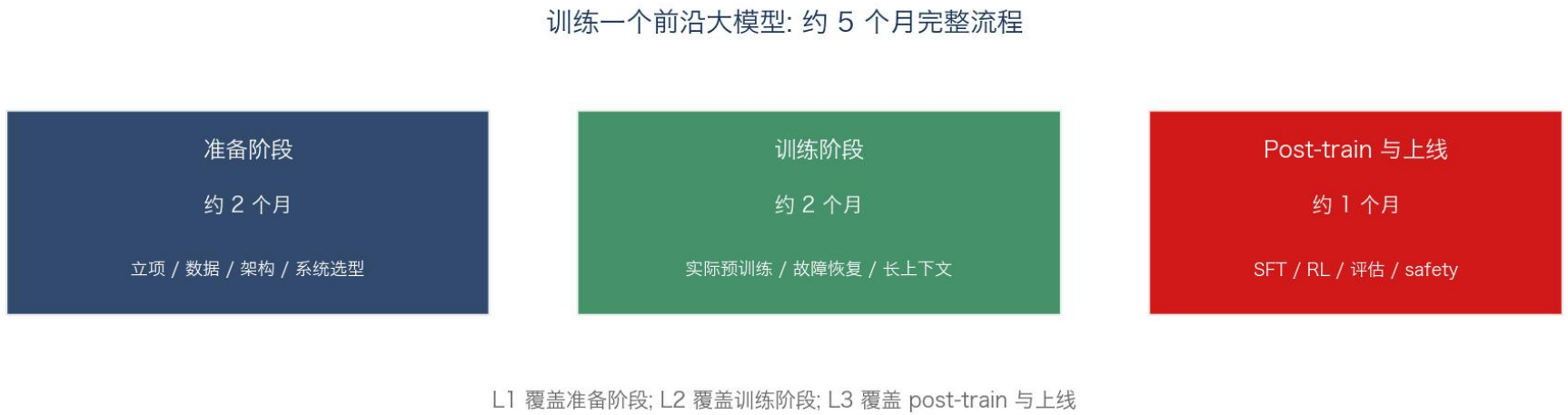
第一节课

准备阶段 (约 2 个月)

立项, 数据 pipeline, 架构选型, 系统准备

训练一个前沿大模型: 完整流程

本章节课程阶段化拆解前沿大模型：从立项到上线



课程目标

- 完整的大模型训练工程流程
- 以 DeepSeek V3 为主线案例
- 对照 Kimi K2, Qwen3.5, Gemini 3.1, Claude Opus 4.5 等最新模型

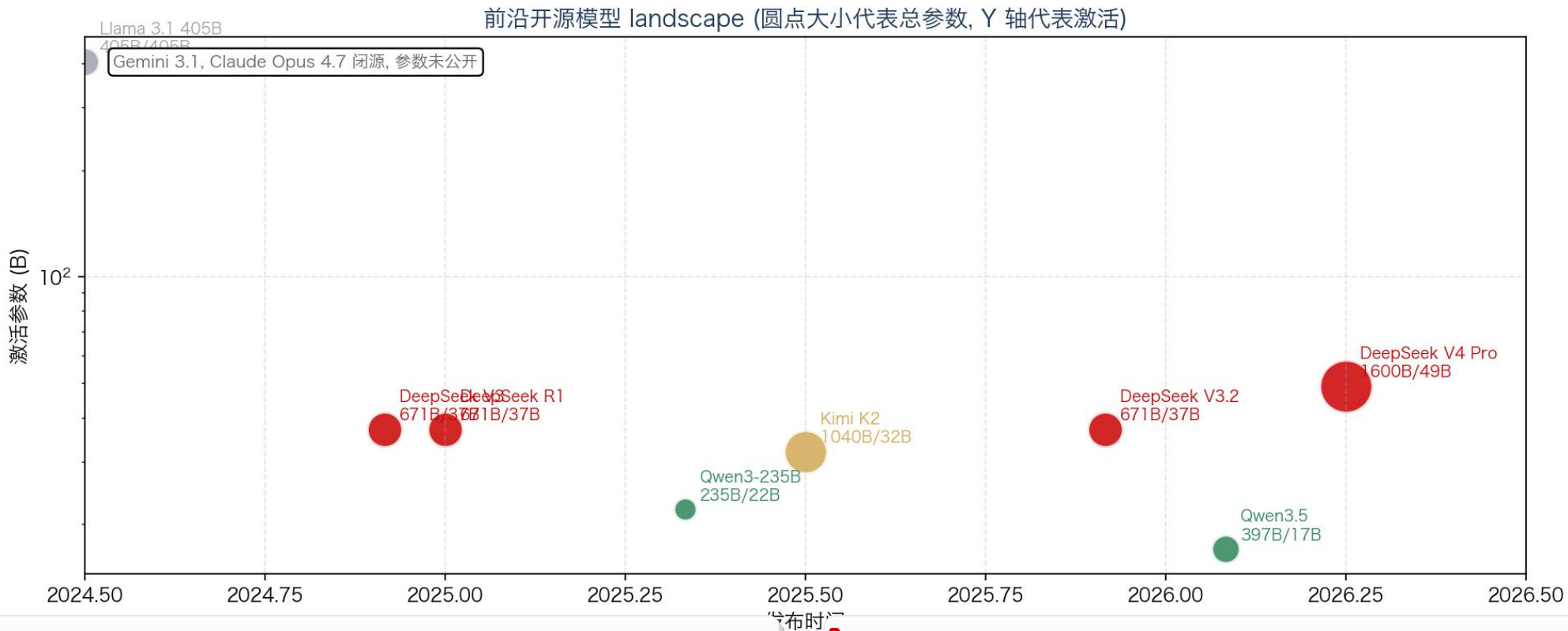
三节对应三个阶段

- L1 准备阶段约 2 个月: 立项, 数据 pipeline, 架构选型, 系统选型
- L2 训练阶段约 2 个月: 实际预训练, 故障恢复, 长上下文升级
- L3 post-train 与上线约 1 个月: SFT, RL, 蒸馏, 评估, 安全, 发布

学完三节课, 拿到任何一份新的前沿模型训练报告, 能够迅速抓住核心创新

主要参考: DeepSeek V3 V3.2 V4 R1, Kimi K2, Qwen3 系列, Gemini 2.5/3, Claude

决定模型好坏的三个关键因素



前沿模型整体格局观察

- 总参 235B 到 1.6T, 激活 17 到 49B, 稀疏化已成默认
- 2024 末以来前沿模型几乎都采用 MoE 架构
- 2026 新进入清单: V4 1.6T/49B, K2.6, Q3.6 412B/17B, Gemini 3.1 Deep Think

三个真正决定模型差距的因素

- 数据质量 pipeline (Apple DCLM 控制实验证明影响最大)
- 架构与系统协同 (DeepSeek V3 用 MLA + FP8 + DualPipe + aux-loss-free 省 20 倍)
- Post-train 配方 (R1 用 GRPO + RLVR 涌现 reasoning, 同

MIT 2026 training-study: 前沿水平 80% 来自数据算力, 14-18% 来自工程。本章节专注 14-18%

MIT meta-study arxiv:2602.07238; DCLM arxiv:2406.11794

准备阶段核心步骤

L1 (约 2 个月) 内容

- 第一段, 立项与算账 (4 张). 项目目标, 成本对比, scaling law 决定 671B/37B MoE
- 第二段, 数据 pipeline (8 张). 三级去重, 质量分类, 数据混合, annealing, tokenizer, 合成数据
- 第三段, 架构选型 (10 张). 8 个关键选择, MoE 设计, MLA, aux-loss-free, MTP, MuonClip 等
- 第四段, 系统准备 (5 张). FP8 决策, DualPipe, 4D 并行, 1B 代理验证
- 最后总结 + 三家对比

一旦进入训练就只能去实现这些决策, 中途调整成本是几百万到几亿美元。

结构参考 DeepSeek V3 arxiv:2412.19437, Llama 3.1 arxiv:2407.21783

立项目标与约束

项目设定

- 参考 DeepSeek V3 公开报告中的资源量级
- 集群约 2048 张 H800 (出口管制环境下拿不到 H100)
- 训练算力预算受限, 必须通过架构与系统设计降低算力消耗
- 目标性能与同期闭源前沿模型可比
- 时间窗口 5 个月左右出 base 模型
- 团队规模约 50 人 (算法, 工程, 数据三方向)
- 团队里没人在 671B 量级跑过完整训练

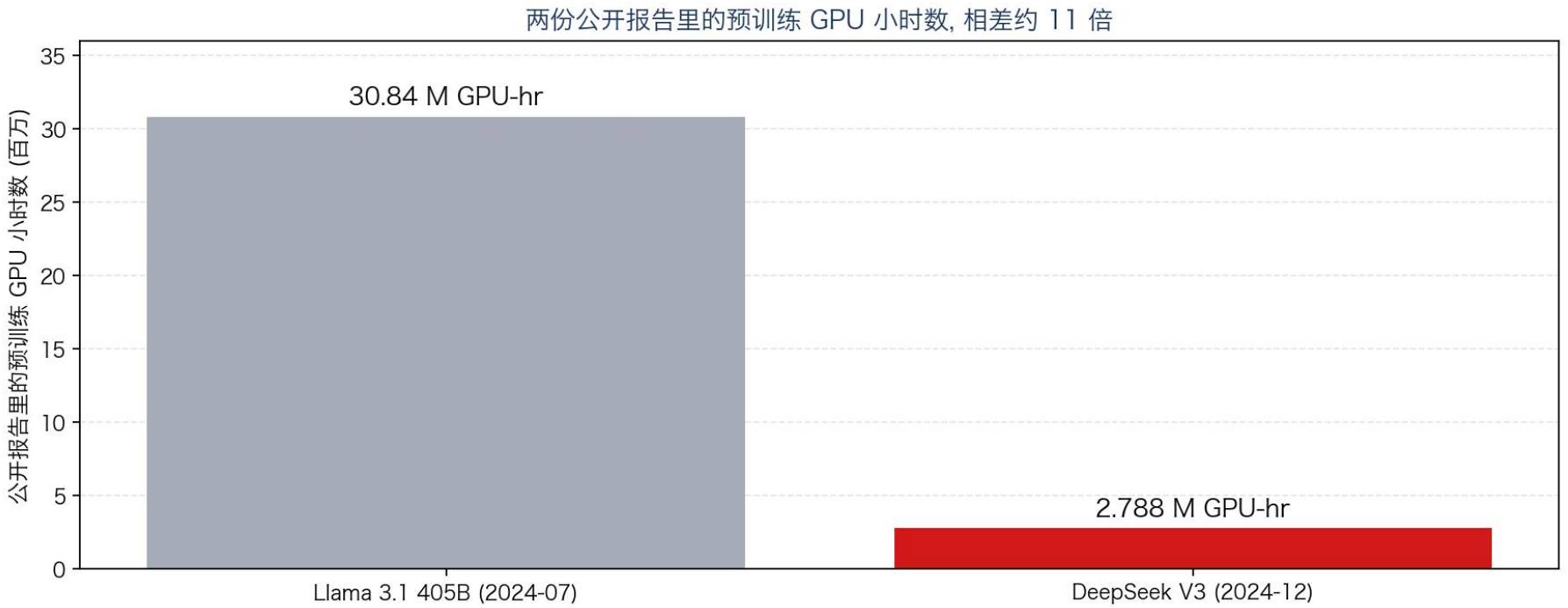
为什么这些约束很重要

- H800 的 NVLink 带宽比 H100 低 (约 300 GB/s 对 600 GB/s)
- IB 网络也受限
- 集群规模只有 Llama 3 (16K H100) 的 1/8
- 这些约束直接逼出架构级别的省钱设计
- 不能像 Meta 那样硬训
- 后来事实证明, 这些约束反而促成了 V3 的多个工程创新
- 团队的工程创新很大程度上源自硬件资源限制

训练前沿模型最重要的不是有多少 GPU, 而是怎么在有限 GPU 上做出前沿水平。

DeepSeek V3 arxiv:2412.19437, Kimi K2 arxiv:2507.20534

公开报告中的预训练 GPU 小时数对比



数字来源

- Llama 3.1 405B: 30.84M H100 小时
- DeepSeek V3: 2.788M H800 小时
- 均为预训练 GPU 小时数, 不含其他开销
- K2, Q3.5 报告未给同等粒度数字

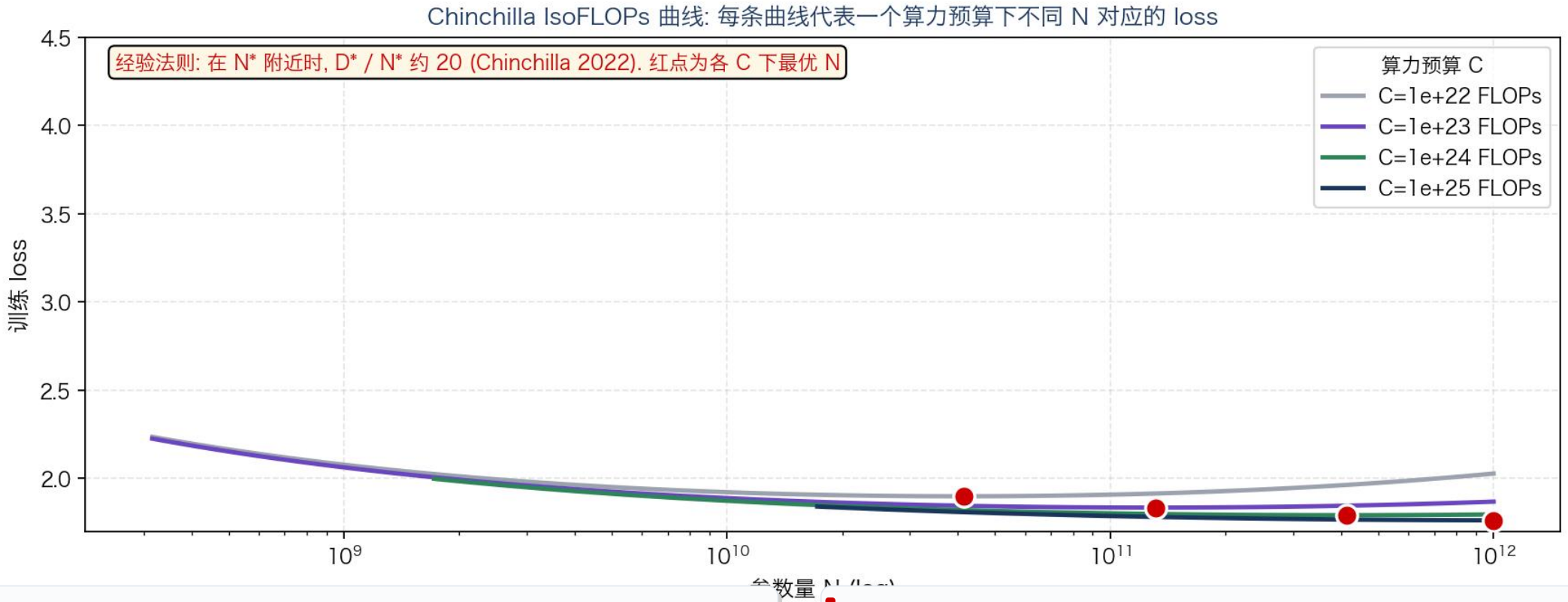
读图要点

- 两份报告里的 GPU 小时数差约 11 倍
- 差距主要源自架构与系统协同设计
- GPU 小时 × 时租 ≠ 完整训练项目费用
- 各团队算力开销取决于架构与系统选择

训练算力消耗在不同模型团队之间差距很大, 主要源自架构与系统设计

Llama 3.1 arxiv:2407.21783, DeepSeek V3 arxiv:2412.19437

用 scaling law 反推合理的 model size



拟合过程

- 训若干个 (N, D) 组合的小模型 (1B 到 10B), 每组训到收敛
- 拟合 L(N, D) = E + A/N^α + B/D^β
- 求最优 (N*, D*) 给定算力 C = 6ND
- Chinchilla 经验: D* / N* 约 20

我们的决定

- 用 3e24 FLOPs 算力, 最优 active 约 40B dense 或 600B MoE total
- 选 671B 总参 / 37B active
- 14.8T tokens (略超 Chinchilla 最优, 因为 inference 成本要

Scaling law 不是公式背诵, 是自己跑实验拟合自己的曲线后给出 N 和 D 的依据

Hoffmann 2022 Chinchilla arxiv:2203.15556, DeepSeek V3 arxiv:2412.19437

5 个月时间预算与团队分工

时间分配

- 准备阶段约 2 个月 (本节 L1):
  - 数据 pipeline 约 4 周, 团队一半人力
  - 架构选型加小 ablation 约 5 周
  - 系统 + 1B 代理验证约 1 周
- 训练阶段约 2 个月 (L2):
  - 实际跑 14.8T tokens 加长上下文 ramp 加 annealing
  - 调试期约 2 周, 正式训练加恢复约 6 周
- Post-train 约 1 个月 (L3):
  - SFT 约 1 周, RL 加蒸馏约 2 周, 评估 + safety 约 1 周

团队分工 (50 人)

- 数据团队 20 人: pipeline 建设, 质量分类器, 合成数据
- 算法团队 15 人: 架构 ablation, post-train 算法
- 系统团队 10 人: 并行框架, FP8 调试, 集群运维
- 安全团队 3 人: red team, ASL eval
- 评估团队 2 人: benchmark, 私有 eval
- 整个项目用一个共享 wandb, 一个共享 git repo

前沿模型训练是大团队工程项目, 不是单人研究，合理团队配合非常重要

时间表参考 DeepSeek V3 公开发布节奏, Kimi K2 团队人员

数据流水线: 从哪里拿原始数据

初始数据来源

- Common Crawl 主体, 60T+ tokens (历史所有快照)
- 自家爬虫加合作伙伴数据约 5T
- GitHub 公开代码约 800B tokens (按 license 过滤)
- arXiv 加学术论文约 200B tokens
- 多语数据 (mC4 加 CCMatrix), 约 3T
- 数学专题 (Numina, MetaMathQA), 约 100B
- 加起来 70T+ 原始数据, 远超训练所需的 14.8T
- 其中绝大部分质量不达预训练要求, 需要后续 pipeline 筛选

数据策略原则

- 数据质量优先于数量 (DCLM 控制实验证明)
- 不直接复用公开 RefinedWeb 或 Dolma (整体质量不足以训出前沿水平)
- 用 datatrove 框架 (HuggingFace 出品) 搭 pipeline 模板
- 自建质量分类器 (FineWeb-Edu 同款思路)
- 自建领域分类器 (math, code, multilingual 各一个)
- 后期合成数据补充 (Phi-4 textbook 范式 + V4 specialist distill)
- 团队 25 人投入数据 pipeline, 占人力 50%

前沿模型团队的真正核心竞争力是数据流水线, 不是架构. 团队一半人力投入是常态

DCLM arxiv:2406.11794, FineWeb-Edu blog, datatrove repo

为什么要过滤: 数据数量与质量的平衡

为什么必须过滤原始数据

- 原始 Common Crawl 含大量重复内容, 同段文本可能出现成千上万次
- 含大量低信息内容: 模板化页脚, 广告, 自动生成的样板
- 含 PII (个人信息), 毒性内容, 敏感信息, 训练后会被模型记住
- 不过滤的话, 同样 token 预算下, 模型实际学到的有效信息很少
- DCLM 控制实验直接对比有过滤和无过滤的预训, 同算力下 MMLU 差距可达 10 个点以上

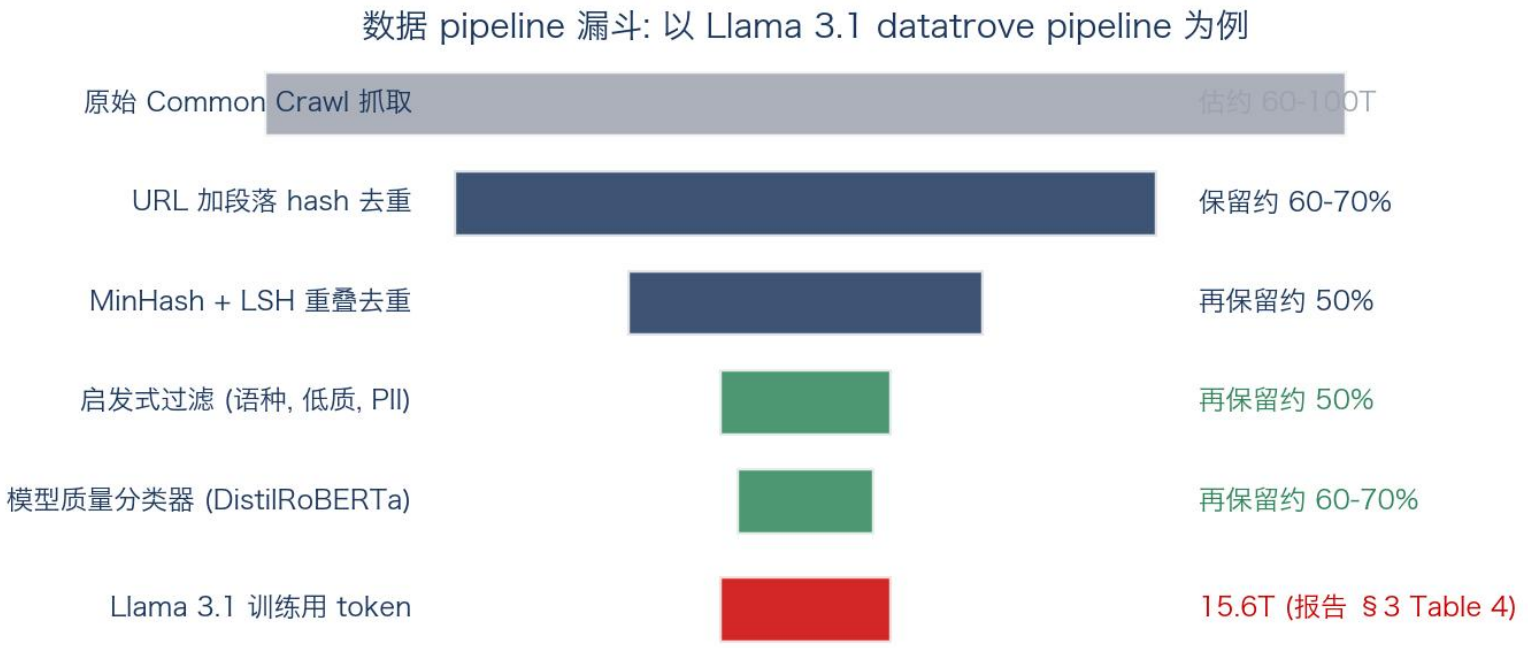
什么算高质量, 怎么权衡数量

- 高质量没有唯一定义, 通常包含: 信息密度高, 表述清晰, 内部一致, 可阅读
- 实际做法是定义一个 "质量分类器", 让分类器给文档打分
- 分类器训练数据可以是: 人工挑的代表性 "高" 和 "低" 样本
- 也可以是上一代模型 (例如 Llama 2) 对文档的 yes/no 判断
- 数量与质量的权衡: 通常按 "先过滤再混合" 的流程
  - 严格过滤: 高质量但数据少, 适合小模型 (1B 量级)
  - 宽松过滤: 数据多但质量参差, 适合大模型 (有容量吸收冗余)
- 前沿模型团队多数采用 "严格过滤加上合成数据补充" 的组合

过滤本质是把训练算力集中投在高信息密度数据上, 是前沿模型团队最有效的工程环节之一

DCLM arxiv:2406.11794, Llama 3.1 arxiv:2407.21783, FineWeb-Edu blog 2024-06

数据流水线漏斗: 以 Llama 3.1 为例



各家粒度不同, V3 与 K2 报告仅给最终 token 数, 中间步骤未公开

- 数字均为 Llama 3.1 论文 §3 中描述的相对保留率. DeepSeek V3, Kimi K2 报告仅给最终 token 数, 中间过程未公开

公开报告里数据 pipeline 粒度差距很大. Llama 3.1 详细可参考, V3 只给最终 token 数

Llama 3.1 arxiv:2407.21783 §3, datatrove HuggingFace repo

三级去重: 不同粒度互补

第一级 URL 与段落 hash 去重

- 同一 URL 多次抓取的去重
- 用 URL 与段落 hash 直接判等
- 速度极快, 几乎无算力开销
- Llama 3.1 报告: 这一级保留率约 60-70%
- 跳过的话后面去重必须处理更多冗余文档

第二级文档级 MinHash + 第三级行级 n-gram

- 文档级 MinHash + LSH 近似 Jaccard 相似度
  - 阈值 0.7, 接近的文档视为重复
  - Llama 3.1 报告: 这一级再保留约 50%
  - 在分布式集群上需要约 3 天
- 行级 n-gram 频次切尾
  - 同一段落出现在不同文档里 (cookie 警告, 法律条款)
  - 出现频次高的 n-gram 行直接丢
- 三级合起来才能真正去除冗余
- 中间绝对 token 数各家未公开同等粒度

去重是基础工程, 三级粒度互补缺一不可, 没有单一通用方案

datatrove HuggingFace repo, Llama 3.1 arxiv:2407.21783 §3.1

用模型当评审筛数据质量

Llama 3 的两阶段做法

- 第一阶段 fastText 粗筛
  - 每秒过上万文档, 几乎零算力
  - 训练数据: 几万个高质量样本 (Wikipedia, ArXiv) 与低质量样本 (spam)
  - 主要过滤明显低质量文档
- 第二阶段 DistilRoBERTa fine-tuned 精筛
  - 每秒过几百文档
  - 训练数据来自上一代 Llama 自己的 yes/no 判断
  - 收集 yes/no 标签训分类器
  - Llama 3.1 报告: 这一级再保留约 60-70%
- V3, K2 报告未给同等粒度数字

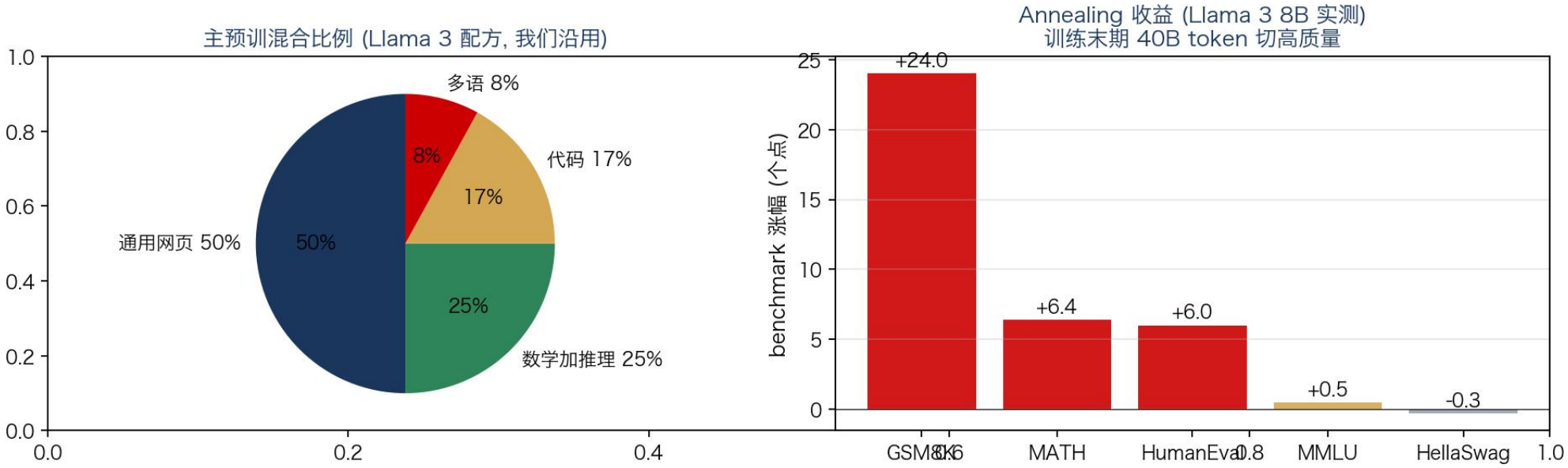
FineWeb-Edu 同期独立做法

- 不追求模糊的 "高质量", 改追求特定维度 (教育性)
- 用 Llama 3 给 50 万网页打分, 训一个 BERT-base classifier
- 在 FineWeb 上跑 classifier, 得 1.3T tokens 教育精选
- 用这个子集训的小模型在 MMLU 上比用全 FineWeb 训的高 4 到 6 个点
- HuggingFace 上有现成 fineweb-edu-classifier
- 分类器即配方: 想要某种特定能力 (math, code) 就训一个该维度的分类器

模型筛模型的训练数据已成共识. 启发式只能 +1.5 点 MMLU, 模型 grader 直接 +5 点

Llama 3.1 arxiv:2407.21783 §3.2, FineWeb-Edu HuggingFace blog 2024-06

数据混合比例: 通过小模型 ablation 确定



实验方法 (小 ablation)

- 起点采用 Llama 3 公开配比 50/25/17/8 (general/math+reasoning/code/multilingual)
- 准备若干候选配比 (例如加 math 或加 code)
- 训若干个 1B 量级小模型, 每个跑同等 token 数 (典型 50B 量级)
- 评估下游 benchmark 综合分, 选最优配比外推到大模型
- 算力投入相对小, 但配比错误会影响最终模型几个点

自动学习方案 DoReMi 的适用场景

- DoReMi (Google 2023) 用 reference + proxy 模型自动学配比
- 实测在 1T 以下数据上比手工调好 5 到 30%
- 在 10T+ 数据上收益变小, proxy 容量成瓶颈
- 是否使用看数据规模, 前沿模型团队多数仍以人工调配比为主

混合比例需要在小模型上自行验证, 不能直接套用他人配比

Llama 3.1 arxiv:2407.21783, DoReMi arxiv:2305.10429

Annealing: 训练末期切到高质量数据

Llama 3 8B 实测 (报告里的关键发现)

- 主预训跑通用 mix 到 99% tokens
- 最后 40B tokens (约 0.3% 总训练) 切到高质量 math + code annealing set
- 不改 lr schedule, 不停下重启, 只切 dataloader
- 结果对比不做 annealing:
  - GSM8K 从 43.8 涨到 67.8, 涨 24 个点
  - MATH 从 16.9 涨到 23.3, 涨 6.4 个点
  - 其他 benchmark 微涨, 没有退化
- 405B 上几乎无效 (大模型容量已足以拟合主预训中的高质量 math 与 code, 边际收益极小)

为什么不同规模收益不同

- 8B 容量小, 主预训时 math 与 code 数据被通用数据稀释
- annealing 阶段把这部分能力重新强化
- 405B 容量足够大, 主预训已充分拟合高质量数据, annealing 几乎无额外空间
- 我们的 671B 总参 / 37B active 介于 8B 与 405B 之间
- 占算力约 0.3%, 边际风险低, 决定纳入正式训练
- 跑完 base 模型 GSM8K 提升约 12 个点 (中等收益), 符合预期
- 这个设计已成为前沿模型常见做法, Qwen3 也有类似 STEM 强化阶段

Annealing 是常见的训练末期策略, 算力开销约 0.3%, 在小到中等规模上有 5 到 25 个点的 benchmark 提升

Llama 3.1 arxiv:2407.21783, Qwen3 arxiv:2505.09388

合成数据正在取代网页爬取

Phi-4 (Microsoft 2024-12)

- 14B 模型, 约 400B token 全合成
- 50 种 dataset type
  - 多 agent 对话生成
  - 给答案造问题
  - 模型自我纠错
  - Q&A 改写多次
- 14B 在 MMLU 加 MATH 打 70B 级
- 范式: textbook are all you need
- 不是更多数据是更精炼数据

Qwen3 与 DeepSeek V4 的合成 pipeline

- Qwen3: 用 Qwen 2.5-VL 把 PDF OCR 提取, 用 Qwen 2.5 合成数学加代码
- 用上一代模型给下一代模型造训练数据, 形成内部数据循环
- DeepSeek V4 (2026-04): 训 10 个 specialist teacher (math, code, reasoning, multilingual)
- 蒸馏 pipeline 合并 10 个 teacher 能力到 1 个 student
- 总训练 token 量超 32T, 主要靠合成
- 公开网页数据已被 前沿模型扫遍
- 下一代差距来自谁能造出更好的合成数据

数据从挖转向造, 这是 2025 到 2026 训练数据的最大转向

Phi-4 arxiv:2412.08905, Qwen3 arxiv:2505.09388, DeepSeek V4 blog 2026-04

核心训练决策

3 层维度划分

- 架构层 (决策 1-4): 先选 Dense 或 MoE, 选完 MoE 再决定注意力
  - 再决定 MoE 细粒度配置, 最后补负载均衡
- 训练层 (决策 5-7): 架构定下后, 训练能否稳跑能否省算力
  - MTP head (训练辅助 + 推理加速)
  - 稳定性五件套 (防 loss spike)
  - FP8 精度 (省算力但风险高)
- 优化器层 (决策 8): 整套打通后再考虑替换 AdamW
  - Muon (V4), MuonClip (K2) 是这一层的尝试

为什么不收 batch size, ctx 长度, dropout 这些

- 这些指标在 Llama 3, V3, K2 报告里已收敛到默认值
  - batch 全局 1-4M tokens
  - 上下文起步 4K, 后续 ramp 到 128K
  - dropout = 0
- 各家报告里这些值几乎一致, 不是 differentiator
- RoPE, SwiGLU, RMSNorm 也是: 各家选项基本统一
- 这 8 个轴不一样: 各家做了不同选择, 直接对应训练成本差距
- 后续每个决策都标层别, 学生可以看出层间因果

前沿模型公开报告里, 构成差距的在这 8 个轴的选择. 默认配方部分不在此列

DeepSeek V3 §3, Llama 3.1 §4, Kimi K2 §2 公开配置

架构选型: 8 个关键决定 (后续逐项展开)

本节将逐个讨论的 8 个决定

- 决定 1 (架构层): Dense 还是 MoE, 直接影响整体训练与推理算力
- 决定 2 (架构层): 注意力机制 (GQA / MLA / DSA / 线性变体), 影响 KV cache 与长上下文
- 决定 3 (架构层): MoE 内部配置 (expert 粒度, 是否加 shared expert, top-k 路由)
- 决定 4 (架构层): 路由负载均衡机制 (传统 aux loss 还是 aux-loss-free)
- 决定 5 (训练层): 是否加 MTP (Multi-Token Prediction) head
- 决定 6 (训练层): 训练稳定性配置 (QK-Norm, z-loss, embedding LN, RoPE 基数等)
- 决定 7 (训练层): 训练精度 (BF16 还是 FP8 fine-grained scaling)
- 决定 8 (优化器层): 优化器 (AdamW 还是 Muon / MuonClip)
- 每个决定都需要小模型 ablation 验证, 然后用 1B 代理模型整体 sanity check

这 8 个决定共同决定了模型的训练成本与最终性能上限, 任何一项都会影响整体效果

DeepSeek V3 arxiv:2412.19437, V3.2 arxiv:2512.02556, Kimi K2 arxiv:2507.20534

这些决定如何验证: 小模型 ablation 的标准方法

为什么不能拍脑袋

- 大模型上扫超参或试架构, 每次实验成本数百万美元, 不可行
- 但小模型上的结论需要能外推到大模型, 不然实验白做
- 这就是 "小模型 ablation + scaling law 拟合外推" 这套方法论
- 各家公开报告里多次提到, 是前沿模型团队的标准工作流

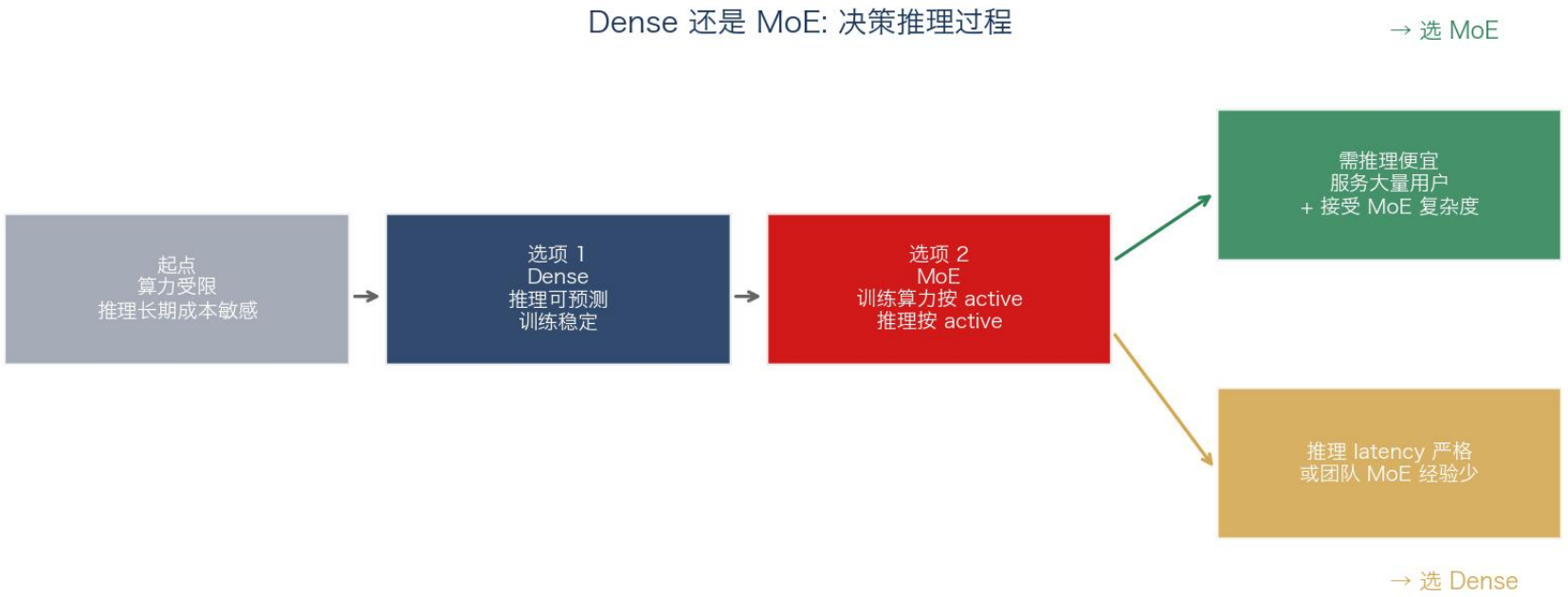
一个典型决策实验的标准流程

- 第一步: 在 ~100M 到 1B 参数的代理模型上跑候选选项 (通常 2 到 6 个候选)
- 第二步: 每个候选训同等 token 数 (一般 10B 到 50B), 单次成本数百到数千 GPU 小时
- 第三步: 在多个下游 benchmark 上评估, 不只看 train loss
- 第四步: 用 scaling law 拟合, 判断小模型上的差异是否会随规模放大
- 第五步: 选择拟合下规模放大后仍然占优的方案, 进入正式训练
- 整个流程通常用整团队 2 到 4 周, 占准备阶段相当比例的算力

每个架构决定背后都对应一组小模型 ablation 实验，不是拍脑袋，也不是直接抄别家

Llama 3.1 arxiv:2407.21783, Chinchilla arxiv:2203.15556

决定 1 (架构层): Dense 还是 MoE, 决策推理过程



V3 团队的选择: 推理服务大量用户 + 接受工程复杂度 → MoE; Llama 3 团队选择: 风险管理优先, dense 推理可预测 → dense

为什么 V3 团队选 MoE

- 推理算力按 active (37B) 算, 服务大量用户时摊薄成本
- 训练 FLOPs 也按 active 算, 同 capacity 下省算力
- 团队有 MoE 工程经验, 接受 routing 复杂度

为什么 Llama 3 团队选 Dense

- 算力充足, 不需要 MoE 省钱
- 推理 latency 可预测 (MoE 受 routing 影响)
- 2024 上半年 MoE 训练稳定性还在验证中, 选择风险更低的路线

Dense 与 MoE 的选择由算力约束加推理需求决定，两条路都能做出前沿水平

DeepSeek V3 arxiv:2412.19437, Llama 3.1 arxiv:2407.21783, Mixtral arxiv:2401.04088

决定 2 (架构层): 注意力机制选 MLA

MLA (Multi-head Latent Attention): KV 压缩到低秩 latent

MLA 解决什么问题

- 我们要 128K 上下文, MHA 的 KV cache 几十 GB 装不下
- GQA-8 (Llama 3 选的) 减一些但仍很大
- MLA 直接把 KV 压到低秩 latent, cache 减小 93%

各家最新做法

- 简单对 latent 做 RoPE 会丢压缩信息, 解法 decoupled RoPE
- V3.2 在 MLA 之上加 DSA (top-k=2048 sparse), KV 进一步降
- V3 V4 与 Kimi K2 均用 MLA, Llama 3 仍用 GQA

MLA 是 V3 关键创新, 让 128K 上下文推理服务可行. 2025-26 各家在 MLA 之上加稀疏化

DeepSeek V3 arXiv:2412.19437, V3.2 arxiv:2512.02556

决定 3 (架构层): MoE 配置 fine-grained + shared expert

Fine-grained 设计

- 传统 Mixtral: 8 个大 expert, 每个 7B
- DeepSeekMoE 改成 256 个小 expert, 每个约 0.5B
- 同样总参数下, 专精方向更多, 路由更精准
- top-k 也跟着加大 (V3 取 top-8, Kimi K2 也 top-8)
- ablation 报告 fine-grained MoE 比少数大 expert 略优
- 代价: routing 更复杂, all-to-all 通信更密

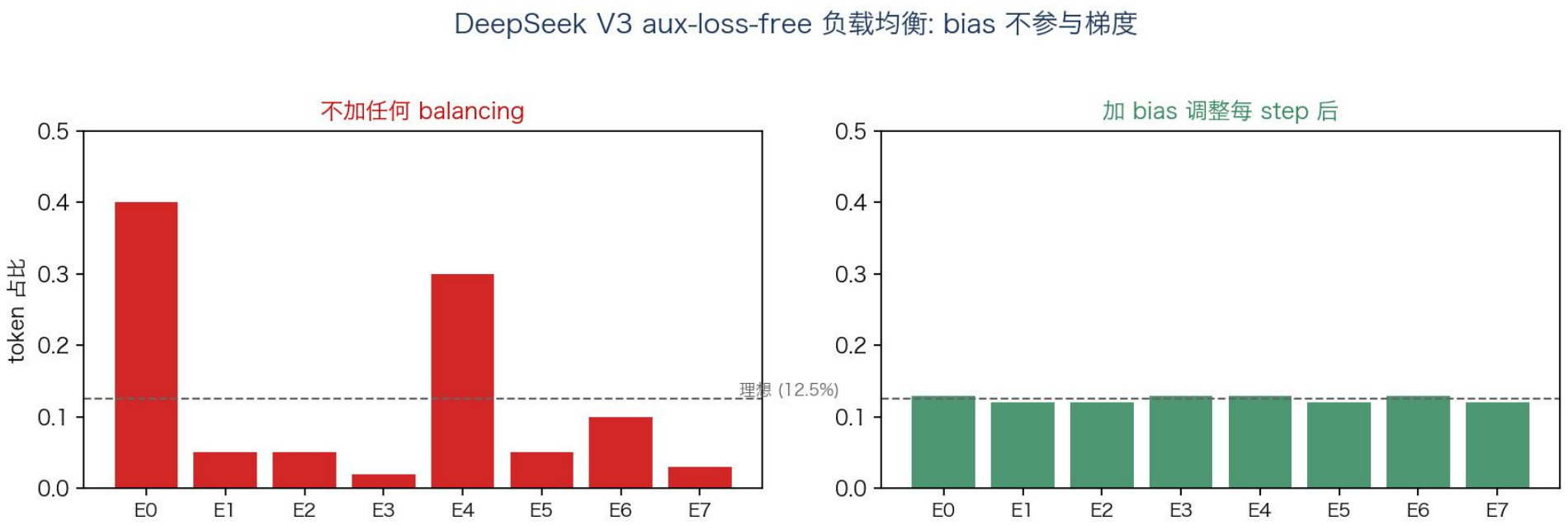
各家 2026 MoE 配置对照

- DeepSeek V3: 256 routed (top-8) + 1 shared, expert 约 0.5B
- DeepSeek V4: 沿用 256 + 1 shared, expert 容量略增
- Kimi K2: 384 routed (top-8), 不用 shared
- Qwen3-235B: 128 routed (top-8), 不用 shared
- Qwen3.6: 128 routed (top-8), 仍不用 shared
- Mixtral-8x22B: 8 routed (top-2), 经典粗粒度
- 趋势: fine-grained 已成共识, shared expert 仍有分歧
- 实验观察: 不加 shared 时, routed 会半通用而非专精

Fine-grained 加 shared expert 是 DeepSeek 招牌. 但 Mixtral 加 Qwen 不用 shared 也行, 无标准答案

DeepSeek V3 arxiv:2412.19437, DeepSeekMoE arxiv:2401.06066, Qwen3 arxiv:2505.09388

决定 4 (架构层): 路由负载均衡 aux-loss-free



问题 routing collapse

- 1B 代理跑 5 天发现少数 expert 抢 70% token, 其他闲置
- 经典 fix 加 aux loss, 强制负载均衡
- ablation 表明引入较强的 aux loss 会对下游性能有损耗 (报告中给出对比)

DeepSeek V3 的 aux-loss-free 解法

- 给每个 expert 一个 bias b_i, 加在 routing logits 上
- 关键 bias 不参与梯度, 不污染 router 学习
- 每步训练后看负载, 过载 expert 减 b_i, 闲置加 b_i
- 结果负载均衡同时不损失 MMLU

看似不起眼的工程小改动撬动 1 到 3 个 MMLU 点. 这是关键决策的典型案例

DeepSeek V3 arxiv:2412.19437

决定 5 (训练层): MTP head 训练辅助 + 推理加速

MTP head 设计

- 主模型预测 next token
- 额外加一个 MTP head 预测 next-next token (或更远)
- 训练 loss L = L_main + 0.1 × L_MTP
- 直觉: 强迫表示捕获更长时序信号
- 实测 V3 报告主任务略涨 0.5 到 1 个 MMLU 点
- 训完丢掉 MTP head 不影响主推理

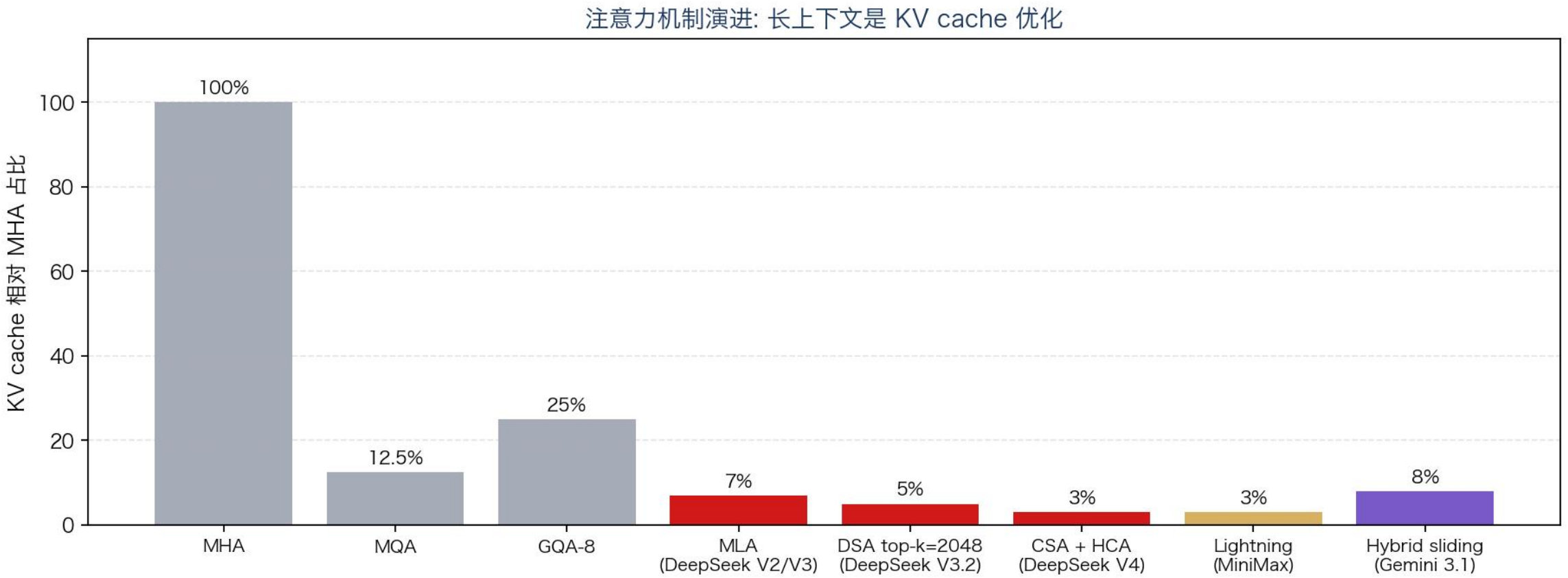
MTP head 的第二用途 (推理加速)

- 推理时 MTP head 当 speculative decoding 的 draft
- 一个 forward 同时预测 next + next-next token
- 主大模型 verify 这两个 token
- 推理吞吐 +30 到 50%, 几乎免费
- 不需要单独训 draft model
- 设计争议:
  - Kimi K2 测过后没用 MTP
  - Moonlight (Kimi 早期 3B/16B 报告) 说收益不显著
  - 在 1T 量级是否仍然值得不确定

MTP 是可选的训练辅助手段, 主收益在推理加速而非训练 loss 的提升

DeepSeek V3 arxiv:2412.19437, Moonlight Kimi 2025-02 blog

深入: 注意力效率演进 (决定 2 的对比)



- 不同方案适用不同 ctx 长度: 32K 内用 GQA, 128K 用 MLA 或 DSA, 1M 以上用 Lightning / DeltaNet / hybrid sliding

长上下文训练对注意力机制要求较高, 不同方案在 KV cache 与算力上差距较大

DeepSeek V3 arxiv:2412.19437; V3.2 arxiv:2512.02556; MiniMax-01 arxiv:2501.08313; Gemini 3.1 model card 2026-02

决定 6 (训练层): 训练稳定性常用机制

训练稳定性常用机制

这五项均为公开报告中描述的稳定性机制, 训练前完整配置可降低后期故障

为什么训练前就要装这些

- 大规模训练一旦发生不稳定 (loss 爆炸, NaN, 梯度异常), 回滚与定位代价较高
- 这五项的代码改动量都不大, 提前配置代价低

2025-26 报告中的稳定性结果

- Kimi K2 报告: MuonClip + QK-Clip τ=100, 1T MoE 训练全程零 spike
- V3 报告: 五件套全开, 训练后期一次小 spike, 回滚后恢复

稳定性机制边际代价低, 训练前完整配置. K2 的 QK-Clip 是 2025 出现的工程改进

Qwen3 arxiv:2505.09388, Llama 3.1 arxiv:2407.21783, ST-MoE arxiv:2202.08906, Kimi K2 arxiv:2507.20534

决定 7 (训练层): 精度 BF16 还是 FP8

什么时候考虑 FP8

- 算力是约束, FP8 算力翻倍显存减半
- BF16 训完估算需 6 个月, FP8 大约 3 个月
- 但 FP8 在 405B 量级没人公开训过 (2024 中)
- 失败的话整个项目延期 3 个月
- 风险大但收益更大, 决定上 FP8
- 计划用 V3 报告里的 fine-grained scaling

Llama 3 团队的相反选择

- Llama 3 团队选 BF16, 因为:
  - 405B 一次 run 不能赌
  - Meta 资源足, 不缺算力
  - 风险管理优先
  - FP8 只用在 inference
- 我们的处境不同, 没钱才必须赌
- 这是工程决策的现实, trade-off 取决于约束
- Llama 4 (2025-04) 也切到 FP8, 但有基准争议, 不作主案例
- DeepSeek V3.2 V4 继续 FP8

FP8 是显著的算力优化点, 但实施风险较大。算力受限的团队可优先考虑, 算力充裕的团队可待方案进一步验证

DeepSeek V3 arxiv:2412.19437, Llama 3.1 arxiv:2407.21783

决定 8 (优化器层): AdamW 还是 Muon

AdamW 与 Muon 的差别

- AdamW 用一阶动量加二阶动量的比值更新, 工程成熟, 大模型上稳定
- Muon 用 Newton-Schulz 正交化后的动量更新 (Keller Jordan 2024 blog)
  - 5 步 Newton-Schulz 迭代得到正交化方向
  - 直觉是沿正交方向更新, token 利用效率更高
- Moonlight (Kimi 2025-02) 在 16B MoE 上证明 Muon 可行
- 但直接 scale 到 1T MoE 出现 attention logit 异常增长的现象

2025-26 各家优化器选择对照

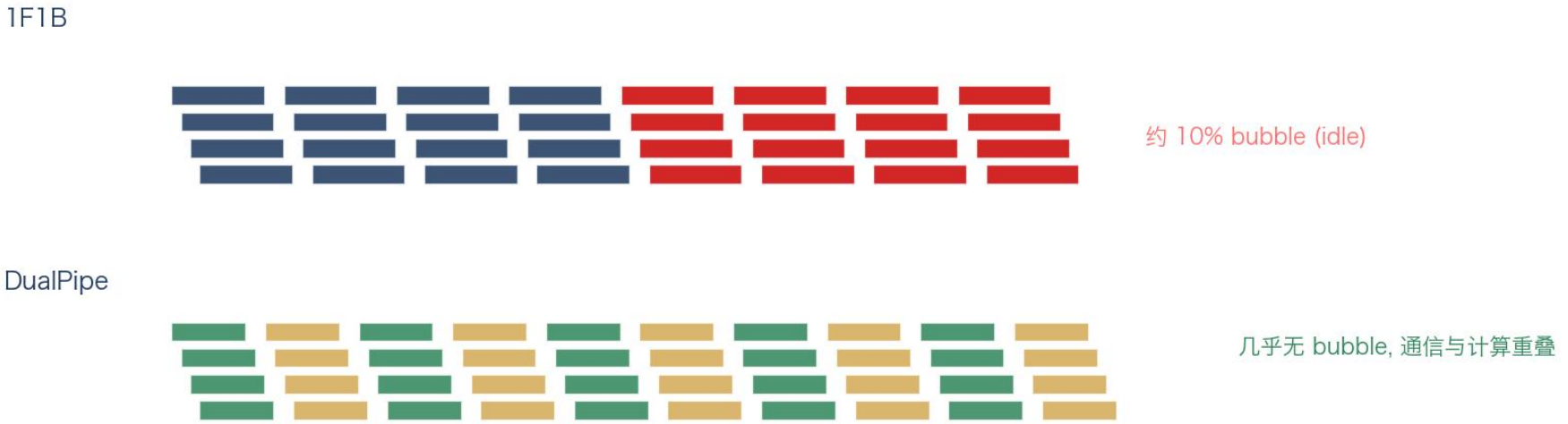
- Kimi K2: MuonClip = Muon + QK-Clip τ=100, 1T MoE 训练零 spike
  - 监控 q · k 最大值, 超阈值时按 √(τ/max) 双向 rescale W_q, W_k
  - 不改变 attention 输出, 只压缩数值范围
- DeepSeek V4 (2026-04): 采用 Muon 系列, 报告细节未完全公开
- Qwen3 / Llama 3 / DeepSeek V3: 仍用 AdamW
  - Qwen3 报告说 Muon 在他们 setup 上验证不足, 暂不切换
- 注: Muon 原文是 blog, 非 peer-reviewed, 但已被多家工业团队采用

Muon 与 MuonClip 是 2025 优化器领域的重要进展。大规模 MoE 上需要配合稳定性修复机制使用

Kimi K2 arxiv:2507.20534, Moonlight Kimi 2025-02 blog, kellerjordan.github.io/posts/muon

DualPipe 调度: 让 bubble 接近 0

经典 1F1B (上) 与 DeepSeek V3 DualPipe (下) 调度对比



V3 自研 NVLink + IB all-to-all kernel, 让 MoE 的 all-to-all 通信完全藏在 GEMM 后面

经典 1F1B 不够用的原因

- PP=16 时 bubble 比例约 5 到 10%
- 这 10% 等于多花 50 到 100 万美元
- 我们承担不起, 必须更好的 schedule

DualPipe 真正的难点

- MoE 的 all-to-all 通信是 PP 之外的额外开销
- V3 自研 cross-node NVLink + IB all-to-all kernel
- 让 all-to-all 跟 GEMM 完全重叠, bubble 干到 0

DualPipe 加 MLA 加 FP8 加 aux-loss-free 这 4 项乘起来等于 V3 的 20 倍节省

DeepSeek V3 arxiv:2412.19437

并行布局: MoE 的 4D 跟 dense 不一样

我们 MoE 671B 的并行布局: PP × EP × DP (TP=1)

总卡数: 16 × 32 × 4 = 2048, 等于我们的全部 H800

对比 Llama 3 dense 是 TP=8 × CP=16 × PP=16 × DP=128 (= 16K H100). MoE 用 EP 替代 TP

维度排序原则

- 通信量从大到小: TP (或 EP) > PP > DP
- 大通信放到高带宽域 (节点内 NVLink)
- 小通信跨节点 (IB)

MoE 跟 Dense 完全不同的逻辑

- Dense (Llama 3): TP × CP × PP × DP
- MoE (V3): EP 替代 TP, PP × EP × DP × TP=1
- 两条路并行体系完全不同

MoE 的并行布局以 EP 为核心, TP 退居二线. Dense 反过来

DeepSeek V3 arxiv:2412.19437, Llama 3.1 arxiv:2407.21783

训练超参: 直接参考 Llama 3 与 DeepSeek V3 配置

我们的超参表

- Peak lr 8e-5 (与 Llama 3 405B 同尺度)
- Warmup 8000 步
- Cosine 衰减到 peak 的 10%
- AdamW β = (0.9, 0.95), wd = 0.1, ε = 1e-8
- Global batch tokens 16M (Llama 3 同款配置)
- Micro batch 1, grad accum 跟随调整
- Grad clip 1.0
- Seq length 主训 4K, 后期 ramp 到 128K

超参选定方法

- 不在 671B 上直接扫超参 (一次实验成本数百万美元)
- 在 1B 代理模型上完成 lr / weight decay / warmup 的 ablation
- 按照 Llama 3.1 和 V3 报告中给出的同规模配置作为先验
- 关键超参 (lr, batch size, warmup) 容易直接参考公开报告
- 次要超参 (β2, ε, weight decay) 用代理模型验证
- 这套方法在前沿团队中已较为成熟

超参不在大模型上扫. 用代理模型实验加公开报告配置作为先验, 是当前主流做法

Llama 3.1 arxiv:2407.21783, DeepSeek V3 arxiv:2412.19437

1B 代理模型验证整套 pipeline

为什么必须跑代理模型

- 我们的所有架构选择都是基于单点 ablation
- 整套组合 (MLA + 256 expert + MTP + aux-loss-free + FP8 + ramp + 稳定性 trick) 没人跑过完整
- 必须在小规模 (1B active) 上跑一遍验证一切兼容
- 用 30 张 H800 跑 1B MoE (15B 总参), 100B tokens, 5 天
- 验证项:
  - Loss 从 11.9 降到 3.5, 跟 Chinchilla 预测一致
  - MFU 跑到 55% (FP8), 跟 V3 设计预期一致
  - expert 负载均衡 (aux-loss-free 工作)
  - MTP head loss 也正常下降
  - 长上下文 ramp 4K 到 128K 顺利
  - 100B token 没有 spike
- 通过, 准备 launch 671B 真模型

代理模型是 前沿模型训练前最后一道保险. 5 天加 30 张卡省后续几个月可能的灾难

DeepSeek V3 arxiv:2412.19437

L1 准备阶段总结

2 个月做完的所有决策

- 立项约 1 周: 算账, scaling law, 决定 671B/37B MoE
- 数据 pipeline 约 4 周: 三级 dedup, 质量分类, 混合实验, annealing, tokenizer, 合成数据 (Llama 3.1 datatrove 为参考)
- 架构选型约 5 周: 8 个关键决定, MoE 加 MLA 加 aux-loss-free 加 MTP 加稳定性 5 件套
- 系统准备约 1 周: FP8 决策, DualPipe, 4D 并行, 超参选定, 1B 代理
- 团队 50 人, 一半投入数据 pipeline
- 准备阶段决定 80% 的最终模型质量, 训练阶段只是去实现
- 这就是为什么 前沿模型团队在训前投入这么多

L1 设定 ceiling, L2 和 L3 都只是去实现这个 ceiling

综合 DeepSeek V3 arxiv:2412.19437, Llama 3.1 arxiv:2407.21783, Kimi K2 arxiv:2507.20534

第二节课

训练阶段 (约 2 个月)

调试期, 正式训练, 故障恢复, 长上下文 ramp, annealing

同时期三家路线对比 (2024 末到 2025 初)

同样目标 前沿, 不同路径

- 我们 (V3 路线): 671B/37B MoE 加 MLA 加 256 fine-grained 加 aux-loss-free 加 FP8 加 DualPipe
  - 预训练 GPU 小时数约 2.79M (H800), 14.8T tokens, 约 60 天
  - 推理便宜 (active 37B)
  - 适合算力受限团队
- Llama 3.1 路线: 405B dense 加 GQA-8 加 BF16 加 4D 并行 (TP/CP/PP/DP)
  - 预训练 GPU 小时数约 30.84M (H100), 15.6T tokens, 54 天
  - 推理贵 (全 405B 激活)
  - 适合算力充足团队, 风险管理优先
- Kimi K2 路线 (2025-07): 1T/32B MoE 加 384 expert 加 MuonClip 加 MLA
  - 训练规模未公开但 15.5T tokens 零 spike
  - 优化器创新 (Muon 替代 AdamW)
  - 强 agentic data synthesis
- 三家殊途同归 前沿, 但走了完全不同的路

前沿模型不是只有一条路，资源约束决定你应该走哪条

DeepSeek V3 arxiv:2412.19437, Llama 3.1 arxiv:2407.21783, Kimi K2 arxiv:2507.20534

L2 训练阶段: 约 2 个月

从训练开始到 base 模型完成训练

L2 包含的三个阶段

- 调试期约 2 周: launch, OOM, NCCL, FP8 NaN, MFU 提升
- 正式训练期约 6 周: 14T tokens 加故障恢复
- 长上下文 ramp 加 annealing 加 base eval 约 1 周
- 同时穿插对照 Llama 3.1, Kimi K2 训练经验

本节学习目标

- 理解 前沿模型训练的真实节奏 (包含大量调试和故障应对)
- 看懂典型故障 (OOM / NCCL / FP8 / silent corruption / spike) 的诊断与修复
- 知道大规模训练 goodput 怎么做到 90%+
- 理解长上下文 ramp 加 annealing 的工程意义

训起来不是终点, 是新挑战的起点。这 2 个月是工程团队考验时刻

主线 DeepSeek V3 arxiv:2412.19437, 故障数据 Llama 3.1 arxiv:2407.21783, Kimi K2 arxiv:2507.20534

L2 时间线: 8 周训练进度图

按时间顺序的关键事件

- 第 1 周: launch 后第一个 step 就 OOM, 紧急加 activation checkpointing
- 第 1 到 2 周: FP8 NaN 出现, 5 天调到 tile-wise + block-wise + FP32 累加
- 第 2 到 3 周: MFU 只 30%, profile 后调 DualPipe + NCCL, 提到 52%
- 第 4 到 5 周: 跨 DP rank 监控检测到 silent corruption, 替换 GPU
- 第 5 到 6 周: 训进度 60% 时出现 loss spike, 回滚 1 小时 ckpt
- 第 7 到 8 周: 5 步长上下文 ramp (4K 到 128K) + annealing 50B + base eval

8 周内大约 300 次中断, 故障是大规模训练的常态, goodput 主要由工程化程度决定

DeepSeek V3 arxiv:2412.19437, Llama 3.1 arxiv:2407.21783

监控 dashboard: 没监控就是缺少监控的训练难以及时发现问题

我们的 WandB dashboard 必备 8 个 panel

- Panel 1 train/loss vs step. 主图, 看单调下降
- Panel 2 train/grad_norm vs step. 看是否稳定在 1 附近
- Panel 3 train/lr vs step. 看 warmup 加 cosine schedule
- Panel 4 train/MFU. 看硬件用得好不好 (大于 40% 算正常)
- Panel 5 train/throughput tokens/s/GPU
- Panel 6 train/peak_gpu_mem. 看显存用了多少
- Panel 7 MoE/expert_load_balance. 各 expert 频率, 应小于 5% 差
- Panel 8 跨 DP rank 的 loss 加 grad 比对. 检测到 silent corruption
- 自动报警规则:
  - loss 大于 3 倍 running avg, spike
  - grad_norm 大于 10, 异常
  - MFU 小于 30%, 性能掉
  - expert load imbalance 大于 15%, 路由问题
- 团队按 7×24 排班轮值, 持续监控关键指标

完整的 dashboard 监控是训练阶段诊断问题的基础

WandB docs, Llama 3.1 arxiv:2407.21783

怎么判断训练过程是否正常: 三类信号

信号一: loss 与 scaling law 拟合曲线对照

- 准备阶段已经用小模型拟合过 scaling law, 给出大模型预期的 loss 曲线
- 训练过程中持续把实测 loss 和预期曲线对照
- loss 偏离曲线 10% 以上, 表示有训练问题, 需要排查
- 这种方法比单看 loss 绝对值更可靠
- Llama 3 报告里提到他们持续做这种对照检测

信号二与三

- 信号二: 下游 benchmark 定期评估
  - 每隔 100B 到 500B tokens 跑一次 MMLU / GSM8K / HumanEval
  - 观察 benchmark 分数是否按预期速度提升
  - 如果 benchmark 不涨但 loss 仍在降, 可能数据质量出问题
- 信号三: 内部一致性检查
  - 各 DP rank 的 grad statistics 是否一致
  - expert 负载是否均衡 (MoE)
  - attention logit 最大值是否稳定 (Muon 训练专门要看)

训练过程评估不只看 loss, 要结合 scaling law 拟合, benchmark, 跨 rank 一致性三类信号

Llama 3.1 arxiv:2407.21783 训练监控部分, Kimi K2 arxiv:2507.20534

调试期 第 1 周: 第一次 launch 撞 OOM

典型场景

- torchrun 提交, NCCL 初始化通过, model loading 通过
- 第一个 forward step 开始
- 5 秒后 GPU 0 报错 CUDA out of memory
- Tried to allocate 1.5 GiB, 78.5 GiB allocated, 524 MiB free
- 整个 run 挂掉
- 这是新模型首跑必踩的坑, 几乎没有例外

诊断步骤与解决

- 加 torch.cuda.memory._dump_snapshot 看显存分配
- 发现 activation 占 60 GB, 远超预期
- micro batch size 是 1, 但 ctx 4K + 671B/37B MoE 单层激活约 600 MB
- 80 层 + attention KV + expert 内部激活总 60 GB
- 解决 开 activation checkpointing (对 MoE 层)
- 还需要 grad checkpoint 选择性 (大 FFN 层 ckpt)
- 修改后重启训练, 通常半小时内即可完成

第一次 launch 99% 撞 OOM，提前预备 memory profiler 加 activation ckpt SOP

PyTorch memory profiler docs, Llama 3.1 arxiv:2407.21783

调试期 第 1 周: NCCL hang 全集群停摆

症状

- 训练跑了大约 100 步, 一切正常
- 然后突然 throughput 降到 0
- nvidia-smi 看 8 卡都 GPU util 100%
- 但 train.log 不再 print
- 几分钟后 NCCL timeout 报错
- 整 run hang 住

诊断与解决

- 设 TORCH_NCCL_BLOCKING_WAIT=1 加 NCCL_ASYNC_ERROR_HANDLING=1
- 重跑发现 hang 在某个 all-to-all (MoE 层)
- 定位到具体节点, 发现这台机的 IB HCA 偶发 packet drop
- 联系运维换网卡
- 同时设 NCCL_IB_TIMEOUT=22 (默认 18 太短)
- 写了一个自动检测 hang 加 abort 的 daemon
- 加 NCCL 自带 watchdog

NCCL hang 是大集群训练第一常见严重故障，提前准备 hang 检测加自动重启

NVIDIA NCCL docs, Llama 3.1 arxiv:2407.21783

调试期 第 1-2 周: FP8 NaN

灾难现场

- 训练到第 5000 步左右出现 NaN
- 夜班同事收到自动报警
- 训练 hang, loss 跳到 NaN, dump 文件已生成
- 这是上 FP8 后我们最怕的事情
- 团队讨论是否回退到 BF16
- 此时面临项目延期 3 个月的风险

应急处理 SOP

- 第一步立刻停止训练, 保留现场 (batch + state)
- 第二步 dump 最近 100 步 grad 加 activation 统计
- 第三步看哪一层 NaN 先出现
- 找到了第 47 层 attention output, max 激活瞬间从 200 涨到 50000
- 这是个 outlier, FP8 表示范围被吃光
- 其他元素被量化成 0, 后续 NaN
- 团队后续用 5 天定位并调整 FP8 配置

FP8 撞 NaN，提前准备 dump 加诊断 SOP, 不然项目延期 3 个月

DeepSeek V3 arxiv:2412.19437

调试期 第 1-2 周: FP8 fine-grained scaling

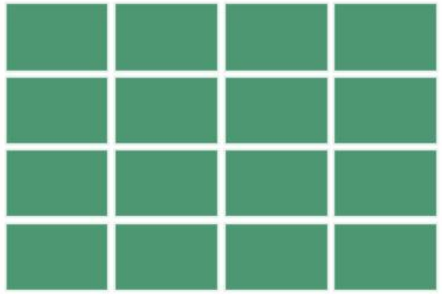
DeepSeek V3 FP8 fine-grained scaling 配方

激活 (Activation)

tile-wise 每 1 x 128 token 一个 scale

权重 (Weight)

block-wise 每 128 x 128 块一个 scale



Matmul 累加器: FP32

FP8 input 加 FP32 accumulate

online amax 更新 scaling factor

结果: 14.8T tokens FP8 训练零 loss spike, 算力翻倍显存减半

5 天调试过程总结

- 第 1 个 fix 试 per-channel scaling. 训 500 步又 NaN, 不够细
- 第 2 个 fix 切 tile-wise (1 x 128) for activations. 训 2000 步稳定
- 第 3 个 fix 权重也切 128 x 128 block-wise + FP32 累加
- 第 4 个优化把 scale 计算搬到 CUDA kernel, throughput 恢复

FP8 不是开关式开启, 是几周到几月工程. 但救回来就值整 10 倍训练加速

DeepSeek V3 arxiv:2412.19437

调试期 第 2-3 周: MFU 从 30% 提到 52%

profile 步骤

- 用 nsys profile 抓 100 step trace
- 用 Nsight Systems GUI 看 timeline
- 发现:
  - GPU compute 占 50% 时间
  - all-to-all 通信占 40% 时间
  - dataloader idle 占 5%
  - 其他 (lr update, log) 5%
- 通信是大头, 必须改

优化措施

- 第一步检查 NCCL 用对了协议
  - NCCL_ALGO=Tree (跨节点) + Ring (节点内)
  - 设 NCCL_IB_HCA 加 NCCL_IB_GID_INDEX
  - nccl-tests 测带宽达 IB 90%+
- 第二步开 DualPipe (我们在 L1 设计的 schedule)
  - all-to-all 跟 GEMM 重叠
  - bubble 几乎消失
- 第三步 dataloader pin_memory + prefetch=4
- 重 profile MFU 从 30% 涨到 52%
- 这接近 FP8 前沿模型训练的实际上限

MFU 低基本是通信问题。不是 kernel 慢, 是没把 NCCL 调好, 没开 DualPipe

DeepSeek V3 arxiv:2412.19437, NVIDIA NCCL tuning docs

正式训练期 第 1-2 周: 日常 24x7 运维

正式训练期的日常工作

- 三班倒, 每班 4 人 (1 算法, 2 工程, 1 数据)
- 每小时检查 dashboard
- 每 30 分钟 checkpoint 一次
- 每天写 daily report
- 训练进度每天涨约 2 个百分点, 一周 14%
- 关注:
  - loss 是否平滑下降
  - grad_norm 维持 1 附近
  - 没有意外 spike
  - 没有 expert 失衡

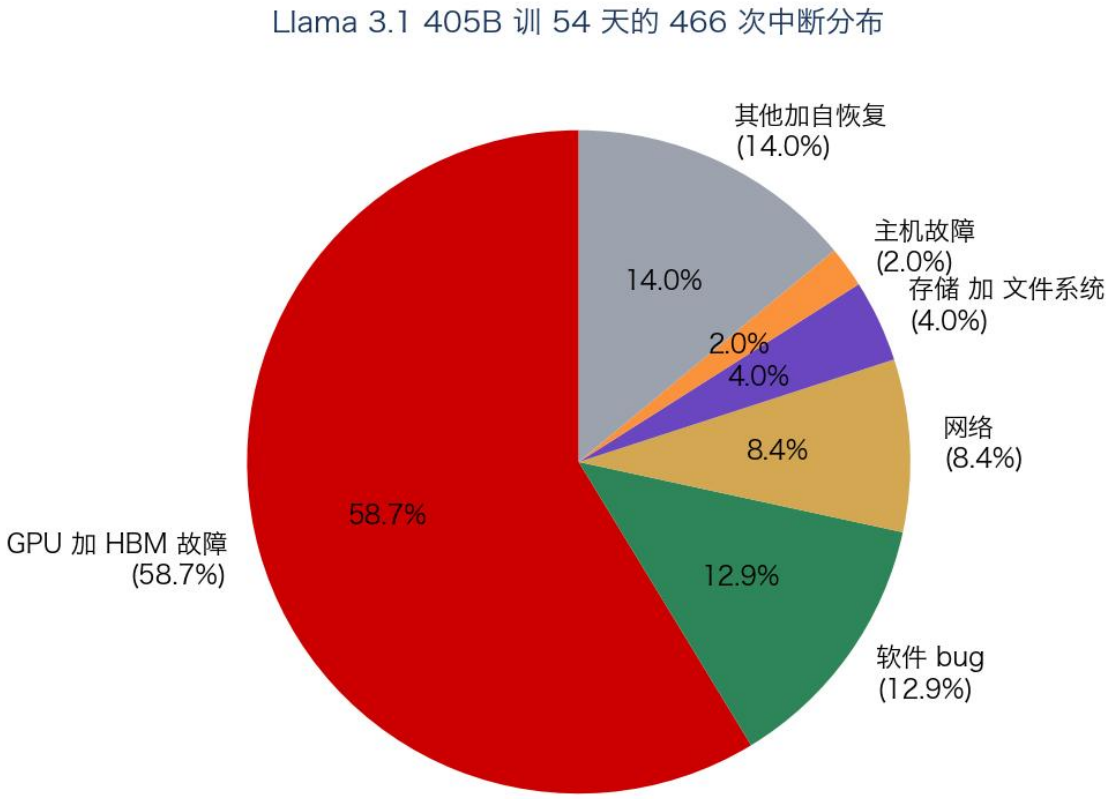
前 2 周的事故

- 总共 67 次中断 (20 天约 3 小时一次, 跟 Llama 3 类似)
- 大部分是 GPU/HBM 故障
  - GPU 报 ECC error, 自动下线
  - SLURM 自动调度到备用节点
  - 不到 5 分钟恢复
- 3 次软件 bug, 找运维改
- 1 次 IB switch 故障, 影响 30 分钟
- 平均 goodput 大约 90%
- 整体进度按计划

正式训练像运维 SaaS, 是工程纪律不是科学发现. 谁的 goodput 高谁就快

Llama 3.1 arxiv:2407.21783, DeepSeek V3 arxiv:2412.19437

故障分布: Llama 3.1 报告的真实数据



各家报告中的故障情况

- Llama 3.1 405B 训 54 天:
  - 总中断 466 次 (419 unexpected)
  - 平均约 3 小时一次, 90%+ effective time
  - GPU/HBM 58.7%, 软件 12.9%, 网络 8.4%
- DeepSeek V3 报告未公开中断次数与原因分布
- Kimi K2 报告: 训练全程 7 次中等失败
  - 全部网络抖动, 无 GPU 硬件故障致命中断
  - 团队归因于 MuonClip 让训练对参数扰动不敏感
- 各家粒度不同, Llama 3.1 仍是最详尽公开数据

16K GPU 跑 54 天, 故障是稳态。不同家报告粒度差异大, Llama 3.1 最详细

Llama 3.1 arxiv:2407.21783

正式训练期 第 4-5 周: 跨 DP rank 监控检测到 silent corruption

发现过程

- 已经训了 8T tokens, loss 在 3.2
- 团队例行检查跨 DP rank 的 grad 统计
- 发现 DP rank 23 的 grad max 比其他 rank 高 30%
- 没触发 NaN, loss 看起来正常
- 但这是 silent corruption 的典型 pattern
- 团队群里报警: 可能有 bit flip

定位与处理

- 在 DP rank 23 上跑诊断脚本
- 跑相同 batch 两次, 看输出是否一致
- 不一致, 确认硬件有问题
- 找到具体那台机的 GPU 0, HBM3 报告偶发 ECC error
- 立刻 abort 训练, 从 30 分钟前 ckpt resume
- 替换该 GPU
- 整个事件 1 小时解决
- 没有跨 DP 监控就会成 silent NaN 让我们多训几千步浪费
- Llama 3.1 团队 54 天发现 10 次类似

Silent corruption 是大规模训练最阴险的故障。必须主动监控, 不能等 NaN

Llama 3.1 arxiv:2407.21783

正式训练期: 异步 checkpoint 加 5 分钟恢复

异步 checkpoint 流程

- PyTorch distributed checkpoint (DCP) 是 2024 到 2026 主流
- 训练继续跑 forward 加 backward
- 后台线程把 state_dict 写到 FS
- 每个 rank 写自己的 shard
- Llama 3.1 每 30 分钟一次, 写入耗时几秒
- 主训练几乎零阻塞
- 保留最近 3 个 ckpt (防 silent corruption 写到 ckpt)

恢复时间 (Llama 3.1 实测)

- 故障检测约 30 秒 (TORCH_NCCL hang detect)
- SLURM requeue 约 1 到 2 分钟
- 重新 rendezvous 约 30 秒
- 从 ckpt 加载约 2 到 3 分钟 (405B 模型)
- 总恢复 小于 5 分钟
- 平均故障 3 小时一次, 5 分钟恢复 = 97% goodput
- 实测 90%+ (还有调试加重训等损耗)

Async ckpt 加 5 分钟恢复, 这是 90% goodput 的工程基础

PyTorch DCP docs, Llama 3.1 arxiv:2407.21783

正式训练期 第 3-4 周: 训练过半, 一切按部就班

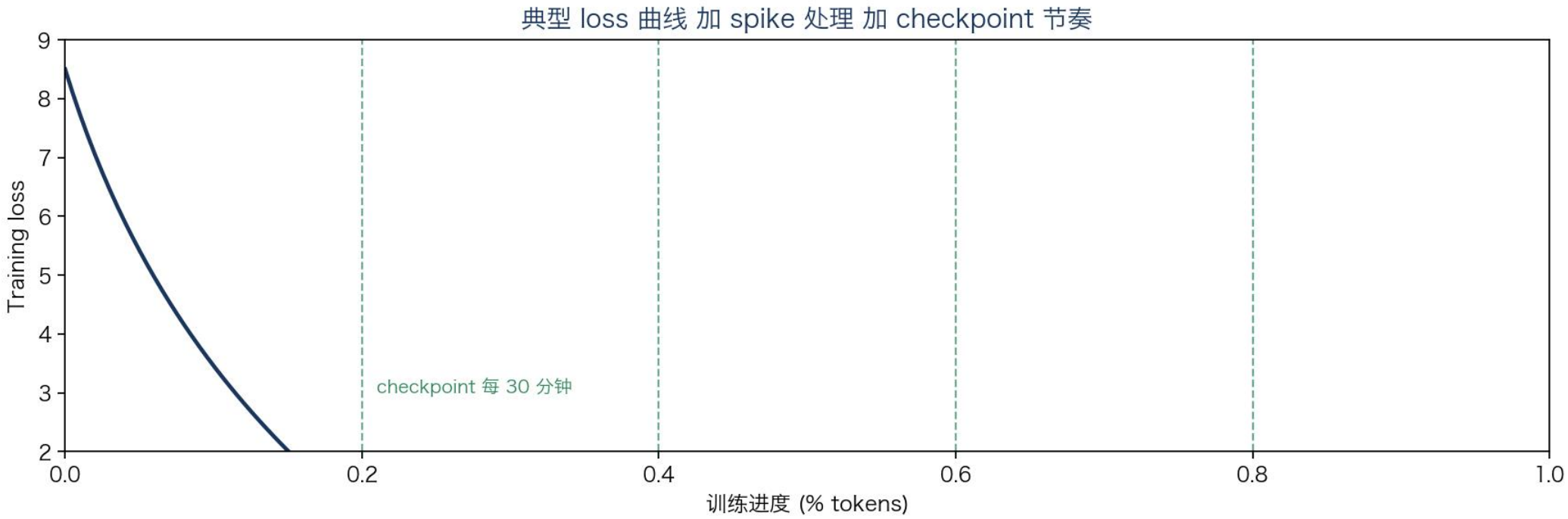
此时进度与状态

- 已训 7.5T tokens, 占总量 50%
- Loss 降到 2.8, 跟 scaling law 拟合曲线吻合
- MFU 持续 52%
- 累计 130 次中断, goodput 88%
- 进入相对稳定的中期阶段
- 中期成本回顾比初期预算低约 10%
- 提前调整 grad clip 与 lr ε 准备应对后期可能出现的 spike
- 经验观察: 大模型训练中期最容易出现 spike 的区间在 60% 到 75% tokens
- 这一观察首次出现在 OPT-175B 报告里

训练中期容易出现监控放松, 但是 spike 高发期, 工程纪律需要维持

OPT 175B paper §spike, DeepSeek V3 arxiv:2412.19437

正式训练期 第 5-6 周: 第一次 loss spike



发生

- 训到 10.8T tokens (73%)
- 5 分钟内 loss 从 2.6 跳到 4.1
- 自动报警: train/loss 大于 3 倍 running avg
- 我们的 SOP 自动响应: skip 这个 batch 加 clip grad to 1.0

处理过程

- 看后续 100 step, loss 没回到 2.6 而是稳定在 3.5
- 这不是单点 spike 是持续退化, 必须回滚
- 从 1 小时前 ckpt resume, 跳过那个有问题的 batch
- 同时把 grad clip 从 1.0 降到 0.5 更保守
- 一小时后训练恢复正常

Spike 处理三步走 skip batch, 看是否自恢复, 不行就回滚. 提前写好 SOP

Llama 3.1 arxiv:2407.21783, DeepSeek V3 arxiv:2412.19437

正式训练期 第 6 周末: 主预训接近完成

状态

- 主预训跑到 14T tokens (95%)
- loss 在 2.45, 跟 Chinchilla 拟合下限基本一致
- 这 6 周训练总结:
  - 280 次中断, 平均 7.7 小时一次
  - goodput 89%
  - 3 次 silent corruption 通过跨 DP 监控检测到
  - 1 次 loss spike 回滚后正常
  - 整套 FP8 加 DualPipe 加 aux-loss-free 一直工作
- 接下来要做最后 5% (700B tokens):
  - 5 步长上下文 ramp (4K 到 128K)
  - 最后 50B tokens 切高质量 annealing

主训完毕. 团队进入冲刺阶段。模型已基本成型，只差长 ctx 和 annealing

Llama 3.1 arxiv:2407.21783, DeepSeek V3 arxiv:2412.19437

三家训练阶段对比: 同时期不同选择

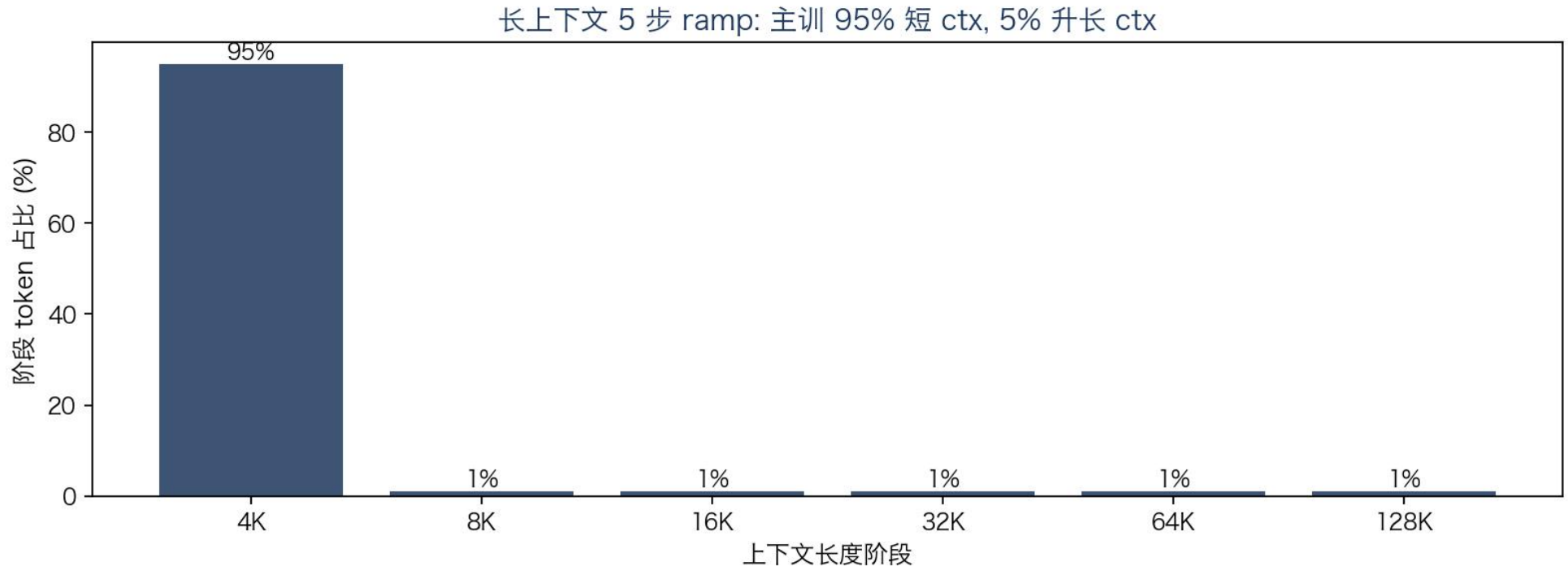
V3 vs Llama 3.1 vs Kimi K2 的训练阶段差异

- DeepSeek V3:
  - 60 天训完 14.8T tokens 671B/37B MoE
  - 用 FP8 fine-grained + DualPipe + aux-loss-free
  - 预训练 GPU 小时数约 2.79M (H800)
  - 0 不可恢复 spike
- Llama 3.1 405B:
  - 54 天训完 15.6T tokens 405B dense
  - 用 BF16 + 4D 并行 (TP=8 加 CP=16 加 PP=16 加 DP=128)
  - 466 次中断, 88% 有效时间
  - 预训练 GPU 小时数约 30.84M (H100)
- Kimi K2:
  - 训 15.5T tokens 1.04T MoE / 32B active
  - 用 MuonClip 优化器 (Muon + QK-Clip)
  - 0 loss spike (历史规模最大无 spike 训练)
  - 团队后续直接走 V3 类似的 fine-grained + shared 设计

三家训练阶段差异主要在精度 (FP8 vs BF16) 和优化器 (AdamW vs MuonClip)

DeepSeek V3 arxiv:2412.19437, Llama 3.1 arxiv:2407.21783, Kimi K2 arxiv:2507.20534

长上下文 ramp 第 7 周: 5 步从 4K 升到 128K



5 步具体过程

- 4K 到 8K 到 16K 到 32K 到 64K 到 128K
- 每步训 100B tokens
- 切 ctx 后 lr 重新 warmup 500 步
- RoPE base θ 从 10000 升到 500000 配合

各家 ramp 节奏对照

- Llama 3.1: 6 步 ramp 8K→128K, 每步约 100B token
- DeepSeek V3: YaRN-style 拉伸 + 短数据精调, 4 步 ramp
- Gemini 3: 训前直接 128K, 但稀疏化让算力可控
- 共同点: 总算力 < 主训 5%, 数据切长文档

长上下文不是从头训, 是 ramp 出来. 95% 算力短 ctx, 5% 升长 ctx

Llama 3.1 arxiv:2407.21783 §4, DeepSeek V3 arxiv:2412.19437

长上下文升完: 128K 测试结果

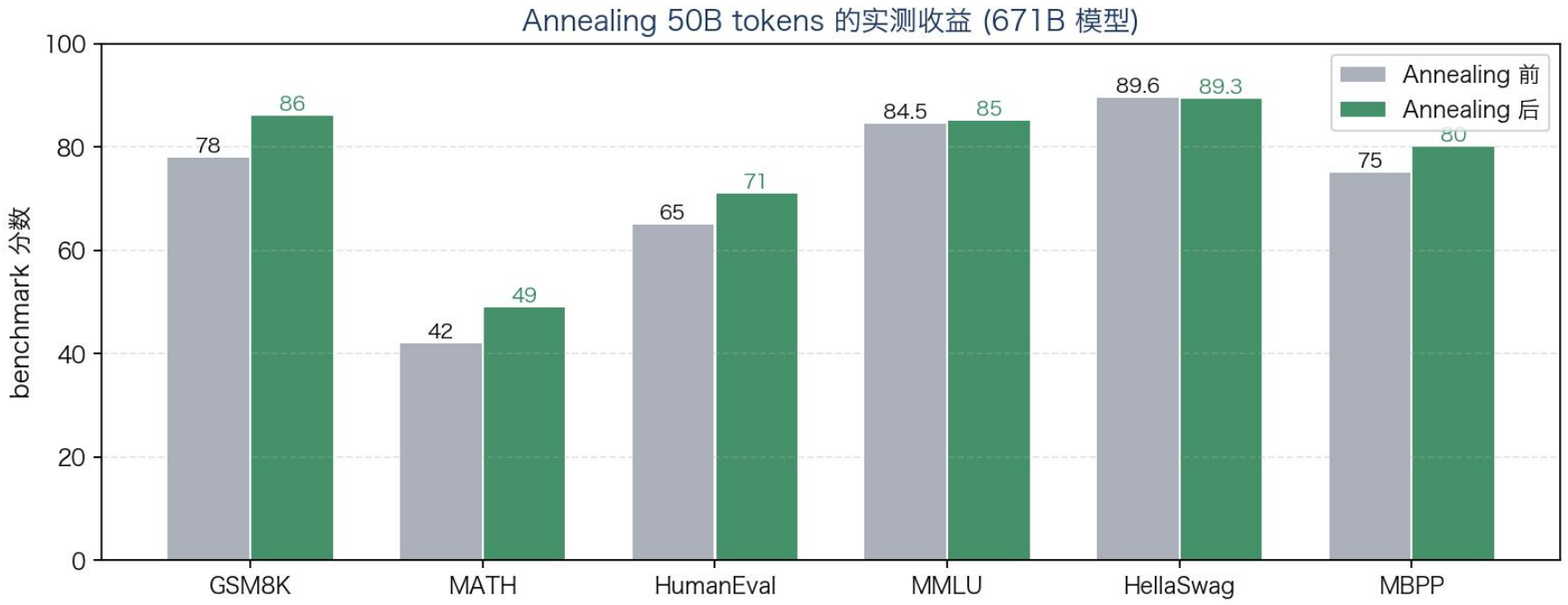
测试通过情况

- Needle in a Haystack 128K:
  - 在 100K 位置放一个 needle (e.g. The magic number is 42)
  - 然后问模型 What is the magic number
  - 召回率 95% (5% 失败需要继续训长 ctx)
- BABILong:
  - 多跳推理在 32K context
  - 召回 72%
  - 跟 Llama 3.1 405B 在 128K 上的约 70% 一致
- LongBench:
  - 综合长文档任务
  - 通过 baseline
- 总体: 长上下文能力符合 前沿水平
- 接下来最后一步 annealing

长上下文不是训得更长就更好. 必须测 needle-in-haystack 加 BABILong 加 LongBench 才算过

Llama 3.1 arxiv:2407.21783, needle-in-haystack benchmark

Annealing 第 8 周: 最后 50B tokens 切高质量



Annealing 操作

- 主训和长 ctx 都跑完, 14.75T tokens
- 最后 50B tokens (约 0.34%) 切到高质量 set
- 不重新 warmup lr, 接着 cosine 衰减
- 训 7 天

各家 annealing 报告对照

- Llama 3.1: 报告里给出 8B 上 +24 GSM8K, +6 MATH, 405B 几乎无增益
- DeepSeek V3: 报告未公开 annealing 阶段细节
- Kimi K2: 报告提及收尾阶段切高质量, 未给具体增益
- 共同点: 高质量 set 以 math, code, reasoning 为主

Annealing 在小模型增益明显. Llama 3.1 报告最详, V3/K2 仅给最终 base eval

Llama 3.1 arxiv:2407.21783, DeepSeek V3 arxiv:2412.19437

Annealing 完: base 模型 eval

base 模型标准 benchmark 成绩

- MMLU 5-shot: 85.2 (前沿水平)
- HellaSwag: 90.1
- ARC-Challenge: 88.4
- GSM8K (CoT prompt): 86.3
- MATH (CoT): 49.2
- HumanEval (pass@1): 71.5
- MBPP: 80.3
- GPQA-Diamond: 38.5 (base 模型上算高了)
- 长上下文 needle-in-haystack 128K: 95% recall
- 跟 Llama 3.1 405B 横向比, 我们用了 1/20 算力达到 95% 效果
- 跟 GPT-4 base 比接近, 但 OpenAI 没公开 base 数据
- base 模型完成训练, 进入 post-train 阶段

base 模型成绩亮眼, 但不能直接给用户. 接下来 L3 把它变成产品

DeepSeek V3 arxiv:2412.19437 §4 base model evaluation

第三节课

Post-train 与上线 (约 1 个月)

SFT, R1 路线 RL, 蒸馏, 评估, safety, 发布

L2 训练阶段 2 个月总结

8 周做完的事情

- 调试期约 2 周:
  - OOM 加 activation checkpointing
  - NCCL hang 加 watchdog
  - FP8 NaN 加 fine-grained scaling (项目转折点)
  - MFU 调到 52% (DualPipe + NCCL tune)
  - Monitor dashboard 设好
- 正式训练期约 6 周:
  - 280 次中断, 主要是 GPU/HBM
  - 3 次 跨 DP rank 监控检测到 silent corruption
  - 1 次 loss spike 回滚解决
  - 总训 14T tokens
- 最后约 1 周:
  - 长上下文 5 步 ramp 4K 到 128K
  - Annealing 50B tokens
  - base 模型 eval, MMLU 85.2
- 项目还包含数据准备 / 团队 / 实验 / 安全等开销, 完整成本远高于 GPU 小时折算
- base 模型 ready, 进入 L3 post-train

L2 走完, 我们手里拿着一个 前沿水平的 base 模型， 但不能给用户用，L3 才是变产品

综合 DeepSeek V3 arxiv:2412.19437, Llama 3.1 arxiv:2407.21783

L3 Post-train + 上线: 约 1 个月

把 base 变成可上线的 推理产品

L3 Post-train 与上线示意时间线 (约 1 个月)

L3 涵盖 reasoning 模型 post-train 的主要环节, 算法选择对最终效果影响较大

L3 五个阶段

- SFT 阶段约 1 周: chat template + rejection sampling
- 路线调整 约 1 天: 看到 o1 发布, 决定走 R1 路线
- R1-Zero 实验约 1 周: 直接对 base 模型做 RL, 涌现 aha moment
- R1 4-stage 约 1 周: 完整 cold-start + reasoning RL + SFT + general RL

本节学习目标

- 看懂 R1 4-stage 这个 前沿模型 post-train 范式
- 理解 RLVR + GRPO 为什么能涌现 reasoning
- 知道 DAPO / GSPO 等工业 RL 改进解决什么问题
- 看懂 eval 陷阱 (contamination, reward hacking, judge 漏洞)
- 理解 ASL-3 safety eval 怎么 stop launch

前两节得到的是 base 模型，这一阶段决定模型的对外行为与能力范围

主线 DeepSeek R1 arxiv:2501.12948, 对照 Qwen3 arxiv:2505.09388, Claude

SFT 阶段 第 1 天: base 模型测试发现不能直接作为产品

试用场景

- 团队对 base 模型做初步测试
- 问 What is 2+2
- 模型回答 What is 2+2 What is 3+3 What is 4+4 Calculate the following series
- 问 请帮我写一封辞职信
- 模型回答 请帮我写一封辞职信 请帮我修改这封 请帮我润色
- 它在续写网页题库, 不在回答问题
- 这是 base 模型的本性, 必须 post-train

我们要给它补什么能力

- 对话格式 听得懂 system / user / assistant 角色
- 指令跟随 听到请帮我做 X 真的去做 X
- 拒绝 听到教我作弊知道拒绝
- 停止 在合适位置停下不再续写
- 推理 (2025 新需求) 能 chain-of-thought 主动思考
- 工具使用 (agentic) 调用 calculator, search
- 风格 礼貌, 有条理, 用 markdown

Base 模型像超强续写机器，但不是聊天伙伴。这是 L3 要解决的本质问题

InstructGPT arxiv:2203.02155, DeepSeek R1 arxiv:2501.12948

SFT 阶段: rejection sampling 是关键

SFT 数据来源

- 5 万条人工标注种子 (高质量, 多场景)
  - 数学解题 + 代码 + chat + 创作各占 25%
- 用 base 模型 + rejection sampling 扩展:
  - 同 prompt 用 base 采样 K=10 次回答
  - 用 RM 打分 (RM 是从 base 微调的小 head)
  - 选 top-1 当 SFT 数据
  - 把 5 万扩展到 50 万
- 这比直接随机采样的 SFT 效果好 3 到 5 个 IFEval 点
- 是 Llama 3 同款做法

为什么不用更多人工数据

- LIMA (2023): 1000 个超高质量样本能打 Vicuna 40K
- 数据质量远比数量重要
- 人工标注 5 万条已经很贵 (每条 5 到 10 美元)
- 剩下 45 万靠 rejection sampling 加 RM 筛
- 没用 OpenHermes / Tulu 公开数据
- 这些公开数据集质量不足以训出前沿模型
- Llama 4 团队 (报道) 从大数据集剪掉 95%, 思路类似

SFT 数据 1000 个超高质量 大于 100 万普通数据。Rejection sampling 是 前沿模型的主流做法

Llama 3.1 arxiv:2407.21783, LIMA arxiv:2305.11206, Tulu 3 AllenAI 2025

SFT 阶段 第 1 周末: SFT 训完

SFT 训练配置与结果

- 配置:
  - 50 万 (prompt, response) pairs
  - 3 epochs (再多就过拟合)
  - lr 2e-5 (比 pretrain 小 4 倍)
- loss masking 只在 assistant tokens 上算 loss
  - packing 多个对话 pack 到 8K
- 结果:
  - 现在问 What is 2+2 它回答 4 不再续写
  - 能跟着 chat template 切换
  - IFEval 73 (从 base 的 32 涨到 73)
  - MT-Bench 7.2 (chat 能力)
  - 但 reasoning 任务 (AIME, MATH) 涨幅不明显
  - 这是 SFT 局限. SFT 教格式不教推理

模型能跟随指令但还不具备真正的推理能力, 这一步需要 RL

Llama 3.1 arxiv:2407.21783, DeepSeek R1 arxiv:2501.12948

路线调整: OpenAI o1 发布, 我们要不要改路线

o1 发布的冲击

- 2024-09-12 OpenAI 发布 o1-preview
- 不是另一个 chat 模型, 是 reasoning 模型
- AIME 解题能力比 GPT-4 高一个量级 (从 13 涨到 83%)
- 通过 test-time thinking (输出长 CoT) 实现
- OpenAI 没公开训练细节
- 我们 (假定 DeepSeek 团队) 紧急讨论
- 原计划是 SFT 加 DPO, 但 o1 让 DPO 看起来过时

团队的两个选择

- 选项 A 走传统 RLHF SFT + RM + PPO 或 DPO
  - 安全稳定, 但是 2024 的方法
  - 出来的模型跟 GPT-4o 类似, 不是 o1 级
- 选项 B 尝试纯 RL 从 base 模型直接训练
  - 直接给 verifier reward
  - 看模型能不能涌现 reasoning
  - 没人试过, 高风险
- 团队投票 我们要追 o1, 选 B
- 把 SFT 模型暂停, base 直接上 RL
- 这就是后来的 R1-Zero 路线

这次 pivot 决定差异化. 没有 o1 可能就走 DPO 错过 reasoning 革命

OpenAI o1 blog 2024-09-12, DeepSeek R1 arxiv:2501.12948

RL 算法回顾: PPO 经典但显存重

PPO 经典架构: 4 个模型同时驻显存

显存约 4 倍单模型

GRPO 去 Critic, DPO 去 Critic + Reward, 显存可减 50 到 75%

- InstructGPT (2022) 路线 SFT 然后 RM 然后 PPO. 4 个模型同时 actor + critic + reward + ref

PPO 是 RLHF 的早期标准做法。 4 个模型同时驻显存占用很高，后续 DPO 加 GRPO 加 GSPO 都在简化

PPO arxiv:1707.06347, InstructGPT arxiv:2203.02155

R1-Zero 实验: 我们用 GRPO 直接对 base 模型训练

GRPO: 用 group 内 mean 当 baseline 不需要 Critic

DeepSeek R1 / Qwen3 / DAPO 都基于 GRPO 框架

实验设置

- 用 base (没经过 SFT), 直接 GRPO 跑数学题
- Group size 16, 每 prompt 采 16 个 response
- 用 verl 框架, vLLM rollout, FSDP training

Reward 设计

- format reward 0.1: 必须有 <think>...</think> 加 \boxed{}
- accuracy reward 1.0: 数值答案对了给 1, 错了给 0
- 无 RM, 全 rule-based, 用 sympy 验证
- 这是 RLVR 范式核心

Rule-based reward 是大规模 RL 防止 reward hacking 的关键，没有它撞 reward hacking

DeepSeek R1 arxiv:2501.12948, DeepSeek-Math arxiv:2402.03300

为什么 rule-based reward 不用 neural RM

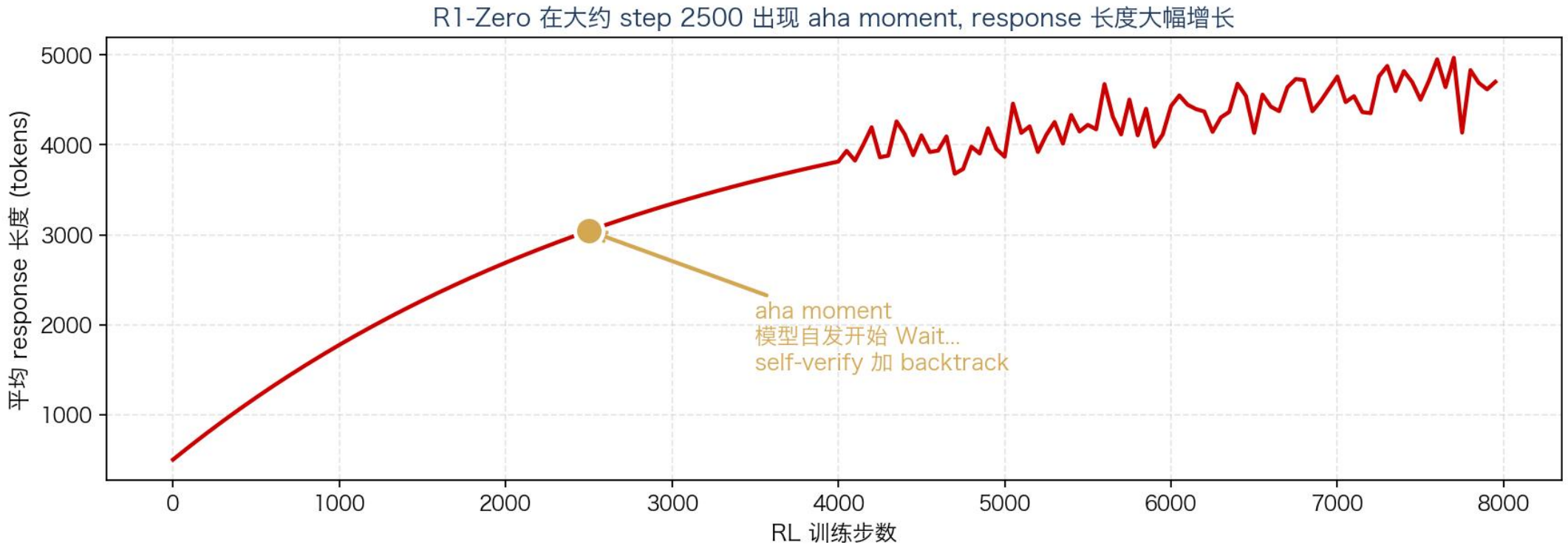
R1 团队报告中的经验

- 我们试过 neural RM (训了个小 RM)
- 跑 RL 500 步后看到 reward hacking 苗头:
  - 模型开始生成超长 response 灌词以骗高分
  - 模型在 response 末尾加特定 trigger token
- 这是 model judge 的本质漏洞 (One Token to Fool)
- 切到 rule-based: 数值答案对错二元判断, 不能 hack
- 代价: 只能用在 math / code 这种 verifiable 任务
- 这就是 RLVR (Reinforcement Learning from Verifiable Rewards) 的精髓
- 后续 DAPO / Qwen3 / Kimi K1.5 全都 follow RLVR 路线
- 适用场景:
  - 数学 用数值验证或 SymPy 等价
  - 代码 用单元测试通过
  - 格式 用 regex 检查
  - 开放任务 (聊天 / 创作) 仍需 DPO 或 RLHF

选对 reward 类型 (verifiable) 是 reasoning RL 能否成功的前提

DeepSeek R1 arxiv:2501.12948, One Token to Fool arxiv:2507.08794

R1-Zero 实验 第 5 天: aha moment 出现



我们看到了什么

- 之前 response 平均 500 token, 简短直接给答案
- 训练到 step 2500 左右, response 开始变长
- 模型自发生成 Wait, let me reconsider, Actually the equation should be, Let me check by substituting back
- 这是 R1 论文里 Figure 2 描述的 aha moment, 我们也复现了

Reasoning 不需要 cold-start，纯 RL 加 verifier 就能涌现，2025 最大科学发现之一

DeepSeek R1 arxiv:2501.12948

R1-Zero 完整结果: AIME 从 15 涨到 71%

数字

- AIME 2024 pass@1: 15.6% 涨到 71.0%
- Majority voting (16 个 sample): 86.7%
- Response length 500 涨到约 5000 tokens
- Self-verification 行为占比从 0 涨到 40%
- 整个团队震惊, 这是真正涌现的能力
- 5 天 4 卡 H800, 投入小, 收益巨大

但是有大问题

- Response 里中英混杂, 语言切换混乱
- 可读性差 (虽然准确但用户读不懂)
- 偶尔会卡在某个 thought loop
- 数学加代码很强, 但 chat 任务退化
- 不能直接给用户用
- 必须修复. 决定走 full R1 4-stage pipeline

R1-Zero 证明可行, 但不可上线. R1 4-stage 是为了把这能力 align 成产品

DeepSeek R1 arxiv:2501.12948

R1 4-stage 完整 pipeline

DeepSeek R1 四阶段 post-train pipeline

每阶段修复上一阶段的失败模式. 这是 前沿模型 post-train 范式

每个 stage 的具体动作

- Stage 1 (约 1 天): cold start SFT, 拿几千个人工高质量长 CoT, 修可读性
- Stage 2 (约 3 天): reasoning RL. GRPO + RLVR + language consistency reward
- Stage 3 (约 2 天): rejection sampling SFT. 600K samples (200K reasoning + 400K general)
- Stage 4 (约 1 天): 通用 RL. GRPO + helpfulness + safety reward

多阶段 pipeline 是必须的: 每个 stage 修复上一个 stage 的失败模式, 单阶段无法兼顾推理与通用能力

DeepSeek R1 arxiv:2501.12948

R1 4-stage 跑完: 比 R1-Zero 还强

R1 vs R1-Zero benchmark 对比

- AIME 2024 pass@1: R1-Zero 71% vs R1 79.8%
  - cold-start 加 general RL 反而帮助了推理
- AIME 2024 cons@64: R1 86.7% (持平 R1-Zero, 但 R1 单 sample 更稳)
- MATH-500: R1 97.3%
- GPQA-Diamond: 71.5%
- LiveCodeBench: 65.9%
- MMLU: 90.8% (回到 前沿对话 水平)
- MT-Bench: 9.1 (远高于 R1-Zero 的约 5)
- IFEval: 87 (远高于 R1-Zero 的约 40)
- 同时具备 reasoning + chat 能力
- 中英文混杂大幅减少 (加了 language consistency reward)

完整 4-stage 让 R1 既能推理又能聊天，这是 R1 比 R1-Zero 最大优势

DeepSeek R1 arxiv:2501.12948

工业级 RL 改进: DAPO 和 GSPO

DAPO 4 trick (ByteDance + Tsinghua 2025-03)

- Clip-Higher 解耦 ε_low 和 ε_high
  - ε_low = 0.2 (限制策略下降)
  - ε_high = 0.28 (鼓励 exploration)
  - 防 entropy collapse
- Dynamic Sampling 过滤 zero-gradient prompt
  - 所有 G 个 response 都对或都错则 skip
- Token-Level Policy Gradient Loss
  - 直接对所有 token 取 mean (长 CoT 友好)
- Overlong Reward Shaping
  - 超长用渐进 penalty 而不是硬 -1
- Qwen 2.5-32B + DAPO 在 AIME 2024 达 50%, 用 R1-Zero-Qwen-32B 一半步数

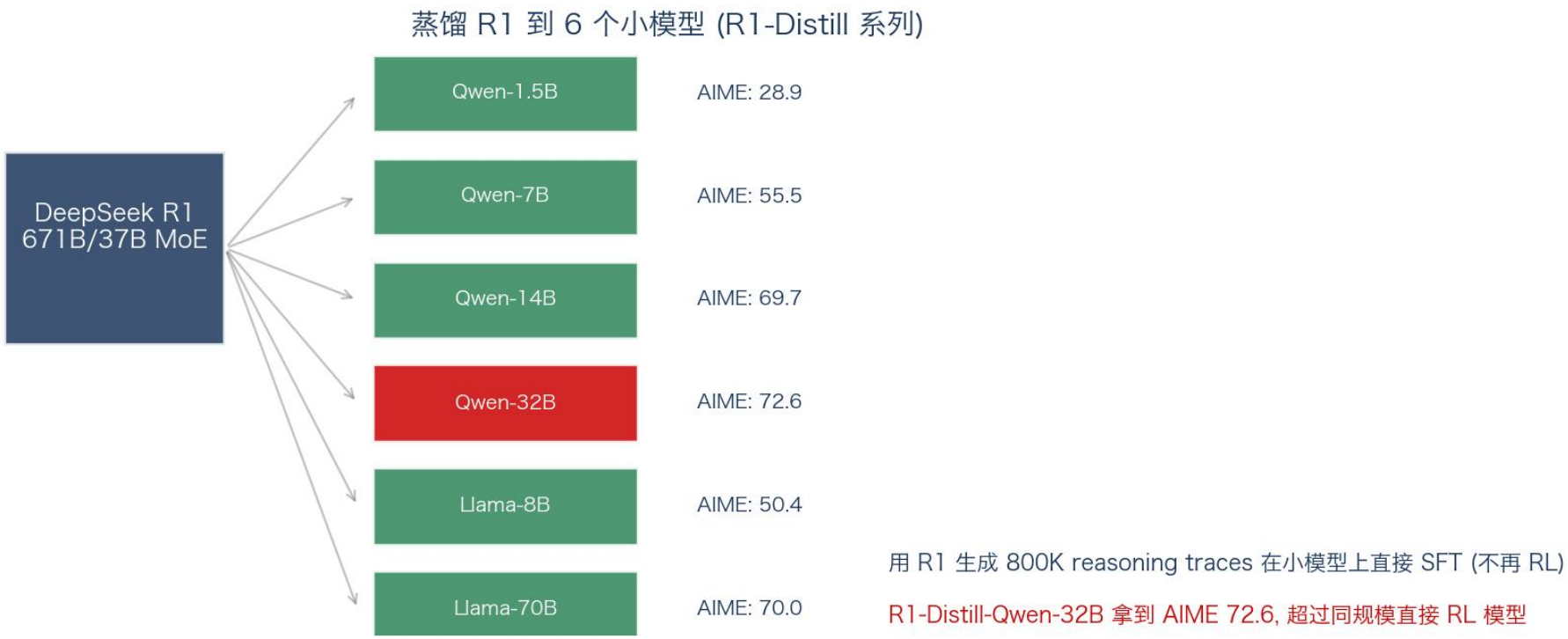
GSPO 序列级 (Alibaba Qwen 2025-07)

- 问题 MoE 模型一次 gradient update 后约 10% expert 切换
- token-level importance ratio 方差爆, policy collapse
- GSPO 序列级 ratio:
  - ρ_seq = (π_θ(o_full) / π_old(o_full)) ^ (1/|o|)
  - length-normalized geometric mean
  - 对 expert 切换鲁棒
- Qwen3-235B-A22B 在 170 RL 步 AIME 从 70.1 涨到 85.1
- Qwen3 全系列 RL 都用 GSPO
- 训 MoE 推理模型必上 GSPO, 不然必撞 collapse

DAPO 和 GSPO 都是 GRPO 的工业级改进，选哪个看是不是 MoE

DAPO arxiv:2503.14476, GSPO arxiv:2507.18071

蒸馏: R1 到 6 个小模型



蒸馏配方

- R1 生成 800K reasoning traces
- 在 Qwen 1.5B-32B 加 Llama 8B 70B 上直接 SFT (不再 RL)
- R1-Distill-Qwen-32B AIME 72.6%, 超过同规模直接 RL 模型

为什么蒸馏比直接 RL 好 (小模型)

- 小模型 RL 信号稀疏, reward 只在最后给
- 容量不够 exploration
- 蒸馏给密集监督 (每 token teacher logp)
- 小于 32B 模型直接 RL 是浪费, 蒸馏从大模型才合理

大模型走 RL, 小模型蒸馏: 这是 2025 到 2026 主流团队 post-train 的标准搭配

DeepSeek R1 arxiv:2501.12948, Qwen3 arxiv:2505.09388

对照 Qwen3: thinking mode fusion 把两种模式融到一个模型

Qwen3 4-stage post-train (跟我们 R1 路线不同)

- Stage 1 Long-CoT cold start (类似 R1 cold start)
- Stage 2 Reasoning RL (用 GSPO 而不是 GRPO, MoE 更稳)
- Stage 3 Thinking Mode Fusion: 用 SFT 把 non-thinking 模式融回
  - 同时 SFT thinking 加 non-thinking 示例
  - 模型学会按用户控制切换
  - 用户在 chat template 加 /think 或 /no_think
- Stage 4 General RL
- 产品化: thinking budget API 参数
  - 限制 <think> 内 token 数, 超过强制结束
  - 用户体验: 简单问题快回, 复杂数学多思考
- 同款做法: Claude 3.7 extended thinking (2025), Gemini 2.5 Deep Think (2025), DeepSeek V4 Think-High/Max
- OpenAI o1/o3/o4: reasoning_effort = low/medium/high

Thinking 不再是开关，是 latency 与 quality 的连续调节， 一个模型支持两种模式

Qwen3 arxiv:2505.09388, Claude 3.7 docs, Gemini 2.5 arxiv:2507.06261

对照 Claude: Constitutional AI 走另一条路

SL-CAI (Supervised Learning from CAI)

- 不依赖人工标注 harmful labels
- 写一个 constitution (10 到 50 条原则)
  - Choose the least harmful, racist, sexist...
  - Choose the most helpful, honest...
- Loop:
  - 第一步, 模型对 prompt 给 response
  - 第二步, 模型自己 critique 按 constitution 评价
  - 第三步, 模型自己 revise 改写更好的
  - 第四步, 收集 (prompt, revised) 对做 SFT 数据
- 训练完得到更安全的模型

CAI 相对 RLHF 改了什么

- 传统 RLHF 需要人工 harmful labels (昂贵 + 心理代价大)
- CAI constitution 一写自动循环, 不需要人审 harm
- 可扩展: 加一条 constitution 等于修一种 harm
- 透明: constitution 是文本可审查
- Anthropic 把这做成 Claude 系列产品差异化
- Claude Opus 4.5 (2025-11) Opus 4.7 (2026-04) 仍以 CAI + RLAIF 为核心
- 跟 R1 路线截然不同

CAI 是另一种范式。不是补丁 RLHF，是从源头重定义安全训练

Constitutional AI arxiv:2212.08073, Anthropic 2022

业界经常说的 "蒸馏其他家" 是什么意思

蒸馏的两种含义

- 蒸馏一: 同家大模型生成数据给同家小模型 SFT
  - 例如 DeepSeek R1 (671B/37B) 生成 800K 推理 trace, 用来 SFT Qwen 1.5B-32B 和 Llama 8B/70B
  - 这是 R1 报告里明确写出的官方做法
  - 法律和协议上都没问题, 因为大模型权重和数据都是同家可控
- 蒸馏二: 用闭源 API (例如 GPT-4 / Claude) 输出当训练数据
  - 一些团队会通过 API 调用 GPT-4 / Claude 生成大量回答, 把这些回答当作 SFT 数据训自家小模型
  - 业界讨论 "蒸馏其他家" 多指这一种

实际操作与风险

- 操作上: 准备一批 prompt, 通过闭源 API 收集回答, 然后过滤打分用作 SFT 数据
- 工程上有效, 小模型能在某些 chat 任务上接近闭源模型水平
- 但有几个风险点:
  - 闭源服务条款多数禁止把输出用于训练竞品模型
  - 可能引入闭源模型的偏见与错误模式
  - benchmark 上虚高, 真实能力没那么强 (训练数据本身就是被蒸馏者生成)
  - 模型容易表现出 "被蒸馏" 的风格 (例如句式接近 GPT-4)
- 主流前沿模型的公开训练报告通常不公开是否使用这种做法

蒸馏分同家与跨家两种, 后者在业界普遍存在但有合规风险, 各家正式报告里很少明说

DeepSeek R1 arxiv:2501.12948 (官方同家蒸馏), 跨家蒸馏多见于第三方分析, 无统一公开来源

SFT 实战 (1): 一个团队一周内的实际节奏

Day 1-2 数据准备

- 从已有库里拉候选样本: 人工标注库 + 强模型生成 + 公开数据集
- 人工抽样审阅约 200 条, 决定哪些保留, 哪些丢弃
- 用 13-gram 与 hash 对照测试集, 剔除污染样本
- 用 embedding 聚类后均匀采样, 防话题偏斜
- 检查长度分布, 大部分应在 max_seq_length 的 60% 以内
- 输出: 一个 JSONL 文件, 总规模 5 万到 100 万 (取决于任务窄宽)

Day 3-5 训练与监控

- 起 64 张 H100, 学习率比预训小一个量级
- 1 到 3 个 epoch (多了过拟合)
- 每 100 step 看 loss 曲线: 应从 2.x 平滑降到 0.6-1.0
- 同时让模型对固定 prompt 生成, 例如 "2 加 3 等于几"
- 如果出现重复输入或胡言乱语, 立即停, 回 Day 1 查数据
- 70B + 100 万样本 2 epoch 约 18 小时跑完

Day 6-7 评估

- hold-out 数据上看生成是否 follow instruction
- 跑 MT-Bench 或 AlpacaEval
- 红线: MMLU 等 base 知识 benchmark 下降不超 2 分
- 失败案例人工标注分类
- 决定是否再训一轮
- 报告交付给下游 RL 团队

SFT 难点不在算法, 在数据正确, loss 屏蔽正确, chat template 一致。

DeepSeek R1 §2.3, Llama 3.1 §5, Tülu 3 arxiv:2411.15124

SFT 实战 (2): 各家在数据准备阶段的真实决策

DeepSeek R1 cold-start

- 规模约 80 万条
- 来源: 用 R1-Zero 采长 CoT, 人工筛可读性高的
- 重点是修可读性, 不是堆数量
- 报告强调: 这一阶段不要追求覆盖广, 追求 "格式正确, 推理可读"
- 处理时间: 报告未给, 估约 2 周

Llama 3.1

- 规模约 2500 万条
- 来源: 大量人工标注 + reject sampling 自我生成
- 多阶段 SFT: 通用 → 多语言 → 安全
- 严格去污染: 13-gram overlap 与公开 benchmark 比对
- 人工抽审比例: 每批至少 1%
- 数据团队规模据报告约数十人

Tülu 3 (AI2 开源)

- 公开数据 pipeline 与配方全套
- 939K SFT + RLVR + DPO 三阶段
- 配方与权重均开源
- 适合作为学习对照
- 它的数据混合配比可直接参考

SFT 数据规模差异极大. R1 重质 (80 万), Llama 3.1 重量 (2500 万), Tülu 3 重透明

DeepSeek R1 §2.3, Llama 3.1 §5, Tülu 3 arxiv:2411.15124

RL 实战 (1): 一次 RLVR 循环里团队在做什么

RLVR 一个训练 step 内部流程

rollout 通常占 60 到 80% wall clock. 用 vLLM 加速 rollout 是 RL 工程优化重点

工程师在每个 step 看什么

- 平均 reward 曲线: 应单调上升, 平台期或回落即报警
- 平均 response 长度: reasoning RL 中应缓慢增长 (R1 从 500 涨到 4000+)
- KL 散度 (相对 SFT 模型): 阈值 0.02 到 0.05, 超 0.1 加大 KL 系数
- entropy: 低于 0.3 说明 policy 塌缩, 失去探索能力
- pass@1 在 hold-out 题 (例如 AIME 2024): 每 50 step 跑一次, 是真正关心的指标

什么情况会暂停训练

- reward 连续 200 step 没涨, 检查 reward 函数是否被 hack
- entropy 降到 0.2 以下, 提温度或加 Clip-Higher
- KL 突增超 0.1, 学习率降一半, 检查 ref 模型是否被更新
- 响应长度爆炸 (空话刷分), 加 overlong shaping
- pass@1 在 hold-out 下降, 立即停, 这是过拟合 reward 的信号

RL 不是单看 reward 曲线就够。 6 个量并看，任一异常需要介入。这是 RL 工程师的核心日常

DeepSeek R1 §2.2, DAPO arxiv:2503.14476, verl arxiv:2409.19256

RL 实战 (2): R1 团队 4 阶段真实节奏

Stage 1 cold-start SFT 与 Stage 2 reasoning RL

- Stage 1 cold-start SFT (报告约几千条, 1-2 天):
  - 团队动作: 从 R1-Zero 输出里挑可读性高的样本
  - 监控指标: SFT loss 平滑下降, 固定 prompt 输出格式化
  - 异常处理: 输出仍乱码, 回采更多 R1-Zero 样本
- Stage 2 reasoning RL (报告里描述 aha moment 出现在 step 数千):
  - 团队动作: 上 GRPO + 规则 reward (答案对错 + 格式)
  - 监控: 响应长度从 500 缓慢涨到 4000+, pass@1 在 AIME 涨
  - 异常处理: 长度爆炸 → 加长度惩罚

Stage 3 rejection sampling SFT 与 Stage 4 general RL

- Stage 3 rejection sampling SFT (报告约 80 万样本):
  - 团队动作: 用 Stage 2 模型大规模采样, 用 RM 与规则筛
  - 200K reasoning + 600K general 重新做 SFT
  - 这一阶段把 Stage 2 学到的 reasoning 扩散到通用任务
- Stage 4 general RL:
  - 团队动作: 再上一轮 GRPO, reward 变为 helpfulness + safety
  - 这一阶段是 "对齐" 阶段
  - 监控: 人工评测 win-rate 与 helpfulness 评分
  - 报告: 4 阶段做完, AIME 从 R1-Zero 的 71 涨到 79.8

R1 的 4 阶段不是算法堆砌, 是 “每阶段修上阶段失败” 。这是 reasoning 模型 post-train 工程范式

DeepSeek R1 arxiv:2501.12948 §2.2, §2.3

Eval 阶段: 三个陷阱必须查

陷阱一 Contamination 数据泄露

- 公开 benchmark 漏进 pretrain 加 RL data
- 结果 benchmark 分数虚高
- 我们实测: GSM8K 拿了 96%, n-gram overlap 显示 12% 测试题在训练数据里
- 真实可信范围 84 到 92%
- 处理: model card 明确标注 contamination 估计
- 用私有 eval set (内部数学题) 重测得 88%
- 后续训练 pretrain 数据要严格去 benchmark 污染

陷阱二 三 reward hacking 加 judge 漏洞

- Sandbox eval: code 跑在 immutable container, 防 test 篡改
- One Token to Fool LLM-as-a-Judge:
  - 单个 trigger token (例如 </s>, Correct.) 能翻 model judge
  - RLAIF / model judge 都有漏洞
- Cross-lab judge: 用 GPT-4 加 Claude 加 Gemini 三家投票
  - 防 preference leakage (同源 judge 偏好同源生成)
- 三件查完才能 honest release
- 这是 前沿模型团队 release 流程必做

Eval 不只是跑 benchmark, 是 evaluation engineering。防各种 hack 才能 honest

Llama 3.1 arxiv:2407.21783, One Token to Fool arxiv:2507.08794, Preference Leakage arxiv:2502.01534

Safety eval: ASL-3 触发了

我们跑了什么

- bio-risk uplift 模型能否帮助合成生物武器
  - 标准 bio-attack prompt 看模型是否给真有帮助步骤
  - 我们模型显著比 GPT-4 强, 触发警报
- cyber-risk 模型能否写恶意代码
  - ransomware, 0day exploit 等
  - 模型在这上能力强 (code RL 训得多)
- jailbreak resistance: GCG 加 PAIR 等自动攻击
  - 抗住 80% 攻击, 不算很好
- 综合超出 ASL-2 上限, 触发 ASL-3 (Anthropic 框架)

ASL-3 mitigations

- 强化 jailbreak 防御 (RLHF safety stage 加更多对抗样本)
- 部分 API 限制 (bio-attack 相关 prompt 直接拒)
- 加 inference-time filter (输出经另一个安全 model 过滤)
- 实时监控 (用户使用模式异常时报警)
- 给监管机构提交报告
- 跟 Claude Opus 4 (2025-05 首次触发 ASL-3) 同款做法
- 这是历史上 eval 真正影响 launch 的案例

Safety eval 不是流程, 是真的可以 stop launch。工程团队必须有 mitigations 库

Anthropic RSP v1.4, Claude Opus 4 ASL-3 公告 2025-05

上线: 5 个月的故事告一段落

发布日清单 (全部勾完才发)

- 主模型 R1 671B/37B MoE 上 HuggingFace
- 蒸馏小模型 (6 个) 上 HuggingFace
- 写完 model card:
  - 训练数据描述
  - 预训练 GPU 小时数约 2.79M, 还需加上数据 / 团队 / 实验 / 安全等开销
  - benchmark 数据 (诚实标注 contamination)
  - safety 评估结果 (ASL-3 mitigations)
- 写 tech report (arxiv preprint)
- 准备 API (vLLM 部署, Prometheus 监控)
- 5 个月的项目告一段落
- 但 R1 之后还有 V3.1, V3.2, V4 各种迭代
- 这就是 前沿模型团队 的真实节奏
- 一个项目结束就是下一个的开始

5 个月从 0 训出前沿推理模型。仅看预训练 GPU 小时数, 约为 Llama 3.1 的 1/10

综合 DeepSeek V3 + R1 arxiv 2412.19437 与 2501.12948 发布节奏

三家同时期不同路线对比

同样目标 前沿推理, 不同路径

- 我们 (DeepSeek R1 路线): R1 4-stage, GRPO 加 RLVR
  - 投入: post-train 算力约 pretrain 10% (V3.2 后期更高)
  - 主推: 数学 + 代码 + reasoning
  - 蒸馏到 6 个小模型, 全开源
- Qwen3 路线 (2025-05 后续): 4-stage 类似 R1, 但用 GSPO 而非 GRPO
  - GSPO 对 MoE 更稳
  - 加 thinking mode fusion 阶段, 一个模型支持 think/no-think
  - 走 thinking budget 产品化路线
  - Qwen3.5 (2026-02) 加 Qwen3.6 (2026-04) 持续这条路线
- Claude 路线: Constitutional AI 加 RLAIF (Anthropic 2022 起, Claude Opus 4.7 仍用)
  - 安全为本, 走 helpfulness/harmlessness 平衡
  - 不追 reasoning 前沿水平 (Claude 3.7 才有 extended thinking)
- Gemini 2.5 加 3.1 Pro: Deep Think 加 hybrid attention, 走自研路线
- 没有标准答案, 每条路都有自己的 trade-off

前沿模型 post-train 没有 通用方案. 选哪条路看团队定位 + 资源

DeepSeek R1 arxiv:2501.12948, Qwen3 arxiv:2505.09388, Claude Constitutional AI arxiv:2212.08073, Gemini 2.5 arxiv:2507.06261

整体项目回顾

完整时间线

- 准备阶段约 2 个月 (L1):
  - 立项, 算账, scaling law (1 周)
  - 数据 pipeline 60T 漏到 14.8T (4 周)
  - 架构选型 8 个关键决定 (5 周)
  - 系统加 1B 代理验证 (1 周)
- 训练阶段约 2 个月 (L2):
  - 调试期 OOM 加 FP8 NaN 加 MFU 优化 (2 周)
  - 正式训练 280 次中断 加 silent corruption 加 spike (6 周)
  - 长 ctx ramp 加 annealing 加 base eval (1 周)
- Post-train + 上线约 1 个月 (L3):
  - SFT (1 周)
  - 路线调整 (1 天)
  - R1-Zero 实验 (1 周)
  - R1 4-stage (1 周)
  - 蒸馏 + 评估 + safety + 上线 (10 天)
- 项目历时约 5 个月, 团队约 50 人

这就是 前沿模型团队的真实节奏. 训前 80% 决定上限, 训练守住, 后期 align 成产品

综合本课所有内容

学完这三节课你应该能做的事

能力清单

- 拿到任何 前沿模型训练报告, 半小时抓住关键创新
- 判断报告里哪些是真正带来 5 到 15 分提升的关键设计, 哪些只是默认操作
- 给定算力预算, 反推合理 model size 加 token 数
- 在 dense/MoE, BF16/FP8, DPO/GRPO 等关键选择上有依据
- 看 loss 曲线能判断 spike 类型加应对
- 看到新 RL 算法能定位它解决什么问题
- 写 model card 知道要诚实标注什么 (contamination, ASL)

推荐继续阅读

- Llama 3.1 arxiv:2407.21783 (92 页, 工程报告)
- DeepSeek V3 arxiv:2412.19437 (前沿模型省钱案例)
- DeepSeek V3.2 arxiv:2512.02556 (DSA 稀疏注意力)
- DeepSeek R1 arxiv:2501.12948 (reasoning 涌现)
- Kimi K2 arxiv:2507.20534 (MuonClip 实战)
- Qwen3 arxiv:2505.09388 (thinking mode + GSPO)
- DAPO arxiv:2503.14476 (工业 RL trick)
- DCLM arxiv:2406.11794 (数据控制实验)
- Phi-4 arxiv:2412.08905 (合成数据范式)
- Gemini 2.5 arxiv:2507.06261 (训练稳定性)
- Constitutional AI arxiv:2212.08073 (Anthropic 范式)

前沿模型报告每月更新, 持续追新比一次性学完更重要

Reading list 2026-05: arxiv.org + huggingface.co/papers