

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



11-768 · AI AGENTS | TRAINING 1

Supervised Fine-Tuning for Agents

Turning recorded trajectories into weights

Carnegie Mellon University
Language Technologies Institute

Yueqi Song
Language Technologies Institute

1

11-768 · AI智能体 | 训练 1

智能体的监督微调

将记录下来的轨迹转化为权重

Yueqi Song

语言技术研究所 (Language Technologies Institute)

智能体的监督微调

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Today

- Weight updates as a capability
- Trajectories as token sequences
- Choosing the trajectories
- One format for many datasets
- Running the training
- Knowing it worked
- Handing off to RL

2

今天

将权重更新作为一种能力

将轨迹表示为词元序列

选择轨迹

一种格式适配多种数据集

运行训练

判断训练是否奏效

交接至强化学习（RL）

今日内容

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

A trajectory as recorded data



- The agent received an issue and a couple of tools in a sandbox: **bash** and a file editor.
- It produced thoughts, commands, and the observations those commands returned.
- The repository's own tests decided whether the run succeeded.

Pan et al., 2024 · SWE-Gym, § 4.2 ↗

一条轨迹，作为记录数据

system: 执行框架 (harness) 提示词 → user: 问题文本 → assistant: 思考 + bash → observation: 测试输出 → assistant: 思考 + 编辑 → observation: 差异 (diff) → assistant: 提交.....

智能体在沙箱中收到一个问题和几个工具: bash 与文件编辑器。

它生成了思考、命令，以及这些命令返回的观测。

仓库自身的测试决定了这次运行是否成功。

Pan 等, 2024 · SWE-Gym, 第 4.2 节 ↗

一条作为记录数据的轨迹

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

Methods of updating the agent

LOCATION	PROS	CONS
Context window SEP 1	high fidelity to what happened	costly, noisy, does not decide what matters
External artifacts SEP 3	inspectable, editable, retrievable, portable	must be induced, selected, and maintained
Model weights THIS LECTURE	faster inference, broad behavior change	slower updates, opaque, model-specific

- Change the weights once and every call after that gets the new behavior, with nothing extra in the prompt.
- The first two rows update in seconds. This row costs a training run every time you change it.
- And you cannot open the weights to see what changed, or move the change to a different model.

4

ZH / 中文译文 · 第 1/1 部分

更新智能体的方法

位置 | 优点 | 缺点

上下文窗口（第 1 部分） | 高度忠实于实际发生的事情 | 成本高、噪声大，而且不会判断哪些内容重要

外部工件（第 3 部分） | 可检查、可编辑、可检索、可移植 | 必须通过诱导生成、筛选并维护

模型权重（本讲） | 推理更快，能广泛改变行为 | 更新较慢、过程不透明，且特定于模型

权重一经改变，此后每次调用都会获得新行为，无须在提示词中添加任何额外内容。

前两行所述的更新只需几秒。第三行每次改变都要付出一次训练运行的成本。

而且你无法打开权重查看发生了什么变化，也无法把这项变化迁移到另一个模型。

更新智能体的方法

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

What the trajectories teach

- How to write a tool call this harness can actually run.
- How to keep going when the conversation gets long, because the trajectories it learned from were long.
- What a tool result looks like, so it waits for one instead of writing one itself.
- And whatever else was in the data: extra commentary, repeated commands, one way of solving a problem where several would work.

5

轨迹教会智能体什么

如何编写这种 执行框架（harness） 确实能够运行的工具调用。

如何在对话变长时继续进行，因为它学习所依据的轨迹本身很长。

工具结果是什么样的，让它会等待工具结果，而不是自己编造一个。

以及数据里包含的其他一切：额外的评论、重复的命令，以及在多种可行解法中采用的一种解题方式。

轨迹教会智能体什么

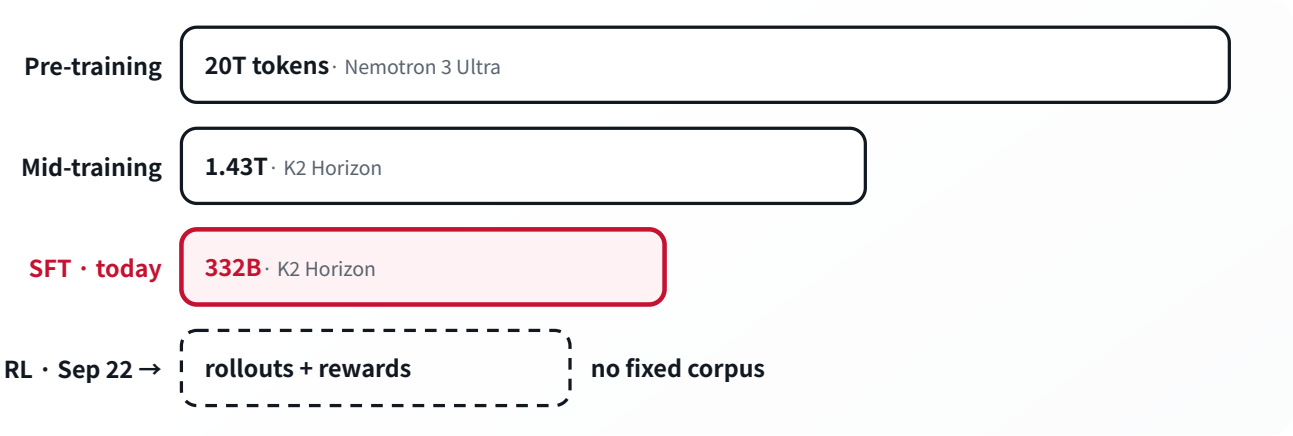
第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

The full training pipeline

Drawn to token scale: every step down is fewer tokens, each one more heavily curated.



[Nemotron 3 Ultra ↗](#)

[K2 Horizon · Training Overview ↗](#)

6

完整训练流程

完整训练流程

按词元尺度绘制：每向下一步，词元数量就更少，筛选也更严格。

预训练：20T 词元 · Nemotron 3 Ultra

中期训练：1.43T · K2 Horizon

监督微调（SFT）· 今日：332B · K2 Horizon

强化学习（RL）· 9 月 22 日起 → 轨迹采样 + 奖励 · 没有固定语料库

Nemotron 3 Ultra ↗ · K2 Horizon · Training Overview ↗

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Cold start before reinforcement learning

"Cold start" is the checkpoint RL begins from. Today's stage is how K3 makes that checkpoint good.



- In Kimi K3, the model learns to call tools and finish long tasks during SFT, before any RL.
- Some might ask: why not just RL, without SFT? DeepSeek-R1-Zero ran RL directly on the base model, with no SFT at all.
- Their very next model put SFT back in: R1 starts from a few thousand curated examples, because R1-Zero's answers mixed languages and were hard to read.

"The SFT stage establishes a high-quality cold-start policy for the subsequent RL stage."

[Kimi Team, 2026 · Kimi K3, § 4.1 ↗](#) [DeepSeek-R1 ↗](#)

7

强化学习之前的冷启动

“冷启动”指强化学习开始时所用的检查点。本讲讨论的是 K3 如何把这个检查点打磨好。

监督微调：初始化基线智能体能力

强化学习：以不同推理强度培养领域专家

策略上的在线蒸馏：将这些专家整合进一个模型

在 Kimi K3 中，模型在监督微调阶段就学会调用工具并完成长任务，此时强化学习尚未开始。

有人可能会问：为什么不直接使用强化学习，而不做监督微调？
DeepSeek-R1-Zero 曾直接在基础模型上运行强化学习，完全没有监督微调。

他们紧接着发布的下一个模型又加入了监督微调：R1 从几千个经过筛选的示例开始，因为 R1-Zero 的回答中语言混杂，难以阅读。

“监督微调阶段为后续强化学习阶段建立了高质量的冷启动策略。”

[Kimi Team, 2026 · Kimi K3, 第 4.1 节 ↗](#) · [DeepSeek-R1 ↗](#)

强化学习之前的冷启动

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

SFT and RL, side by side

	SUPERVISED FINE-TUNING	REINFORCEMENT LEARNING
learns from	recorded trajectories, whoever produced them	its own rollouts, scored by a reward
signal	a target for every token	one number per rollout
needs	data and GPUs	live environments, sandboxes, a checkable reward
good at	learning tool calls, formats, long tasks	sharpening what the model can already attempt
stuck when	the demonstrations are wrong, narrow, or used up	the start is too weak, or the reward cannot be checked

[OpenThoughts-Agent, Table 11 ↗](#) [Kimi K3 ↗](#) [Kimi K2, § 3.2 ↗](#)

并列比较监督微调与强化学习

监督微调 | 强化学习

从什么学习 | 记录下来的轨迹，不论由谁生成 | 智能体自己的轨迹采样，由奖励打分

训练信号 | 每个词元都有一个目标 | 每条轨迹采样只有一个数值

需要什么 | 数据和 GPU | 实时环境、沙箱，以及可验证的奖励

擅长什么 | 学习工具调用、格式和长任务 | 强化模型已经能够尝试的能力

何时受阻 | 示范有误、范围狭窄或已经用尽 | 起点太弱，或奖励无法验证

OpenThoughts-Agent, 表 11 ↗ · Kimi K3 ↗ · Kimi K2, 第 3.2 节 ↗

并列比较监督微调与强化学习

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

The training lectures

TOPIC	WHEN
Supervised fine-tuning	today
Reinforcement learning	Sep 22 · Sep 29 · Oct 1

9

本课程的训练主题

主题 | 时间

监督微调 | 今天

强化学习 | 9 月 22 日 · 9 月 29 日 · 10 月 1 日

本课程的训练主题安排

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

将轨迹表示为词元序列

Trajectories as token sequences

10

将轨迹表示为词元序列

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

The chat template applied to the trajectory

system
harness prompt

user
issue text

assistant
thought + bash

observation
test output

assistant
thought + edit

observation
diff

...

assistant
submit

one token sequence · every boundary above is a template token

<|im_start|>tool↵3·failed,·41·passed↵<|im_end|>↵<|im_start|>assistant↵The·test·compares·floats·Fix·it.↵<tool_call>↵str_replace(...)↵</tool_call>↵<|im_end|>

“.” = space, “↵” = newline

▪ Feed the trajectory through the template and you get one string of tokens.

▪ The <|...|> markers are single special tokens. The rest is what the agent wrote and saw.

Transformers · chat templating ↗

Kimi K3 · XTM, § 4.1.1 ↗

11

应用于轨迹的聊天模板

system: 执行框架 (harness) 提示词 → user: 问题文本 → assistant: 思考 + bash → observation: 测试输出 → assistant: 思考 + 编辑 → observation: 差异 (diff) → assistant: 提交.....

一个词元序列 · 上方每个边界都是模板词元

<|im_start|>tool ↵ 3·failed,·41·passed ↵ <|im_end|> ↵ <|im_start|>assistant ↵ The·test·compares·floats·Fix·it. ↵ <tool_call> ↵ str_replace(...) ↵ </tool_call> ↵ <|im_end|>

“.” 表示空格, “↵” 表示换行。

把轨迹输入模板, 就会得到一个词元字符串。

<|...|> 标记是单个特殊词元。其余部分则是智能体写出并看到的内容。

Transformers · 聊天模板 ↗ · Kimi K3 · XTM, 第 4.1.1 节 ↗

应用于轨迹的聊天模板

个人学习译文 · 非 CMU 官方译本 · 译注独立标记 · 原始课件来自 cmu-agents.com

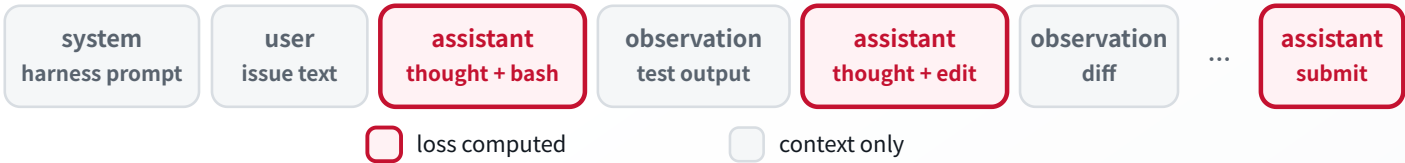
双语 PDF 11 | 原页 11

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Which tokens carry the loss



Cross-entropy over the assistant tokens only

$$\mathcal{L}(\theta) = - \sum_{t: m_t=1} \log p_{\theta}(x_t \mid x_{<t})$$

- Mask: one bit per token, saying whether that token's prediction counts.
- Minimizing it raises the probability of each recorded action, given everything before it.
- Observations are conditioning context, not prediction targets.

12

哪些词元参与损失计算

system: 执行框架 (harness) 提示词 → user: 问题文本 → assistant: 思考 + bash → observation: 测试输出 → assistant: 思考 + 编辑 → observation: 差异 (diff) → assistant: 提交.....

参与损失计算: assistant 词元; 仅作上下文: 其他内容

仅对 assistant 词元计算交叉熵

$$L(\theta) = - \sum_{t: m_t = 1} \log p_{\theta}(x_t \mid x_{<t})$$

掩码: 每个词元对应一位, 用来说明该词元的预测是否计入损失。

最小化该损失会提高每个已记录动作的概率, 条件是此前发生的一切。

观测是条件上下文, 而不是预测目标。

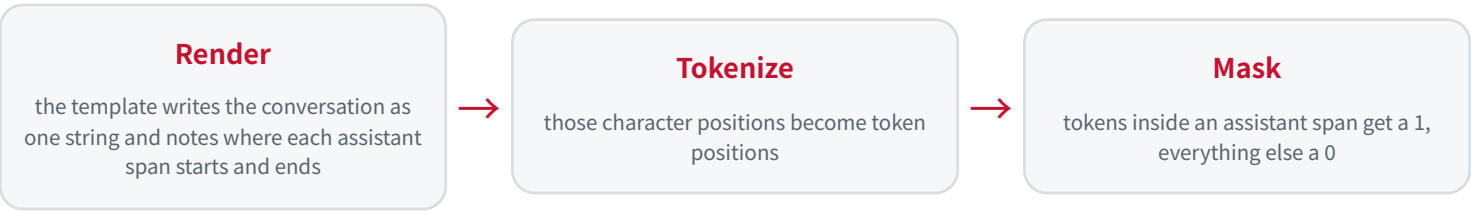
哪些词元参与损失计算

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

How the mask is built



- The template does the bookkeeping as it writes: where assistant text starts, where it stops.
- By the time training starts, the roles are gone. The trainer sees the string and the bits, nothing else.

[Transformers · apply_chat_template ↗](#)

13

掩码如何构建

渲染：模板将对话写成一个字符串，并记录每个 assistant 片段的起止位置 → 词元化：这些字符位置映射为词元位置 → 掩码：assistant 片段中的词元标为 1，其余全部标为 0

模板在写入时负责记录位置：assistant 文本从哪里开始、到哪里结束。

训练开始时，各角色信息已被移除。训练器看到的只有字符串和这些位，除此之外一无所有。

[Transformers · apply_chat_template ↗](#)

掩码如何构建

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Masking controls across frameworks

FRAMEWORK	WHAT YOU SET	HOW THE MASK IS PRODUCED
TRL	assistant_only_loss=True	the template marks the assistant spans as it renders
Axolotl	roles_to_train, train_on_eos	finds each turn’s role header in the rendered string
LLaMA-Factory	train_on_prompt, mask_history	tags whole turns as train-or-ignore while encoding

- Same decision in all three. Only TRL puts it in the template.

不同框架中的掩码控制

框架 | 你设置什么 | 掩码如何生成

TRL | assistant_only_loss=True | 模板在渲染时标记 assistant 片段

Axolotl | roles_to_train、train_on_eos | 在渲染后的字符串中查找每一轮的角色标头

LLaMA-Factory | train_on_prompt、mask_history | 编码时将完整轮次标记为训练或忽略

三者作出的判断相同。只有 TRL 把它放进模板中。

TRL · SFTTrainer ↗ · Axolotl · 对话格式 ↗ · LLaMA-Factory · 数据参数 ↗

不同框架中的掩码控制

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

The end-of-turn token



“.” = space, “↵” = newline · red = loss computed, gray = context only

The role header is supplied at inference. It is never a target. **The model produces the stop token, so it has to be inside the mask.**

"End of turn" is three events, not one: end this message, hand off to a tool, or finish the task. Kimi K3's template spells them apart with separate tokens:



"the model may not learn to stop."

轮次结束词元

`<|im_start|>assistant ↵ I'll run the failing test. ↵ <tool_call> ↵`
`bash(cmd='pytest -q') ↵ </tool_call> ↵ <|im_end|> ↵`

“.” 表示空格， “↵” 表示换行 · 红色表示参与损失计算，灰色表示仅作上下文
角色标头由推理时提供，永远不是目标。

模型会生成停止词元，因此它必须包含在掩码中。

“轮次结束” 包含三个事件，而非一个：结束此消息、移交给工具，或完成任务。
Kimi K3 的模板用不同词元将它们区分开：

`[open]message·role=assistant[sep] [open]think[sep] ... [close]think[sep]`
`[open]response[sep] Use·math.isclose. [close]response[sep] [end_of_msg]`

“模型可能学不会停止。”

[TRL · 训练聊天模板 ↗](#) · [TRL · SFTTrainer 源码 ↗](#) · [Kimi K3 · XTMl，第 F 节 ↗](#)

轮次结束词元

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

Masking the reasoning block



“.” = space, “↵” = newline · red = loss computed, gray = context only

- Two switches, set separately: does the reasoning stay in the context, and does it get trained on.
- In the context, the reasoning explains the action that follows. Trained on, it becomes the style the model itself writes.
- Nemotron keeps budget-truncated reasoning in the context, and masks the artificial cut-off out of the loss.

ZH / 中文译文 · 第 1/1 部分

对推理区块应用掩码

<|im_start|>assistant ↵ <think> ↵ The·assert·compares·floats. ↵ </think>
↵ Use·math.isclose. ↵ <|im_end|>

“.” 表示空格，“↵” 表示换行 · 红色表示参与损失计算，灰色表示仅作上下文

有两个独立开关：推理是否保留在上下文中，以及是否对其进行训练。

在上下文中，推理会解释后续动作。若对其训练，它就会成为模型自身生成的风格。

Nemotron 会将按预算截断的推理保留在上下文中，并将人为截断部分从损失中屏蔽掉。

NVIDIA, 2026 · Nemotron 3 Ultra, 第 3.1.1 节 ↗

对推理区块应用掩码

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Weighting inside the assistant turn

Models	Modules		Single Turn	Multi Turn
	SSB	HDR		
Qwen2.5-Coder-Inst+SFT	×	×	81.73	38.25
	✓	×	82.20	41.62
	×	✓	83.85	43.12
	✓	✓	84.00	47.00
Llama-3.2-3B-Inst+SFT	×	×	73.67	33.25
	✓	×	76.82	34.62
	×	✓	78.28	35.75
	✓	✓	78.99	36.12

Table 5: Performance comparison of Qwen2.5-Coder-Inst and Llama-3.2-3B-Inst, across different components of modules (SSB and HDR) in Single-Turn and Multi-Turn on BFCL.

Two components from one paper: SSB is a learned weight that rebalances reasoning tokens against tool-call tokens in the loss, and HDR resamples the hard examples.

- Long reasoning dominates the loss, and the short tool call is what you actually need correct.
- Rebalancing helps more on multi-turn than on single-turn, on both base models.

Hao et al., 2025 · BalanceSFT, Table 5 ↗

assistant 轮次内部的权重分配

assistant 轮次内部的权重分配

表 5: Qwen2.5-Coder-Inst 与 Llama-3.2-3B-Inst 在 BFCL 的单轮与多轮表现比较，展示不同模块组件（SSB 和 HDR）的影响。

模型 | SSB | HDR | 单轮 | 多轮

Qwen2.5-Coder-Inst+SFT | × | × | 81.73 | 38.25

Qwen2.5-Coder-Inst+SFT | ✓ | × | 82.20 | 41.62

Qwen2.5-Coder-Inst+SFT | × | ✓ | 83.85 | 43.12

Qwen2.5-Coder-Inst+SFT | ✓ | ✓ | 84.00 | 47.00

Llama-3.2-3B-Inst+SFT | × | × | 73.67 | 33.25

Llama-3.2-3B-Inst+SFT | ✓ | × | 76.82 | 34.62

Llama-3.2-3B-Inst+SFT | × | ✓ | 78.28 | 35.75

Llama-3.2-3B-Inst+SFT | ✓ | ✓ | 78.99 | 36.12

一篇论文中的两个组件：SSB 是一种学习得到的权重，用于在损失中重新平衡推理词元与工具调用词元；HDR 则对困难样例重新采样。

长推理主导了损失，而真正需要正确的是较短的工具调用。

在两个基础模型上，重新平衡对多轮任务的帮助都大于对单轮任务的帮助。

Hao 等, 2025 · BalanceSFT, 表 5 ↗

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Samples versus tokens

Capability	Sample weight	Token weight
STEM and Coding	56%	89%
Agentic Capability	11%	9%
General Helpfulness and Safety	33%	2%

Table 10. Consolidation SFT data mixture composition by capability.

They chose the mixture by counting examples. The loss is computed per token.

- The mixture was balanced by counting examples, but STEM and coding traces run far longer, so they take nearly all the tokens.
- Whatever you balance by counting trajectories, check what it turned into in tokens.

Microsoft AI, 2026 · MAI-Thinking-1, Table 10 ↗

样例数与词元数

能力 | 样例权重 | 词元权重

STEM 与编程 | 56% | 89%

智能体能力 | 11% | 9%

通用帮助性与安全性 | 33% | 2%

表 10：按能力划分的整合 SFT 数据混合组成。

他们按样例数量来选择数据混合比例。损失却是按词元计算的。

数据混合按样例数量均衡，但 STEM 与编程轨迹要长得多，因此几乎占据了全部词元。

无论按轨迹数量怎样做均衡，都要检查最终折算成词元后是什么情况。

Microsoft AI, 2026 · MAI-Thinking-1, 表 10 ↗

样例数与词元数

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Open problems in supervising a trajectory

- We could not find a published experiment that changes the mask for agent SFT and measures what happens.
- The careful masking experiments are single-turn instruction tuning, not agents.
- The one case where unmasking the prompt helped had long prompts, short answers, and little data. Agent trajectories are the opposite on all three.
- At 8B scale this experiment is cheap, and it is sitting there unrun.

Chatterjee et al., 2025 · WIT ↗ Shi et al., 2024 · Instruction Modelling ↗

轨迹监督中的开放问题

我们没有找到已发表的实验，专门改变智能体 SFT 的掩码并测量其影响。

细致研究掩码的实验针对的是单轮指令微调，而非智能体。

唯一一个取消提示词掩码后有所帮助的案例，具有提示词长、回答短、数据量少这三个特点。智能体轨迹在这三方面恰好相反。

在 8B 规模上，这个实验成本很低，却至今无人运行。

Chatterjee 等, 2025 · WIT ↗ · Shi 等, 2024 · Instruction Modelling ↗

轨迹监督中的开放问题

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



选择轨迹

选择轨迹

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Where the trajectories come from

WHO RAN THE TASK	YOU KEEP EVERYTHING	YOU KEEP THE RUNS THAT PASSED
a stronger model ran it	plain distillation	SWE-Gym · Nemotron · OT-Agent
the model ran it itself	MAI, on its own reasoning runs	expert iteration · Sep 22

- Most agent SFT data sits in the top right box: a stronger model writes better trajectories than the student could, and the task's own tests make keeping the good ones cheap.
- In the bottom row the model makes its own training data. Repeat that loop and you get expert iteration, which will be covered in future lectures.

21

轨迹来自哪里

谁运行了任务 | 全部保留 | 只保留通过的运行

由更强的模型运行 | 朴素蒸馏 | SWE-Gym · Nemotron · OT-Agent

模型自己运行 | MAI，基于其自身推理运行 | 专家迭代 · 9 月 22 日

大多数智能体 SFT 数据都处于右上格：更强的模型能写出优于学生模型自身能力的轨迹，而且任务自带的测试能以很低成本筛选出好的轨迹。

在下面一行中，模型生成自己的训练数据。重复这个循环就形成专家迭代，我们将在后续课程中讲到。

轨迹来自哪里

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Distillation from a stronger model

Model Size	Empty Patch (% , ↓)			Stuck in Loop (% , ↓)			Avg. Turn(s)			Resolve Rate (% , ↑)		
	zero-shot	fine-tuned	Δ	zero-shot	fine-tuned	Δ	zero-shot	fine-tuned	Δ	zero-shot	fine-tuned	Δ
SWE-Bench Lite (300 instances)												
7B	40.3	29.7	-10.7	47.0	31.0	-16.0	20.3	22.2	+1.9	1.0 (± 1.0)	10.0 (± 2.4)	+9.0
14B	49.7	18.1	-31.6	31.7	27.1	-4.6	23.2	21.4	-1.8	2.7 (± 1.9)	12.7 (± 2.3)	+10.0
32B	27.0	18.1	-8.9	16.7	18.1	+1.5	15.5	29.3	+13.9	3.0 (± 1.4)	15.3 (± 2.5)	+12.3
SWE-Bench Verified (500 instances)												
7B	45.8	33.8	-12.0	39.6	21.0	-18.6	21.9	35.3	+13.4	1.8 (± 1.1)	10.6 (± 2.1)	+8.8
14B	44.9	14.5	-30.4	32.1	21.3	-10.7	25.5	30.1	+4.6	4.0 (± 1.6)	16.4 (± 2.0)	+12.4
32B	9.5	13.8	+4.3	29.4	23.8	-5.6	24.6	31.6	+7.0	7.0 (± 1.3)	20.6 (± 2.1)	+13.6

- SEP 10 GPT-4o and Claude ran the SWE-Gym tasks; 491 runs passed the repository tests.
- A 32B Qwen model was fine-tuned on those 491 runs alone.
- Its scores on both SWE-Bench splits rose by more than ten points.

Pan et al., 2024 · SWE-Gym, § 4.2 and Table 3 ↗

从更强的模型进行蒸馏

从更强的模型进行蒸馏

模型规模 | 空补丁 (% , ↓) : zero-shot / 微调后 / Δ | 陷入循环 (% , ↓) : zero-shot / 微调后 / Δ | 平均轮数: zero-shot / 微调后 / Δ | 解决率 (% , ↑) : zero-shot / 微调后 / Δ

SWE-Bench Lite (300 个实例)

7B | 40.3 / 29.7 / -10.7 | 47.0 / 31.0 / -16.0 | 20.3 / 22.2 / +1.9 | 1.0 (±1.0) / 10.0 (±2.4) / +9.0

14B | 49.7 / 18.1 / -31.6 | 31.7 / 27.1 / -4.6 | 23.2 / 21.4 / -1.8 | 2.7 (±1.9) / 12.7 (±2.3) / +10.0

32B | 27.0 / 18.1 / -8.9 | 16.7 / 18.1 / +1.5 | 15.5 / 29.3 / +13.9 | 3.0 (±1.4) / 15.3 (±2.5) / +12.3

SWE-Bench Verified (500 个实例)

7B | 45.8 / 33.8 / -12.0 | 39.6 / 21.0 / -18.6 | 21.9 / 35.3 / +13.4 | 1.8 (±1.1) / 10.6 (±2.1) / +8.8

14B | 44.9 / 14.5 / -30.4 | 32.1 / 21.3 / -10.7 | 25.5 / 30.1 / +4.6 | 4.0 (±1.6) / 16.4 (±2.0) / +12.4

32B | 9.5 / 13.8 / +4.3 | 29.4 / 23.8 / -5.6 | 24.6 / 31.6 / +7.0 | 7.0 (±1.3) / 20.6 (±2.1) / +13.6

9 月 10 日: GPT-4o 和 Claude 运行了 SWE-Gym 任务; 491 次运行通过了仓库测试。

一个 32B Qwen 模型仅使用这 491 次运行进行了微调。

它在 SWE-Bench 的两个数据划分上的分数都提升了十多个百分点。

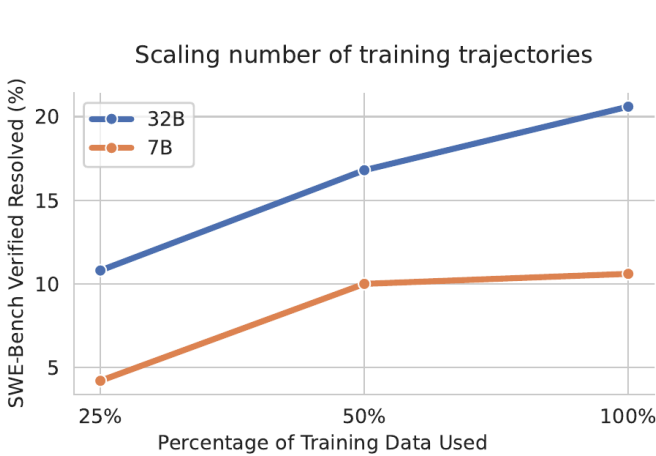
Pan 等, 2024 · SWE-Gym, 第 4.2 节与表 3 ↗

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Trajectory count and saturation



Same task set throughout. Only the number of sampled trajectories changes.

- Accuracy was still rising when the sampling budget ran out.
- They had tasks to spare. What they ran out of was compute to sample more runs.
- Every point in this plot uses the same tasks. What happens when you add new tasks comes a few slides later.

Pan et al., 2024 · SWE-Gym, § 5.2 and Figure 5 ↗

轨迹数量与饱和

训练轨迹数量的扩展

纵轴：SWE-Bench Verified 解决率（%）；横轴：使用的训练数据比例（25%、50%、100%）；系列：32B、7B

整个过程中使用相同的任务集。变化的只有采样轨迹的数量。

采样预算耗尽时，准确率仍在上升。

任务还有很多。真正耗尽的是采样更多运行所需的计算资源。

图中的每个数据点使用的都是相同任务。增加新任务会发生什么，将在几页之后讲到。

Pan 等, 2024 · SWE-Gym, 第 5.2 节与图 5 ↗

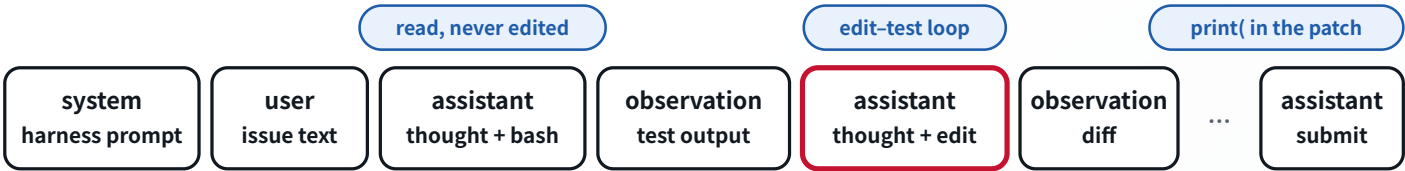
轨迹数量与饱和

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Undesirable behaviors in successful rollouts



- Every run on this slide passed its tests.
- But look at how: one edited and re-ran tests in a loop for most of its turns, another left print statements in the final patch.
- The loss copies every action, good or not. Train on these runs and the model learns the mess together with the fix.

"would teach undesirable behaviors if used directly as SFT data"

NVIDIA, 2026 · Nemotron 3 Ultra, § 3.1.1 ↗

24

成功轨迹中的不良行为

成功轨迹中的不良行为

只读、从未编辑 | 编辑—测试循环 | 补丁中留下 print(

system: 执行框架 (harness) 提示词 → user: 问题文本 → assistant: 思考 + bash → observation: 测试输出 → assistant: 思考 + 编辑 → observation: 差异 (diff) → assistant: 提交.....

本页中的每次运行都通过了测试。

但看看它们是怎么做到的：有一次运行的大部分轮次都在编辑并重复运行测试；另一次则在最终补丁中留下了 print 语句。

损失会照单全收每个动作，无论好坏。用这些运行进行训练，模型会把混乱和修复方案一起学进去。

“若直接用作 SFT 数据，会教会模型不良行为。”

NVIDIA, 2026 · Nemotron 3 Ultra, 第 3.1.1 节 ↗

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Filtering rollouts by shape

Rank	Trajectory Filter	Benchmarks			Average	
		SWE-bench Verified (100)	OT-TB Lite	Terminal-Bench 2.0	Raw	Normalized
1	Min turns ≥ 5	29.00 ^{1.56}	19.10 ^{1.57}	11.61 ^{1.30}	19.90	+1.25
2	Filter timeouts	26.67 ^{1.70}	18.03 ^{1.60}	10.49 ^{1.18}	18.39	-0.35
3	Filter subagent traces	23.00 ^{1.63}	17.31 ^{1.58}	10.86 ^{1.35}	17.06	-0.90

Table 7: **Filtering agent rollouts: keeping longer trajectories helps.** The minimum-turns filter outperforms timeout and subagent filters across all three benchmarks.

- The best filter here just counts turns and drops every run shorter than five.
- Turn count says how much work a run took, not whether the work was any good.
- A four-turn clean fix gets dropped. A forty-turn mess gets kept. On average it still helped.

Raoof et al., 2026 · OpenThoughts-Agent, Table 7 ↗

按轨迹形态筛选轨迹采样

按轨迹形态筛选轨迹采样

排名 | 轨迹筛选器 | SWE-Bench Verified (100) | OT-TB Lite | Terminal-Bench 2.0 | 平均原始分 | 平均归一化分

1 | 最少轮数 ≥ 5 | 29.00 (1.56) | 19.10 (1.57) | 11.61 (1.30) | 19.90 | +1.25

2 | 筛除超时 | 26.67 (1.70) | 18.03 (1.60) | 10.49 (1.18) | 18.39 | -0.35

3 | 筛除子智能体轨迹 | 23.00 (1.63) | 17.31 (1.58) | 10.86 (1.35) | 17.06 | -0.90

表 7：筛选智能体轨迹采样：保留更长轨迹有所帮助。最少轮数筛选器在全部三个基准上都优于超时筛选和子智能体筛选。

这里效果最好的筛选器只统计轮数，并删除所有少于五轮的运行。

轮数说明一次运行做了多少工作，并不能说明工作做得好不好。

四轮就完成的干净修复会被删掉，四十轮的混乱过程却会被保留。尽管如此，平均而言仍然有帮助。

Raoof 等, 2026 · OpenThoughts-Agent, 表 7 ↗

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Choosing the teacher

Rank	Teacher Model	Benchmarks			Average	
		SWE-bench Verified (100)	OT-TB Lite	Terminal-Bench 2.0	Raw	Normalized
1	GLM 4.7 (Quantized)	28.00 ^{1.76}	17.86 ^{1.60}	8.61 ^{1.40}	18.16	+0.73
2	Kimi K2.5	33.33 ^{1.83}	14.19 ^{1.51}	8.24 ^{1.06}	18.59	+0.66
3	GLM 5	33.00 ^{1.60}	14.30 ^{1.78}	7.50 ^{1.18}	18.27	+0.66
4	GLM 4.6 (Quantized)	26.00 ^{1.67}	16.99 ^{1.80}	6.37 ^{1.24}	16.45	+0.08
5	GPT-5.3-Codex	21.67 ^{1.33}	10.42 ^{1.55}	3.75 ^{0.99}	11.94	-1.47

Table 6: **Teacher model ablation: stronger model $\neq$ better teacher.** Despite GPT-5.3-Codex being the strongest model on these benchmarks, it is the weakest teacher, underperforming GLM 4.7 AWQ by $\sim$ 5pp on Terminal-Bench 2.0.

- The paper reports GPT-5.3-Codex was the strongest of these five models on the benchmarks themselves. Its runs produced the weakest student.
- Leaderboard rank told them the wrong thing here. Trying two teachers and keeping the better student is cheap by comparison.

Raoof et al., 2026 · OpenThoughts-Agent, Table 6 ↗

选择教师模型

选择教师模型

排名 | 教师模型 | SWE-Bench Verified (100) | OT-TB Lite | Terminal-Bench 2.0 | 平均原始分 | 平均归一化分

1 | GLM 4.7 (量化) | 28.00 (1.76) | 17.86 (1.60) | 8.61 (1.40) | 18.16 | +0.73

2 | Kimi K2.5 | 33.33 (1.83) | 14.19 (1.51) | 8.24 (1.06) | 18.59 | +0.66

3 | GLM 5 | 33.00 (1.60) | 14.30 (1.78) | 7.50 (1.18) | 18.27 | +0.66

4 | GLM 4.6 (量化) | 26.00 (1.67) | 16.99 (1.80) | 6.37 (1.24) | 16.45 | +0.08

5 | GPT-5.3-Codex | 21.67 (1.33) | 10.42 (1.55) | 3.75 (0.99) | 11.94 | -1.47

表 6：教师模型消融实验：更强的模型 $\neq$ 更好的教师。尽管 GPT-5.3-Codex 在这些基准上是最强模型，却是最弱的教师；在 Terminal-Bench 2.0 上，它比 GLM 4.7 AWQ 低约 5 个百分点。

论文报告称，GPT-5.3-Codex 在这五个模型中自身的基准表现最强。可它生成的运行却训练出了最弱的学生模型。

排行榜排名在这里给出了错误指引。与此相比，试用两个教师并保留表现更好的学生模型，成本很低。

Raoof 等，2026 · OpenThoughts-Agent，表 6 ↗

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Task sources

Rank	Strategy	Benchmarks			Average	
		SWE-bench Verified (100)	OT-TB Lite	Terminal-Bench 2.0	Raw	Normalized
1	SWESmith	32.33 ^{1.83}	17.63 ^{1.63}	6.37 ^{1.18}	18.78	+1.92
2	StackExchange SuperUser	13.33 ^{1.41}	16.68 ^{1.63}	10.86 ^{1.35}	13.62	+1.51
3	StackExchange Tezos	16.33 ^{1.41}	16.94 ^{1.83}	9.36 ^{1.06}	14.21	+1.45
4	IssueTasks	24.00 ^{1.76}	16.44 ^{1.49}	6.74 ^{1.24}	15.73	+1.37
95	AgentTuning-OS	0.00 ^{0.00}	5.64 ^{1.12}	0.37 ^{0.37}	2.00	-2.31

Table 2: **Task source choice has the largest spread of any pipeline stage.** Issue-resolution tasks and human-written infrastructure questions dominate the top, but improvements are spiky across benchmarks. Full ranking in Appendix A.1.

- The choice of task source changed results the most.
- Different sources help different benchmarks, so spend your effort here before tuning anything else.
- Mix the best four to eight sources. Going wider than that did not help them.

Raoof et al., 2026 · OpenThoughts-Agent, Table 2 ↗

任务来源

任务来源

排名 | 策略 | SWE-Bench Verified (100) | OT-TB Lite | Terminal-Bench 2.0 | 平均原始分 | 平均归一化分

1 | SWESmith | 32.33 (1.83) | 17.63 (1.63) | 6.37 (1.18) | 18.78 | +1.92

2 | StackExchange SuperUser | 13.33 (1.41) | 16.68 (1.63) | 10.86 (1.35) | 13.62 | +1.51

3 | StackExchange Tezos | 16.33 (1.41) | 16.94 (1.83) | 9.36 (1.06) | 14.21 | +1.45

4 | IssueTasks | 24.00 (1.76) | 16.44 (1.49) | 6.74 (1.24) | 15.73 | +1.37

.....

95 | AgentTuning-OS | 0.00 (0.00) | 5.64 (1.12) | 0.37 (0.37) | 2.00 | -2.31

表 2：任务来源的选择在所有流程阶段中造成的结果差异最大。问题解决任务和人工撰写的基础设施问题在排名靠前的位置占主导，但各基准上的提升并不均衡。完整排名见附录 A.1。

任务来源的选择对结果影响最大。

不同来源有助于不同基准，因此应先把精力放在这里，再调整其他内容。

混合表现最好的 4 至 8 个来源。扩大到更多来源并没有带来帮助。

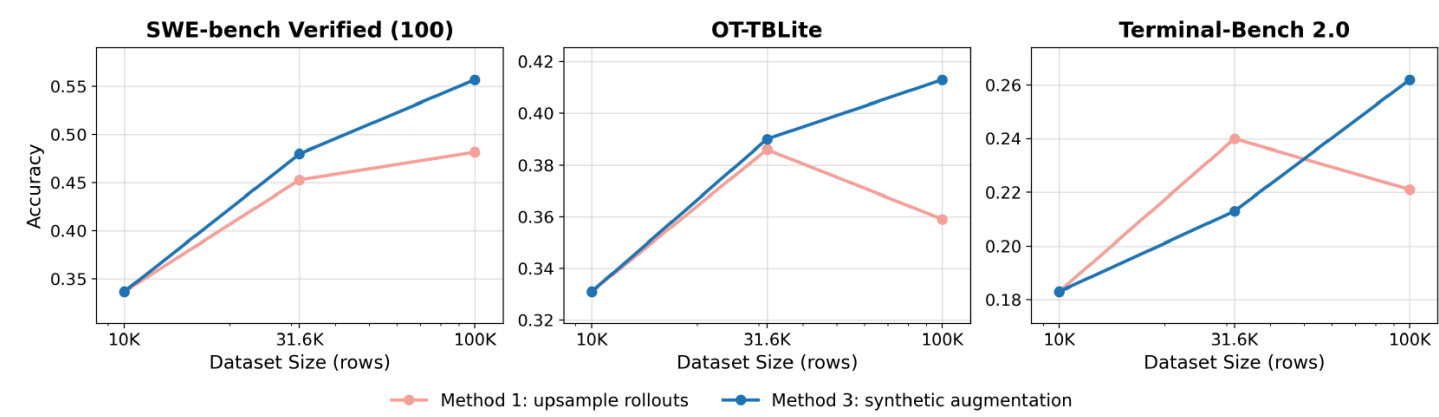
Raoof 等, 2026 · OpenThoughts-Agent, 表 2 ↗

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Scaling the data



- Both curves start from the same 10K dataset and differ only in where the extra data came from.
- The pink curve adds more runs of the same tasks, and it flattens out. The blue curve adds new tasks, and it keeps climbing.

Raoof et al., 2026 · OpenThoughts-Agent, Figure 3 ↗

28

扩展数据规模

SWE-Bench Verified (100) : 准确率; OT-TB Lite: 准确率; Terminal-Bench 2.0: 准确率。三幅图的横轴均为数据集大小 (行数) : 10K、31.6K、100K。

图例: 方法 1——对轨迹采样进行过采样; 方法 3——合成数据增强。

两条曲线都从相同的 10K 数据集开始, 区别仅在于额外数据的来源。

粉色曲线增加了相同任务的更多运行, 随后趋于平缓。蓝色曲线增加了新任务, 并持续上升。

Raoof 等, 2026 · OpenThoughts-Agent, 图 3 ↗

扩展数据规模

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Discussion

Three runs of the same task just came back. You are deciding what goes into the training set.

- A. A short run that solved the issue in four turns.

B. A long run that made a wrong edit, saw the test fail, and recovered.

C. A run that solved the issue after eleven attempts that changed nothing.
- Which of these do you keep?
 - For the ones you keep, which actions carry the loss?
 - Which of your answers would a minimum-turns filter get wrong?

29

讨论

同一个任务刚刚返回了三次运行结果。你正在决定哪些内容进入训练集。

- A. 一次较短的运行，四轮就解决了问题。
- B. 一次较长的运行，先做了错误编辑，看到测试失败后又恢复正确。
- C. 一次运行，经过十一次毫无变化的尝试后才解决问题。

你会保留其中哪些？

对于保留的轨迹，哪些动作参与损失计算？

你哪些判断会被“只保留最少轮数”的筛选器弄错？

讨论：哪些轨迹应进入训练集

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

一种格式适配多种数据集

—
One format for many datasets

30

一种格式适配多种数据集

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/2 部分

Heterogeneity across agent datasets

Dataset	Variety	Count	Source	Note
AgentInstruct (Zeng et al., 2023)	C T W	1.9K	synthetic	Mixture of Browsing, Database, OS, etc.
Code-Feedback (Zheng et al., 2024a)	C	66.4K	manual	Code generation with runtime feedback loops
CodeActInstruct (Wang et al., 2024b)	C	7.1K	synthetic	Code generation and tool use with execution
Go-Browse (Gandhi & Neubig, 2025)	W	9.5K	rollout	Structured exploration web rollouts
Mind2Web (Deng et al., 2023)	W	1.0K	manual	Human web demos on real websites
Nebius SWE Trajectories (Golubev et al., 2024)	S	13.4K	rollout	SWE-agent trajectories from Nebius relying solely on open-weight models
NNetNav-live (Murty et al., 2024)	W	5.0K	rollout	Retroactively labeled live web exploration
NNetNav-wa (Murty et al., 2024)	W	4.2K	rollout	Retroactively labeled WebArena exploration
openhands-feedback (All Hands AI, 2024)	C T W	0.2K	rollout	Recorded OpenHands agent trajectories with human feedback
Orca AgentInstruct (Mitra et al., 2024)	T	1046.1K	synthetic	Large-scale synthetic tool-use instructions data
SWE-Gym (Pan et al., 2025)	S	0.5K	rollout	Agent trajectories solving real GitHub repo tasks
SWE-smith (Yang et al., 2025b)	S	5.0K	synthetic	Trajectories of agents on synthesized bug-fix tasks
Synatra (Ou et al., 2024)	W	99.9K	synthetic	Synthetically created web demos of tutorials

- Each dataset was built by a different group, in its own format.
- For a web page, some save the HTML and others save the accessibility tree.
- To train on two of them together, you first write two customized converters.

Song et al., 2026 · Agent Data Protocol, Table 1 ↗

智能体数据集之间的异质性

智能体数据集之间的异质性

数据集 | 多样性 | 数量 | 来源 | 说明

AgentInstruct (Zeng 等, 2023) | C、T、W | 1.9K | 合成 | 浏览、数据库、操作系统等内容的混合

Code-Feedback (Zeng 等, 2024a) | C | 66.4K | 人工 | 带运行时反馈循环的代码生成

CodeActInstruct (Wang 等, 2024b) | C | 7.1K | 合成 | 带执行能力的代码生成与工具使用

Go-Browse (Gandhi 与 Neubig, 2025) | W | 9.5K | 轨迹采样 | 结构化的网页探索轨迹采样

Mind2Web (Deng 等, 2023) | W | 1.0K | 人工 | 真实网站上的人工网页操作示范

Nebius SWE Trajectories (Golubev 等, 2024) | S | 13.4K | 轨迹采样 | 来自 Nebius、完全依靠开放权重模型的 SWE-agent 轨迹

NNetNav-live (Murty 等, 2024) | W | 5.0K | 轨迹采样 | 对实时网页探索进行回溯式标注

NNetNav-wa (Murty 等, 2024) | W | 4.2K | 轨迹采样 | 对 WebArena 探索进行回溯式标注

openhands-feedback (All Hands AI, 2024) | C、T、W | 0.2K | 轨迹采样 | 记录下来的 OpenHands 智能体轨迹，带有人类反馈

Orca AgentInstruct (Mitra 等, 2024) | T | 1046.1K | 合成 | 大规模合成的工具使用指令数据

SWE-Gym (Pan 等, 2025) | S | 0.5K | 轨迹采样 | 智能体解决真实 GitHub 仓库任务的轨迹

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 2/2 部分

Heterogeneity across agent datasets

Dataset	Variety	Count	Source	Note
AgentInstruct (Zeng et al., 2023)	C T W	1.9K	synthetic	Mixture of Browsing, Database, OS, etc.
Code-Feedback (Zheng et al., 2024a)	C	66.4K	manual	Code generation with runtime feedback loops
CodeActInstruct (Wang et al., 2024b)	C	7.1K	synthetic	Code generation and tool use with execution
Go-Browse (Gandhi & Neubig, 2025)	W	9.5K	rollout	Structured exploration web rollouts
Mind2Web (Deng et al., 2023)	W	1.0K	manual	Human web demos on real websites
Nebius SWE Trajectories (Golubev et al., 2024)	S	13.4K	rollout	SWE-agent trajectories from Nebius relying solely on open-weight models
NNetNav-live (Murtu et al., 2024)	W	5.0K	rollout	Retroactively labeled live web exploration
NNetNav-wa (Murtu et al., 2024)	W	4.2K	rollout	Retroactively labeled WebArena exploration
openhands-feedback (All Hands AI, 2024)	C T W	0.2K	rollout	Recorded OpenHands agent trajectories with human feedback
Orca AgentInstruct (Mitra et al., 2024)	T	1046.1K	synthetic	Large-scale synthetic tool-use instructions data
SWE-Gym (Pan et al., 2025)	S	0.5K	rollout	Agent trajectories solving real GitHub repo tasks
SWE-smith (Yang et al., 2025b)	S	5.0K	synthetic	Trajectories of agents on synthesized bug-fix tasks
Synatra (Ou et al., 2024)	W	99.9K	synthetic	Synthetically created web demos of tutorials

- Each dataset was built by a different group, in its own format.
- For a web page, some save the HTML and others save the accessibility tree.
- To train on two of them together, you first write two customized converters.

Song et al., 2026 · Agent Data Protocol, Table 1 ↗

智能体数据集之间的异质性

SWE-smith (Yang 等, 2025b) | S | 5.0K | 合成 | 智能体解决合成缺陷修复任务的轨迹

Synatra (Ou 等, 2024) | W | 99.9K | 合成 | 合成的教程网页演示

每个数据集都由不同团队以各自的格式构建。

对于网页，有些数据集保存 HTML，另一些则保存可访问性树。

要将两个数据集一起用于训练，首先得分别编写两个定制转换器。

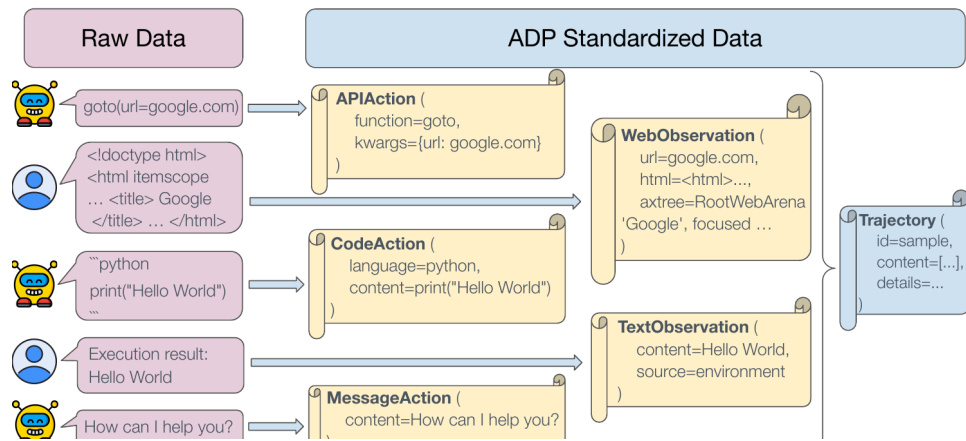
Song 等, 2026 · Agent Data Protocol, 表 1 ↗

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

One representation for a trajectory



- You read each dataset once and rewrite it as typed actions and observations.
- For the web page we just saw, keep the HTML and the accessibility tree, instead of picking one.
- Each dataset converts once, each harness once, not once per pair.

Song et al., 2026 · Agent Data Protocol, ICLR oral ↗

32

一种轨迹表示方式

一种轨迹表示方式

原始数据 → ADP 标准化数据

原始数据示例: goto(url=google.com) → APIAction(function=goto, kwargs={url: google.com})

原始数据示例: <!doctype html>、<html itemscope> ... <title> Google </title> ... </html> → WebObservation(url=google.com, html=<html>... , axtree=RootWebArena 'Google', focused ...)

原始数据示例: '''python print("Hello World")''' → CodeAction(language=python, content=print("Hello World"))

原始数据示例: Execution result: Hello World → TextObservation(content=Hello World, source=environment)

原始数据示例: How can I help you? → MessageAction(content=How can I help you?)

统一轨迹对象: Trajectory(id=sample, content=[...], details=...)

你只需读取每个数据集一次，然后将其改写为带类型的动作与观测。

对于刚才看到的网页，同时保留 HTML 和可访问性树，而不是二选一。

每个数据集只转换一次，每种 执行框架（harness） 只转换一次，而非每一对组合各转换一次。

Song 等, 2026 · Agent Data Protocol, ICLR 口头报告 ↗

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

Harness-specific rendering

- Once the data is in one format, you still have to write it back out for the harness you are training for.
- OpenHands, SWE-Agent and AgentLab take different actions, so the same trajectory comes out looking different in each.
- This is also where you set the system prompt and decide what to do when the context gets long.
- Then you check the result: do the tool calls parse, does each call come with a thought, does the conversation end the way it should.

[Song et al., 2026 · Agent Data Protocol, § 3.3 ↗](#)

33

ZH / 中文译文 · 第 1/1 部分

针对特定执行框架进行渲染

数据一旦进入统一格式，你仍需将其重新写出为目标训练 执行框架（harness）所需的格式。

OpenHands、SWE-Agent 和 AgentLab 采用不同的动作，因此同一条轨迹在每个框架中渲染出来都不一样。

这也是设置 system 提示词，以及决定上下文变长时如何处理的地方。

然后检查结果：工具调用能否解析？每次调用是否都附带思考？对话是否以正确方式结束？

Song 等, 2026 · Agent Data Protocol, 第 3.3 节 ↗

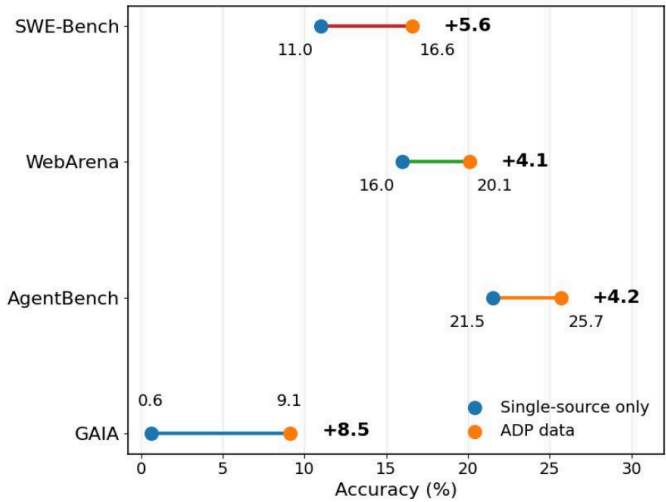
针对特定执行框架进行渲染

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Diversity versus task-specific data



- Same harness, same model, same evaluation. Only the training mixture changes.
- Train on the mixture and you beat the matching single-domain set, even on that domain's own benchmark.

[Song et al., 2026 · Agent Data Protocol, cross-task transfer ↗](#)

34

多样性与任务专用数据

图表：准确率（%）；基准与单一来源数据 / ADP 数据的结果：SWE-Bench：11.0 / 16.6（+5.6）；WebArena：16.0 / 20.1（+4.1）；AgentBench：21.5 / 25.7（+4.2）；GAIA：0.6 / 9.1（+8.5）。

图例：仅单一来源；ADP 数据。

执行框架相同、模型相同、评估方式相同。只有训练数据混合发生了变化。

使用混合数据训练后，表现超过与目标领域对应的单一领域数据集，甚至在该领域自己的基准上也是如此。

Song 等, 2026 · Agent Data Protocol, 跨任务迁移 ↗

多样性与任务专用数据

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

运行训练

Running the training

35

运行训练

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Real SFT configurations

	SEQUENCE	BATCH	LEARNING RATE
Nemotron 3 Ultra	packed 294,912 then 515,000	64	1.5e-5 → 1e-6
MAI self-distillation	packed 128k	2,048	1.7e-5 → 5.2e-6, 2% warmup
K2 Horizon	512K, three phases	—	decayed on a high-quality subset

[Nemotron 3 Ultra, § 3.1 ↗](#)

[MAI-Thinking-1, § 3.1.4 and § 3.5 ↗](#)

[K2 Horizon model card ↗](#)

真实的 SFT 配置

真实的 SFT 配置

模型 | 序列长度 | 批次 | 学习率

Nemotron 3 Ultra | 打包后 294,912, 然后 515,000 | 64 | 1.5e-5 → 1e-6

MAI 自蒸馏 | 打包后 128k | 2,048 | 1.7e-5 → 5.2e-6, 预热 2%

K2 Horizon | 512K, 三个阶段 | — | 在高质量子集上衰减

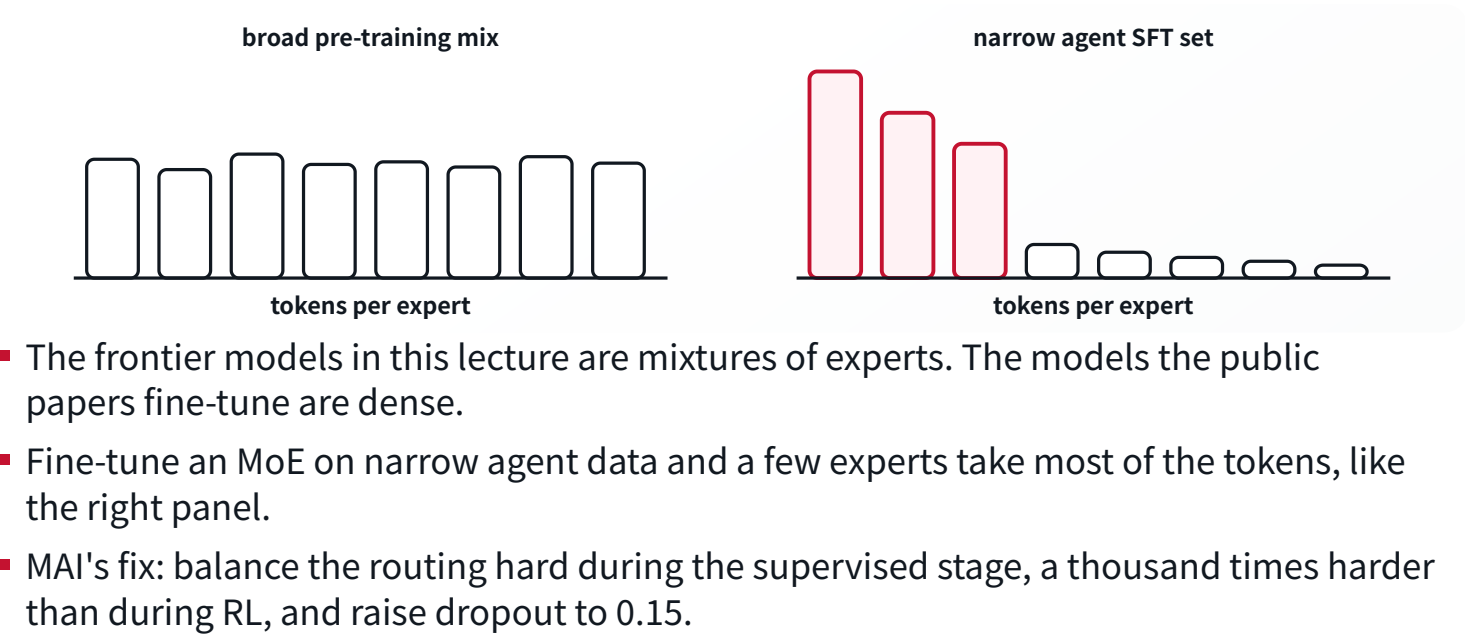
Nemotron 3 Ultra, 第 3.1 节 ↗ · MAI-Thinking-1, 第 3.1.4 与 3.5 节 ↗ · K2 Horizon 模型卡 ↗

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Expert routing during SFT



Microsoft AI, 2026 · MAI-Thinking-1, § 3.1.5 ↗

SFT 期间的专家路由

左图：宽泛的预训练数据混合；右图：狭窄的智能体 SFT 数据集；纵轴：每个专家处理的词元数量。

本讲中的前沿模型采用混合专家（MoE）架构。公开论文中进行微调的模型则是稠密模型。

在狭窄的智能体数据上微调 MoE 时，少数专家会处理大部分词元，就像右侧图示一样。

MAI 的解决办法：监督阶段要强力平衡路由，其力度是 RL 阶段的一千倍，并把 dropout 提高到 0.15。

Microsoft AI, 2026 · MAI-Thinking-1, 第 3.1.5 节 ↗

SFT 期间的专家路由

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Packing whole conversations

Training runs on fixed-length sequences. Packing fills each one with several conversations, instead of padding most of it away.



- Each conversation is assigned to the pack whose remaining capacity it most tightly fits.
- No conversation is split or truncated, which is a deliberate choice against hallucination.
- Identical prompts are kept out of the same pack, and packs are shuffled afterwards.

NVIDIA, 2026 · Nemotron 3 Ultra, § 3.1.2 ↗

打包完整对话

训练使用固定长度的序列。打包会用多段对话填满每条序列，而不是把大部分位置都填充掉。

打包 1：19k 运行、15k 运行、11k 运行、8k 运行、未使用空间。

打包 2：29k 运行、17k 运行、7k 运行、未使用空间。

每段对话都会分配给剩余容量最适合容纳它的打包序列。

不会拆分或截断任何对话，这是有意作出的选择，以避免产生幻觉。

相同的提示词不会放进同一个打包序列，之后还会打乱各个打包序列的顺序。

NVIDIA, 2026 · Nemotron 3 Ultra, 第 3.1.2 节 ↗

打包完整对话

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Template drift

- The template you train with has to be the template you serve with, down to the whitespace.
- Render the same trajectory twice, once with a tool message after it and once without, and you can get two different strings.
- When a tool message is appended, a conditional thinking block changes the earlier assistant turn, and the prefix no longer matches.

"The chat template should always match the format the model was trained with."

[Transformers · writing chat templates ↗](#)

[TRL · training chat templates ↗](#)

39

模板漂移

训练时使用的模板必须与部署服务时使用的模板一致，连空白字符也要一致。

将同一条轨迹渲染两次，一次在后面接工具消息，一次不接，可能得到两个不同的字符串。

追加工具消息时，条件式思考区块会改变较早的 assistant 轮次，导致前缀不再匹配。

“聊天模板应始终与模型训练时使用的格式相匹配。”

[Transformers · 编写聊天模板 ↗](#) · [TRL · 训练聊天模板 ↗](#)

模板漂移

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Released training artifacts

- K2 Horizon's model card has a stage-by-stage training table: steps, tokens, sequence length, and what each phase was for. That table is where the numbers on the configurations slide came from.
- It also links the Weights & Biases run, so you can look at the actual loss curves, and it ships intermediate checkpoints for every stage.
- Nemotron released the post-trained checkpoints together with the training data and the recipe.
- Reading one of these teaches a lot.

[Nemotron 3 Ultra ↗](#)

[K2 Horizon ↗](#)

40

已发布的训练工件

K2 Horizon 的模型卡提供了逐阶段的训练表格，列出训练步数、词元数、序列长度，以及各阶段的目的。配置页中的数字就来自那张表。

模型卡还链接到 Weights & Biases 运行记录，因此你可以查看实际的损失曲线；它也提供每个阶段的中间检查点。

Nemotron 发布了后训练检查点，以及训练数据和训练方案。

阅读这些资料会让人学到很多。

[Nemotron 3 Ultra ↗](#) · [K2 Horizon ↗](#)

已发布的训练工件

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

判断训练是否奏效



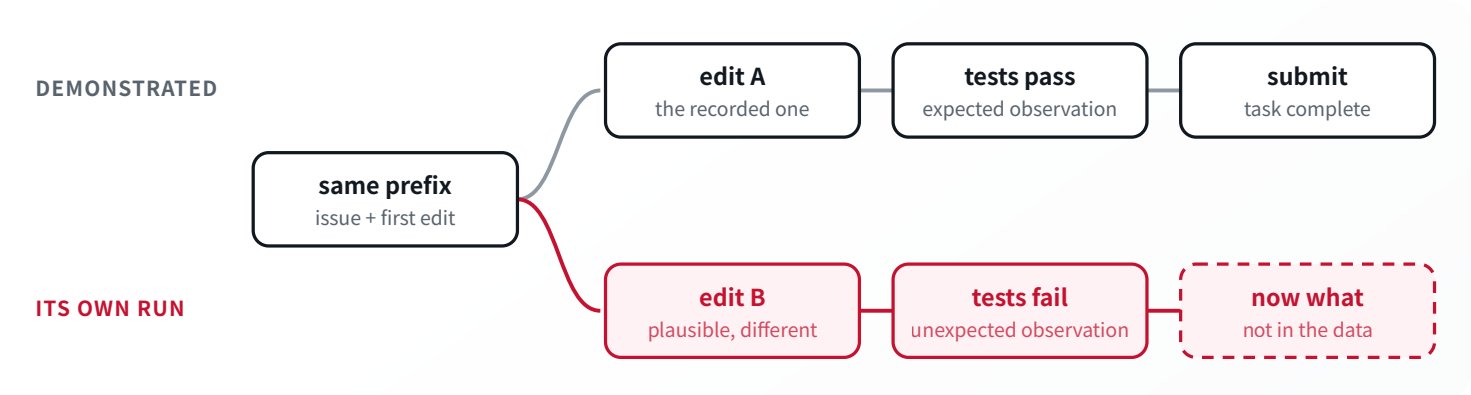
判断训练是否奏效

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

The mismatch between training and running



- Training grades one action at a time, always continuing the recorded run.
- Deployed, there is no recording. Every action the model takes changes what it sees next.
- One different edit and it is in a state no training run ever visited. Low loss promised nothing about what happens there.

Ross et al., 2011 · DAgger ↗

训练与运行之间的不匹配

示范运行：编辑 A → 测试通过 → 提交；记录下来的运行 → 预期观测 → 任务完成。

相同前缀：问题 + 第一次编辑。

智能体自己的运行：编辑 B → 测试失败 → 接下来怎么办？；看似合理但不同 → 意外观测 → 训练数据中没有。

训练时逐个评估动作，并始终沿着记录下来的运行继续。

部署后没有预先记录的轨迹。模型采取的每个动作都会改变它接下来看到的内容。

只差一个编辑，就会进入任何训练运行都未曾到达的状态。低损失并不能保证模型在这种状态下会如何表现。

Ross 等, 2011 · DAgger ↗

训练与运行之间的不匹配

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Evaluation in the target harness

- Before trusting any number, check that you can overfit twenty examples. If the loss will not drop, the bug is upstream.
- Then evaluate by running the model in a harness, on tasks with checkable outcomes.
- Read published numbers carefully. Sometimes they are the best of several harnesses.
- Match the split to the claim. Keep all runs of one task on the same side, and split by repository if you want to say it generalizes to new ones.

[OpenThoughts-Agent, Table 1 ↗](#)

[SWE-Gym, § 5.2 ↗](#)

43

在目标执行框架中评估

在相信任何数字之前，先检查模型能否过拟合 20 个样例。如果损失降不下来，问题出在前面的流程。

然后让模型在 执行框架（harness） 中运行，并使用结果可验证的任务进行评估。

仔细阅读已发表的数字。有时它们是多个 执行框架（harness） 中的最佳结果。

让数据划分与所声称的结论相匹配。同一任务的所有运行都要放在同一侧；若要声称模型能泛化到新仓库，则按仓库划分。

OpenThoughts-Agent, 表 1 ↗ · SWE-Gym, 第 5.2 节 ↗

在目标执行框架中评估

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Robustness across harnesses

- Trajectories collected in one system teach that system's conventions along with the task.
- Collecting across several harnesses is how labs avoid training a model that only works in one.

[NVIDIA, 2026 · Nemotron 3 Ultra, § 3.1.1 ↗](#)

跨执行框架的稳健性

在一个系统中收集的轨迹，不仅教会模型任务，也教会它该系统的惯例。

跨多个 执行框架（harness） 收集数据，是实验室避免训练出只能在一种框架中运行的模型的方法。

NVIDIA, 2026 · Nemotron 3 Ultra, 第 3.1.1 节 ↗

跨执行框架的稳健性

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Regressions outside the target domain

- Train on one capability and you move the others, whether or not you measure it.
- Your mixture is the control here, and balancing it by examples is not the same as balancing it by tokens.
- So evaluate the things you were not trying to improve, too.

Microsoft AI, 2026 · MAI-Thinking-1, § 3.5 ↗

目标领域之外的性能退化

针对一种能力进行训练，也会改变其他能力，无论你是否测量了这些变化。

数据混合在这里充当对照；按样例均衡与按词元均衡并不是一回事。

因此，也要评估那些你原本没有打算提升的能力。

Microsoft AI, 2026 · MAI-Thinking-1, 第 3.5 节 ↗

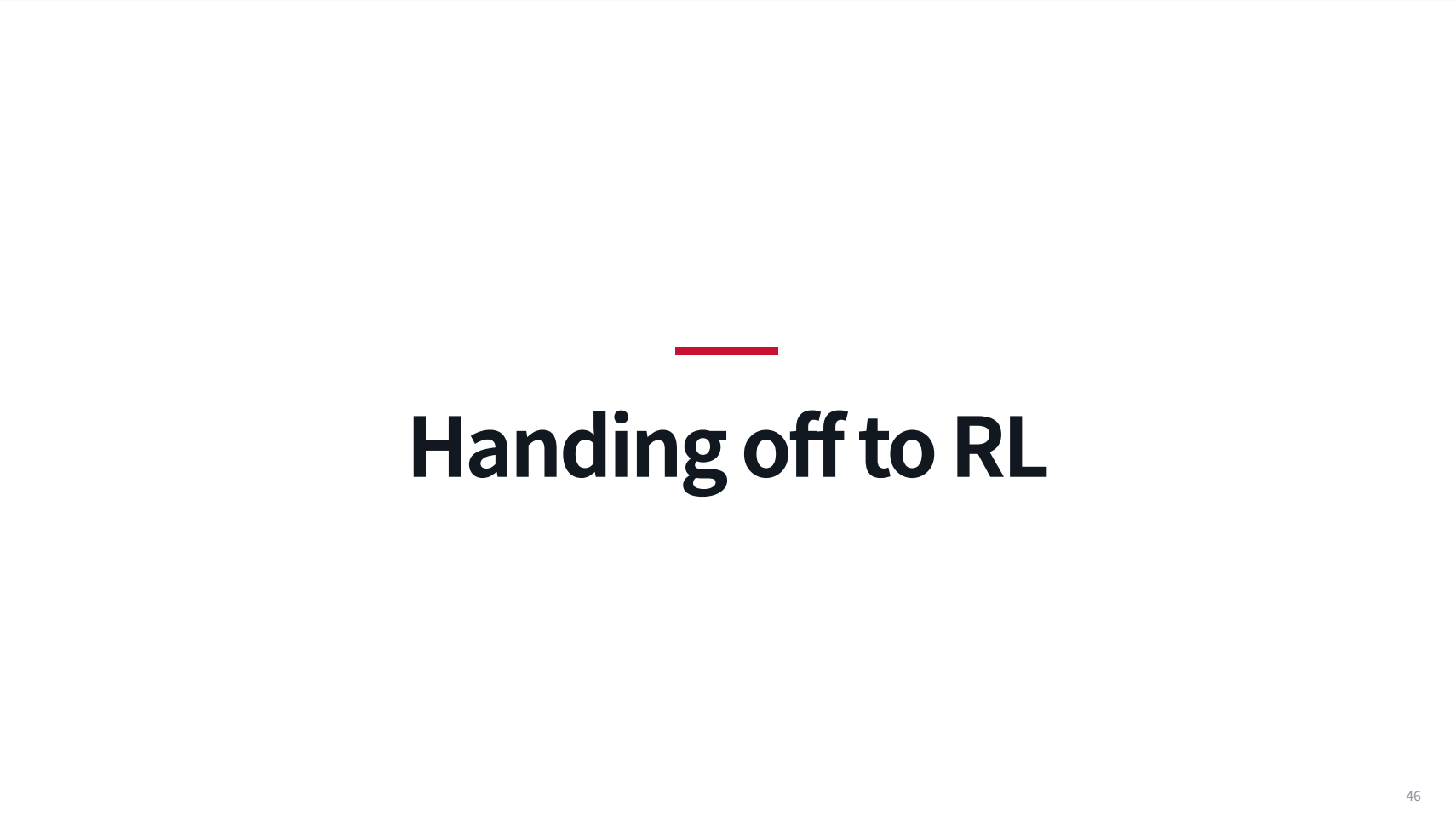
目标领域之外的性能退化

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

交接至强化学习（RL）



交接至强化学习

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

How much SFT before RL

Rank	Model (training recipe)	Benchmarks			Average	
		SWE-bench Verified (100)	OT-TB Lite	Terminal-Bench 2.0	Raw	Normalized
1	OT-Agent-ColdSFT+RL-8B	35.7 ^{1.8}	16.0 ^{1.6}	13.5 ^{1.5}	21.7	+1.2
2	OT-Agent-SFT-8B (10K)	24.3 ^{1.9}	15.6 ^{1.5}	7.9 ^{1.4}	15.9	+0.5
3	OT-Agent-ColdSFT	23.7 ^{1.6}	14.8 ^{1.4}	6.7 ^{1.1}	15.1	+0.4
4	RL on Qwen3-8B (no SFT)	1.0 ^{0.5}	7.8 ^{1.1}	1.9 ^{0.7}	3.6	-0.9
5	Qwen3-8B	5.3 ^{1.1}	1.3 ^{0.3}	1.5 ^{0.4}	2.7	-1.1

- RL applied to no SFT at all barely moves the base model.
- SFT alone stops well short of what the pair reaches.
- The winning recipe starts RL from a checkpoint that was deliberately not trained to its own best score.

"RL provides the most gains when the SFT model is selected with RL in mind."

[OpenThoughts-Agent, Table 11 ↗](#)

[MAI-Thinking-1, § 3.1.4 ↗](#)

47

强化学习之前需要多少监督微调

排名 | 模型 (训练方案) | SWE-Bench Verified (100) | OT-TB Lite | Terminal-Bench 2.0 | 平均原始分 | 平均归一化分

1 | OT-Agent-ColdSFT+RL-8B | 35.7 (1.8) | 16.0 (1.6) | 13.5 (1.5) | 21.7 | +1.2

2 | OT-Agent-SFT-8B (10K) | 24.3 (1.9) | 15.6 (1.5) | 7.9 (1.4) | 15.9 | +0.5

3 | OT-Agent-ColdSFT | 23.7 (1.6) | 14.8 (1.4) | 6.7 (1.1) | 15.1 | +0.4

4 | 在 Qwen3-8B 上进行 RL (无 SFT) | 1.0 (0.5) | 7.8 (1.1) | 1.9 (0.7) | 3.6 | -0.9

5 | Qwen3-8B | 5.3 (1.1) | 1.3 (0.3) | 1.5 (0.4) | 2.7 | -1.1

完全不做 SFT、直接进行 RL，几乎无法改变基础模型。

单独使用 SFT 的效果远不及 SFT 与 RL 组合。

最佳方案从一个检查点开始做 RL，而这个检查点刻意没有训练到自身最佳分数。

“当 SFT 模型是以 RL 为目标进行选择时，RL 带来的增益最大。”

OpenThoughts-Agent, 表 11 ↗ · MAI-Thinking-1, 第 3.1.4 节 ↗

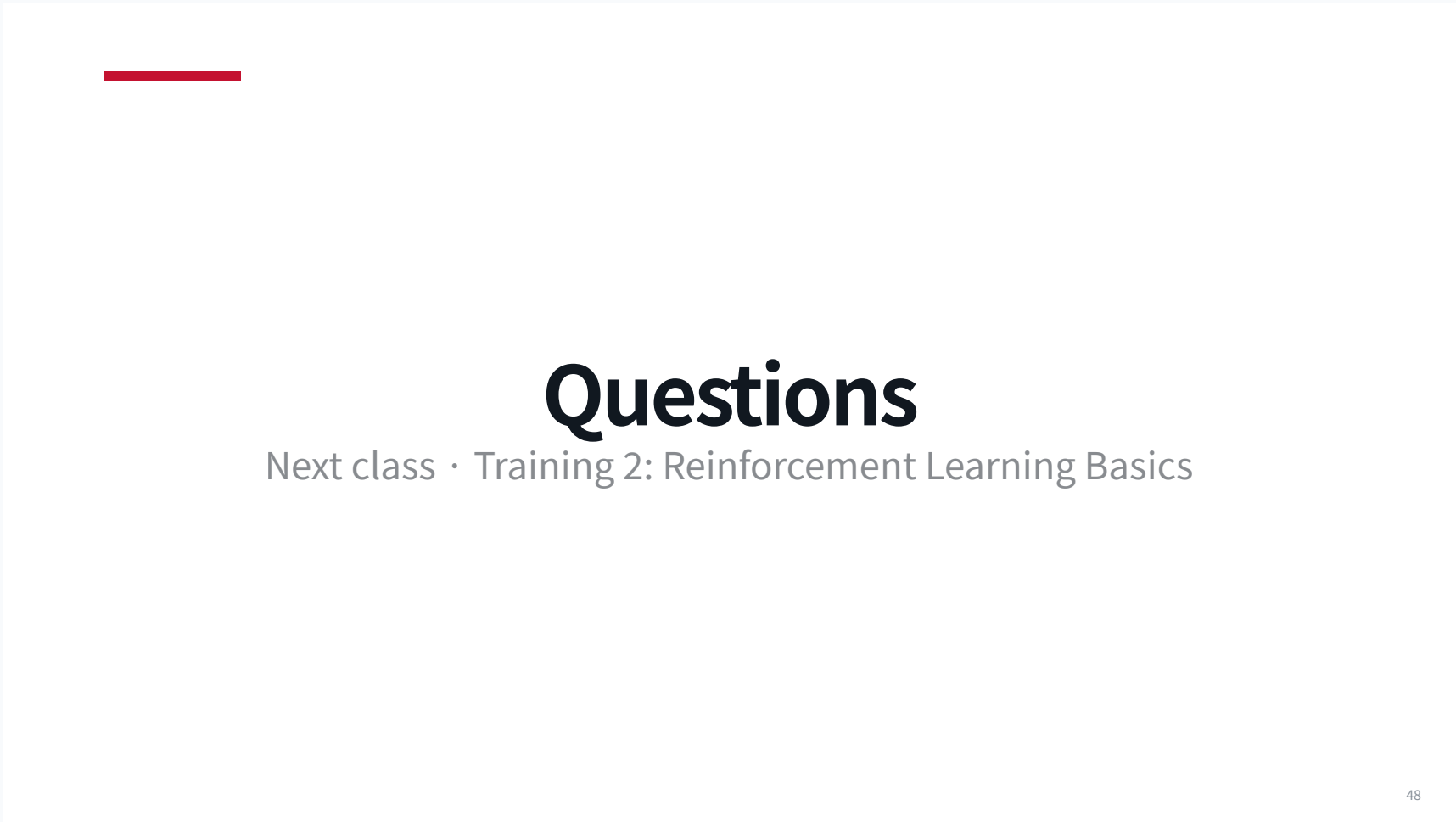
第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

问题

下一讲 · 训练 2：强化学习基础



问答

第 8 讲 · 智能体的监督微调

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Backup · Exact masking flags

FRAMEWORK	FLAG	BEHAVIOR
TRL	assistant_only_loss=True	needs {% generation %} markers; known families get patched templates
TRL	completion_only_loss	prompt-completion datasets; on by default there
Axolotl	roles_to_train: ["assistant"]	default; loss on the listed roles only
Axolotl	train_on_eos: turn	the EOS of each trained turn is in the loss
LLaMA-Factory	train_on_prompt: false	default; prompt tokens masked
LLaMA-Factory	mask_history: true	last turn only; truncation keeps the final turn

49

附录 · 精确的掩码标志

框架 | 标志 | 行为

TRL | assistant_only_loss=True | 需要 {% generation %} 标记；已知系列会对模板进行修补

TRL | completion_only_loss | 用于提示词—补全数据集；在此类数据集中默认开启

Axolotl | roles_to_train: ["assistant"] | 默认值；仅对列出的角色计算损失

Axolotl | train_on_eos: turn | 每个受训练轮次的 EOS 都计入损失

LLaMA-Factory | train_on_prompt: false | 默认值；提示词词元会被屏蔽

LLaMA-Factory | mask_history: true | 仅训练最后一轮；截断时保留最后一轮

附录：精确的掩码标志