Let It Flow: Agentic Crafting on Rock and Roll

Arxiv：https://arxiv.org/pdf/2512.24873，阿里

分享：汪婧昀

CLI 任务

- CLI ＝ Command Line Interface（命令行界面）

- 你可以理解为：让 AI 坐在一台 Linux 终端前，像程序员一样敲命令、跑代码、调 Bug、修仓库。

CLI 任务

- **典型的 CLI 任务例子：**
  - 用户说："帮我找到这个 Python 项目的 Bug 并修复它"
  - AI 需要：
    - 1. ls / cat 查看文件结构
    - 2. 读懂代码
    - 3. 定位问题（可能在哪个文件哪一行）
    - 4. 写 patch 修复
    - 5. 跑测试验证
    - 6. 如果失败 → 反思 → 重试

CLI 任务

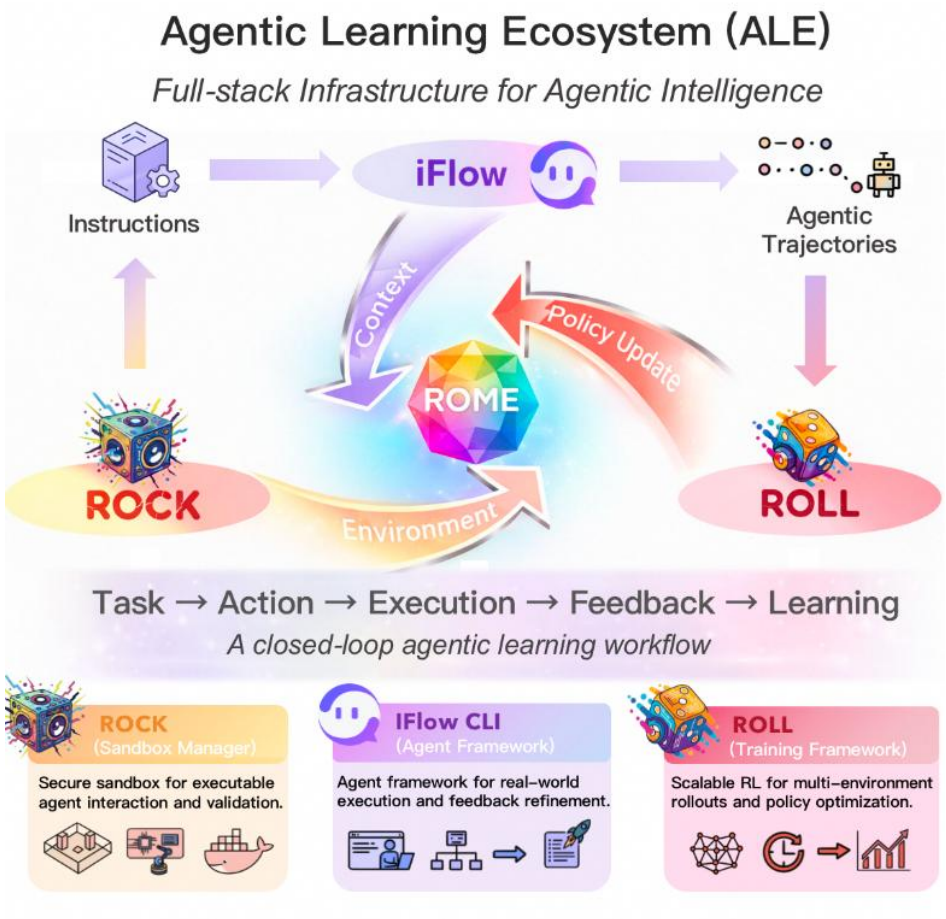
- 与普通LLM/Agent的核心区别：

维度	普通 LLM	CLI Agent（本文研究对象）
交互轮次	单轮输出	多轮（几十轮）迭代
反馈来源	无	真实运行结果（报错/通过）
工具使用	无	bash、文件系统、测试框架
目标	生成文本	完成任务（可验证）

CLI 任务

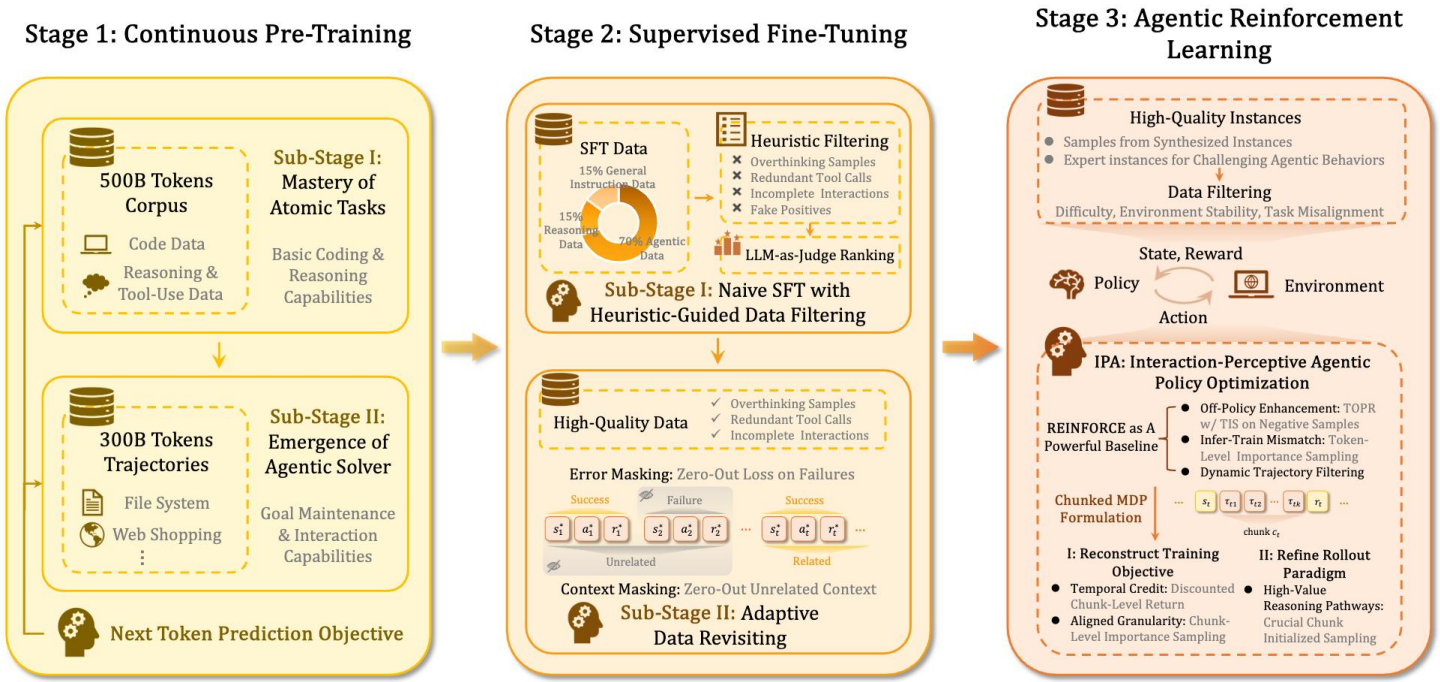
- **核心评测集：**
  - SWE-bench Verified（修复真实 GitHub Issue）
  - Terminal-Bench（纯终端操作任务）

整体框架



- 三大系统的分工
  - ROCK：提供安全的沙箱环境，AI 在里面跑代码不会炸服务器
  - iFlow CLI：是 AI 和环境交互的"操作系统"，管理上下文、工具调用
  - ROLL：RL训练引擎，把刷出来的轨迹转化为模型能力
  - ROME：最终模型，30B-MoE，激活参数 3B

训练流程



- 三个阶段

阶段一：CPT（持续预训练）

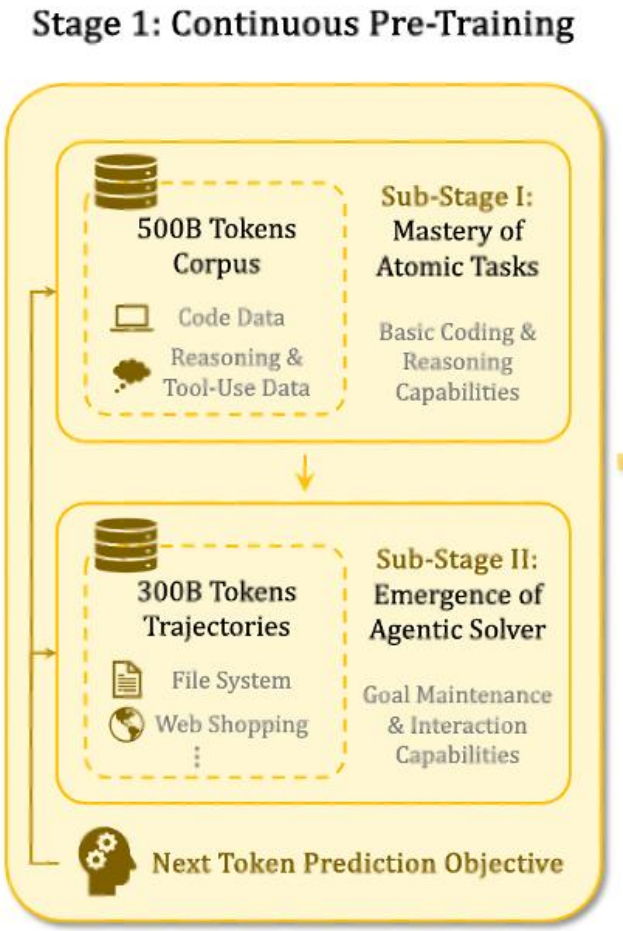
↓

阶段二：SFT

↓

阶段三：RL（IPA 算法）

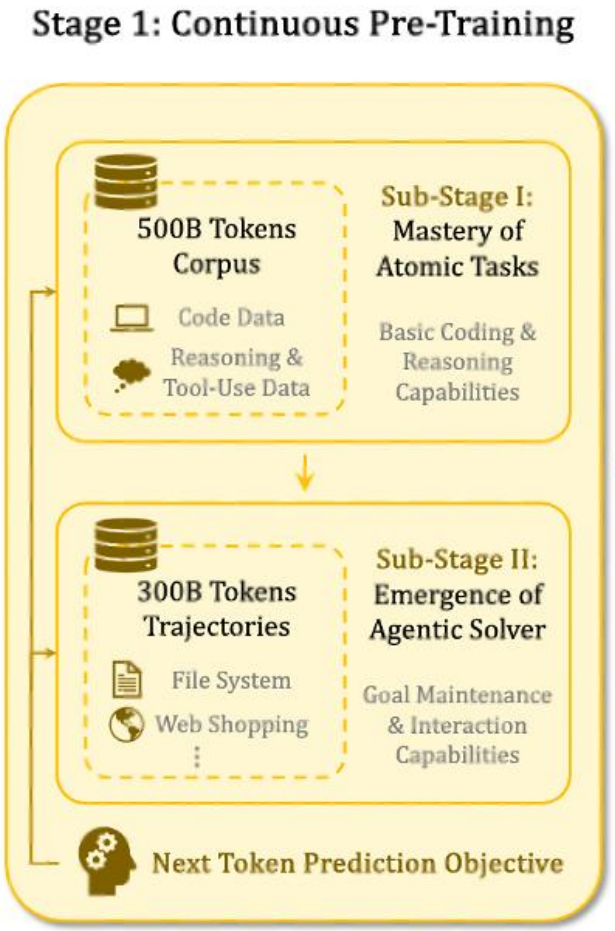
预训练（CPT）



- 核心目标：让模型从"会写代码"升级到"会做工程任务"
- 子阶段
  - Stage I（500B tokens）：原子任务掌握
  - Stage II（300B tokens）：智能体涌现

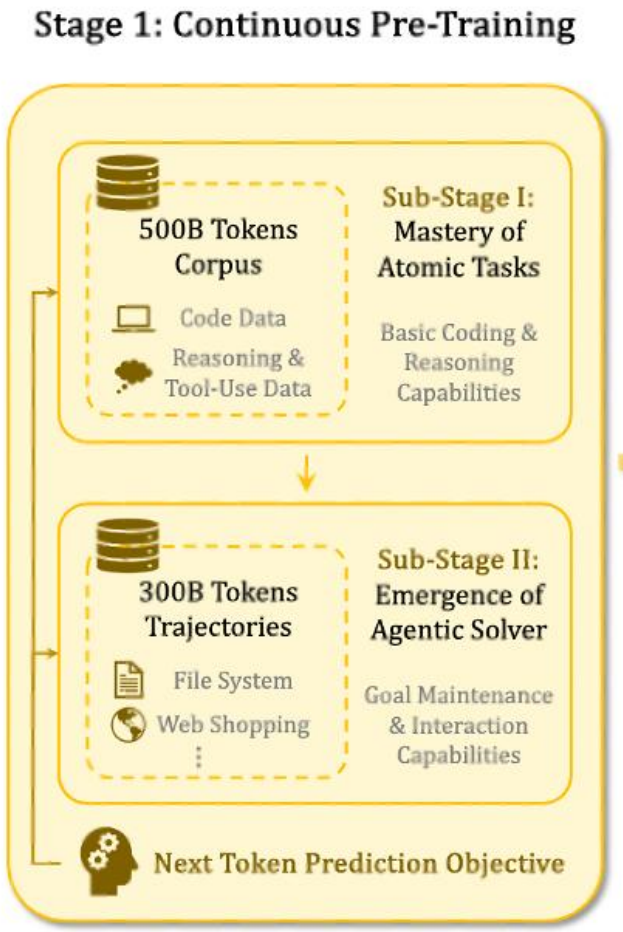
预训练（CPT）— Stage Ⅰ

让模型学会静态分析



- Stage Ⅰ 数据构成：
  - 代码工程任务（主要）
    - 任务：
      - Bug 定位：给 Issue + 仓库结构 → 找哪个文件出问题
      - 代码修复：给出 search-and-replace 格式的修改
      - 单元测试生成：针对修复写验证测试
    - 数据来源：PR 评论作为 feedback，commit 作为 response
    - 蒸馏大模型的CoT
  - 通用推理数据（辅助）
    - 数学推理
    - 逻辑谜题
    - 工具使用示例

预训练（CPT） — Stage II Mid-Train，学会动态执行

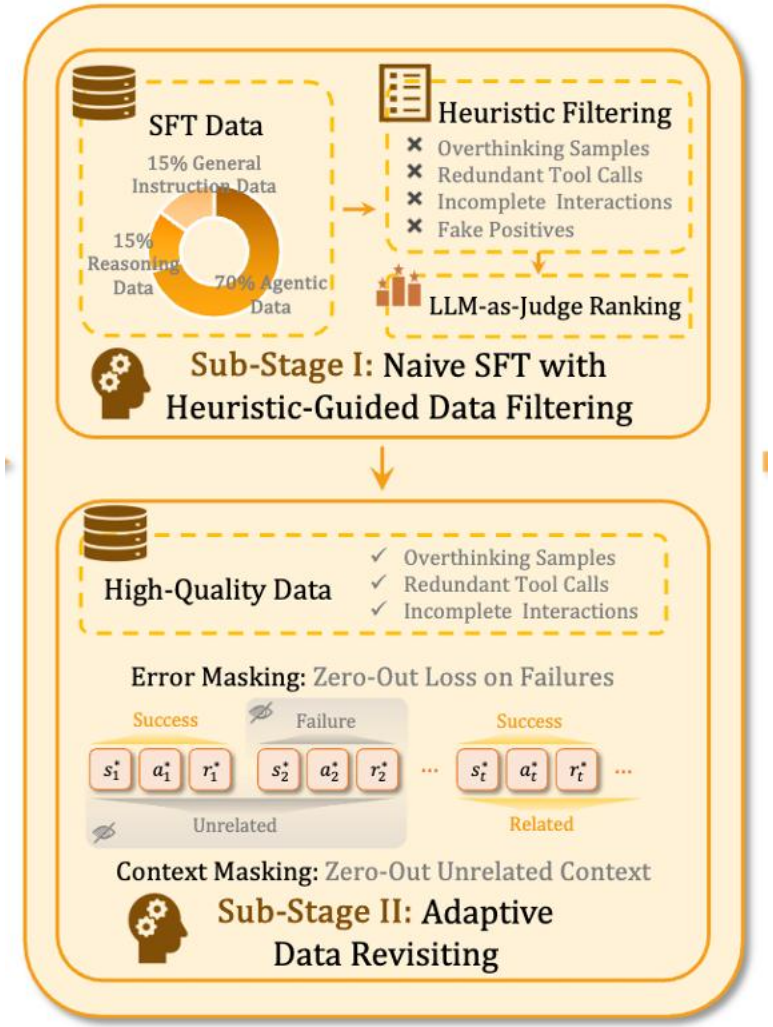


- Stage II 数据构成：孤立代码片段 -> 完整交互轨迹
  - 轨迹来源
    - 强教师模型（Qwen3-Coder-480B、Claude）在沙箱环境中的执行轨迹
    - 环境包括：文件系统、网络购物模拟器等
    - 同时包含：成功轨迹 + 失败后修正的轨迹（让模型学会从错误恢复）
  - 关键能力
    - 维持长期目标（不迷失方向）
    - 探索高维决策空间
    - 从环境反馈中自我修正

SFT

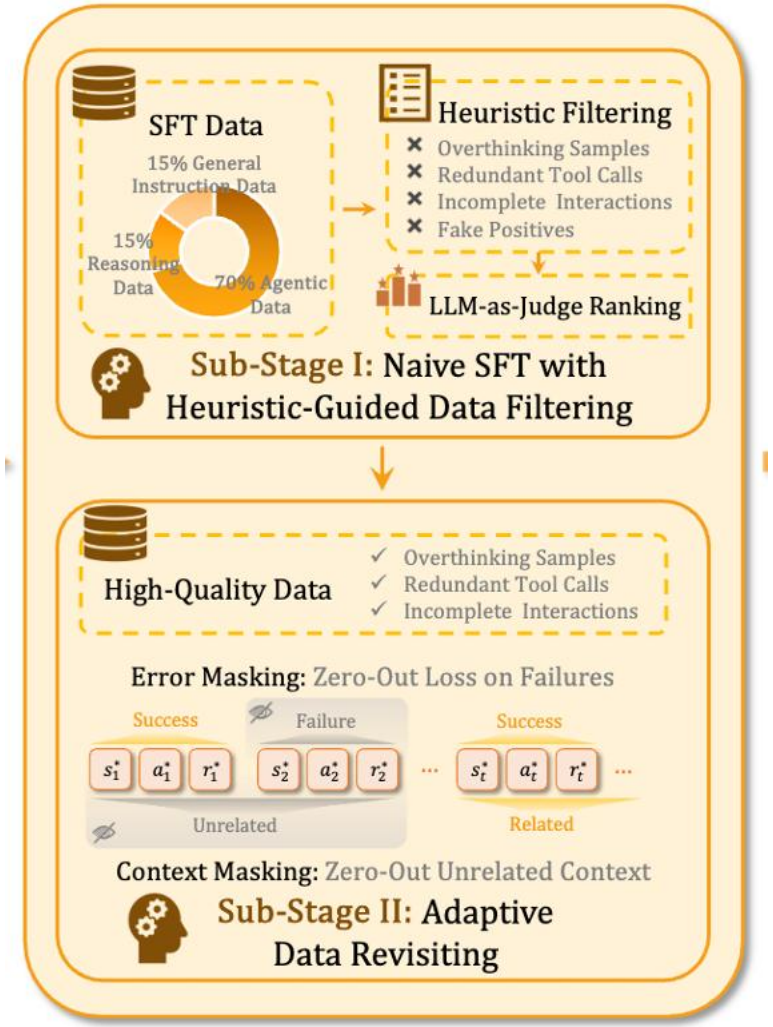
- 传统SFT在CLI／Agent 场景下的问题
  - 1．执行失败的 turn 也会产生梯度 → 模型会"学会犯错"
  - 2．多个子任务的上下文混在一起 → 模型学到混乱的对齐行为
  - 3．过度思考（verbose reasoning）→ 降低执行效率
  - 4．假正例（fake positives）→ 通过了测试但逻辑有错

SFT



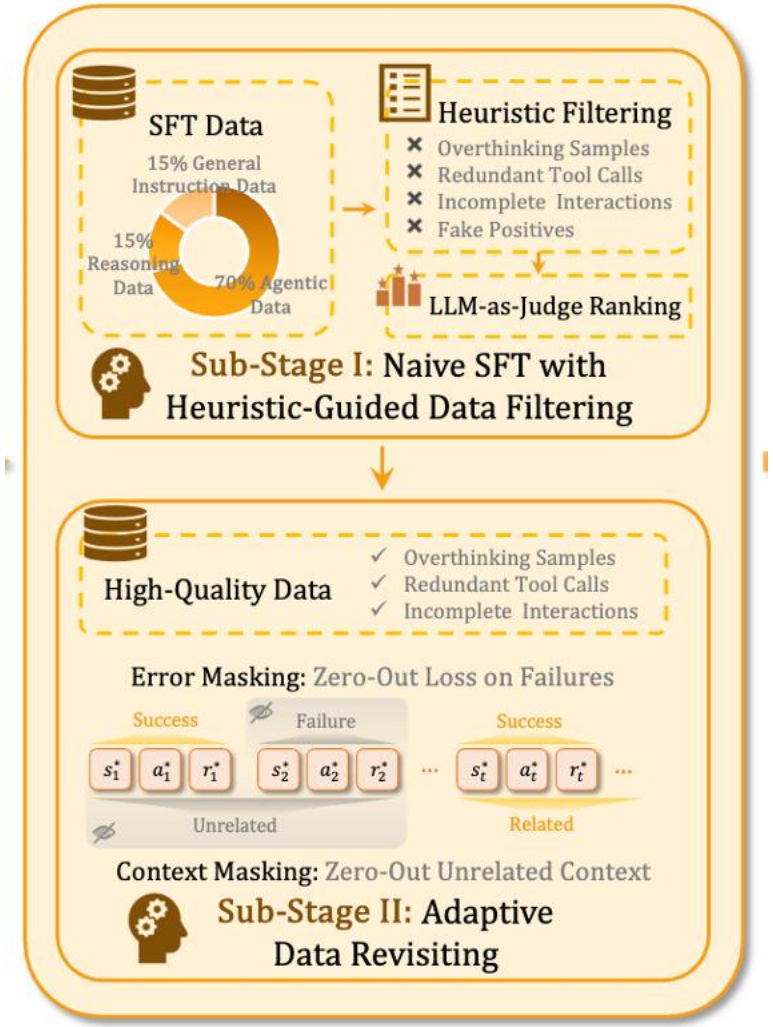
- SFT Stage 1：Naive SFT（带启发式过滤）
- SFT Stage 2：高质量数据 revisiting（自适应精炼）

SFT— SFT Stage 1



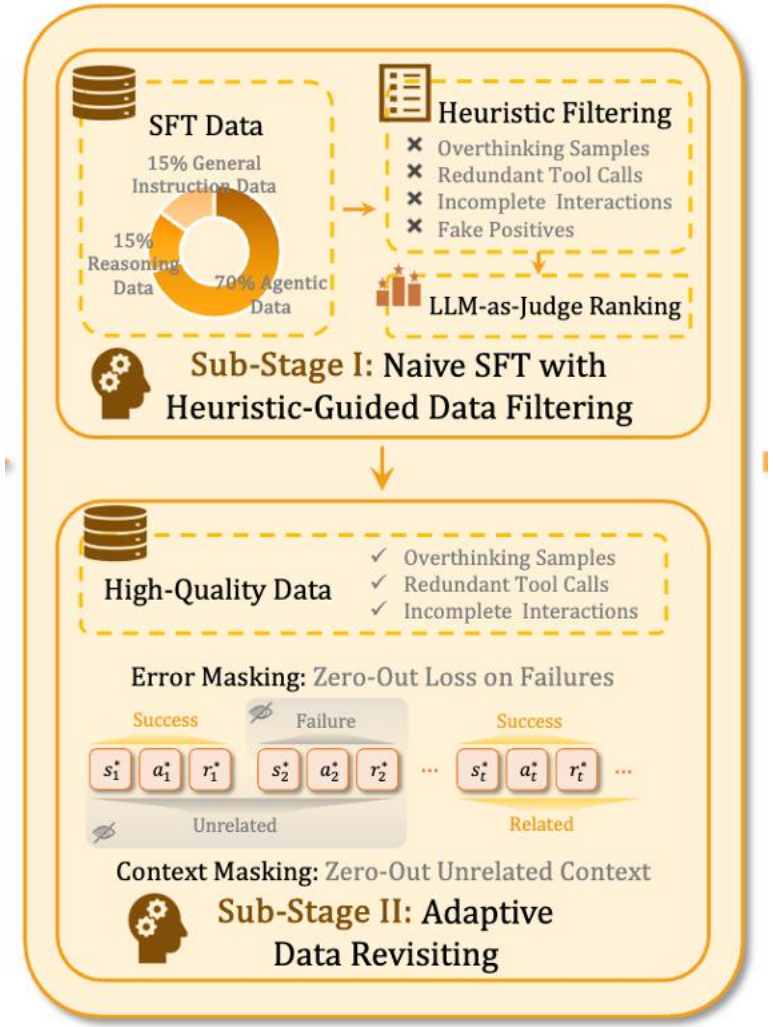
- 数据过滤：
  - ① 剔除"过度思考"样本（冗长自相矛盾的推理链）
  - ② 去除重复工具调用序列
  - ③ 丢弃截断/不完整的交互
  - ④ 过滤自我修复循环中卡住的轨迹
  - ⑤ 标记"假正例"（表面通过但逻辑错的）

SFT— SFT Stage 2



- **高质量数据获取**
  - **目的：强调可验证性、风格一致性、可复现性**
  - 可验证交互轨迹
  - 专家审计示范
  - 偏好精炼样本：每个任务多个候选轨迹，用规则/reward model评分

SFT— Loss



- Error Masking Loss
  - 标准 SFT 对所有 token 一视同仁地反向传播，会**让模型学到错误行为**

$$m_k^{\text{err}} = \mathbf{1}[\neg \text{Err}(k)]$$

- Task—Aware Context Masking
  - 模型聚焦于"因果上有影响的交互"，提升样本效率，使行为与真实软件开发工作流保持一致

$$m_k^{\text{task}} = \mathbf{1}[\text{Rel}(k)]$$

Mid—Train v.s. SFT

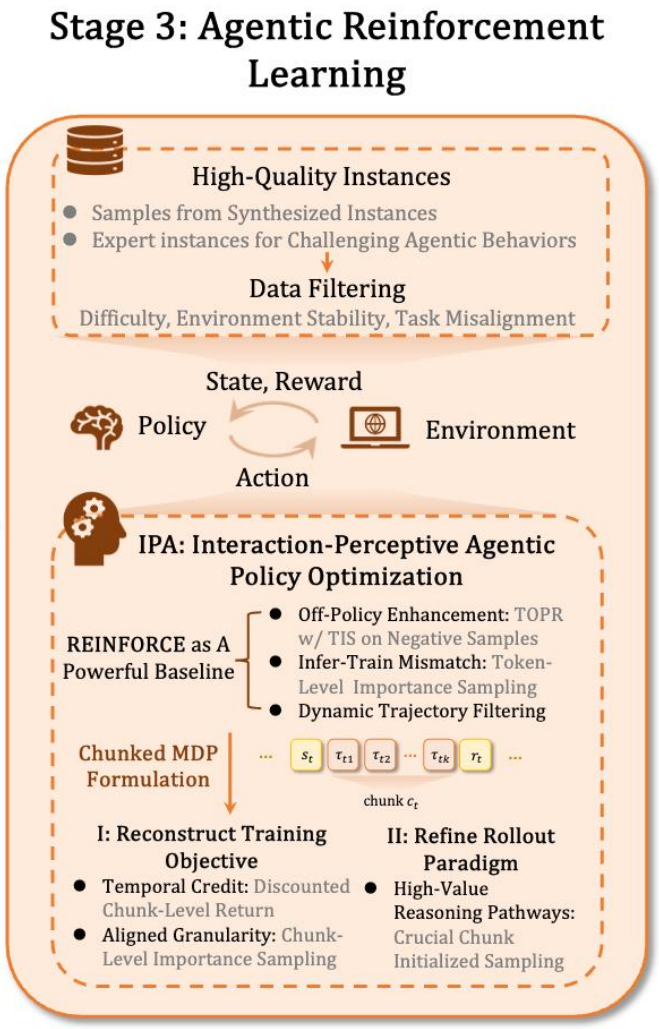
- Mid—train：让模型"见过"智能体行为是什么样的（学习分布）
- SFT：让模型"知道"哪些智能体行为是对的（学习对齐）
  - SFT 1：大部分输出是好行为，但还有噪声
  - SFT 2：行为分布精确集中在RL 优化器容易工作的区域

RL

- 现有RL方法在agentic场景下失效原因

问题	表现	影响
策略更新不稳定	大规模负样本采样导致梯度爆炸	策略崩溃
时序信用分配低效	逐个token优化，前序token的优化权重指数衰减； 句子级优化粒度太粗	长轨迹学习效果差
轨迹采样效率低	复杂任务正样本极其稀疏	学习信号近乎为零

RL

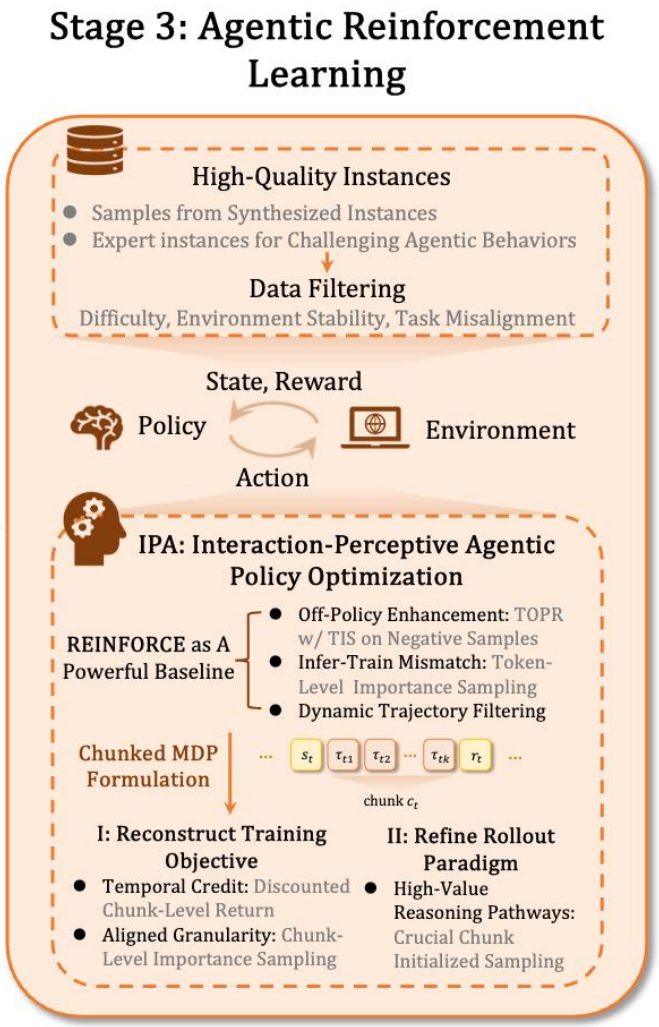


- 增强 REINFORCE算法
  - 引入 TIS（Truncated Importance Sampling）
    - 工业级 RL 训练异步（推理服务器持续产生数据，训练服务器持续更新参数），导致**策略偏移**
    - Importance Sampling

$$\mathbb{E}_{x\sim p}[f(x)] = \mathbb{E}_{x\sim q}\left[\frac{p(x)}{q(x)}f(x)\right]$$

$$\nabla J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_{\theta_{\text{old}}}}\left[\underbrace{\frac{\pi_\theta(\tau)}{\pi_{\theta_{\text{old}}}(\tau)}}_{\text{IS ratio}} \cdot R(\tau)\nabla\log\pi_\theta(\tau)\right]$$

RL



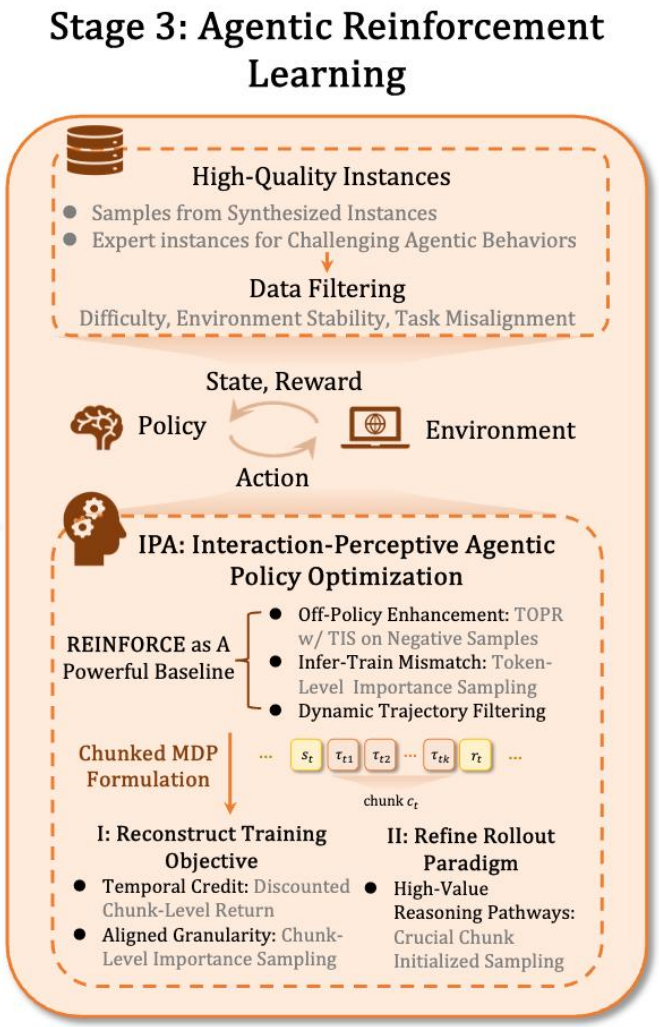
- 增强 REINFORCE算法
  - 引入 TIS（Truncated Importance Sampling）
    - Importance Sampling：**方差爆炸**
    - 对于一条长度 T 的轨迹，IS 权重是**所有 token 概率比的乘积**

$$\rho(\tau) = \prod_{t=1}^{T} \frac{\pi_\theta(\tau_t|\tau_{<t})}{\pi_{\theta_{\text{old}}}(\tau_t|\tau_{<t})}$$

  - Truncated IS $\quad [\rho(\tau)]_0^1 = \min(\rho(\tau), 1)$
  - 几何平均

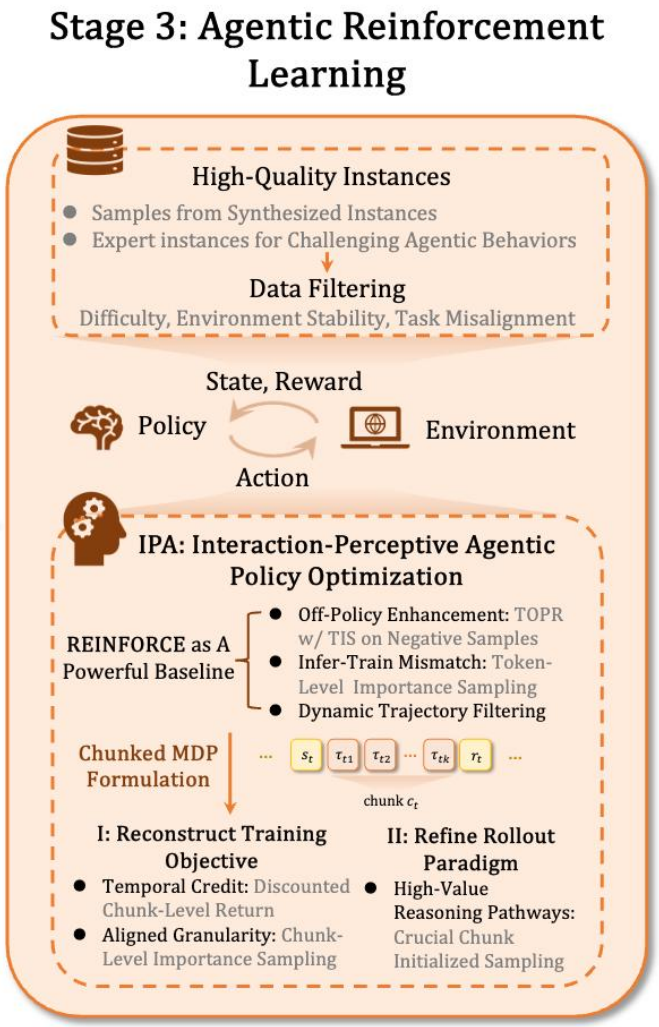
$$\rho(\tau) = \exp\left(\frac{1}{|\tau|}\sum_{t\in\tau}\log\frac{\pi_\theta(\tau_t)}{\pi_{\theta_{\text{old}}}(\tau_t)}\right)$$

RL



- Chunked MDP
  - Chunk 典型结构：reason → format API call → trigger execution
  - Chunk-Level Discounted Return
  - Chunk-Level Importance Sampling
  - Chunk-Level Mismatch Masking

RL


Stage 3: Agentic Reinforcement Learning
High-Quality Instances
Samples from Synthesized Instances
Expert instances for Challenging Agentic Behaviors
Data Filtering
Difficulty, Environment Stability, Task Misalignment
State, Reward
Policy
Environment
Action
IPA: Interaction-Perceptive Agentic Policy Optimization
REINFORCE as A Powerful Baseline
Off-Policy Enhancement: TOPR w/ TIS on Negative Samples
Infer-Train Mismatch: Token-Level Importance Sampling
Dynamic Trajectory Filtering
Chunked MDP Formulation
chunk c_t
I: Reconstruct Training Objective
Temporal Credit: Discounted Chunk-Level Return
Aligned Granularity: Chunk-Level Importance Sampling
II: Refine Rollout Paradigm
High-Value Reasoning Pathways: Crucial Chunk Initialized Sampling

- 数据

实验—CLI Benchmark

Table 1: Performance on Terminal-Based Benchmarks (Normal Models).

Benchmark	ROME	Qwen3-Coder 30B-A3B-Instruct	Devstral Small 2	GPT-OSS-120B	Gemini-2.5 Flash	GLM-4.5 Air	GPT-5 Mini
Architecture	MoE	MoE	Dense	MoE	-	MoE	-
# Total Params	30B	30B	24B	117B	-	106B	-
# Activated Params	3B	3B	-	5.1B	-	12B	-
Terminal-Bench 1.0	41.50	28.50	28.33	31.25	23.75	30.00	33.75
Terminal-Bench 2.0	24.72	13.48	18.20	21.12	16.40*	17.30	20.97
SWE-Bench Verified	57.40	46.33	51.87	43.93	28.73*	56.20	59.30
SWE-Bench Multilingual	40.00	30.00	27.00	34.84	11.50	38.16	49.67
Terminal-Bench-Pro-Public	40.50	26.00	32.17	32.00	23.67	33.00	34.75
Terminal-Bench-Pro-Private	21.50	11.33	17.00	27.83	15.17	15.83	29.50
Avg.	**37.60**	**25.94**	**29.10**	**31.83**	**19.87**	**31.75**	**37.99**

Table 2: Performance on Terminal-Based Benchmarks (Large Models).

Benchmark	ROME	Qwen3-Coder Plus	Qwen3-Coder 480B-A35B-Instruct	DeepSeek V3.1	GLM-4.6	Kimi-K2	Claude-Haiku-4.5
Architecture	MoE	MoE	MoE	MoE	MoE	MoE	-
# Total Params	30B	-	480B	671B	355B	1043B	-
# Activated Params	3B	-	35B	37B	32B	32B	-
Terminal-Bench 1.0	41.50	39.58	37.92	38.75	41.25	39.25	47.08
Terminal-Bench 2.0	24.72	32.36	26.97	28.47	26.29	30.90	34.83
SWE-Bench Verified	57.40	65.87	65.20	62.20	62.67	64.80	69.60
SWE-Bench Multilingual	40.00	54.16	49.50	48.16	53.84	48.67	60.34
Terminal-Bench-Pro-Public	40.50	39.67	38.33	39.33	41.50	40.50	45.83
Terminal-Bench-Pro-Private	21.50	28.50	26.50	28.33	29.17	29.00	35.33
Avg.	**37.60**	**43.36**	**40.74**	**40.87**	**42.45**	**42.19**	**48.84**

实验—Tool—Use Benchmark

Table 3: Performance on Tool-Use Benchmarks (Normal Models).

Benchmark	ROME	Qwen3-Coder 30B-A3B-Instruct	Devstral Small 2	GPT-OSS-120B	Gemini-2.5 Flash	GLM-4.5 Air	GPT-5 Mini
Architecture	MoE	MoE	Dense	MoE	-	MoE	-
# Total Params	30B	30B	24B	117B	-	106B	-
# Activated Params	3B	3B	-	5.1B	-	12B	-
Tau2-Bench Retail	62.28	59.87	58.33	64.30	64.30*	74.60	74.12
Tau2-Bench Airline	50.50	45.50	30.00	53.50	42.50*	69.00	60.00
Tau2-Bench Telecom	30.92	30.04	20.40	54.61	16.90*	46.90	73.46
BFCL-v3 (Multi-Turn)	43.00	29.75	30.12	53.62	36.25*	66.88	27.25
MTU-Bench (Single-Turn)	62.45	50.69	63.08	54.16	45.93	57.74	59.82
MTU-Bench (Multi-Turn)	47.63	29.38	34.15	58.61	57.01	37.55	55.61
Avg.	**49.46**	**40.87**	**39.35**	**56.47**	**43.82**	**58.78**	**58.38**

Table 4: Performance on Tool-Use Benchmarks (Large Models).

Benchmark	ROME	Qwen3-Coder Plus	Qwen3-Coder 480B-A35B-Instruct	DeepSeek V3.1	GLM-4.6	Kimi-K2	Claude-Haiku-4.5
Architecture	MoE	MoE	MoE	MoE	MoE	MoE	-
# Total Params	30B	-	480B	671B	355B	1043B	-
# Activated Params	3B	-	35B	37B	32B	32B	-
Tau2-Bench Retail	62.28	62.28	59.00	71.50	76.10	67.54	67.32
Tau2-Bench Airline	50.50	48.00	48.00	52.00	65.00	49.00	47.50
Tau2-Bench Telecom	30.92	52.19	58.55	40.35	71.05	86.40	36.40
BFCL-v3 (Multi-Turn)	43.00	27.75	42.38	20.62	67.50	50.63*	53.50
MTU-Bench (Single-Turn)	62.45	56.68	63.87	61.71	49.54	56.21	61.19
MTU-Bench (Multi-Turn)	47.63	37.56	34.85	53.44	37.55	53.31	55.43
Avg.	**49.46**	**47.41**	**51.11**	**49.94**	**61.12**	**60.52**	**53.56**