大模型对齐技术研究及应用

中国人民大学

张静

2025年12月

提纲

Contents

01 背景与动机

02 可扩展、高保真的高效数据合成方法

03 稳定高效的强化学习

04 性能保持的推理加速与模型压缩

05 总结与展望

提纲

Contents

01 背景与动机

02 可扩展、高保真的高效数据合成方法

03 稳定高效的强化学习

04 性能保持的推理加速与模型压缩

05 总结与展望

背景与动机

■ **大模型**习得大量人类知识，展现强通用性，但人类希望它适配完成越来越难的任务

问答

直接提供答案



推理

先思考再给答案



智能体

先思考再调用工具



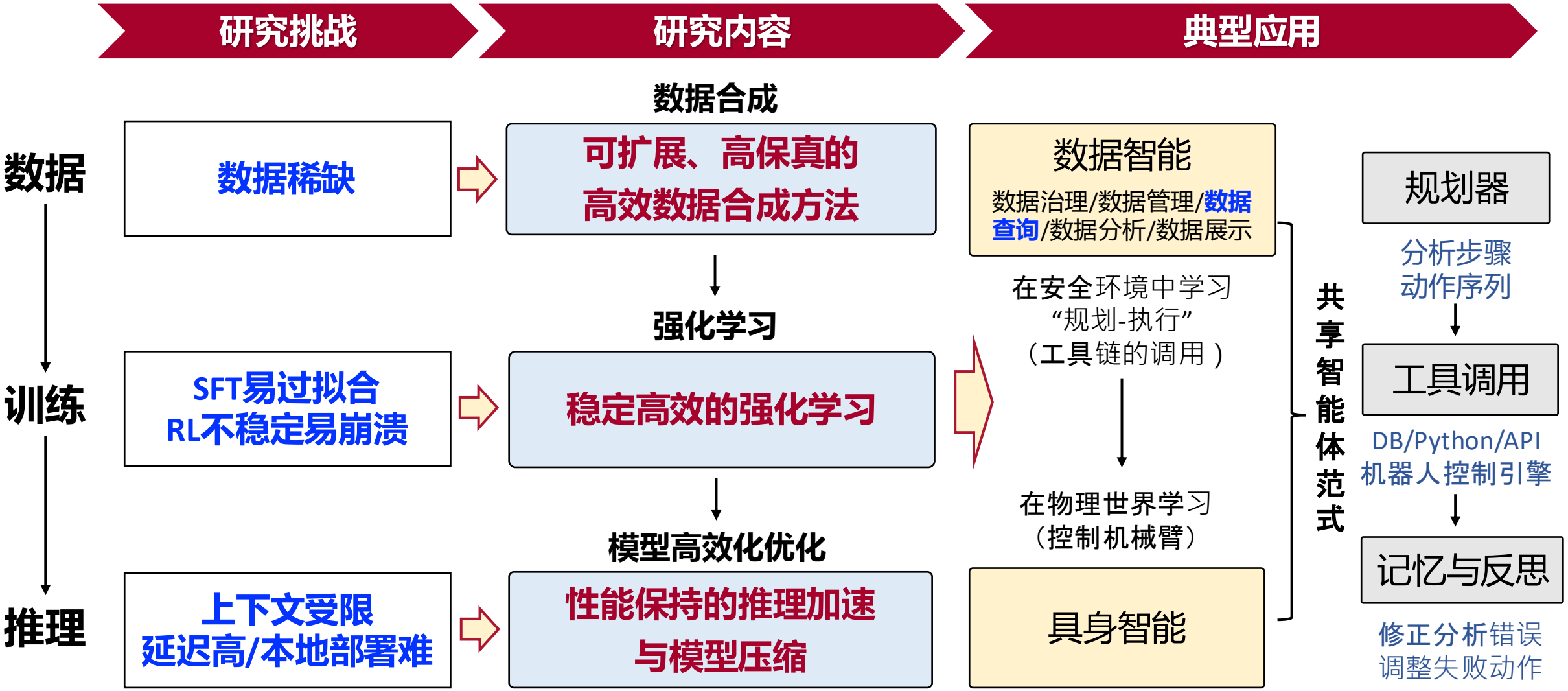
静态回答 → **动态决策**

自主性

长链推理与记忆

环境交互与反馈学习

背景与动机


研究挑战
研究内容
典型应用
数据
训练
推理
数据稀缺
SFT易过拟合
RL不稳定易崩溃
上下文受限
延迟高/本地部署难
数据合成
可扩展、高保真的
高效数据合成方法
强化学习
稳定高效的强化学习
模型高效化优化
性能保持的推理加速
与模型压缩
数据智能
数据治理/数据管理/数据查询/数据分析/数据展示
在安全环境中学习
“规划-执行”
（工具链的调用）
在物理世界学习
（控制机械臂）
具身智能
共享智能体范式
规划器
分析步骤
动作序列
工具调用
DB/Python/API
机器人控制引擎
记忆与反思
修正分析错误
调整失败动作

提纲

Contents

01 背景与动机

02 可扩展、高保真的高效数据合成方法

03 稳定高效的强化学习

04 性能保持的推理加速与模型压缩

05 总结与展望

数据合成

- 关键科学问题
  - **如何**实现数据合成的可扩展性？

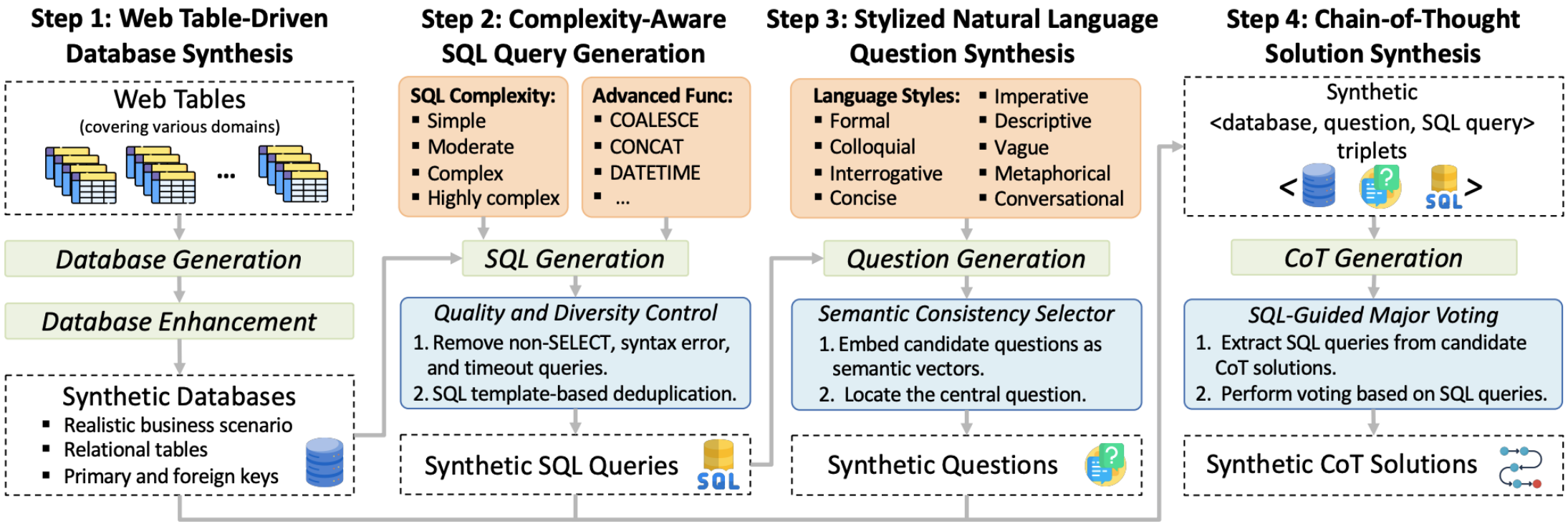
  ——**即如何构建具**备大规模和通用化的数据生成能力。
  - **如何保**证合成数据的保真性？

  ——**即如何确保合成**问题的真实性与合成答案的可靠性。
  - **如何提高数据合成的效率？**

  ——**即如何在有限算力与**资源约束下实现高产出的数据生成过程。

一种可扩展的Text2SQL数据合成框架

- 1. 从Webtable合成DB schema；2.合成复杂且可靠的SQL ；
- 3. 合成多样的自然语言指令；4 合成思维链



Haoyang Li, Shang Wu, Xiaokang Zhang, Xinmei Huang, Jing Zhang, etc. OmniSQL: Synthesizing High-quality Text-to-SQL Data at Scale. VLDB'25

SynSQL-2.5M数据集

Table 2: Overall statistics of different datasets. Note: † denotes that each database in WikiSQL consists of a single table. ∗ indicates that the number of unique SQL queries for BIRD cannot be calculated due to the inaccessibility of its test set.

Dataset	Source	# Example	# Unique SQL	# DB	Lang. Styles	Knowledge	CoT Solution
WIKISQL [85]	Human+Template	80,654	80,257	26,531†	✘	✘	✘
Spider [80]	Human	10,181	4,489	200	✘	✘	✘
BIRD [42]	Human	12,751	-*	95	✘	✓	✘
ScienceBenchmark [81]	LLM-Gen+Human+Template	5,031	3,652	3	✘	✘	✘
EHRSQL [37]	Human+Template	20,108	18,253	2	✘	✘	✘
SynSQL-2.5M	LLM-Gen	2,544,390	2,412,915	16,583	✓	✓	✓

SynSQL-2.5M是第一个百万量级的跨领域Text-to-SQL数据集

Table 4: SQL statistics. "Agg." and "Func." are abbreviations of aggregations and functions. The asterisk(*) means BIRD's SQL statistics are based on training and development sets due to test set inaccessibility.

Dataset	# Tables per SQL	# Joins per SQL	# Func. per SQL	# Tokens per SQL	# Agg.	# Set Operators	# Subqueries	# Window Func.	# CTEs	# Unique Skeletons	# Unique Func.
WIKISQL [85]	1.00	0	0.72	10.32	22,707	0	0	0	0	488	5
Spider [80]	1.52	0.48	0.51	14.85	3,802	348	604	0	0	2,136	5
BIRD* [42]	1.98	0.94	0.67	24.75	5,144	25	843	10	9	4,596	24
ScienceBenchmark [81]	1.42	0.42	0.25	14.61	1,234	2	23	0	0	855	5
EHRSQL [37]	3.60	0.62	**2.63**	41.62	11,673	166	17,989	3,207	0	2,447	11
SynSQL-2.5M	**4.00**	**1.75**	1.49	**57.05**	**1,897,440**	**363,472**	**612,647**	**33,916**	**1,073,483**	**2,190,988**	**83**

合成SQL的复杂度和多样性也远超已有数据集

Table 3: Database statistics. The asterisk(*) means BIRD's database statistics are based on training and development sets due to test set inaccessibility.

Dataset	# DB	# Tbl/DB	# Col/DB	# PK/DB	# FK/DB
WIKISQL [85]	26,531	1.00	6.34	1.00	0.00
Spider [80]	200	5.11	26.82	4.70	4.79
BIRD* [42]	95	7.64	54.56	6.71	6.58
ScienceBenchmark [81]	3	16.67	86.67	14.33	20.33
EHRSQL [37]	2	13.50	92.00	13.50	17.00
SynSQL-2.5M	16,583	10.15	74.07	10.14	9.61

合成数据库的复杂程度达到真实世界水平

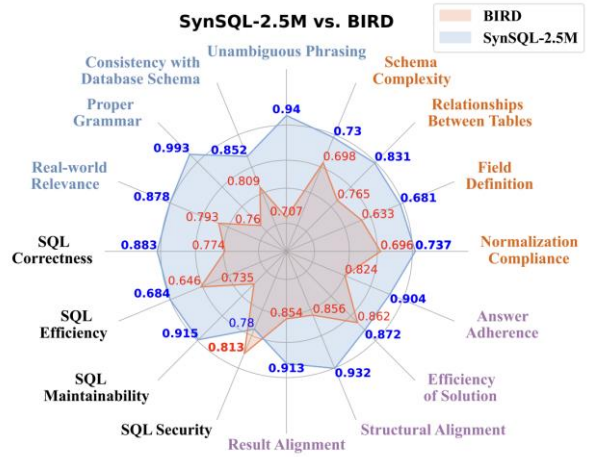
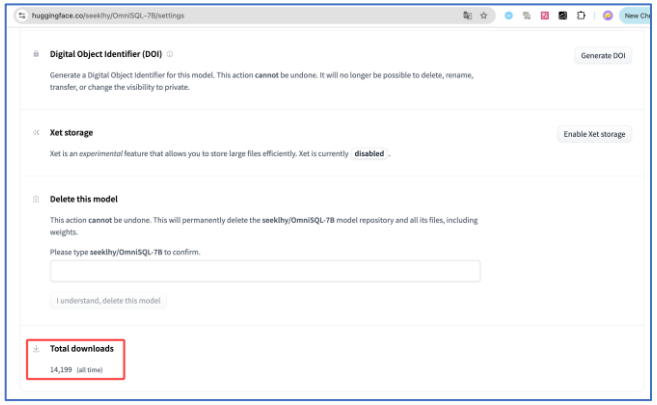


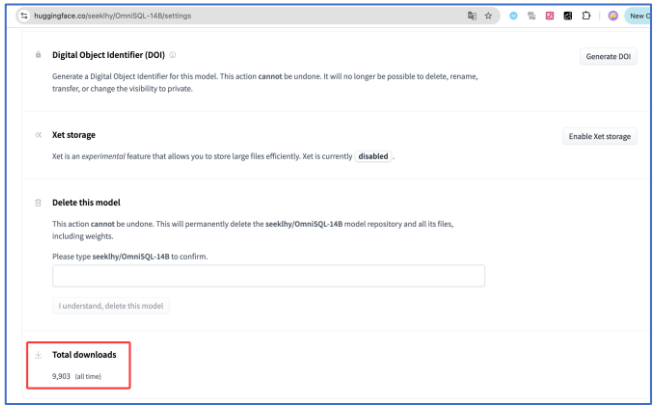
Figure 3: Quality evaluation of SynSQL-2.5M and BIRD.

SynSQL-2.5M的评估质量全面超过了人类标注的数据集BIRD

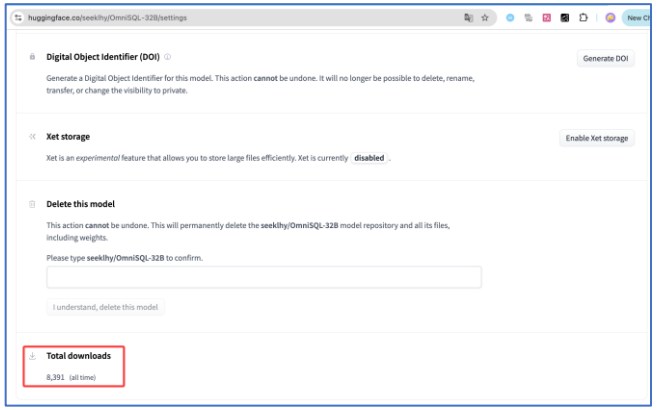
OmniSQL在Huggingface和Modelscope的下载量超10万



OmniSQL7B在huggingface的下载量为14,119



OmniSQL14B在huggingface的下载量为9, 903



OmniSQL32B在huggingface的下载量为8,391



OmniSQL四个尺寸在modelscope的总下载量为3.7万

研发的Text2SQL系列模型RESDSQL/CodeS/OmniSQL在国际公认基准测试集Spider/Bird/Spider2上取得过总计 3个月榜首

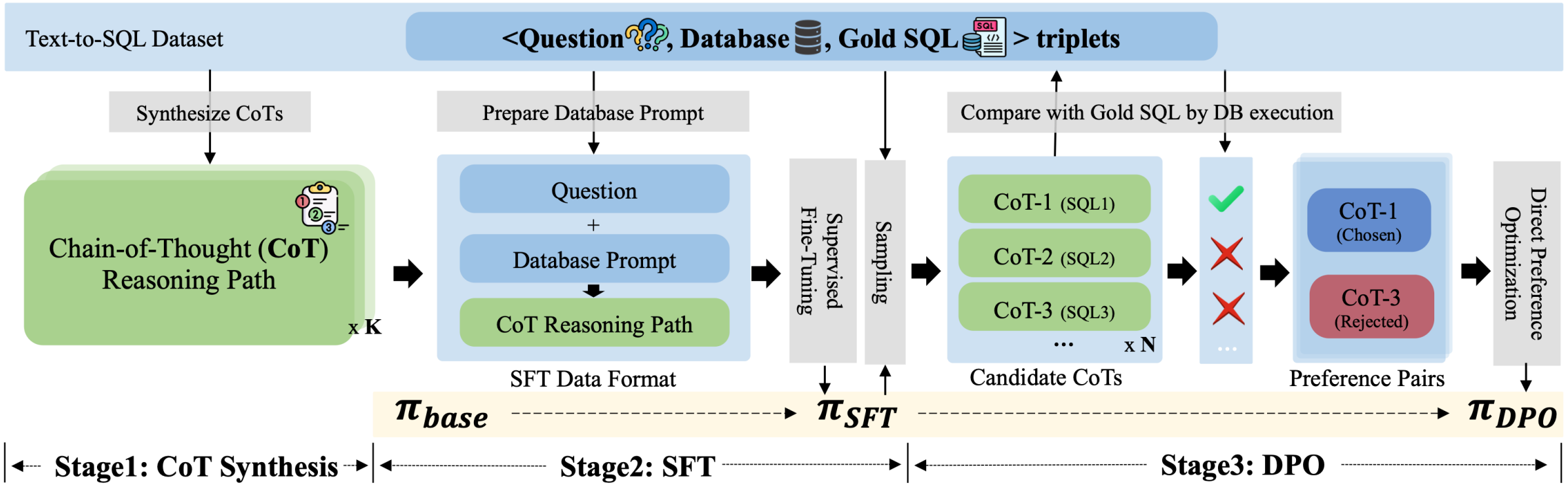
发表 AAAI23 论文被 Google DeepMind 所引用，指出我们的方法是 fine-tuned SOTA

发表 SIGMOD24 论文在SIGMOD当年发表的论文中引用量排名第一。

OmniSQL 在huggingface的、modelscope的下载量突破10万

为什么要合成思维链COT？

- 源自于COT-based DPO在Text2SQL任务上的探索结果



Hanbing Liu, Haoyang Li, Xiaokang Zhang, Ruotong Chen, Haiyong Xu, Tian Tian, Qi Qi and Jing Zhang. Unlocking the Potential of Direct Preference Optimization in Text-to-SQL Through Chain-of-Thought Reasoning. ACL'25.

COT-based DPO 效果显著

- 在Text-to-SQL数据集上直接应用DPO的效果并不好，很多时候性能甚至会下降。
- 而在合成了CoT之后，无论什么模型采用何种推理方式，模型在DPO阶段都可以得到一致的增强。

问题: CoT的加入对DPO具体造成了怎样的影响？

DPO相较于SFT带来的性能增益

	Model	Bird Dev						ΔEX
		Greedy		Pass@1		Maj@16		
		SFT	DPO	SFT	DPO	SFT	DPO	
General Models								
Vanilla	Deepseek-llm-7b-chat	51.8	51.2 (-0.6)	47.9	49.1 (+1.3)	54.5	54.3 (-0.3)	-
	Llama-3.1-instruct-7b	59.0	59.8 (+0.8)	56.1	57.2 (+1.1)	61.4	60.8 (-0.6)	-
	Qwen2.5-7B-Instruct	58.8	59.0 (+0.2)	55.1	55.7 (+0.6)	61.4	60.6 (-0.8)	-
	Qwen2.5-14B-Instruct	64.3	63.5 (-0.8)	62.3	62.6 (+0.3)	64.6	65.1 (+0.5)	-
Syn CoT	Deepseek-llm-7b-chat	54.3	55.9 (+1.6)	51.9	54.8 (+2.9)	59.1	61.0 (+1.9)	54.5 → 61.0 (**+6.5**)
	Llama-3.1-instruct-7b	56.8	61.2 (+4.4)	57.5	59.0 (+1.5)	60.2	61.9 (+1.7)	61.4 → 61.9 (**+0.5**)
	Qwen2.5-7B-Instruct	57.4	61.9 (+4.5)	54.8	59.2 (+4.4)	63.0	64.9 (+1.9)	61.4 → 64.9 (**+3.5**)
	Qwen2.5-14B-Instruct	63.2	65.3 (+2.1)	61.8	64.7 (+2.9)	65.4	67.1 (+1.7)	64.6 → 67.1 (**+2.5**)
Coder Models								
Vanilla	Deepseek-coder-6.7b-instruct	60.6	60.9 (0.3)	56.9	58.8 (+1.9)	59.8	61.0 (+1.2)	-
	CodeLlama-7b-Instruct-hf	57.0	55.7 (-1.3)	54.3	55.5 (+1.2)	59.1	58.5 (-0.6)	-
	CodeLlama-13b-Instruct-hf	60.0	60.2 (+0.2)	56.7	57.9 (+1.2)	61.9	62.0 (+0.1)	-
	Qwen2.5-Coder-7B-Instruct	61.6	61.3 (-0.3)	59.4	60.6 (+1.2)	61.3	62.7 (+1.4)	-
Syn CoT	Deepseek-coder-6.7b-instruct	61.5	63.8 (+2.3)	59.9	62.3 (+4.5)	64.3	65.4 (+1.1)	59.8 → 65.4 (**+5.6**)
	CodeLlama-7b-Instruct-hf	58.2	61.3 (+3.1)	56.9	60.4 (+3.5)	60.2	61.9 (+1.7)	59.1 → 61.9 (**+2.8**)
	CodeLlama-13b-Instruct-hf	62.0	63.9 (+1.9)	59.8	62.5 (+2.7)	63.6	65.8 (+2.2)	61.9 → 65.8 (**+3.9**)
	Qwen2.5-Coder-7B-Instruct	60.8	63.4 (+2.6)	59.1	62.8 (+3.7)	62.5	64.1 (+1.6)	61.3 → 64.1 (**+2.8**)
SQL-Specialized Models								
Vanilla	CodeS-7b	56.8	56.6 (-0.2)	53.7	54.6 (+0.9)	58.1	58.0 (-0.1)	-
	CodeS-15b	58.3	58.2 (-0.1)	55.6	56.2 (+0.6)	60.2	59.1 (-1.1)	-
Syn CoT	CodeS-7b	56.7	57.5 (+0.8)	54.2	55.3 (+1.1)	60.2	61.7 (+1.5)	58.1 → 61.7 (**+2.6**)
	CodeS-15b	58.6	61.1 (+2.5)	56.6	60.5 (+3.9)	62.4	63.2 (+0.8)	60.2 → 63.2 (**+3.0**)

COT-DPO相较于非COT-DPO带来的性能增益

Table 1: Model performance on the Bird development set. **Vanilla**: SFT and DPO on the original Bird training set; **Syn CoT**: SFT and DPO on the CoT-enhanced training set; **ΔEX**: The performance difference in EX between "Syn CoT + DPO" and "Vanilla + SFT" when using the same base model.

COT给DPO带来了什么样的影响？

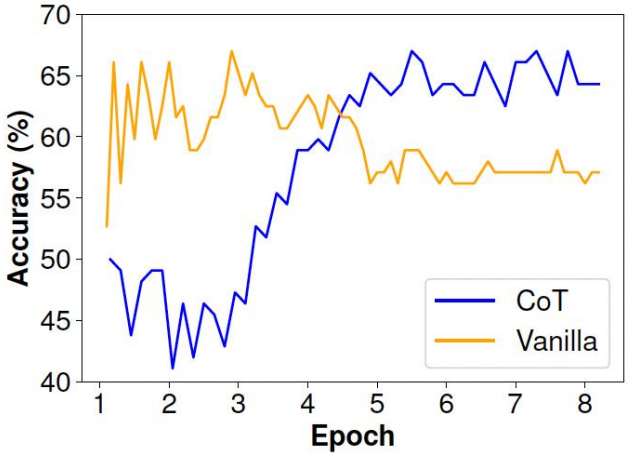
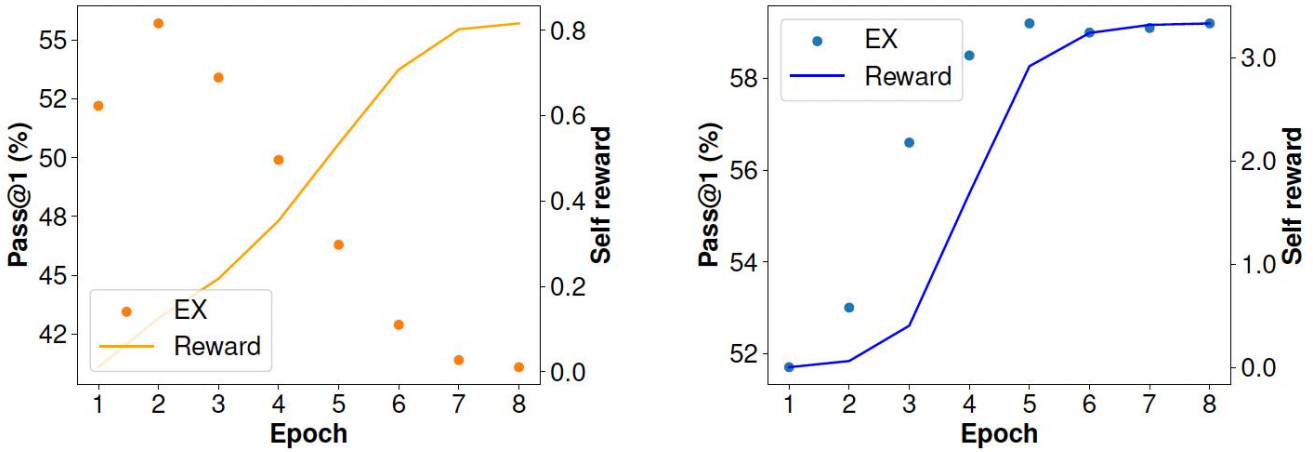


Figure 2: Comparison of model's discriminative ability during DPO (measured by classification accuracy on curated evaluation set).

COT使得DPO产生更具判别能力的隐含奖励模型



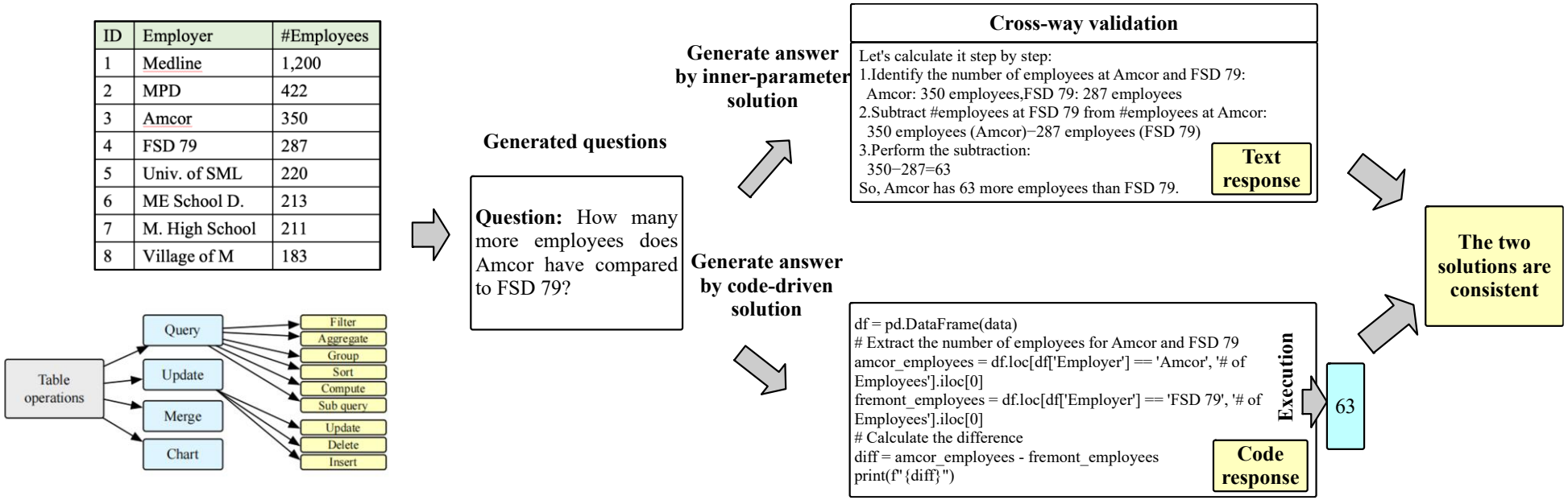
(a) Vanilla (b) Syn CoT

Figure 3: Comparison of model's self-assessed performance (average implicit reward policy model given to its own roll-outs) and real performance (EX) on Bird development set (Pass@1) during DPO training.

COT使得DPO产生更能防止Reward Hacking的隐含奖励模型

一种交叉验证的表格问答数据保真方法

- 学术界存在大量的 TableQA 训练数据，但缺乏可靠的思维链COT
- 提出一种交叉验证的方法对大模型合成的思维链进行可靠性验证



定理：

(1) 如果A和B来自同一分布且满足 $P(Y_a) = P(Y_b) = p > ½$, 那么一致性检查优于单次推理, 即$P(Y|E) >= P(Y_a)$

(2) 如果A和B进一步来自独立分布，那么交叉验证效果会更显著 (从期望的角度来看)

Xiaokang Zhang, Sijia Luo, Bohan Zhang, Zeyao Ma, Jing Zhang, etc. TableLLM: Enabling Tabular Data Manipulation by LLMs in Real Office Usage Scenarios. ACL Findings'25.

表格操作数据集：SpreadsheetBench

- 简单的指令
- 简化的表格
- **缺乏**测试用例

- 真实场景的复杂指令：**收集来自**热门 Excel 论坛的真实用户查询。
- 多样化格式的表格：**包含非**标准表格、多张表格及多种样式。
- OJ 风格的评估指标：**每条指令包含三个**测试用例，并进行全面评估。

Instruction

SheetCopilotBench

Calculate the **sum** of the Subtotal column only for the rows that have "Company A" in the Vendor/Client column in a new row with header "Total Expenses".

Our SpreadsheetBench

How can I **sum** the output of a formula at the end of a row? I have created a spreadsheet to capture the results of an up-coming horse show. There are multiple sections and aggregate awards to calculate. My IF functions appear to be working well but when I try to total up the scores at the end of the row I am completely lost. I have investigated the =VALUE, =SUM and =SUMPRODUCT functions with no luck. In my example, I am attempting to sum the results in the blue cells to the location in the green cell that I have manually totalled.

Spreadsheet

SheetCopilotBench

	A	B	C	D	E	F
1	Date	Vendor/Client	Expense Account	Subtotal	Tax	Total
2	21-Dec-19	Client A	Gas	$313.00		
3	21-Dec-19	Client A	Meals	$134.00		
4	25-Dec-19	Company A	Car Repairs	$1,205.00		

Our SpreadsheetBench

	A	B	C	D	E	F	G	H	I	J	K
1	Relational Table:				Non-standard Table (Nested Header)						
2	Name	Score	Level		Name	Score					
3	Kim	62	A			Math	P.E.		Textual Information		
4	Jone	49	B		Jone	76	92		F7:F10 is the day to be on duty		
5	Martin	23			Kate	33	87				
6	Jane	76			Yao	62	79				
7											
8	Non-standard Table (Incomplete Header)				Non-standard Table (Missing Header)						
9		Monday	Tuesday		Jone	Monday	Yes				
10	Jane	1			Kate	Thursday	Yes				
11	Kim		1		Yao	Friday	No				
12	Bob	1	1		Martin	Tuesday	Yes				

Zeyao Ma, Bohan Zhang, Jing Zhang, etc. SpreadsheetBench: Towards Challenging Real World Spreadsheet Manipulation. Neurips'24.

表格操作数据集：SpreadsheetBench

Benchmark	Ours
Data Source	Forum & Blog
Instructions	912
Ave. Instruction Words	85.7
Spreadsheet Files	2729
Single Sheet	2019
Multiple Sheet	710
Sheets	3917
Non-standard Tables	✓
Multiple Tables	✓
Additional Info.	✓
Evaluation	OJ-style EM
Ave. Test Cases	3

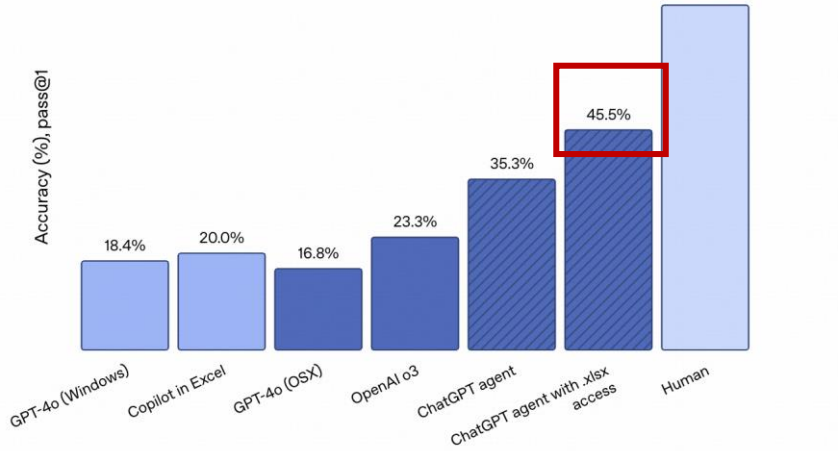
- 2025年7月，OpenAI推出的ChatGPT 智能体，在我们开源的SpreadsheetBench平台上进行测试
- 2025年11月，Claude Opus 4.5在我们开源的SpreadsheetBench平台上进行测试

2.15 SpreadsheetBench

SpreadsheetBench is an evaluation of a model's ability to navigate and manipulate complex spreadsheets, with problems developed using real-world examples.

We used the full 912-problem set. We executed the problems in a custom harness where the model was provided access to a bash tool, string viewing and editing tools, and a Python environment with the openpyxl, libreoffice, pandas, and numpy libraries available.

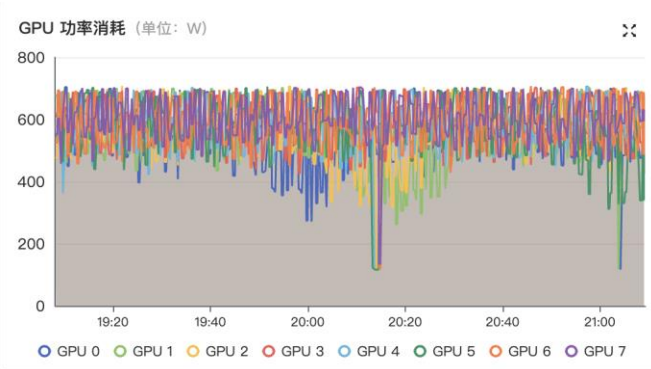
With no extended thinking, and a 200k context window, Claude Opus 4.5 achieved a score of 64.25% on SpreadsheetBench (averaged across 5 trials).

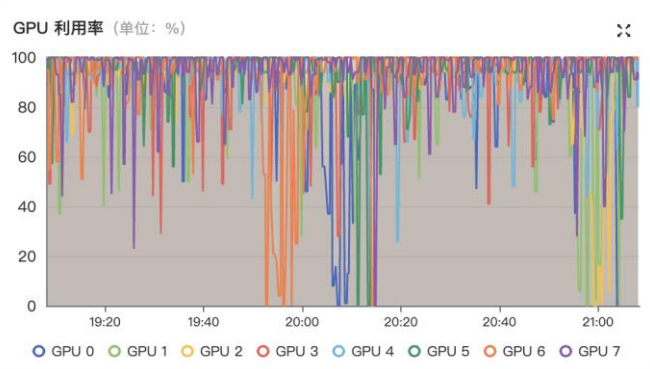


1. https://openai.com/zh-Hans-CN/index/introducing-chatgpt-agent/

一种高效的具身思维链（ECOT）数据合成管道

- 开发高效的ECoT <reasoning, actions to-do, target grounding>合成框架, 增强VLM对具身任务的理解，服务下游Action Expert Model
- 流水线化：采用并行进程架构（Dataloader–Processor–Inference–Consumer）实现数据准备、推理与解析的流水线处理，确保 GPU 高负载运行；加速约 15%。
- 图像去重：利用感知哈希检测相似图像（如静止帧、光影变化），跳过重复数据，标注效率提升约 30%。
- 降低GPU IO: 以视频输入替代逐帧输入，通过关键帧划分子任务减少推理次数，推理速度提升约 2 倍。
- 读写优化：针对 Lerobot 数据集采用异步预取与列式读取，仅加载任务描述与对应图像，吞吐率由 40 → 4000 frame/s， 每节点每日处理约 440 万帧（不含跳帧）



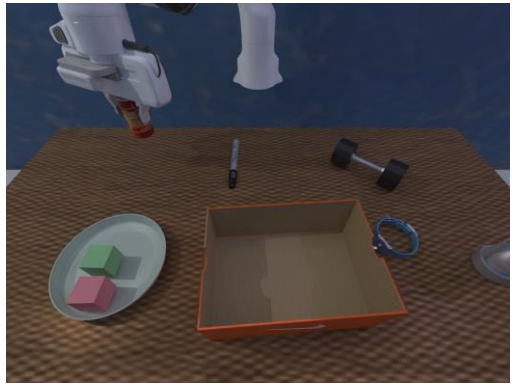


一种与环境交互的可扩展具身规划纠错数据合成方法

- 通过与仿真环境交互合成可扩展的具身规划纠错数据，提升模型的状态判断和错误修复能力。

- 任务构造：包含推理的具身规划任务，多种推理类别如Common Sense、Counting等。列举仿真环境中可用的物体及其属性，通过模型自动化生成大量的任务以及完成任务需要的操作序列（子任务）。
- 纠错数据构造：在操作序列中刻意加入错误的行为并进行恢复，以此得到错误的场景与错误修复的数据，对每个子任务标注推理过程，用于进行状态判断。
- 通过仿真环境的接口将具身原子动作封装成工具，自动化编写执行代码可以得到训练数据，同时提供仿真环境中工具调用的执行环境，为后续提升模型训练作准备。

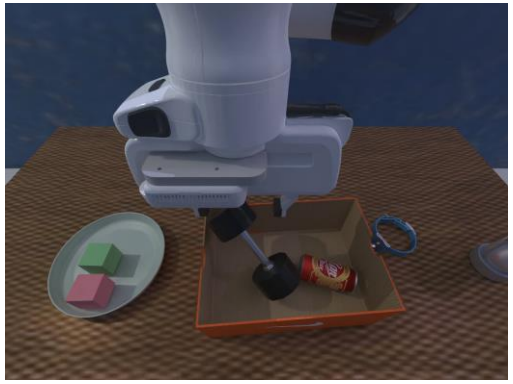
Task: Could you put the small, lightweight toys on the plate and the metallic or heavy non-toy items in the shoebox?



<think>The can is metallic and should be placed in the shoebox. It is picked up now.
<action>place(can, shoebox)



<think>The can is metallic and should be in the shoebox but now is placed on the plate. It needs to be corrected.(recovery)
<action>pick(can,[x1,y1,x2,y2])



<think>The showbox contains a can and a dumbell, which are metallic and heavy non-toy items. The plate contains two blocks wich are small and lightweight toy. The task has complete.
<action>done()

提纲

Contents

01 背景与动机

02 可扩展、高保真的高效数据合成方法

03 稳定高效的强化学习

04 性能保持的推理加速与模型压缩

05 总结与展望

强化学习

- 关键科学问题
  - **如何合理**设定奖励函数？

  ——**即如何构建既能准确反映任**务目标，又具备适当稠密性与可学习性的奖励信号，以缓解稀疏奖励带来的优化困难。

  - **如何**设计高效且稳定的Rollout**策略？**

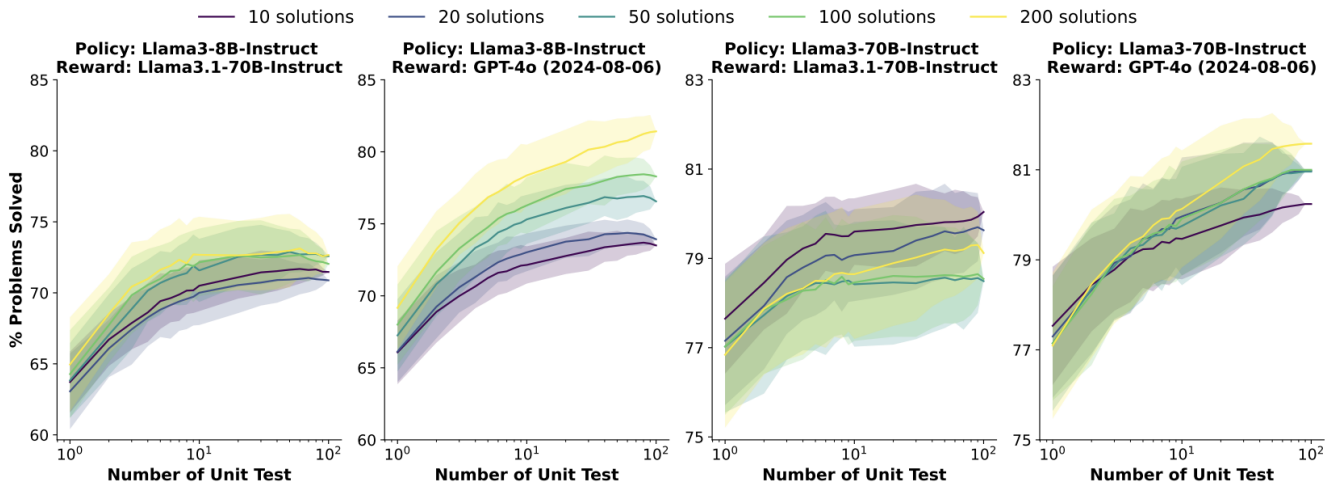
  ——**即如何在**长程、多轮交互情境下，实现具有行为合理性的策略生成机制，确保模型在探索空间中**保持合理行**为路径与稳定收敛。

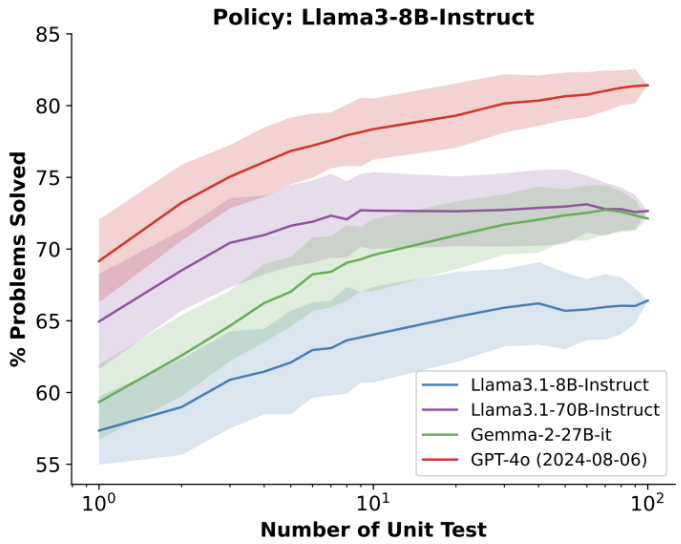
  - **如何加速**Rollout过程？

  ——**即如何通**过推理加速、注意力稀疏化等手段，提高强化学习中Rollout过程的效率，从而在保证策略质量的前提下显著降低计算开销。

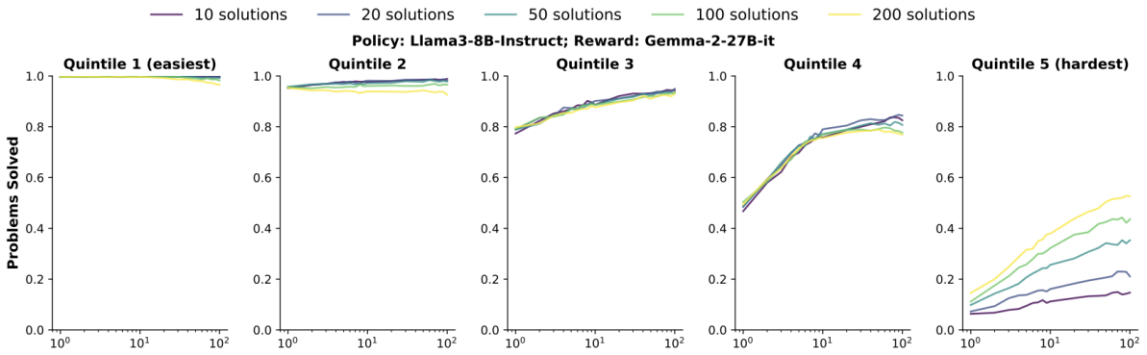
一种可验证奖励信号的扩展性

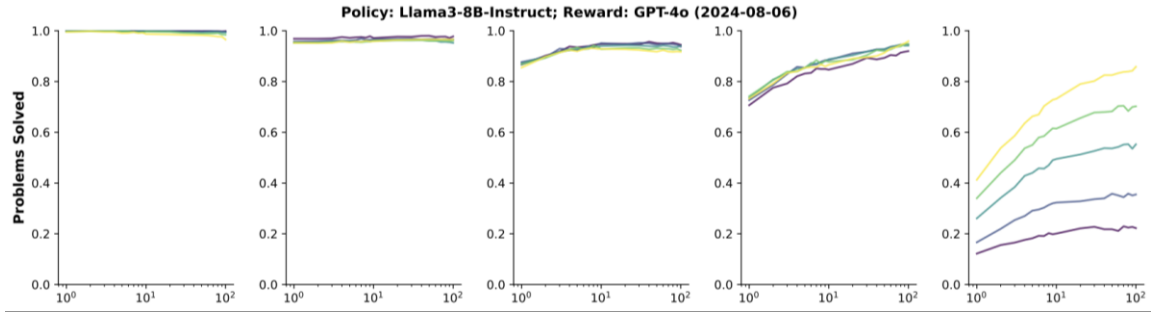
- 针对代码生成任务，将代码通过测试用例与否视作可验证奖励信号
- 发现：扩展单元测试用例的数量能显著提高代码生成的性能：





- 扩展单元测试用例数量在较难的代码题上效果提升更加显著：

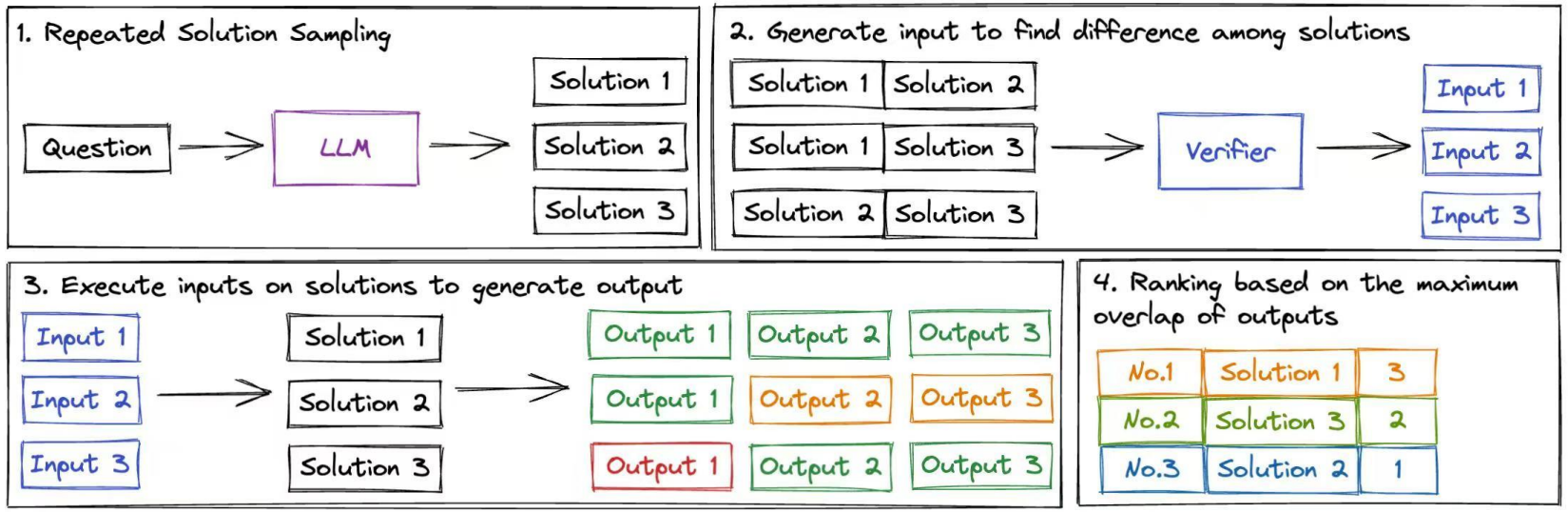


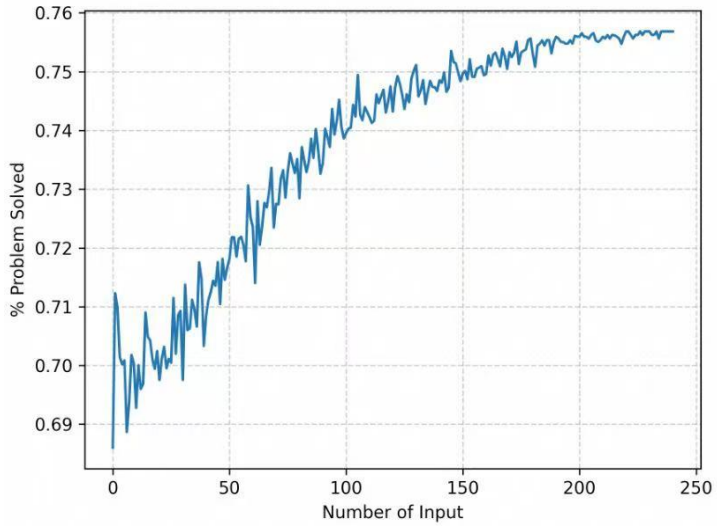


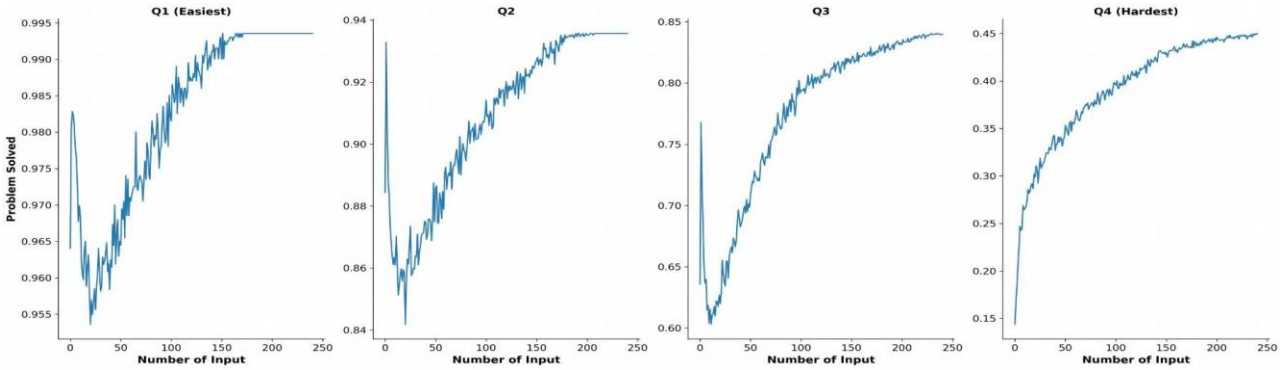
Zeyao Ma, Xiaokang Zhang, Jing Zhang, etc. Dynamic Scaling of Unit Tests for Code Reward Modeling. ACL'25.

测试用例生成模型

- 观察：给定符合代码题目要求的测试用例输入，正确代码的输出一致，错误代码的输出通常不同
- 训练一个测试用例生成模型，目标是生成可以把正确、错误代码答案区分开的测试用例输入。



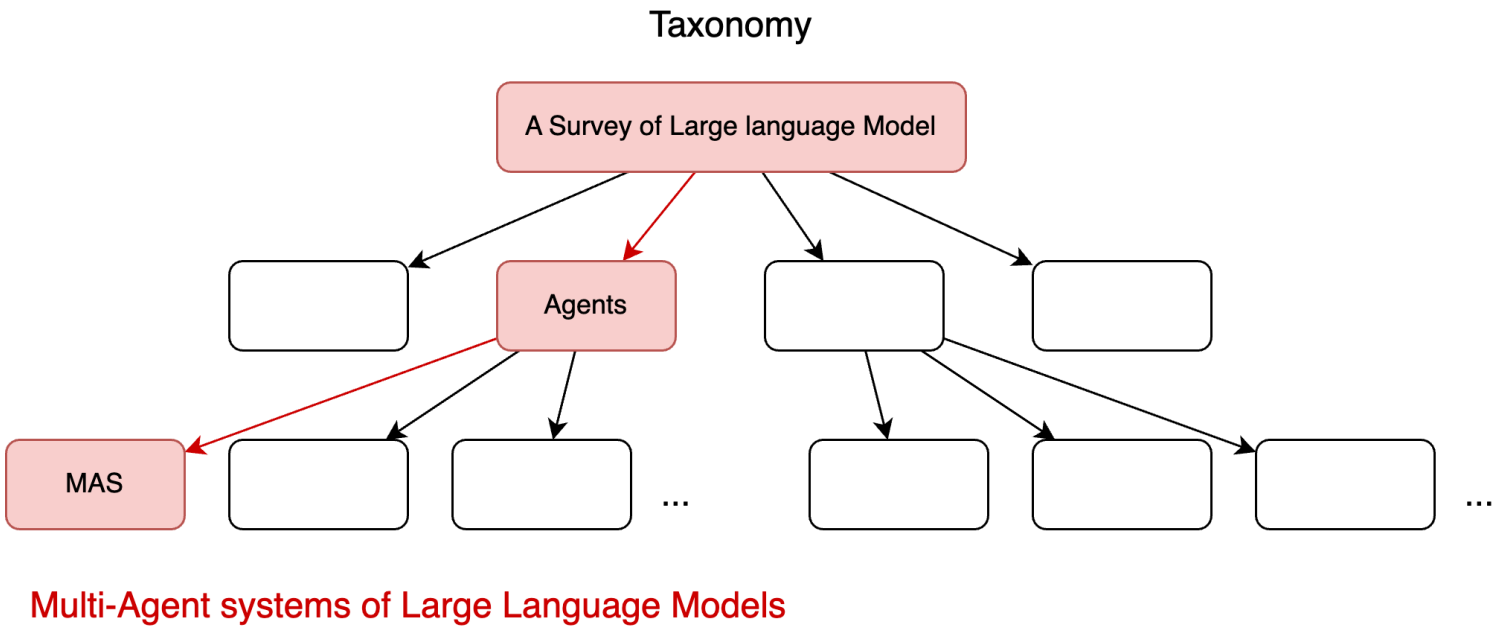




Dataset	LiveCodeBench	CodeForces	OJBench
Direct Infer	63.48	25.95	27.84
+verify	72.82	41.36	32.35
Improved	+9.34	+15.41	+4.51

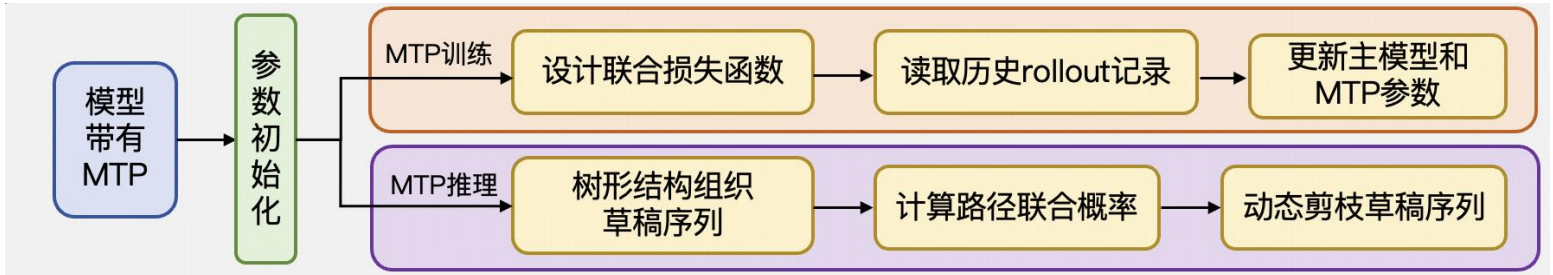
一种外部先验知识引导的合理行为路径Rollout策略

- 以Deep Research场景的论文搜索任务为例：
  - 已有的Rollout策略：论文搜索关键词 -> 调用搜索引擎 -> 得到论文列表
  - 针对Ground Truth论文列表设计的reward会导致reward稀疏问题
    - Rollout出来的关键词得到的论文列表都不符合预期
  - 外部知识引导：借助Survey Paper的Taxonomy来生成搜索关键词，提高Rollout命中率



一种基于投机解码的强化学习Rollout加速方法

- 基于Multi-token Prediction (MTP) 构建加速RL Rollout过程的通用投机解码方法。
- 设计主模型与草稿模型的联合损失函数实现两者协同优化，从而在强化学习过程中持续保持稳定的加速性能。



- 实验结果表明，在强化学习训练中引入交叉熵损失和 MTP 头与主模型输出分布的 KL 散度损失，能够有效约束模型分布偏移，提升生成稳定性与收敛质量，同时显著提高草稿头的接受长度。

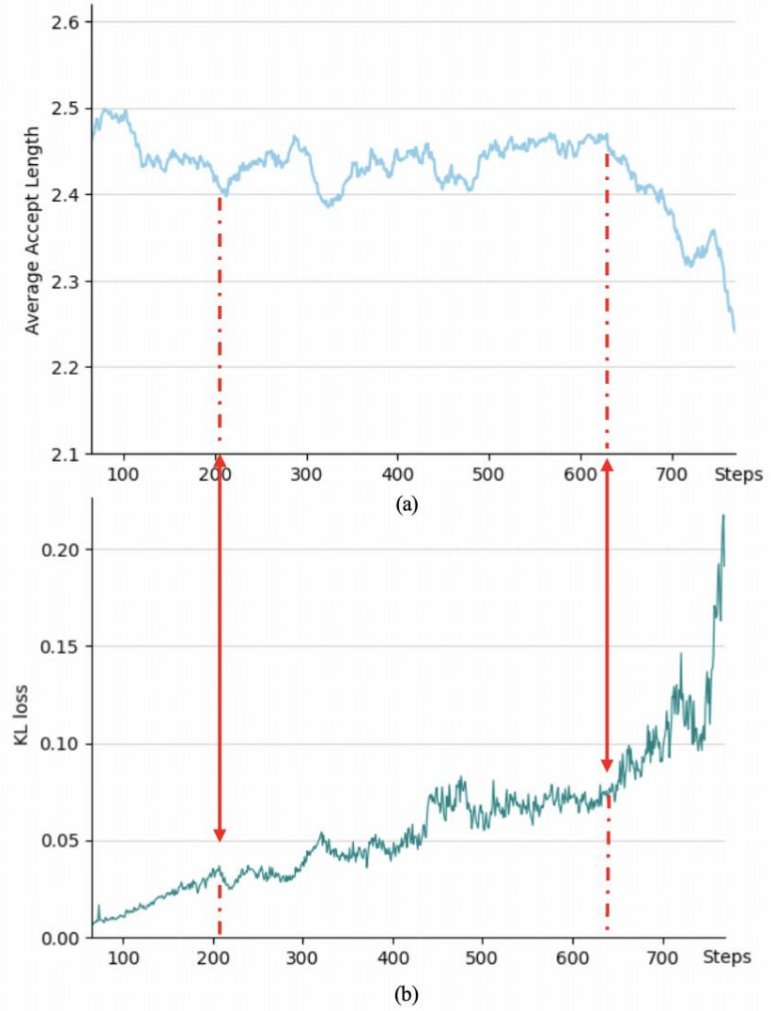


Figure 1: The variation tendency of average accept length and KL loss in RL training with frozen MTP module

提纲

Contents

01 背景与动机

02 可扩展、高保真的高效数据合成方法

03 稳定高效的强化学习

04 性能保持的推理加速与模型压缩

05 总结与展望

推理加速与模型压缩

- 关键科学问题
  - 如何利用外部信息或离线计算实现高效推理优化？

——即如何借助外部辅助模型输出（如小草稿模型）或主模型已有输入特征或离线计算，构建协同推理机制，在保证生成质量的前提下有效降低主模型的推理延迟。

  - 如何组合与优化多种稀疏化策略？

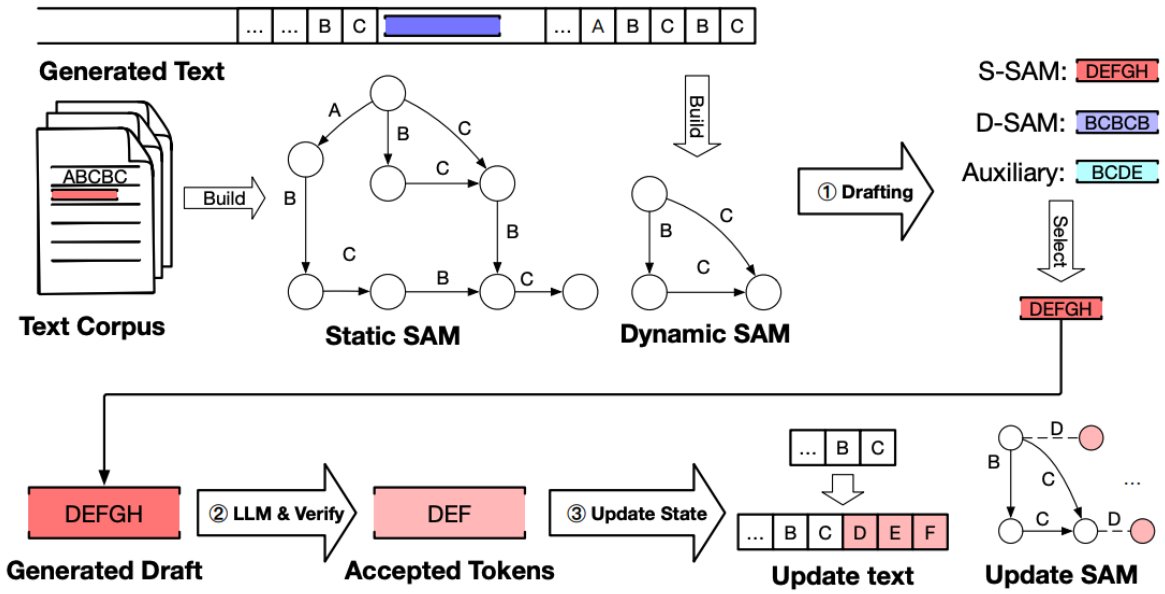
——即如何叠加不同形式的稀疏注意力机制（如 Sliding Window Attention、Compression Attention、Selection Attention 等），以在推理效率与模型性能之间实现最优平衡。

  - 如何保持压缩后的模型性能？

——即如何在模型压缩导致精度退化的情况下，通过数据再训练实现高压缩比条件下的性能稳健性。

一种面向检索型任务的投机解码方法：SAM-Decoding

- 发现：对于总结、润色、代码改写等任务，**模型生成的内容与**输入有较大的重合，因此可以通过输入预测模型可能的输出
- **提出基于**后缀自动机**快速**识别生成内容与输入文本的匹配位置
- **根据匹配**长度**自适**应选择匹配结果和生成结果**作**为草稿



Task: Summarization

Prompt: … On Wednesday, the two Scooby vans arrived at Clinton's appointed location separately, but left together in a seven-car motorcade. UNDER GUARD: Two Secret Service agents with Scooby One in Iowa today. ….

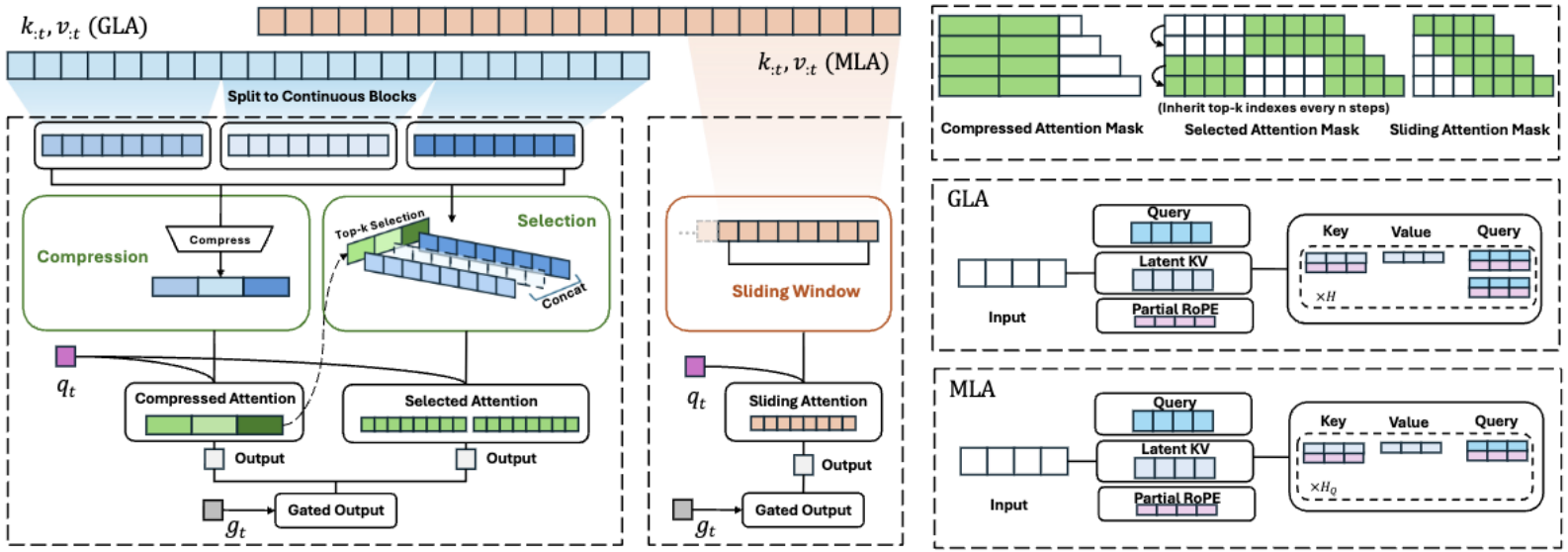
Answer: … The two Scooby vans arrived at Clinton's appointed location separately but left together in a seven-car motorcade. The second Scooby van's driver brought up the rear, …

- SAM Decoding**算法**应用于美团代码编辑产品，推理速度提升~1.5**倍**

Yuxuan Hu, Ke Wang, Xiaokang Zhang, Fanjin Zhang, Cuiping Li, Hong Chen, Jing Zhang. SAM Decoding: Speculative Decoding via Suffix Automaton. ACL'25.

一种层混合原生稀疏注意力模型架构

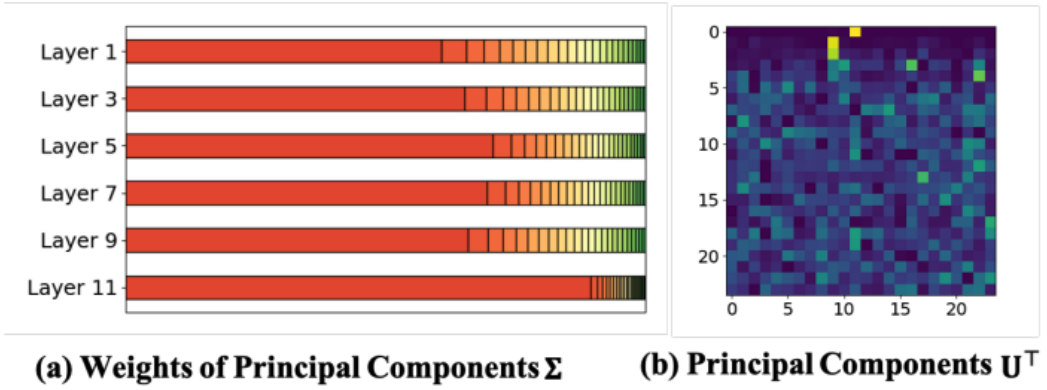
- 发现：
  - Sliding Window Attention在语言建模中起到主导作用
  - Selection Attention用于检索全局信息
  - Compression Attention虽有全局感受野但不能有效捕捉全局信息(仅起到索引作用)
  - Sliding Window Attention会阻碍Compression Attention/Selection Attention的检索能力
- 提出隔层混合Sliding Window Attention和Compression Attention/Selection Attention的模型架构



Yuxuan Hu, Jianchao Tan, Jiaqi Zhang, Wen Zan, Pingwei Sun, Yifan Lu, Yerui Sun, Yuchen Xie, Xunliang Cai, Jing Zhang. Optimizing Native Sparse Attention with Latent Attention an Local Global Alternating Strategies. https://arxiv.org/html/2511.00819v1

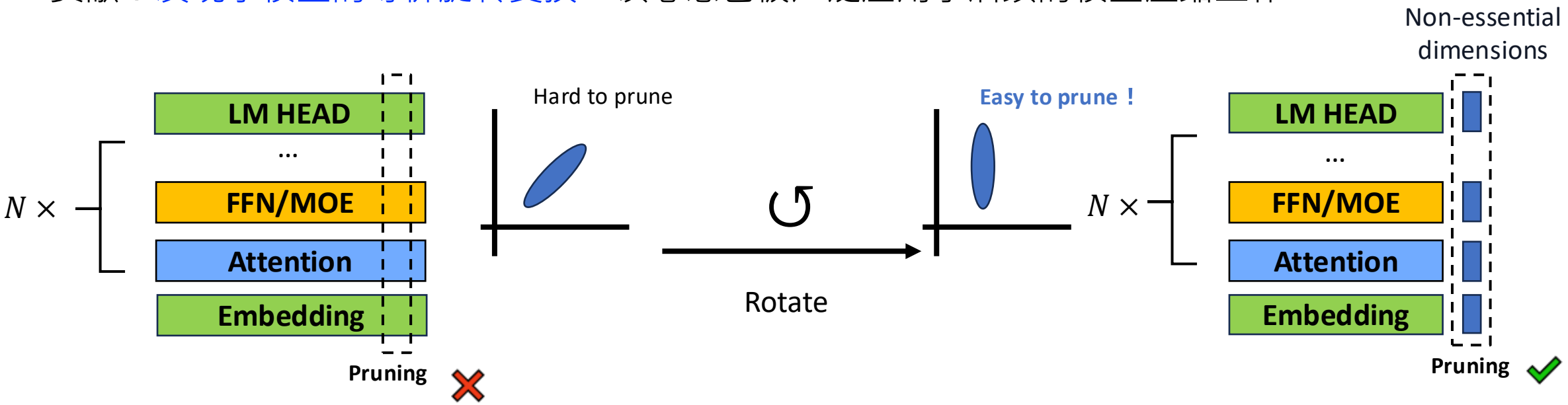
模型压缩: 一种维度剪枝方法

- 观察（对Transformer**模型的特征**进行奇异值分解后）：
  - **少数特征向量存在很大特征值 (主要成分)**
  - **主要成分与大多数**隐藏维度相关



(a) Weights of Principal Components Σ (b) Principal Components $U^\top$

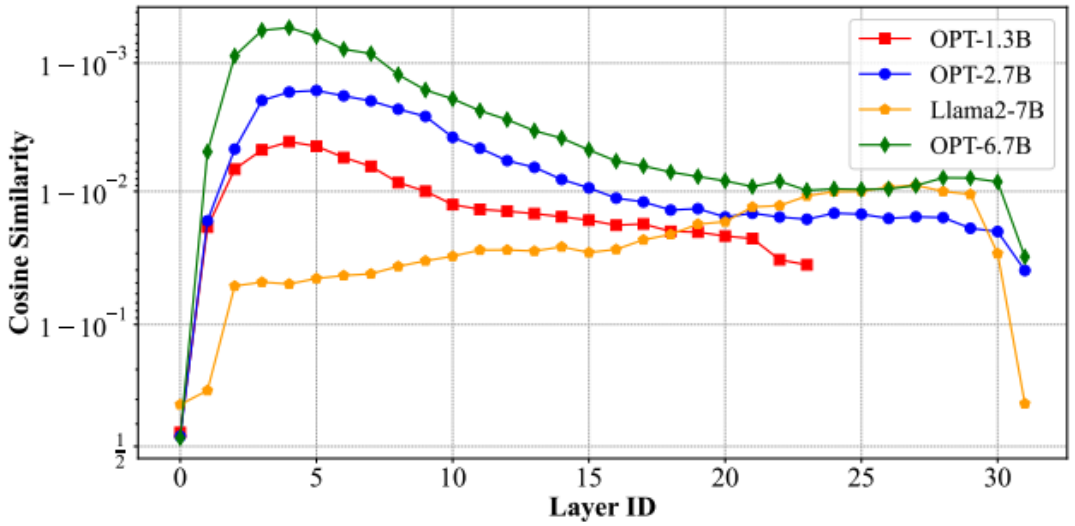
- **方法: 在**计算的过程中引入旋转矩阵，即**使用校准数据构建模型特征矩**阵，通过奇异值分解得到的**特征向量构建旋**转矩阵，**通**过该旋转矩阵对模型进行**等价**变换从而降低模型的剪枝难度
- 贡献：发现了模型的等价旋转变换，该思想也被广泛应用于后续的模型压缩工作。



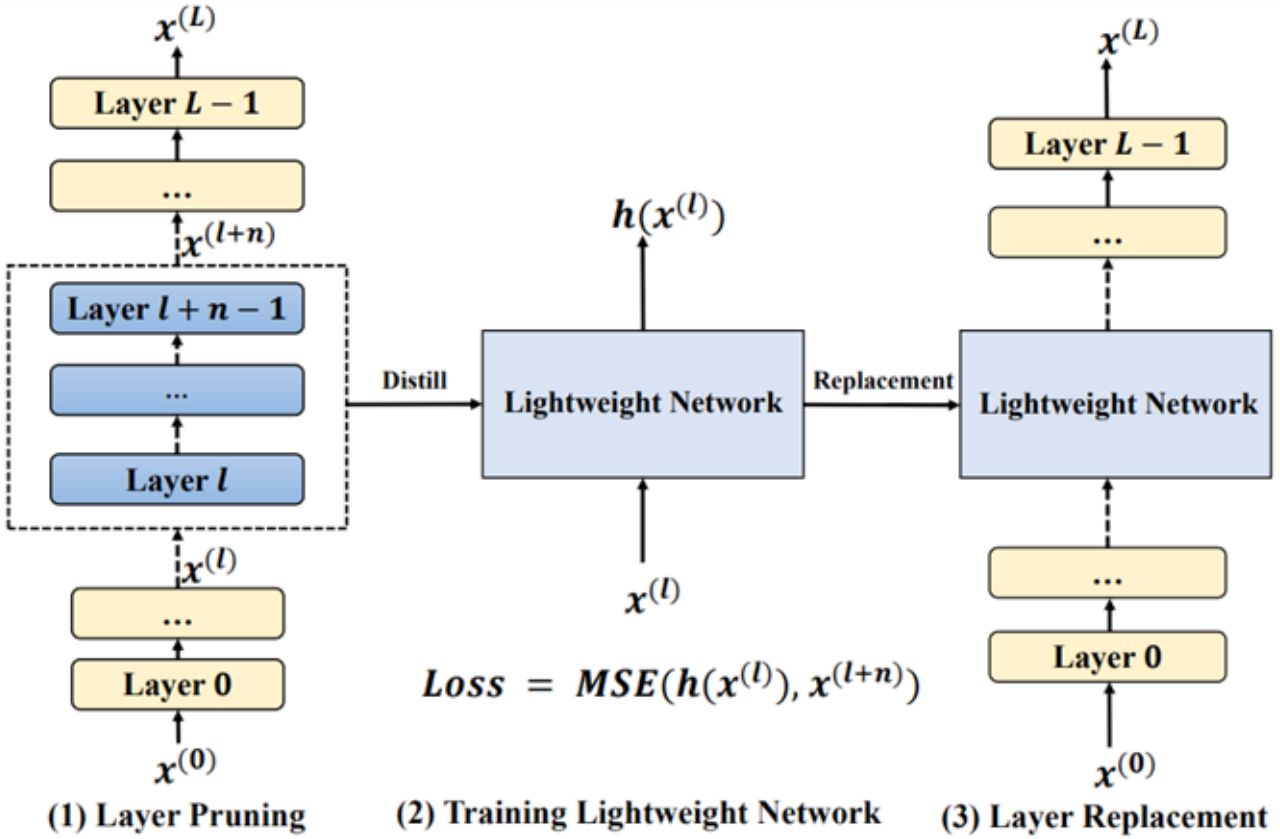
Yuxuan Hu, Jing Zhang, Zhe Zhao, Chen Zhao, Xiaodong Chen, etc. SP³: Enhancing Structured Pruning via PCA Projection. Findings of ACL'24.

模型压缩: 一种层剪枝方法

- 层剪枝：通过计算输入与输出隐藏状态之间的余弦相似度评估层的重要性，**并移除**连续低重要性的层。
- 层替换：使用通过均方误差（MSE）损失训练的轻量级网络来近似被剪枝层的功能。



$$\cos\left(\boldsymbol{x}^{(\ell)}, \boldsymbol{x}^{(\ell+1)}\right) = \mathbb{E}_{(\boldsymbol{x}_i^{(\ell)}, \boldsymbol{x}_i^{(\ell+1)}) \in \mathcal{D}} \left(\frac{1}{L} \sum_{j=1}^{L} \frac{\boldsymbol{x}_{i,j}^{(\ell)} \cdot \boldsymbol{x}_{i,j}^{(\ell+1)}}{\|\boldsymbol{x}_{i,j}^{(\ell)}\| \cdot \|\boldsymbol{x}_{i,j}^{(\ell+1)}\|} \right)$$



Xiaodong Chen, Yuxuan Hu, Jing Zhang, etc. Compressing Large Language Models by Streamlining the Unimportant Layer. ICLR'25.

模型压缩：剪枝后训练的Scaling Law

- 在 Llama-3 和 Qwen-2.5 系列模型上进行实验，使用深度剪枝、宽度剪枝和2:4半结构化剪枝方法，使用SlimPajama作为后训练数据集。
- 在后训练损失曲线中我们观察到了如下趋势：
  - 越小的LLMs的Loss下降越快。
  - 相对后训练损失与剪枝率呈幂律关系。
- 根据这些趋势，我们确定了P^2必须满足的必要条件：

① 后训练损失随数据量增加而降低。

$$\frac{\partial \mathcal{L}}{\partial D} < 0$$

② 在相似的剪枝率下，越小的模型损失下降越快。

$$\frac{\partial}{\partial N_0}\left(\frac{\partial \mathcal{L}}{\partial D}\right) = \frac{\partial^2 \mathcal{L}}{\partial N_0 \partial D} > 0$$

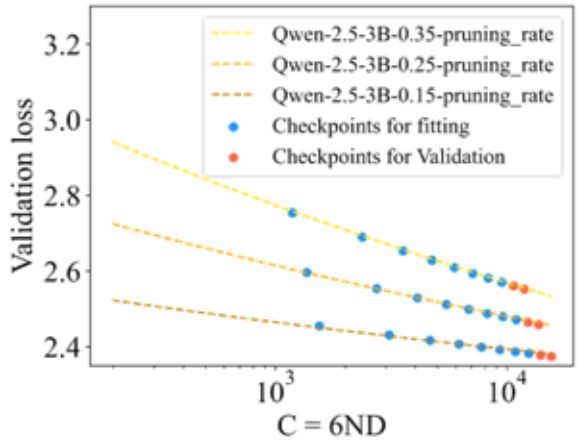
③ 相对后训练损失与剪枝率呈幂律关系。

$$\Delta \mathcal{L} \propto \left(\frac{1}{\rho}\right)^{\gamma}$$

P^2 Law:

$$\mathcal{L}_1(N_0, D, \rho, \mathcal{L}_0) = \mathcal{L}_0 + \left(\frac{1}{\rho}\right)^{\gamma}\left(\frac{1}{N_0}\right)^{\delta}\left(\frac{N_C}{N_0^{\alpha}} + \frac{D_C}{D^{\beta}} + E\right)$$

α, β, γ, δ , N_C, D_C 和E 是常数

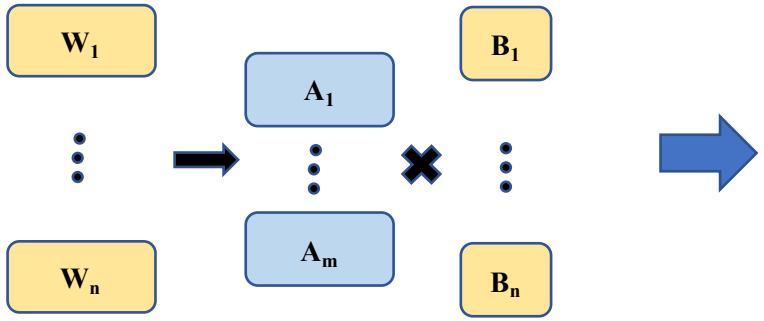


- P^2 法则在更大数据集规模、更大模型以及更高参数量上具有泛化能力

Xiaodong Chen, Yuxuan Hu, Xiaokang Zhang, Yanling Wang, Cuiping Li, Hong Chen, Jing Zhang. P^2Law: Scaling Law for Post-Training After Model Pruning. ACL'25.

模型压缩：一种重参数化MoE权重矩阵的方法

- SVD**分解法**：现有LLMs**中的**权重矩阵有些并非低秩的，**可能会造成**较大的重建误差
- **剪枝法**：**永久性**损失知识并需要大量训练成本来弥补丢失的知识
- **提出不依**赖低秩矩阵分解来减少参数，**而是以**重参数化方式**完成参数**压缩。



$$W_i = (\sum_{j=0}^{m} \alpha_j A_j^B) B_i$$

基底矩阵A负责学习所有注意力头（专家）**的共性，而矩**阵B则学习各注意力头（专家）**的个性**。

LLM	Method	Ratio	General Reasoning			General Knowledge			
			ARC-C	IFEval	GPQA	BBH	MMLU	CEval	CMMLU
Qwen3-30B-A3B-2507	MoE	0%	95.6	86.6	56.8	85.4	87.6	88.2	86.6
	D^2-MoE	24%	93.1	83.5	45.2	69.9	83.3	71.2	68.6
	MoLAE	24%	92.5	79.2	46.3	76.5	80.3	76.0	74.9
	MoNE	24%	92.5	85.1	52.0	80.7	80.1	71.2	68.2
	Sub-MoE	24%	92.5	82.1	46.3	82.8	80.3	70.2	68.5
	FaRE	24%	**96.6**	**86.9**	**52.1**	**83.5**	**85.6**	**85.1**	**83.5**

Xiaodong Chen, Mingming Ha, Zhenzhong Lan, Jing Zhang, Jianguo Li. MoBE: Mixture-of-Basis-Experts for Compressing MoE-based LLMs. Submitted.

提纲

Contents

01 背景与动机

02 可扩展、高保真的高效数据合成方法

03 稳定高效的强化学习

04 性能保持的推理加速与模型压缩

05 总结与展望

总结与展望

大模型对齐技术研究及应用

	研究挑战	研究成果	示范应用	潜在研究
数据	数据稀缺	**数据合成** • 可扩展的Web采集与交互仿真方法 • 高可靠性的交叉验证机制 • 高效的数据合成流程	**数据智能** • RESDSQL / CodeS / OmniSQL 系列 Text2SQL模型在 Spider/Bird/Spider2 榜单连续3个多月 位居榜首 • OmniSQL 在 Hugging Face / ModelScope 下载量 突破 10 万次 • 支持字节跳动、中国移动等企业内部数据库业务	• 多轮工具调用路径的高保真合成 • 具身智能真机下的可扩展合成 • 训推分布不一致 / 工具演化 / Rollout稀疏化 • 面向智能体的稀疏化和Memory机制
训练	SFT易过拟合 RL不稳定易崩溃	**强化学习** • 可扩展的代码生成奖励机制 • 先验引导的路径探索策略 • 基于投机解码的强化学习加速		
推理	上下文受限 延迟高/本地部署难	**模型高效化优化** • 维度、层剪枝与Scaling Law • 面向检索场景的投机解码 • 层混合原生稀疏注意力模型架构		

感谢，请批评指正！