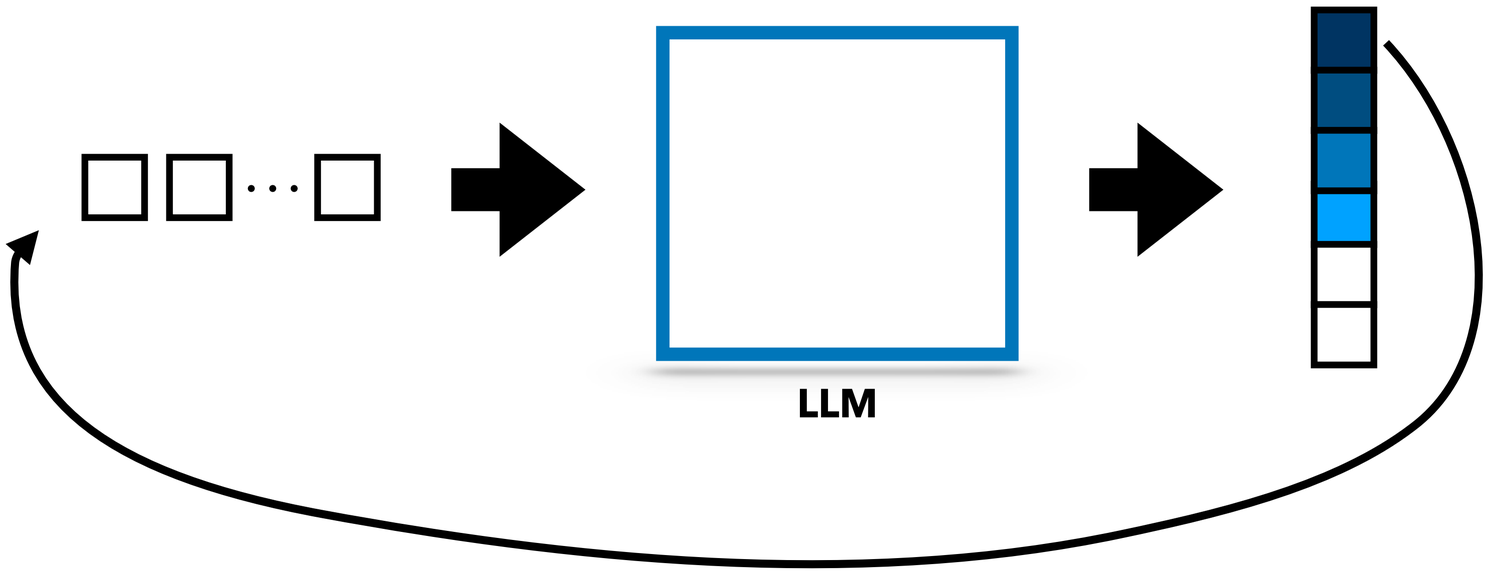
LLM Reasoning

从数据增强到推理增强

LeiWu 2024-11-15

Back to Basic

- 自回归语言模型根据prompt（一串token）预测出一个维度为vocab_size的softmax分布
- 我们根据某种采样规则选择一个token作为next-token，将其加入prompt序列，并重复该过程直到遇到终止条件（max_length, EOS, …）



Back to Basic

我们的目标是让LLM能根据prompt (query)，输出一系列token，由这些token组成我们满意的answer。

Answer本身是一串token的排列组合，假设answer包含 d 个token，于是任务可以抽象为在一个 d 维的空间中搜索出一个满意解。

由于自回归语言建模的特性，每次生成的token被加入下一轮输入序列，于是搜索空间的维度递归地减小。也可以理解成我们不断地在搜索空间中施加约束，限制搜索空间的大小。

Back to Basic

我们能做什么?

- Pre-training
- Post-training
- Inference-time

Back to Basic

- Pre-training 阶段，使用海量语料为模型注入世界知识

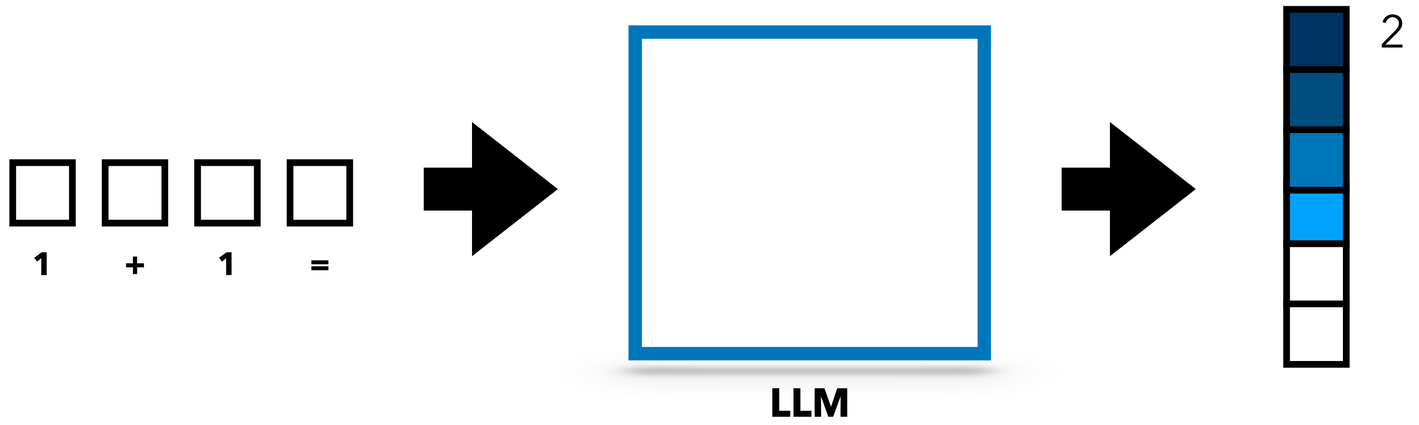
当前主流观点认为，模型在此阶段学习全部所需知识，Post-training 阶段则不再学习新知识而是加强指令跟随能力。若是 Post-training 阶段注入与预训练阶段冲突的知识则可能导致出现幻觉。

- Post-training 阶段，使用 (Instruct, Response) 数据对，使模型内部知识之间建立起联系

Back to Basic

Post-training

- SFT 的目的是让模型在接收到 Instruct 时，确保 Response 中的 tokens 赋予较高的概率
- Preference Learning 的目的是提高合意的 tokens 的概率，降低不合意的 tokens 的出现概率



Back to Basic

Inference Time

理想情况：我们有一个经过预训练与完美对齐的模型，面对 Instruct ，模型能将期望响应的每个 token 都赋予最高概率值，此时通过简单的贪心解码就可以获得正确答案。

真实情况：模型只能保证将一系列有希望的 tokens 赋予一簇概率团。需要设计搜索策略，在一个个概率团间找到最终的解。同时使用 CoT 压缩搜索空间。

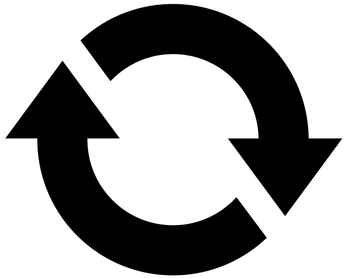
根据搜索空间的粒度可将搜索算法大致分类为：

- token-level：Top-P，Top-K，Beam-Search，…
- Step-level: ToT, MCTS, Lookahead Search …
- Response-level: Self-consistency, Best-of-N, …

Back to Basic

现阶段的工作都做了什么?

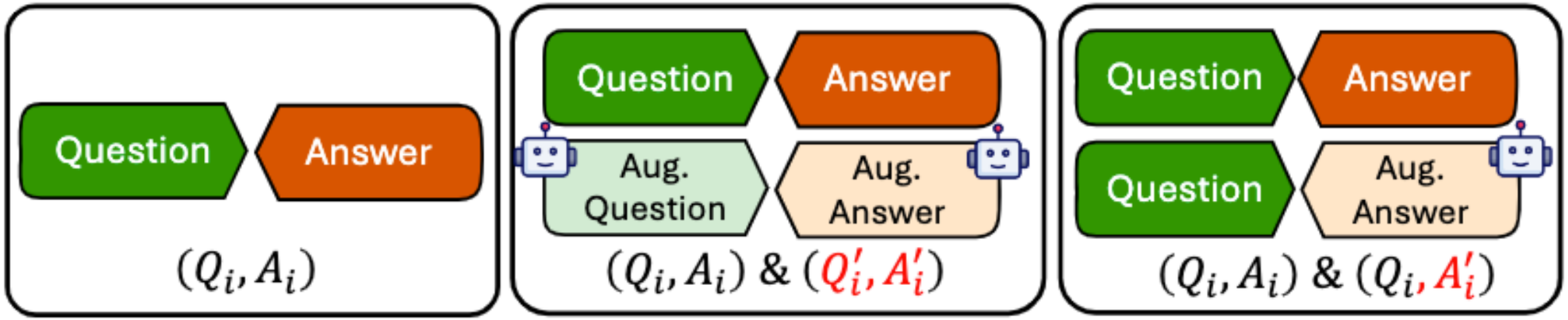
- Pre-training
  - 大规模高质量语料数据
  - 训练数据配比，合成数据（参考Llama3，Qwen2.5-Math 等技术报告）
- Post-training
  - 数据增强 pipeline
  - SFT or Preference Learning (DPO, PPO, …)
- Inference time
  - CoT，PoT，Self-consistency
  - Best-of-N，Beam Search，Lookahead Search（这一排需要奖励模型或 oracle answer）
  - Tree Search：BFS，DFS，MCTS（奖励模型可有可无）



通过推理时搜索到高质量的路径，构建合成数据集，进行迭代 self-improvement

数据增强

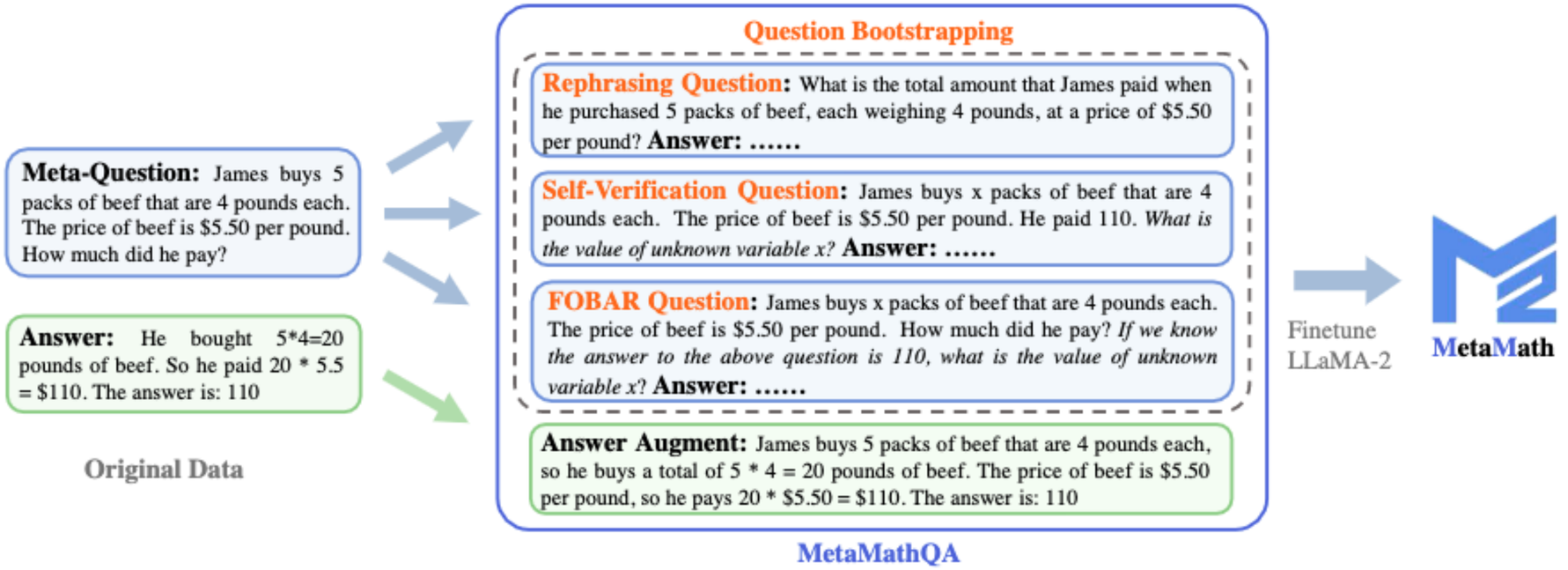
核心思想：从数量和质量上扩展现有的数据集。



(a) Standard data (b) Question augmentation (c) Answer augmentation

数据增强

Question Augmentation

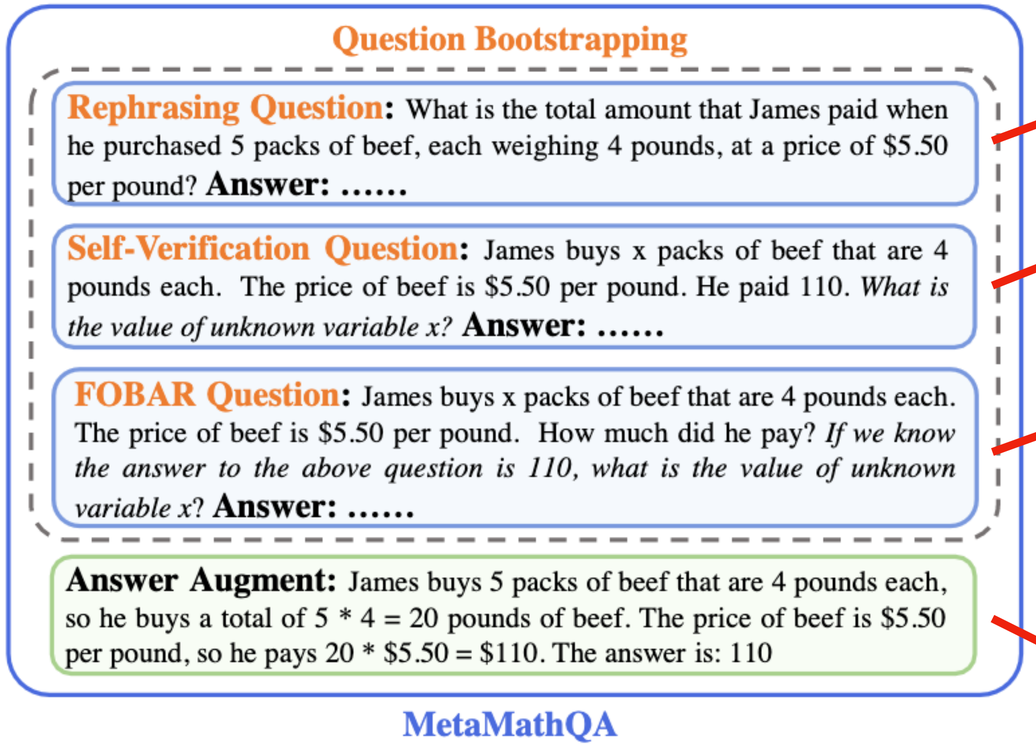


观点：一道数学题仅是底层元知识的一个视角。因此，可以进行一种多视角增强，以实现元知识的迁移。其次，增加问题的多样性可以扩展数据集的分布，以覆盖更广的场景。

MetaMath: Bootstrap Your Own Mathematical Questions For Large Language Models (ICLR 2024)

数据增强

Question Augmentation



使用 GPT-3.5-Turbo 生成问题的答案

重新表述问题，确保LLM准确理解题目

将原问题中的条件设为未知数x，将原始疑问句改为陈述句，要模型求解x

由于有些问题难以转化为连贯的陈述句，所以直接将原始问题中某个条件设为x，并在末尾加上一句 " if we know the answer to the above question is 110 , what is the value of unknown variable x? "

简单的 few-shot CoT 配合高温采样，对同一问题生成多个推理路径。

MetaMath: Bootstrap Your Own Mathematical Questions For Large Language Models (ICLR 2024)

数据增强

Question Augmentation

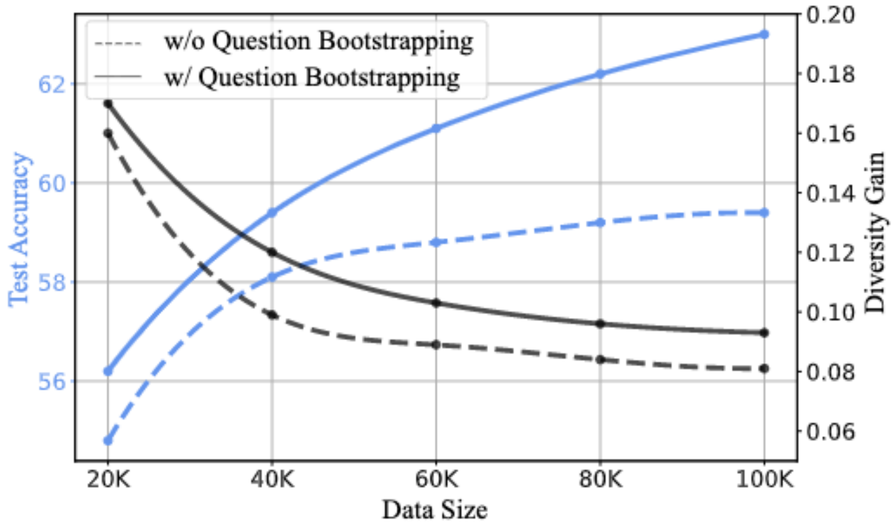


Figure 2: GSM8K accuracy of LLaMA-2-7B finetuned on different sizes of answer augmentation data. A larger diversity gain indicates the question is more diverse compared to the existing questions. Detailed experimental setup is given in Section 4.1.

如何度量Question Bootstrapping带来的数据集多样性增强?

$$d_{\text{gain}} = \frac{1}{N_{\text{new}}} \sum_{x_i \in \mathcal{D}_{\text{new}}} \min_{x_j \in \mathcal{D}_{\text{base}}} \left(\| f(x_i) - f(x_j) \|_2^2 \right)$$

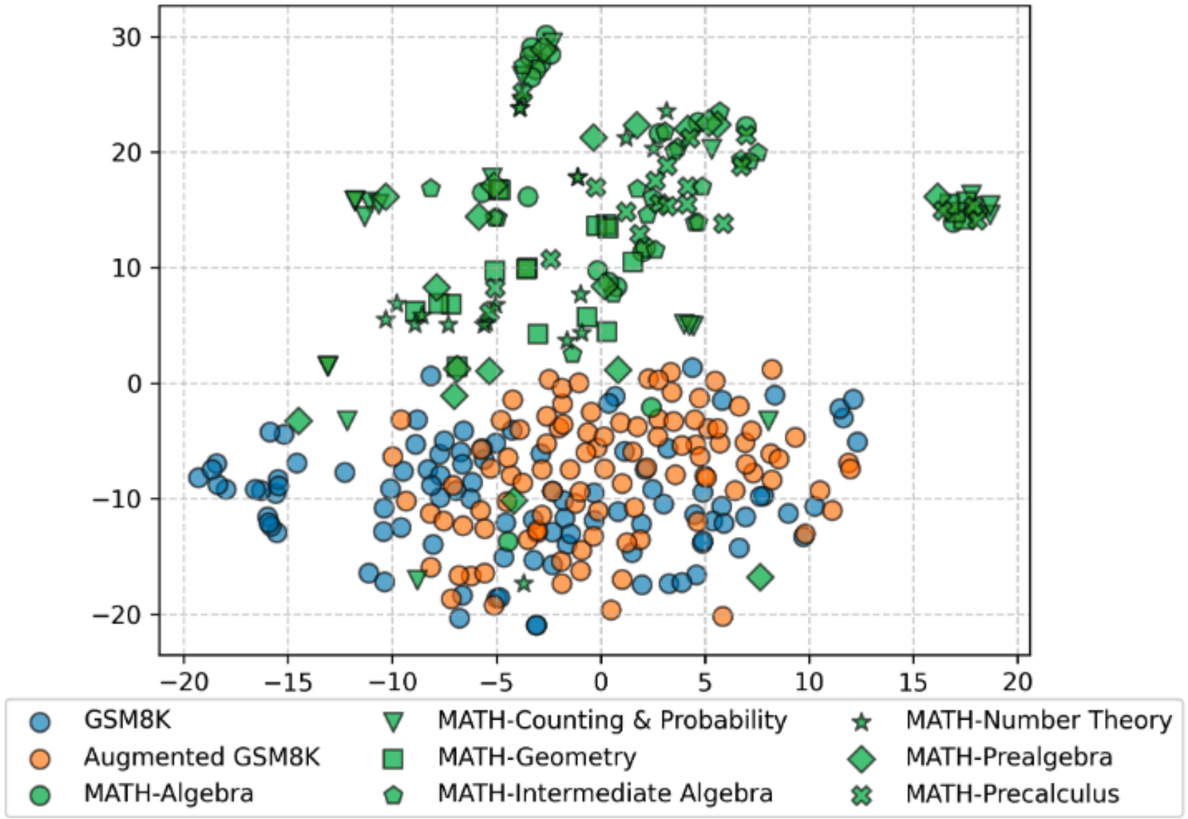
其中f(.)为embedding操作，文章中使用OpenAI Embedding API text-embedding-ada-002

MetaMath: Bootstrap Your Own Mathematical Questions For Large Language Models (ICLR 2024)

数据增强

Question Augmentation

在 MetaMath 的基础上又增加几种问题改写的方式（改变原题数字、引入小数分数等）。缺点在于改变了原始问题的答案，只能寄希望于强大的教师模型有较高的准确率，来为新问题生成正确的答案。



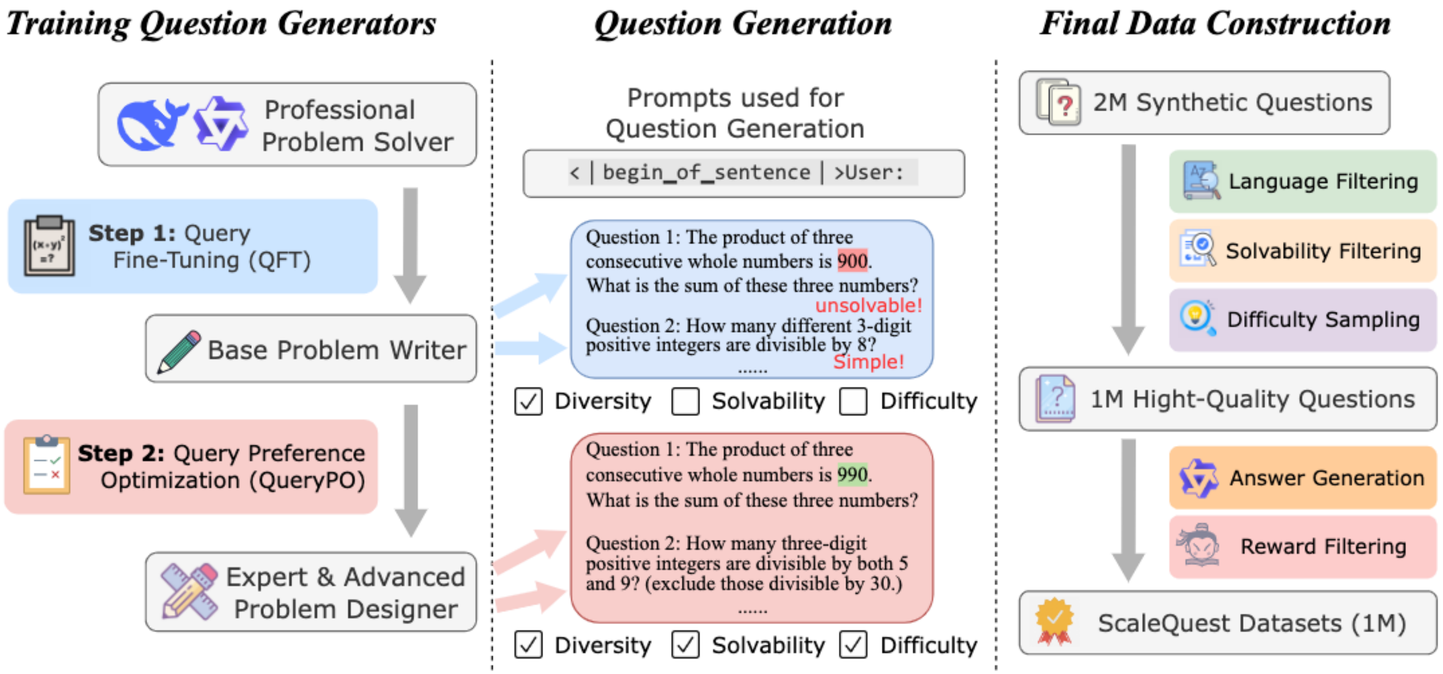
使用 t-SNE 算法对Llama2-7B模型第15层，
各种数据集的embedding可视化

MuggleMath: Assessing the Impact of Query and Response Augmentation on Math Reasoning (ACL 2024)

数据增强

Question Augmentation

微调开源数学模型来大量合成数学题



IDEA：一个专业的数学LLM，在大量数学QA数据上进行过训练，本身就会很出题。相当于花式蒸馏出Qwen2-Math的预训练数学数据集

使用 Qwen2-Math-7B-Instruct 来生成答案

相比上一篇MuggleMath，使用开源奖励模型 InternLM2-7B-Reward 来选出 Best-of-5 来确保生成答案的正确性。

UNLEASHING REASONING CAPABILITY OF LLMS VIA SCALABLE QUESTION SYNTHESIS FROM SCRATCH

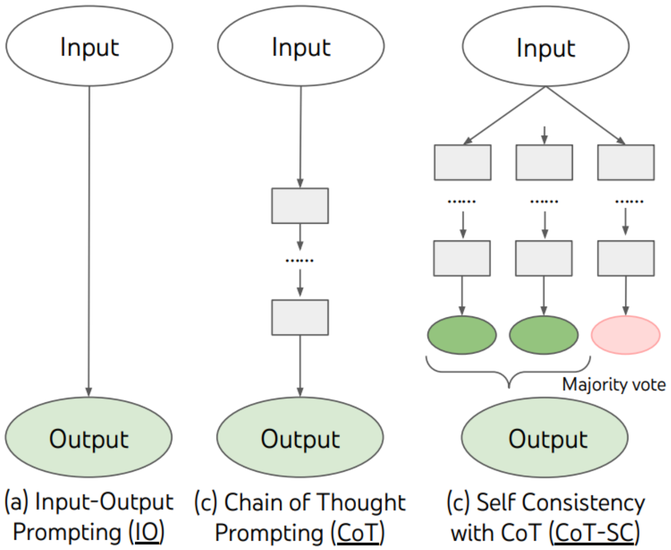
数据增强

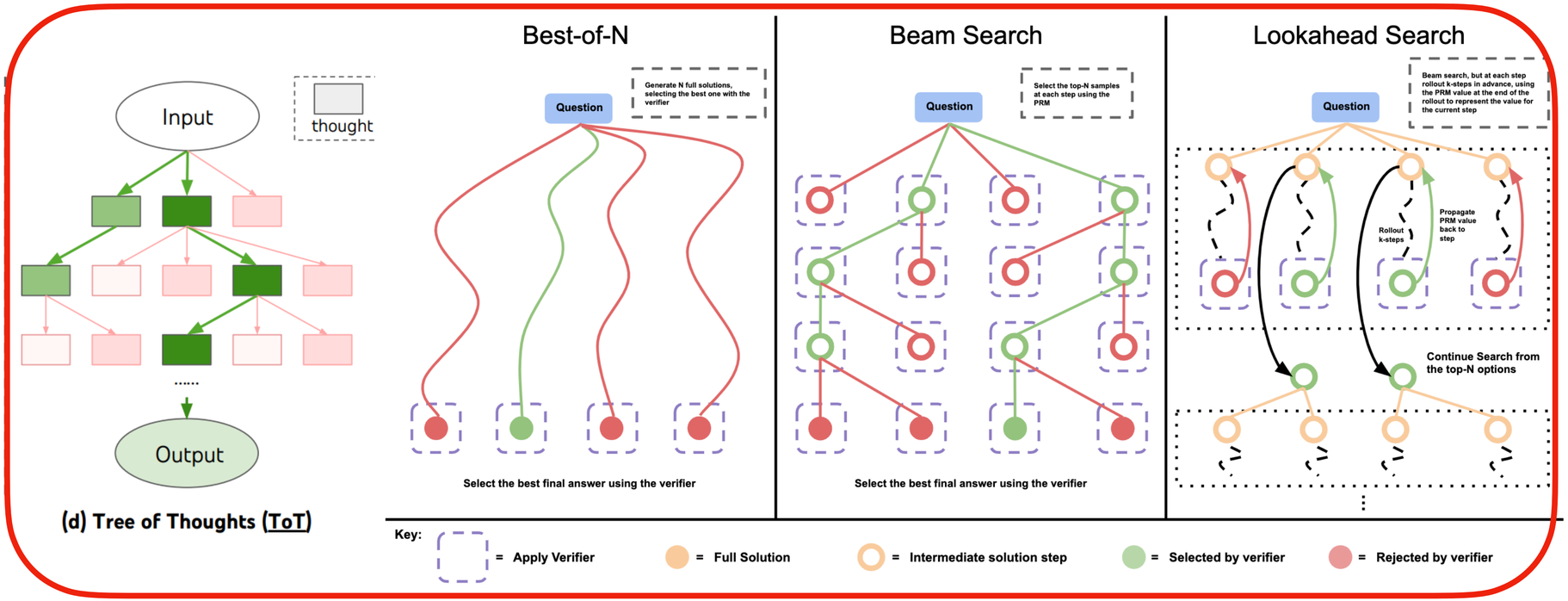
Answer Augmentation

答案增强的核心在于利用强大的LLM，或用推理增强策略助力一般LLM，生成高质量、详细的推理路径。由于使用强大的闭源模型成本较高，所以该类方法一般与推理增强结合使用，利用推理增强技术生成高质量的answer，然后用高质量的answer进一步微调开源模型，实现 self-improvement 或称 self-training

推理增强

Overview





需要奖励模型或Oracle Answer进行评价、剪枝

推理增强

Reward Model

	ORM	PRM	Majority Voting
% Solved (Best-of-1860)	72.4	**78.2**	69.6

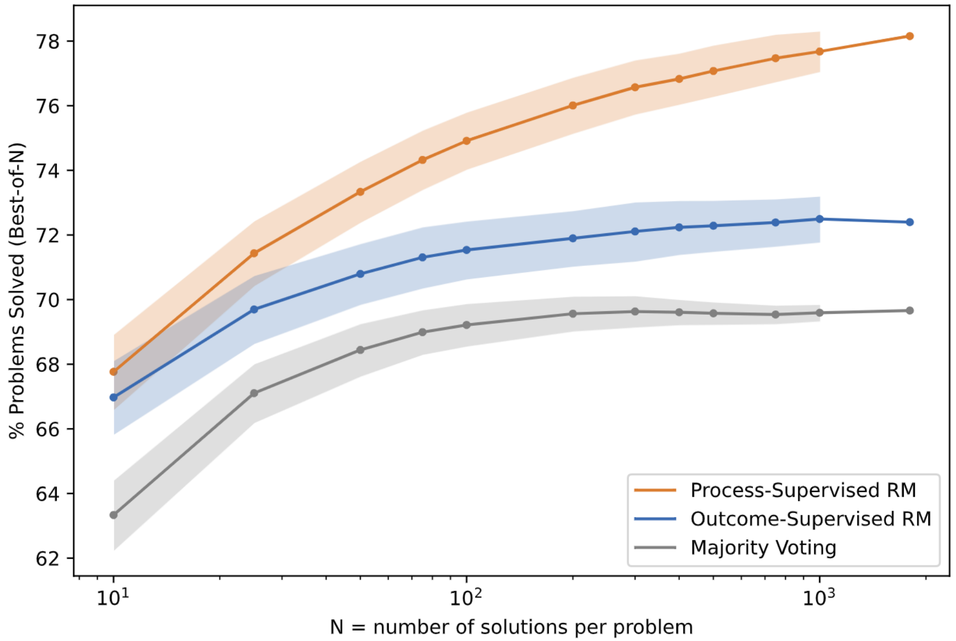


Figure 3: A comparison of outcome-supervised and process-supervised reward models, evaluated by their ability to search over many test solutions. Majority voting is shown as a strong baseline. For $N \leq 1000$, we visualize the variance across many subsamples of the 1860 solutions we generated in total per problem.

Training Method

微调预训练LLM，给定一个部分解，预测类别（-1，0，1）。由于要微调一个在语言建模任务上预训练的LLM执行分类任务，所以存在较大的分布偏移，需要使用较小的学习率来稳定训练过程。

Scoring Method

每步的得分为该步label预测为1的概率。具体执行时需要考虑将0视为positive或者negative，以及决定如何聚合step-level的得分得到solution-level的得分（取各步最小值或各步得分连乘），具体如何设置评分方法，论文中根据实验给出以下结果：

	product	minimum
neutral = positive	78.2%	77.6%
neutral = negative	77.4%	77.8%

Table 4: Best-of-1860 test performance using the PRM with four different scoring strategies.

Let's Verify Step by Step (ICLR 2024)

推理增强

Reward Model

训练 PRM 需要大量昂贵的人工标签，如何避免？-->蒙特卡洛估计

- **蒙特卡洛估计**是一种使用随机采样来估算未知量的统计方法。通俗来说，就是通过反复**随机尝试**得到大量样本，然后用这些样本的平均值来估计一个整体的结果。

- **蒙特卡洛树搜索**是一种用于决策问题（特别是游戏 AI）的算法，结合了随机采样和树结构来寻找最优决策路径。简单来说，它会尝试探索可能的决策路径，并不断优化。它的过程分为几个阶段：

  场景：比如在下围棋或玩象棋时，要决定下一步最优的棋步。

  过程：

  1. **选择**：从当前的棋盘状态开始，顺着已经探索过的路径，找到最有前景的分支。
  2. **扩展**：如果还没走到终点，就在当前路径的末端“扩展”一个新节点，代表一个新的决策状态。
  3. **模拟**：从这个新状态开始随机模拟游戏的后续步骤，直到结束。
  4. **回溯更新**：根据模拟结果更新路径上各个节点的胜率。

  通过多次模拟，MCTS 逐渐找到获胜几率最大的路径，最终选出最优决策。

推理增强

Reward Model with Mento Carlo Estimate

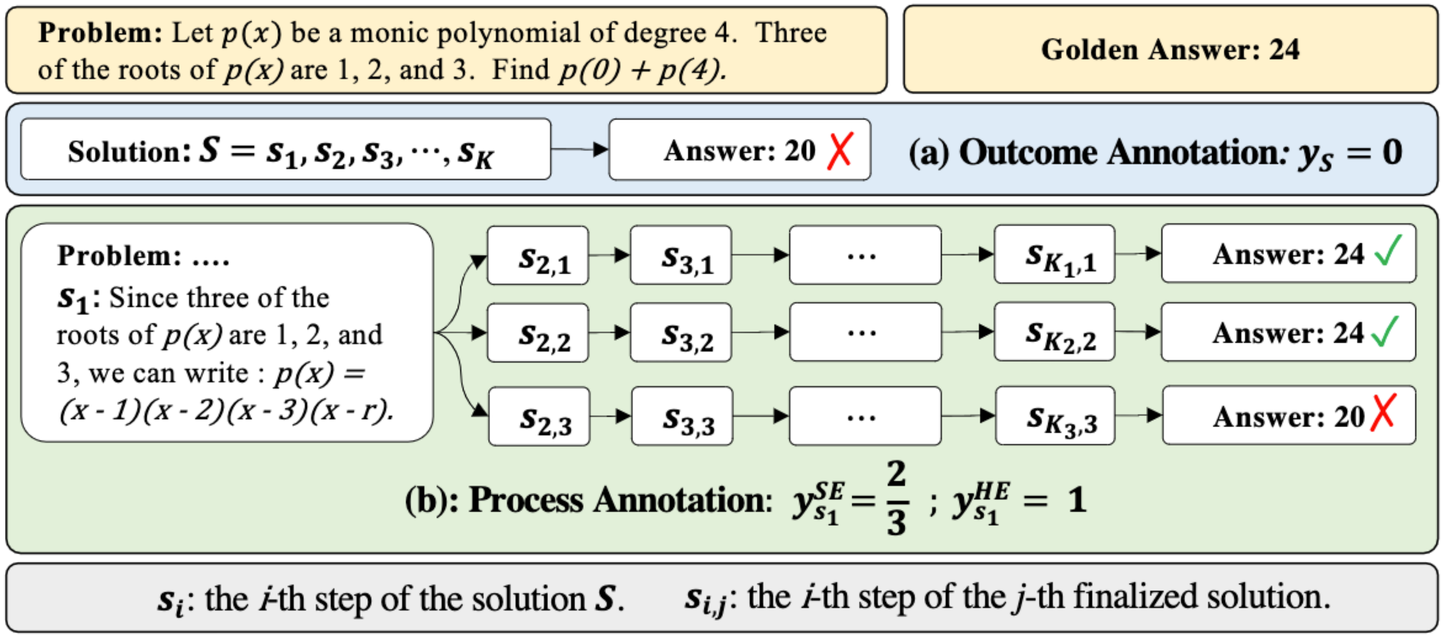


Figure 1: Comparison for previous automatic outcome annotation and our automatic process annotation. (a): automatic outcome annotation assigns a label to the entire solution S, dependent on the correctness of the answer; (b) automatic process annotation employs a 'completer' to finalize N reasoning processes (N=3 in this figure) for an intermediate step (s_1 in this figure), subsequently use hard estimation (HE) and soft estimation (SE) to annotate this step based on all decoded answers.

通过大量采样估计单步过程奖励，将每步推理步骤的质量定义为推理出正确答案的潜力。

具体做法：以每个步骤为起点进行N次随机模拟，推理出N个答案，使用软、硬两种估计方法为该步骤打分：

软估计 (SE)：N次推理，推理出正确答案的概率

硬估计 (HE)：N次只要有一次推理出正确答案，就赋为1

缺点：地毯式的遍历，消耗大量计算资源

Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations (ACL 2024)

推理增强

Reward Model with MCTS

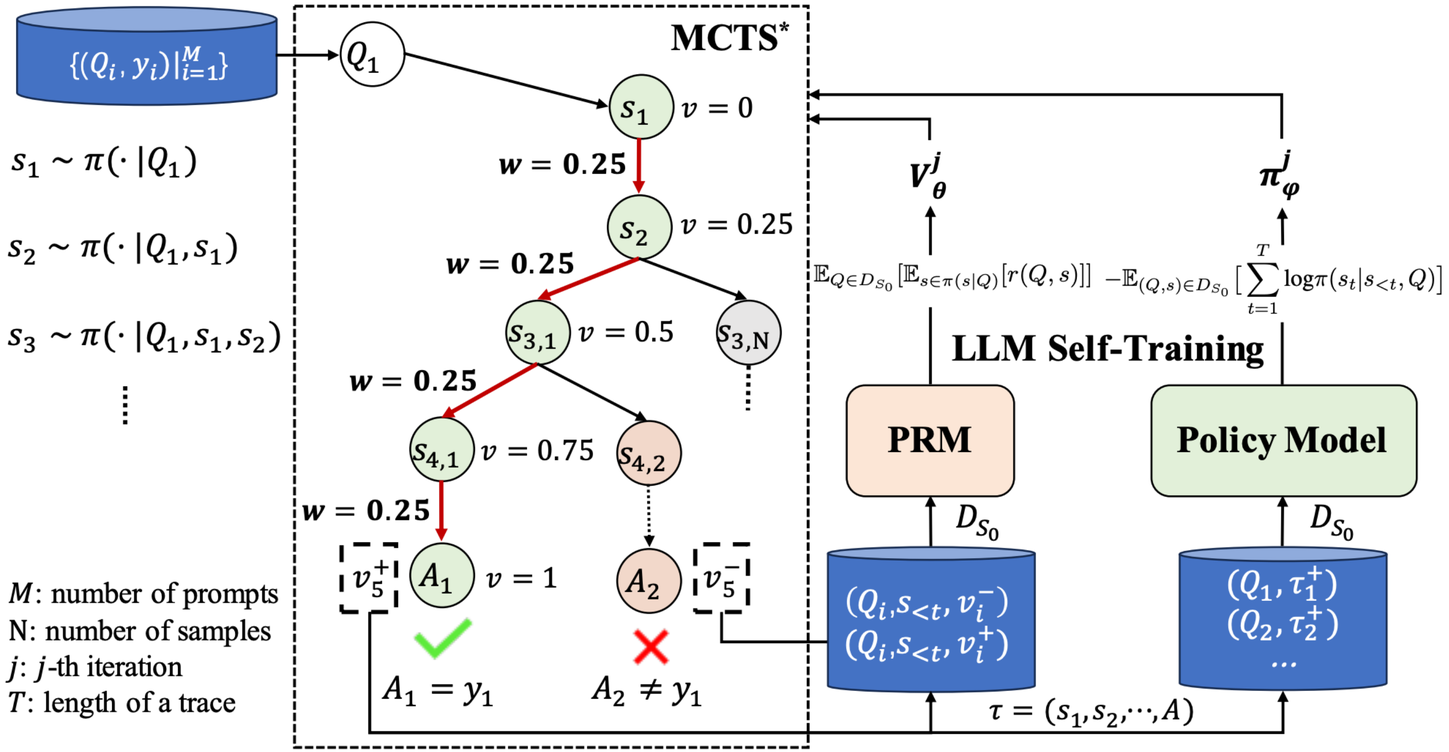


Figure 1: The left part presents the process of inferring process rewards and how we conduct process reward guide tree-search. The right part denotes the self-training of both the process reward model and the policy model.

MCTS相比于普通树搜索的优势：根据每个节点的访问次数与模拟结果，动态地平衡探索与利用，优先探索更有希望的路径。

令 v 作为过程奖励值，在反映当前步骤的正确性的同时，还反映当前步骤距离最终答案有多远，当 v->1 时即可认为当前推理路径已到达正确答案，可以终止探索。

但由于错误路径上，v 将不会接近于1，为了确保能及时终止错误路径的探索使用 self-critic，即令模型在每次节点选择之后根据问题与当前路劲判断是否应该停止探索。

ReST-MCTS*: LLM Self-Training via Process Reward Guided Tree Search

推理增强

Reward Model with MCTS

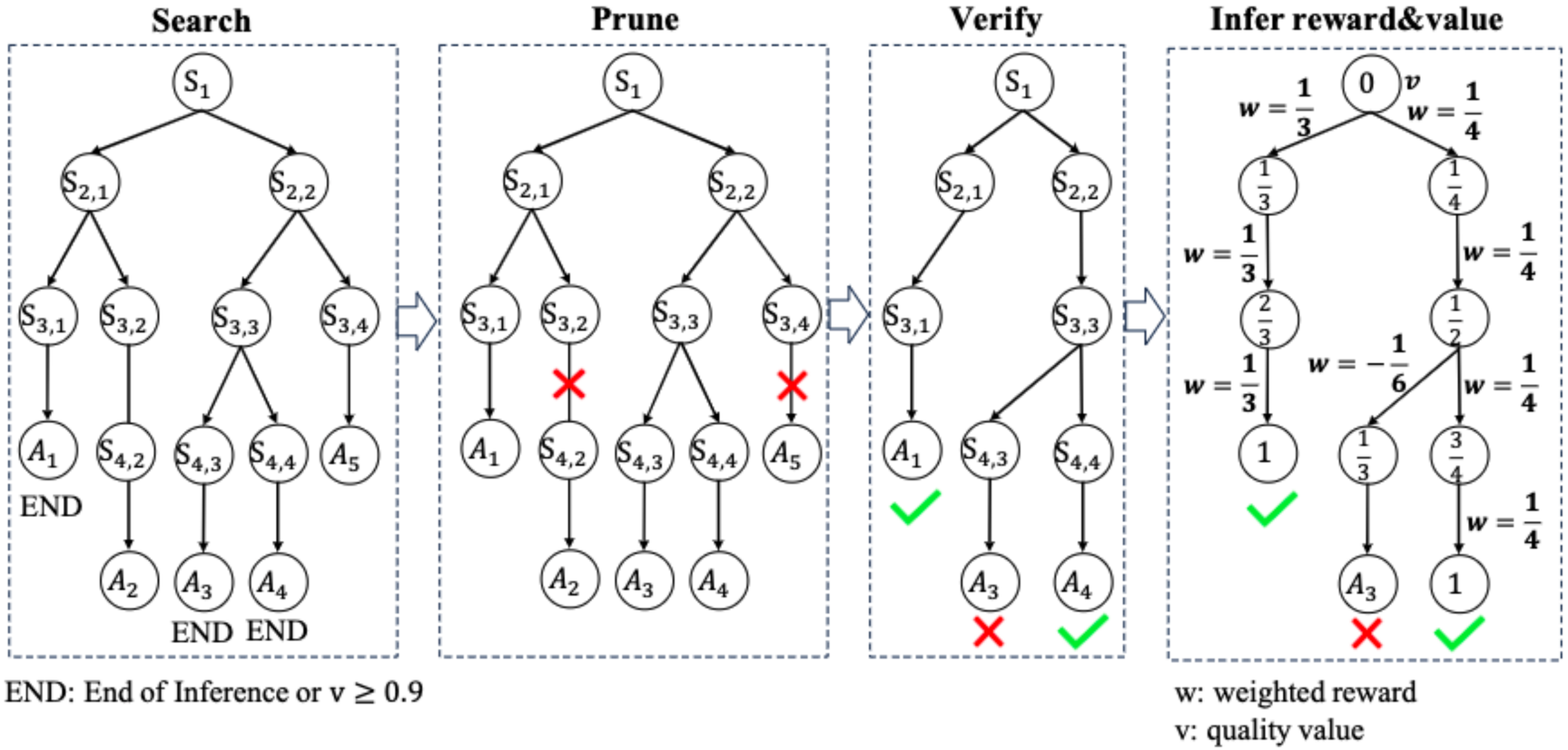


Figure 4: Detailed process of new sample data generation for the self-training framework.

ReST-MCTS∗: LLM Self-Training via Process Reward Guided Tree Search

推理增强

设计一个搜索策略的关键

- 如何设计搜索空间？token, step, solution or problem-specific ?
- 如何剪枝，避免搜索空间爆炸？random rollout with PRM ?
- 以什么策略平衡探索与利用？MCTS' UCB or UCT ?
- 什么时候终止某条路径上的探索？pre-define max depth ?
- 如何确保探索路径的多样性？high temperature sampling ?

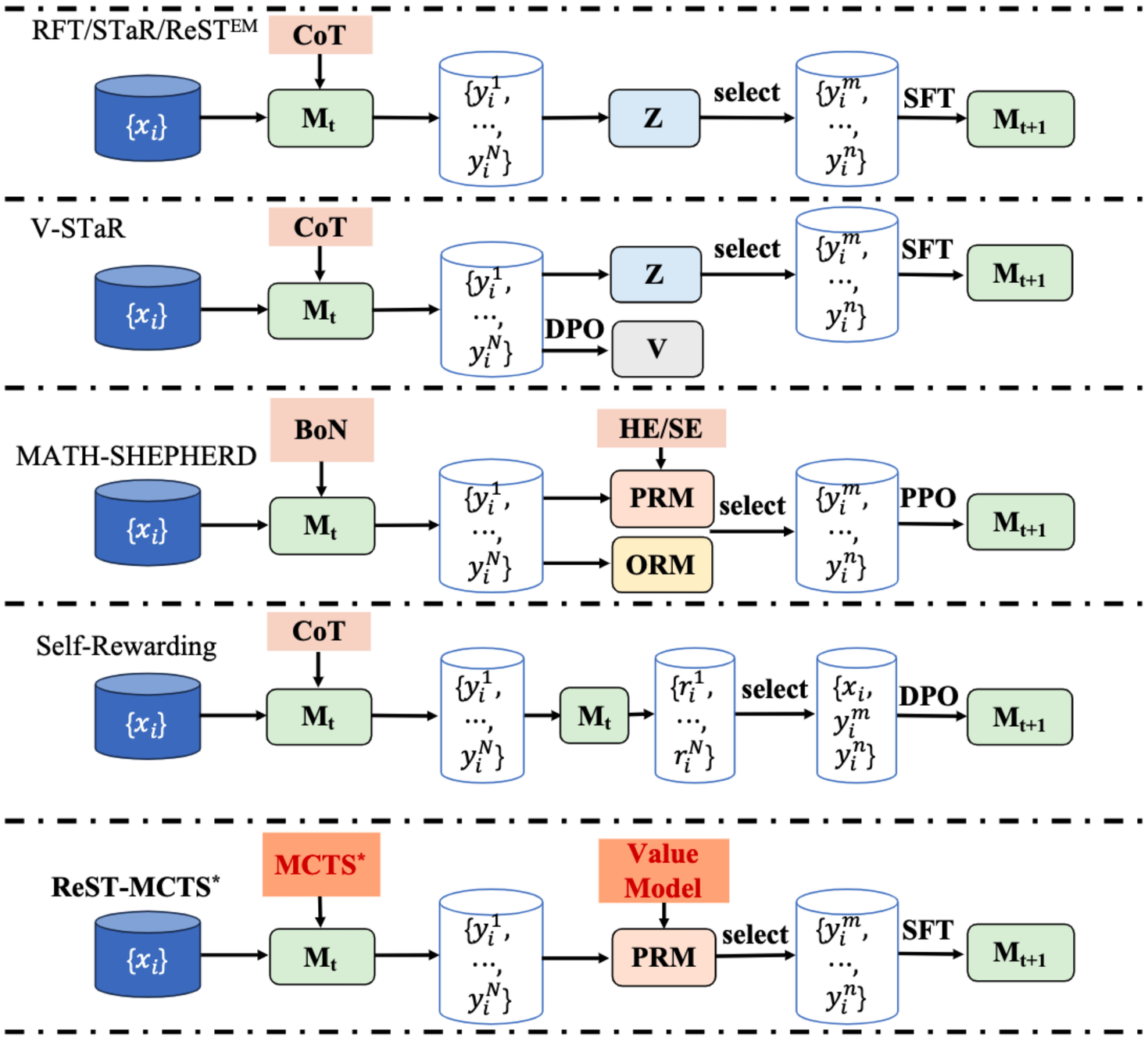


Figure 6: Comparison between existing self-training methods with our proposed ReST-MCTS*.

推理增强

Some Idea

- 迭代自我训练的有效性有待探索

  调研迭代自我训练导致模型崩溃，探明原因，找到解决方法（数据层面、训练算法层面）

- 生成多样化的高质量推理数据是关键

  设计更加高效、有效的搜索策略，帮助模型生成优质的训练数据（A* ？ MCTS 有什么可以改进的地方？）

- 奖励模型的能力迁移

  一个强大的奖励模型可以准确地判断推理步骤的潜力，其内部具有对数学推理很强的“直觉”，可不可以想办法引出这种能力将其转变为强大的推理模型?

推理增强

相似概念辨析

- Self-training VS Self-correct

前者指利用模型自己生成的数据来迭代训练自己；后者指模型在多次尝试中不断改进自己前面犯的错，不断完善自己的回答。

- Self-correct 相关工作

Large Language Models Cannot Self-Correct Reasoning Yet (Google DeepMind 2024/3/14)

Recursive Introspection: Teaching Language Model Agents How to Self-Improve (CMU 2024/7/18)

When Can LLMs Actually Correct Their Own Mistakes? A Critical Survey of Self-Correction of LLMs (2024/6/1)

Training Language Models to Self-Correct via Reinforcement Learning (Google DeepMind 2024/9/20)