

# 通过强化 抽象思维增强 LLM 的推理能力

Silin Gao<sup>1,2</sup>, Antoine Bosselut<sup>2</sup>, Samy Bengio<sup>1</sup>, Emmanuel Abbe<sup>1,2</sup>  
<sup>1</sup>Apple <sup>2</sup>EPFL

## Abstract

最近的研究表明，大型语言模型（LLMs），特别是较小的模型，其推理通常缺乏稳健性。也就是说，当面临分布变化时，如数值或名义变量的变化，或插入干扰性条款时，它们的性能往往会下降。一种可能的应对策略是生成合成数据以进一步“实例化”推理问题的潜在变化。相比之下，我们的方法着重于“抽象化”推理问题。这不仅有助于抵消分布变化，还促进与符号工具的联系以导出解决方案。我们发现这种抽象过程通过强化学习（RL）比简单的监督微调更适合获取，后者常常难以生成可靠的抽象。我们的方法 AbstraL——通过使用 RL 对细粒度抽象数据，促进 LLMs 的抽象推理——显著减轻了在最近 GSM 扰动基准上性能下降的问题。

## 1 介绍

推理能力，即整合知识来得出动态结论而非直接依赖记忆信息的能力 [43]，是人工通用智能的一个重要特质 [45]。为此，最近开发的大型语言模型（LLMs）配备了令人印象深刻的推理能力，无论是在一般领域 [13, 38, 12] 还是在数学等专业领域 [29, 39]。

然而，大多数大型语言模型（LLMs），尤其是较小的模型<sup>1</sup>，在推理时仍然面临鲁棒性挑战，在分布外（OOD）泛化方面仍有相当大的改进空间。最近的研究 [20, 17, 22] 表明，即使是在简单的小学数学任务中，当面对扰动和分布偏移时，LLMs 的性能也会下降。特别是，LLMs 在实例化偏移上容易出现推理错误，例如数值或名义变量被更改为不同的值，即便 LLMs 能够正确回答原始问题。在更具挑战性的干扰偏移上，即问题中加入一个分散注意力的（与主题相关但无用的）条件时，LLMs 的性能下降更加显著。

为了提高推理的鲁棒性，一种可能的学习策略 [3] 是合成更多具有不同表面形式上下文但遵循相同抽象 *i.e.*、高层知识或以符号方式表示的推理模式的推理问题实例。在本文中，我们没有扩大训练实例（这也可能计算成本高），而是教导 LLM 直接构建每个推理问题 [10] 的底层抽象，从而学习一种对于分布转变不变的更普遍的推理方式。

如图 2 所示，我们提出了一个强化抽象学习框架 AbstraL，该框架首先教授大语言模型（LLMs）生成输入问题的抽象，然后将抽象与符号工具连接起来，以稳定地推导输出解，无论具体的输入上下文。抽象的学习依赖于我们从神谕大语言模型提炼出的粒度分解抽象推理（Granular）数据，该数据将符号推理集成在苏格拉底式问题分解 [30] 和链式思维（CoT）解释 [33] 之中。在监督微调（SFT）基础上，AbstraL 使用强化学习（RL）和一组新的无模型奖励来进一步提高生成抽象的可信度。

我们在两个评估数学推理鲁棒性的基准上测试了 AbstraL，分别是 GSM-Symbolic [20] 和 GSM-Plus [17]。各种种子 LLM 的实验结果一致表明，AbstraL 有效地提高了 LLM 的推理

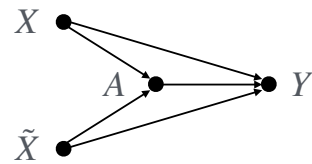


Figure 1: 两个经过改写的查询  $X$  和  $\tilde{X}$ ，具有相同的解决方案  $Y$ ，可以由一个通用的抽象  $A$  来处理。

<sup>1</sup>由我们在 §5 中的分析验证

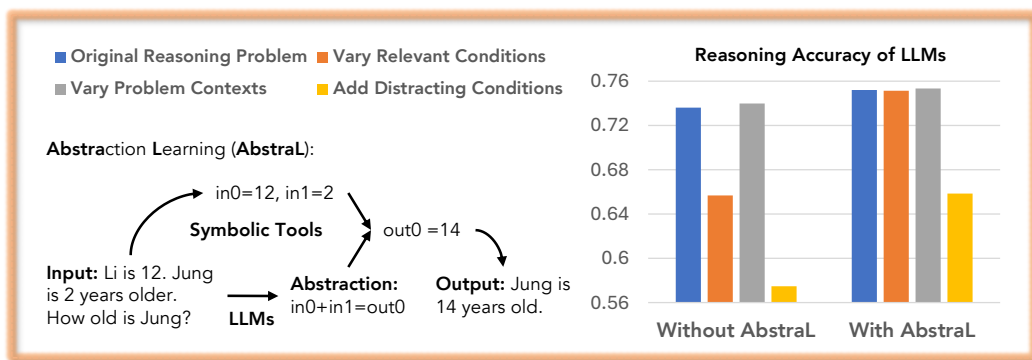


Figure 2: 我们的抽象学习 (AbstraL) 方法有效地提高了大型语言模型的推理鲁棒性, 尤其是在面对相关输入条件的变化和干扰条件的干扰时。我们展示了在不同的 GSM-Plus [17] 测试集上所有测试的大型语言模型的平均准确率, 包括原始 GSM8K 测试集 (原始推理问题)、具有数值变化的测试集 (改变输入条件, 均值为三个部分: 数字扩展、整数-小数-分数转换和数字替换)、具有问题改写的测试集 (改变问题上下文) 以及具有干扰项插入的测试集 (添加干扰条件)。

鲁棒性。如图 2 所示, AbstraL 几乎恢复了因相关输入条件变化而导致的 LLM 性能下降, 并且显著减轻了对扰动测试样本中添加的干扰条件的影响。

## 2 初步: 学习策略以提高推理的鲁棒性

我们假设每个推理数据样本, 由输入问题 (或查询)  $\mathcal{X}$  和输出答案 (或响应)  $\mathcal{Y}$  组成, 都是一个底层符号抽象  $\mathcal{A}$  的实例, 这代表了高层次的知识或推理架构。例如, 在图 3 中, 根据他比 Li 大多少岁来计算 Jung 年龄的问题, 是基于两个数字相加的抽象算术规则。一个稳健的推理者应该掌握该抽象  $\mathcal{A}$ , 因此能够稳定地对任何从  $\mathcal{A}$  隐含派生的问题  $\mathcal{X}$  给予真实的回答  $\mathcal{Y}$ , 而不是仅仅拟合到  $\mathcal{A}$  的一部分实例, 并且易于受到超出该子集的分布转变的影响。提高 LLMs 推理稳健性的常见策略是通过改写 [9, 46] 或模板 [3] 来合成更多的抽象  $\mathcal{A}$  的实例  $\{(\mathcal{X}', \mathcal{Y}'), (\mathcal{X}'', \mathcal{Y}''), \dots\}$  以增强学习数据。例如, 如图 3 (a) 所示, 一个实例中出现的数字和名字可以被其他值替换以创建一个新实例, 该实例具有不同的问题背景, 但遵循相同的抽象算术规则。经过之前研究 [3] 验证, 这种学习策略需要大量合成数据增强, 以有效提升 LLMs 对高层次抽象的掌握, 从而能够抵抗表面形式变化的干扰。

在这项工作中, 我们专注于策略 [14, 10], 教会 LLMs 直接学习每个实例  $(\mathcal{X}, \mathcal{Y})$  基础上的抽象  $\mathcal{A}$ , 并将  $\mathcal{A}$  与方程求解器等符号工具连接, 以稳步推导出问题  $\mathcal{X}$  的答案  $\mathcal{Y}$ , 如图 3 (b) 所示。即使不扩大训练数据, 直接学习抽象仍有效提高了 LLM 推理的稳定性, 这是由于对由抽象表示的更一般推理模式的建模。我们在 §3 中更详细地介绍了我们的方法。

## 3 AbstraL: 强化抽象学习框架

图 3 (b) 展示了我们鼓励 LLM 进行抽象思考的 AbstraL 框架, 由四个步骤组成。

**条件识别** 首先, 我们解析输入问题  $\mathcal{X}$  以识别可以用来回答问题的条件  $\mathcal{C}$ , 并用抽象符号表示的形式  $\mathcal{C}$  表示, 这些符号代表抽象中的输入变量  $\mathcal{A}$ 。例如, 在图 3 (b) 中显示的数学推理问题中, 题中给出的数字 12 和 2 用于推导答案, 并分别分配给符号  $\text{in0}$  和  $\text{in1}$ 。接着构建抽象输入问题, 记作  $\mathcal{X}^{\mathcal{A}}$ , 通过将  $\mathcal{X}$  中的具体文本 (或值) 替换为分配给它们的抽象符号  $\mathcal{C}$ , 并用方括号括起来。该步骤可以通过符号工具 *e.g.* 来执行, 该工具是一个通过正则表达式匹配来搜索数值的脚本, 或是通过提示大型语言模型而设计的任务特定解析器<sup>2</sup>。

我们然后使用构建的 Granular 数据 (§3.1) 基于 SFT 和 RL (§3.2) 教授 LLMs 抽象推理。特别地, 要求 LLMs 为抽象问题  $\mathcal{X}^{\mathcal{A}}$  生成抽象答案  $\mathcal{Y}^{\mathcal{A}}$ 。在  $\mathcal{Y}^{\mathcal{A}}$  中, LLMs 通过其分配的抽象符号在  $\mathcal{X}^{\mathcal{A}}$  中引用输入变量, 并且用抽象符号表示导出的输出变量, 例如在图 3 (b) 中用双尖括号括起的数学推导结果  $\text{out0}$ 。

<sup>2</sup>我们介绍了在 §4 中, 我们提出的框架的具体实验实现。

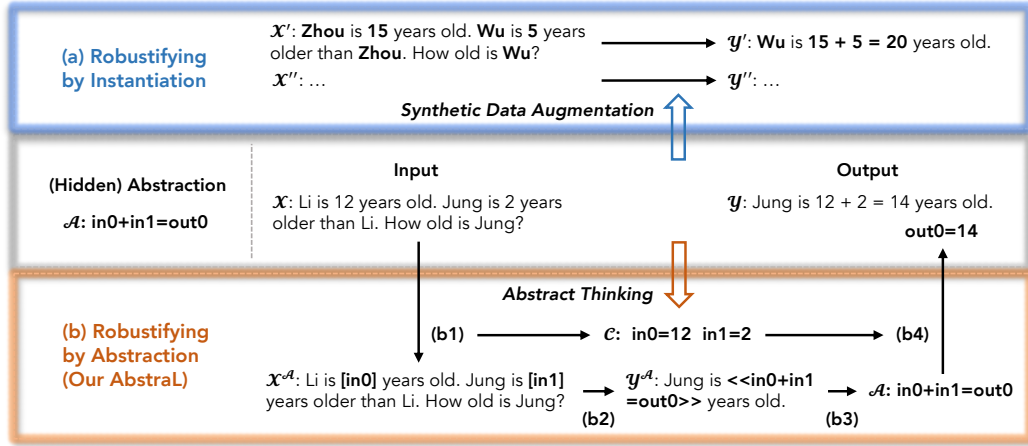


Figure 3: 学习策略以提高推理稳健性，针对分布变化进行改善。(a) 通过合成更多的推理实例来增加学习数据的量。(b) 直接学习基于输入构建基础抽象，包括：(b1) 条件识别，(b2) 抽象推理，(b3) 抽象检索和 (b4) 符号推导。

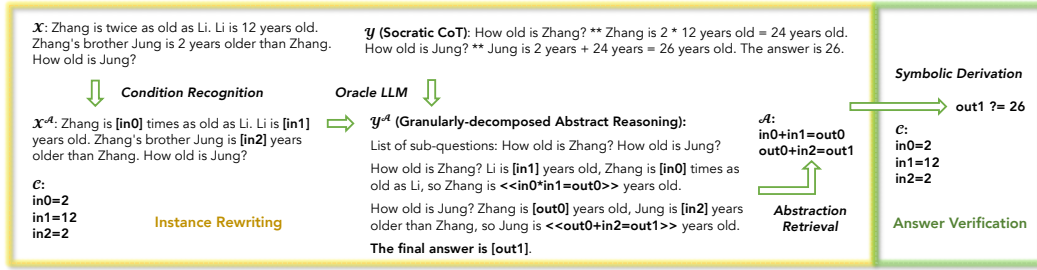


Figure 4: GranularAR 训练数据构建概览，其包含一个实例重写过程，用于将现有的 socratic CoT 数据 ( $\mathcal{X}, \mathcal{Y}$ ) 重写为细粒度抽象推理数据 ( $\mathcal{X}^A, C, \mathcal{Y}^A, \mathcal{A}$ )，接着是一个答案验证过程，用于检查重写的正确性。

基于抽象答案  $\mathcal{Y}^A$ ，我们然后检索从实例 ( $\mathcal{X}, \mathcal{Y}$ ) 中去除语境后的抽象  $\mathcal{A}$ 。与条件识别类似， $\mathcal{A}$  的检索也可以用正则表达式匹配工具或带有特定提示的 LLM 来完成。

**符号推导** 我们最终使用抽象  $\mathcal{A}$ ，结合输入条件  $C$ ，来推导问题的最终答案。这种推导可以通过基于规则的符号解析器 *e.g.*、用于算术运算的方程求解器、或者神经 (LLM) 符号推理器来完成。

我们的框架将从  $\mathcal{X}$  推断  $\mathcal{A}$  的任务分解为一个细粒度的流程  $\mathcal{X} \rightarrow \mathcal{X}^A \rightarrow \mathcal{Y}^A \rightarrow \mathcal{A}$ ，从而促进这一任务的学习。然而，从  $\mathcal{X}^A$  推断  $\mathcal{Y}^A$  的核心抽象推理步骤仍然是不简单的，这需要细粒度的抽象推理数据和适当的强化学习以实现稳健的推断，如下所述。

### 3.1 颗粒化分解的抽象推理 (GranularAR) 数据

**动机** 大型语言模型 (LLMs) 在预训练 [41, 5] 或后训练 [15] 阶段学习了各种细粒度的推理策略，如链式思维 (CoT) [33] 和苏格拉底问题分解 [30] 即为代表。在我们的 GranularAR 训练数据中，我们将抽象推理与这些预先学习的策略整合在一起，使得 LLMs 能够在细粒度推理链中逐步构建抽象，如图 4  $\mathcal{Y}^A$  所示。这种数据格式接近于预训练或后训练的数据分布，因此降低了 LLMs 适应我们新的抽象推理方式的难度。

**GranularAR 格式** 答案  $\mathcal{Y}^A$  首先将问题  $\mathcal{X}^A$  分解为一系列的子问题，从而实现逐步推理的整体规划。在此基础上， $\mathcal{Y}^A$  使用 CoT 和抽象符号回答每个子问题，其中首先引用相关的输入条件（或之前子问题的答案），然后使用引用的符号推导出答案。最后， $\mathcal{Y}^A$  得出结论，以明确哪个输出的抽象符号代表最终的答案。

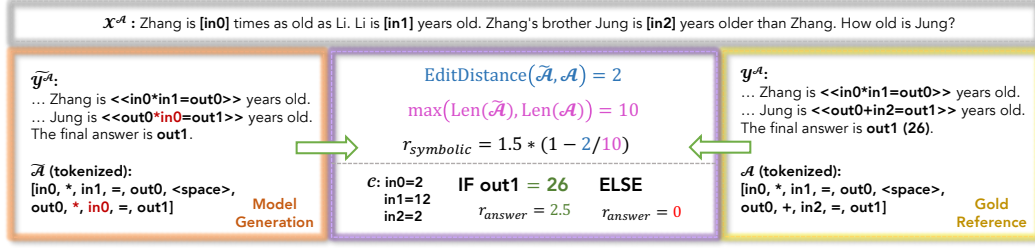


Figure 5: 在我们的强化学习方法中，抽象奖励的说明，包括符号距离奖励  $r_{symbolic}$  和答案正确性奖励  $r_{answer}$ 。

**数据构建** 图 4 说明了我们如何构建 Granular 训练数据。我们首先进行 §3 中描述的条件识别，以从问题  $\mathcal{X}$  中制定条件  $C$  并创建抽象问题  $\mathcal{X}^A$ 。基于此，我们提示一个 oracle LLM 将黄金苏格拉底链 (CoT) 答案  $\mathcal{Y}$  重写成我们所需的抽象答案  $\mathcal{Y}^A$ 。抽象问题  $\mathcal{X}^A$  也被输入 oracle LLM，以补充问题背景并澄清输入变量的抽象符号。给定提炼的抽象答案  $\mathcal{Y}^A$ ，我们然后进行抽象检索以获得去上下文化的抽象  $\mathcal{A}$ 。最后，进行符号推导步骤，以验证  $\mathcal{A}$  与  $C$  是否能得出  $\mathcal{Y}$  中陈述的正确最终答案。我们只保留通过答案验证的重写实例。

### 3.2 抽象推理的学习

**动机** 先前的研究 [10] 表明，大型语言模型 (LLMs) 不擅长通过上下文演示来以抽象的方式进行推理，这表明抽象推理的学习需要通过适当的监督进行训练，而不仅仅依赖于上下文指令和示例。一个直接的方法是通过监督微调 (SFT) 对 LLMs 进行训练。然而，虽然对抽象数据的 SFT 可以教会 LLMs 体面的抽象推理格式，但其自回归训练目标也迫使 LLMs 学习每个训练样本的具体上下文。这妨碍了 LLMs 学习更普遍的抽象思维策略，从而导致在测试时因测试数据中的新上下文偏离了问题而经常失败地生成一致的抽象，正如我们的结果在 §5 中所示。因此，我们建议在 SFT 的基础上进行强化学习 (RL)，以增强 LLMs 构建真实抽象的能力。

**监督微调** 我们通过基于输入问题  $\mathcal{X}^A$  自回归生成我们构建的 Granular 答案  $\mathcal{Y}^A$ ，简单地使用因果语言建模损失来基于前面的标记预测  $\mathcal{Y}^A$  中的每个标记来微调 LLMs。

**抽象奖励的强化学习** 在 SFT 的基础上，我们开发了一种 RL 方法，以进一步提高由 LLMs 生成的抽象的忠实性。我们的 RL 方法提出了一组新的奖励，以从两个方面紧密评价所生成的抽象。首先，给定模型从其生成的答案  $\mathcal{Y}^A$  中创造的抽象  $\tilde{A}$ ，我们进行符号推导以检查在标准参考中给定的条件  $C$  下， $\tilde{A}$  是否可以推出正确的最终答案（表示为  $Ans$ ）。如果是，则对模型给予一个正奖励  $r_{correct}$ （超参数），否则给予零奖励。我们将这个答案正确性奖励记为  $r_{answer}(\tilde{A}, C, Ans)$ 。其次，我们更细致地测量  $\tilde{A}$  与期望的抽象  $A$  的对齐程度，后者从标准答案  $\mathcal{Y}^A$  中提取。具体来说，我们将  $\tilde{A}$  和  $A$  拆分为符号标记列表，其中每个标记要么是一个抽象符号（在条件  $C$  或抽象答案  $\mathcal{Y}^A$  或  $\mathcal{Y}^A$  中创建）用于表示输入或输出变量（如  $in0$ ），要么是连接变量的预定义操作符（如  $=$ ）或者是分隔推导的符号（如  $<space>$ ），如图 5 所示。在此基础上，我们计算一个符号距离奖励：其中， $r_{max}$  表示最大奖励超参数， $EditDistance(\tilde{A}, A)$  表示标记化抽象之间的列表编辑距离 [16]， $\tilde{A}$  和  $A$ ， $Len(\cdot)$  用于计算列表的长度，以将编辑距离规范化到 0 到 1 的范围内。<sup>3</sup> 如果  $\tilde{A}$  更接近（或更相似）于  $A$ ，则模型会获得更高的符号距离奖励，这为模型提供了更细粒度的学习信号，以判断它与创建正确抽象之间的差距。图 5 展示了我们的抽象奖励的一个示例案例。请注意，我们提出的奖励不需要预训练的奖励模型，仅需与黄金标准进行比较。<sup>4</sup> 我们将我们的抽象奖励与 GRPO [29] 相结合，这是一个用于激励语言模型推理能力的高级强化学习算法。附录 ?? 描述了结合我们提出的奖励后 GRPO 的公式。

<sup>3</sup>最大可能的编辑距离是最长列表的长度。

<sup>4</sup>这减少了计算成本，并避免了由奖励模型引入的潜在偏好偏差 [29]。



## 4 实验设置

**AbstraL 框架的实现** 我们的实验研究将 AbstraL 应用于数学推理任务，这是一个具有代表性的推理鲁棒性研究领域 [20, 17, 22]。

特别是，我们创建了一些少样本示例来提示 Llama-3.3-70B-Instruct [12] 模型完成第一个条件识别步骤。如图 4 所示，该 LLM 的任务是将输入问题  $\mathcal{X}$  中的数值用方括号 “[ ]” 标记，然后按顺序将每个标记的值替换为抽象符号，索引为 “in0”、“in1” 等，以创建抽象问题  $\mathcal{X}^A$ 。注意，对于诸如 “一百” 和 “两倍” 等隐含数值，我们还提示 LLM 将它们转换为明确格式 “100” 和 “2 次”，以便对数字进行标记和替换。同时，要求 LLM 使用方程记录这些替换，e.g.，“in0 = 2”，以创建条件  $\mathcal{C}$ 。

然后，我们在构建的 Granular 数据 (§3.1) 上根据我们的 SFT 和 RL 方案 (§3.2) 调整各种 LLM，以执行核心抽象推理步骤。我们测试的 LLM 包括 Llama3 模型系列 [12] (Llama-3.2-1B-Instruct、Llama-3.2-3B-Instruct 和 Llama-3.1-8B-Instruct)、Qwen2.5 模型系列 [38] (Qwen2.5-0.5B-Instruct、Qwen2.5-1.5B-Instruct、Qwen2.5-3B-Instruct、Qwen2.5-7B-Instruct 及其数学专用版本 Qwen2.5-Math-7B-Instruct) 和 Mathstral-7B<sup>5</sup>。所有 LLM 都应该遵循我们的 Granular 训练数据在其生成中索引派生的输出变量，使用 “out0”、“out1” 等形式，并在每个抽象数学推导中用双角括号突出显示，比如 “<< in0 \* in1 = out0 >>”。LLM 还应生成一个固定语句 “The final answer is [ outN ].”，以明确表示代表最终答案的输出变量。在推理阶段，LLM 使用贪婪解码生成其抽象答案。

在抽象检索步骤中，我们简单地使用一个正则表达式匹配脚本来提取封闭在双尖括号内的所有数学推导，这些推导位于模型生成的答案  $\mathcal{Y}^A$ （或在我们的 Granular 数据构建中的  $\mathcal{Y}^A$ ）中。所有提取的数学推导构成问题抽象  $\tilde{\mathcal{A}}$ （或  $\mathcal{A}$ ）。在对抽象进行标记化（用于计算我们 RL 方法中的符号距离奖励）时，推导的标记化列表与一个特殊标记 “<space>” 连接，如图 5 所示。为了执行最终的符号推导，我们将  $\tilde{\mathcal{A}}$ （或  $\mathcal{A}$ ）中的输出数学推导和  $\mathcal{C}$  中的输入条件共同作为一个方程组，输入到 SymPy<sup>6</sup> 方程求解器中，以推导出最终答案。

对于 Granular 训练数据的构建，我们使用 Llama-3.3-70B-Instruct 作为 oracle 大型语言模型，并通过少量示例提示它重写现有的苏格拉底链式推理数据。我们使用来自 OpenAI 的 GSM8K [8] 训练集的苏格拉底版本作为重写的种子数据，其中包含 7473 小学数学文字题。在重写和筛选后，最终保留了 6386 个问题用于训练。

**评估数据集** 我们在两个从 GSM8K 测试样本中提取的数据集上评估我们的方法，它们是 GSM-Symbolic [20] 和 GSM-Plus [17]。

**GSM-Symbolic** 手动从 100 个 GSM8K 测试样本中构建问题模板，并使用这些模板创建新问题，其中原问题中的数字、名称或两者都被更改为不同的值，分别记为 Vary Num.、Vary Name 和 Vary Both。我们按照 GSM-Symbolic 进行 50 轮评估，每轮从每个模板创建 1 个问题（因此每轮对 100 个新问题进行测试），并衡量这 50 轮的平均表现，i.e.，即精度的平均值（带有标准偏差），并检查其是否与原始 100（Origin 100）GSM8K 问题的表现相匹配。

**GSM-Plus** 创建了全 GSM8K 测试集的不同变体，其中每个变体测试集都包含所有 1319 个 GSM8K 测试问题。对于每种类型的变体，每个原始问题创建 1 个变体样本，从而使每种变体进行一次评估。我们在 GSM-Plus 变体<sup>7</sup>的子集中测试我们的方法，包括数字扩展（Digit Ex.），即在数字中添加更多位数（e.g.，从 16 到 1600），整数-小数-分数转换（Int-Dec-Fra），即改变数字的类型（e.g.，从 2 到 2.5），数字替换（Num. Sub.），即用另一个相同位数的数字替换（e.g.，从 16 到 20），重述问题以检查问题理解（Rephrase），以及插入与主题相关但无用的条件（Distract），并将其与模型在原始 GSM8K 测试集上的性能进行比较。

**基线方法** 我们将 AbstraL 与几种基线推理方法进行比较。

**CoT-8S** 使用 GSM-Symbolic [20] 提议的演示模板和用于 GSM8K 评估的常用 8-shot 样例对 LLMs 进行提示，以生成输入问题  $\mathcal{X}$  的 CoT 答案。最后一个数字（通常在 “最终答案是” 之后）被提取为最终答案。

<sup>5</sup><https://huggingface.co/mistralai/Mathstral-7B-v0.1>

<sup>6</sup><https://github.com/sympy/sympy>

<sup>7</sup>一些修改底层抽象的 GSM-Plus 和 GSM-Symbolic 变体，例如添加有用条件以使问题复杂化，被排除在外并超出了我们研究的范围。

Table 1: GSM-Symbolic 的评价结果。△ 表示与 Origin 100 的性能相比，在 Vary Both 上的相对降幅。每个模型的最佳结果以粗体显示，其中 △ 越低越好。多轮评价结果的标准差 (std) 在括号内显示，其中每个模型的最低 std 为 underlined。

Model	Method	Vary Num.	Vary Name	Vary Both	Origin 100	△ ( % )
Llama-3.2-1B-Instruct	CoT-8S	0.3864 (0.030)	0.4250 (0.026)	0.3890 (0.038)	0.4800	18.96
	CoT-RL	0.4788 (0.031)	0.5342 (0.024)	0.4488 (0.038)	0.5700	21.26
	CoA	0.4534 (0.020)	0.4372 (0.026)	0.4242 (0.030)	0.4600	7.78
	AbstraL	0.5912 ( <u>0.016</u> )	0.5838 ( <u>0.023</u> )	0.5804 ( <u>0.027</u> )	0.6000	3.27
Llama-3.1-8B-Instruct	CoT-8S	0.8440 (0.026)	0.8746 (0.021)	0.8236 (0.032)	0.8700	5.33
	CoT-RL	0.7784 (0.029)	0.8710 ( <u>0.014</u> )	0.7540 (0.026)	0.8300	9.16
	CoA	0.7240 (0.019)	0.7086 (0.019)	0.6944 (0.025)	0.7200	3.56
	AbstraL	0.8686 ( <u>0.013</u> )	0.8672 (0.018)	0.8620 ( <u>0.023</u> )	0.8700	0.92
Qwen2.5-0.5B-Instruct	CoT-8S	0.3394 (0.039)	0.3724 ( <u>0.024</u> )	0.3398 (0.033)	0.3800	10.58
	CoT-RL	0.3192 (0.025)	0.3948 (0.032)	0.3228 (0.032)	0.3500	7.77
	CoA	0.2866 (0.025)	0.3060 (0.026)	0.2872 (0.026)	0.2900	0.97
	AbstraL	0.4396 ( <u>0.015</u> )	0.4416 (0.026)	0.4456 ( <u>0.025</u> )	0.4400	-1.27
Qwen2.5-Math-7B-Instruct	CoT-8S	0.8956 (0.021)	0.9108 (0.018)	0.8766 (0.023)	0.9500	7.73
	CoT-RL	0.8942 (0.022)	0.9154 ( <u>0.012</u> )	0.8812 (0.021)	0.9600	8.21
	CoA	0.7122 (0.028)	0.6976 (0.031)	0.6970 (0.033)	0.7100	1.83
	AbstraL	0.9066 ( <u>0.015</u> )	0.9014 (0.013)	0.9022 ( <u>0.016</u> )	0.9100	0.86

CoT-RL 在非重写的 GSM8K 训练数据 ( $\mathcal{X}, \mathcal{Y}$ ) 上调整 LLMs (socratic CoT 版本<sup>8</sup>，包含 7473 个样本)，使用与我们方法相同的 SFT 和 RL 算法 (GRPO [29])。由于 CoT-RL 不在推理链中生成抽象，因此在 RL 中仅使用我们的答案正确性奖励  $r_{answer}$  (§3.2) 作为学习信号，以检查从生成的 CoT 中提取的最终答案 (最后一个数字) 是否与标准答案匹配。

CoA [10] 是另一种抽象推理方法，它同样对大型语言模型进行微调，以生成具有抽象符号的推理链，并与 SymPy 方程求解器结合以推导最终答案。然而，与 AbstraL 不同的是，CoA 推理链中只有输出的数字由抽象符号表示，并没有抽象化输入条件 ( $\mathcal{X}^A$  和  $\mathcal{C}$ )，以及使用我们细粒度分解的推理链 (Granular)，且 CoA 的学习仅基于 SFT，没有结合适当的 RL。与我们的 Granular 训练数据构建类似，CoA 提示一个通用的 LLM 将现有的 CoT 数据改写为抽象格式，其输出数字被替换为抽象符号，然后还使用方程求解器验证改写样本的正确性。为了公平比较，我们使用相同的通用 LLM (Llama-3.3-70B-Instruct) 和种子数据 (GSM8K 训练集的柏拉图式 CoT 版本) 来构建 CoA 训练数据。

## 5 实验结果

表 1 显示了我们在 GSM-Symbolic 数据集上部分有代表性的评估结果。我们报告了在我们的 Llama3 和 Qwen2.5 系列测试中最小和最强模型的性能，而其他模型的结果留在附录 ?? 中。我们发现 AbstraL 有效地提高了所有模型从 0.5B 到 8B 规模的推理鲁棒性，对于数值和名称的变化 (同变)，展示了更好的对分布变化的泛化能力。具体而言，与基线方法相比，具有 AbstraL 的模型在同变样本上获得了一致较高的 (平均) 准确率，在不同测试轮次中的标准差较低。此外，从 Origin 100 转移到同变时，具有 AbstraL 的模型性能下降较小 ( $\Delta$ )，特别是相比于不使用任何抽象推理的 CoT 方法。值得注意的是，在大型 Llama (8B) 和 Qwen (7B) 模型上，尽管 AbstraL 在 Origin 100 和 Vary Name 上的评分低于 8-shot 提示 (CoT-8S)，它在 Vary Num. 和同变扰动样本上超过了提示方法。这表明，使用 AbstraL 学习可能会减轻 LLMs 在预训练阶段由于潜在的数据污染导致对现有输入条件 (或数字) 的过拟合。[19, 35] 相比之下，CoT-RL (无抽象推理) 和 CoA (在我们的 Granular 数据上未抽象输入条件和 RL) 通常未能改善测试的模型，这表明我们基于 RL 的细粒度抽象推理框架在使 LLMs 更稳定地适应鲁棒推理方面更有效。

表格 2 展示了我们在 GSM-Plus 数据集上的代表性评估结果 (完整结果见附录 ??)，该数据集对整个 GSM8K 测试集进行了更丰富的扰动测试。类似于在 GSM-Symbolic (变化数字和变化名称) 上的结果，AbstraL 几乎完全恢复了由于输入数字变化 (数字例、整数-小数-分数和数字替代) 所导致的性能下降，并且在上下文变化 (改述) 方面保持了与基线方法相当的鲁棒性。更有意思的是，AbstraL 显著减轻了问题中添加干扰条件 (干扰) 的影响，而相比之下，采用基线推理方法的模型从原始转移到干扰时的得分急剧下降。AbstraL 在干扰方

<sup>8</sup>我们还进行了一项试点研究，使用非苏格拉底连锁思维数据来训练 LLMs，取得了类似的结果。

Table 2: GSM-Plus 的评估结果。每个模型的最佳结果用粗体显示。

Model	Method	Digit Ex.	Int-Dec-Fra	Num. Sub.	Rephrase	Distract	Original
Qwen2.5-0.5B-Instruct	CoT-8S	0.3601	0.2866	0.3958	0.4359	0.2267	0.4238
	CoT-RL	0.3336	0.2373	0.3571	0.4079	0.1524	0.3798
	CoA	0.2745	0.2267	0.2782	0.3161	0.1266	0.3033
	AbstraL	0.4670	0.4663	0.4670	0.4625	0.3654	0.4670
Qwen2.5-Math-7B-Instruct	CoT-8S	0.8552	0.8218	0.8453	0.9052	0.7627	0.9181
	CoT-RL	0.8757	0.8522	0.8506	0.9037	0.8150	0.9340
	CoA	0.7460	0.6907	0.7180	0.7642	0.5709	0.7809
	AbstraL	0.8916	0.8886	0.8916	0.8992	0.8226	0.8916

面的提升主要归因于其对细粒度分解的抽象推理（GranularAR）数据的学习（在我们的以下消融研究中得到验证），这使得能够在生成开始时整体规划推理步骤，并在每个推理步骤中重新考虑有用的输入条件。

**消融研究** 我们方法的一个自然担忧是 AbstraL 的改进是由于抽象推理的学习，还是仅仅因为集成了强大的符号工具，例如用于条件识别的 *i.e.*、被提示的 Llama-3.3-70B-Instruct、用于抽象检索的正则匹配脚本以及用于符号推导的 SymPy 方程求解器。为了解释这一点，我们进行了一项消融研究（w/o Tools），在我们的框架中用 LLM 本身替换了上述所有符号工具，以每个步骤特定的提示来完成我们的流程的每个步骤。表中 3 的结果表明，没有额外工具的 LLMs 仅遭受轻微的性能下降，这表明在我们的框架中，抽象推理的学习是改进的主要贡献者。

我们还通过测试一个削减框架（w/o Contexts）来研究使用 AbstraL 的流水线在上下文（*i.e.*、 $\mathcal{X} \rightarrow \mathcal{X}^A \rightarrow \mathcal{Y}^A \rightarrow \mathcal{A}$ ）中构建抽象的效果。在该框架中，LLMs 通过 SFT 和相同的 RL 方法被训练为直接根据抽象输入问题  $\mathcal{X}^A$ 、*i.e.*、 $\mathcal{X} \rightarrow \mathcal{X}^A \rightarrow \mathcal{A}$  生成抽象  $\mathcal{A}$ ，而不是显式生成  $\mathcal{Y}^A$  并通过抽象检索后处理它。正如表格 3 所示，学习推理而没有中间上下文的 LLMs 远远落后于未削减的模型。这表明缩小符号抽象和自然语言建模之间的差距对于忠实的抽象思维很重要，这与推理的全局性程度 [1] 观点一致。

我们的第三个消融研究考察了 RL 在 AbstraL 中的贡献，通过完全去除 RL（w/o RL）或仅去除符号距离奖励（w/o  $r_{symbolic}$ ）。在原始和扰动的测试集上，我们发现没有 RL 的 LLMs 得分远低于经过强化学习的 LLMs。在缺少  $r_{symbolic}$ （提示生成的抽象与正确抽象的接近程度）的情况下，LLMs 的性能也显著下降，从而表现不如提示基线（CoT-8S）。这些发现表明，要获得忠实的抽象推理需要通过适当的 RL 方法和里程碑式的奖励来仔细学习，密切监控学习的进展，*e.g.*。

最后，我们对细粒度分解的抽象推理（w/o GranularAR）进行了消融，该推理作为 AbstraL 的训练数据格式。我们将构建的训练数据中的标准抽象答案  $\mathcal{Y}^A$  改回到标准的苏格拉底链式推理格式（如图 4 所示），在这种格式中，我们在推理链开头移除子问题的列表，并在每个推理步骤中去掉输入条件的引用，只保留构成抽象  $\mathcal{A}$  的抽象推导。训练于此消融数据上的大型语言模型在 Distract 测试集上的性能严重下降，表明它们容易受到无干扰项的影响。这验证了我们采用的细粒度推理格式在识别干扰条件方面起到了至关重要的作用，因为它规划了推理步骤和每一步中有用的输入条件。

## 6 相关工作

**推理稳健性** 最近在大型语言模型（LLM）推理方面的进展也发现了相当大的稳健性挑战 [44]，这些挑战在符号推理基准中得以体现，比如逻辑 [2] 和数学 [32, 17, 31, 20] 推理，以及事实性或常识性推理 [40, 22, 23]。上述所有基准一致揭示，大型语言模型容易受到测试时数据分布变化的干扰。受通过数据增强提高稳健性的启发 [26]，之前的工作 [9, 25, 46, 3] 利用各种数据合成技术来扩大训练样本的覆盖范围，从而预见潜在的数据分布变化，这自然增加了开发大型语言模型的计算成本。在这项工作中，我们旨在通过激励模型进行抽象思考来提高大型语言模型的推理稳健性，而不是依赖于大量的推理实例。

**抽象思维和计划** 抽象思维是一般流体智力 [6] 的基本组成部分，也是人类认知和推理 [27] 的关键。它需要基于抽象的基本规则或概念 [4] 进行推断，而不仅仅是记忆概率模式匹配 [11, 36]。最近，已经提出了各种推理（或数据）格式用于让大型语言模型（LLM）学习抽象思维，如思维抽象（AoT）[14]、抽象链（CoA）[10] 等。然而，以上所有方法在简单的监督微调（SFT）之外缺乏一个适当的学习机制，以稳定地使大型语言模型适应抽象思维。另



Table 3: GSM-Symbolic 和 GSM-Plus 的消融研究结果。Num. Pert. 表示在对输入数字进行扰动的三个 GSM-Plus 测试集 (Digit Ex., Int-Dec-Fra 和 Num. Sub.) 上的平均值。

Model	Method	GSM-Symbolic			GSM-Plus			
		Vary Both	Origin 100	$\Delta$ ( % )	Num. Pert.	Rephrase	Distract	Original
Qwen2.5-0.5B-Instruct	AbstraL	0.4456	0.4400	-1.27	0.4668	0.4625	0.3654	0.4670
	- w/o Tools	0.4362	0.4400	0.86	0.4599	0.4564	0.3556	0.4610
	- w/o Contexts	0.2306	0.3000	23.13	0.2702	0.2881	0.1653	0.2866
	- w/o RL	0.2956	0.3600	17.89	0.3803	0.3230	0.2828	0.3571
	- w/o $r_{symbolic}$	0.3760	0.3900	3.59	0.4167	0.3995	0.3131	0.3882
	- w/o GranulAR	0.4262	0.4200	-1.48	0.4236	0.4230	0.2358	0.4238
Qwen2.5-Math-7B-Instruct	AbstraL	0.9022	0.9100	0.86	0.8906	0.8992	0.8226	0.8916
	- w/o Tools	0.8996	0.9100	1.14	0.8860	0.9037	0.8180	0.8893
	- w/o Contexts	0.6926	0.7100	2.45	0.6715	0.6936	0.5513	0.7065
	- w/o RL	0.7854	0.8200	4.22	0.8211	0.7998	0.7445	0.8226
	- w/o $r_{symbolic}$	0.8322	0.8700	4.34	0.8577	0.8234	0.7809	0.8590
	- w/o GranulAR	0.8814	0.8900	0.97	0.8835	0.8976	0.6679	0.8855

一方面，规划也是一种基本的人类推理技能 [34]，这有利于大型语言模型的推理。典型的规划方法，如思维链 (CoT) [33]、思维树 (ToT) [42] 和苏格拉底问题分解 [30]，被广泛应用于大型语言模型以提高推理。我们的工作开发了一种更好的抽象思维学习机制（基于 RL），并将其整合到规划的力量中。

**强化学习** RL 是在最近 LLMs [29, 13] 的发展中用于增强推理能力的一种流行技术。一种有代表性的 RL 方法是 PPO [28]，通常用于通过基于人类偏好标注预训练的奖励模型从人类反馈中学习 (RLHF) [7, 21]。更进一步，DPO [24] 通过优化策略模型的直接偏好简化了 PPO 的实施，摆脱了对额外奖励模型的预训练。我们的方法采用了具有无模型奖励的 GRPO [29]（不依赖偏好反馈），通过使用群体相对优势，进一步去除了利用优势估计的价值模型的训练。

## 7 结论

本文提出了一种方法，AbstraL，以促进大型语言模型 (LLM) 的抽象思维。AbstraL 的设计旨在提高 LLM 的推理稳健性，其基于这样一个自然原则，即抽象思维会导致推理步骤对表面形式变化的更高不变性。我们的抽象机制通过一个适当的强化学习 (RL) 框架实现，其中无模型奖励来自新设计的推理依据 GranulAR，这些依据将苏格拉底的连贯思维与增强的粒度结合在一起。这既能使问题去上下文文化，也能实现符号工具的整合。我们在最新的 GSM 扰动基准上评估 AbstraL，并表明它有效地缓解了由实例化和干扰转移引起的性能下降。

找到一个理论环境来研究抽象对学习的样本/时间复杂度、OOD 鲁棒性以及模型规模要求的影响将是很有趣的。例如，形式化三元组  $(A, X, Y)$  的数据分布，其中  $X \rightarrow Y$  代表一个推理问题  $X$  的目标，该问题具有解  $Y$ ，而  $A$  是去除上下文/非形式化（具有相同解  $Y$ ）的  $X$  的抽象。需要正确定义抽象的概念，区分可能的去上下文文化与推理解决（例如， $A$  不应该是  $Y$ ）。是否存在一个适当的框架（包括一个学习模型类别）可以使以下类型的陈述具有严谨性：(i) 学习  $A \rightarrow Y$  的样本复杂度低于  $X \rightarrow Y$ （正式且适当地定义抽象和实例化过程的结果）？(ii) 通过依赖于适当的抽象  $A$  学习  $X \rightarrow Y$  能够增强对适当建模的实例化转移的鲁棒性？(iii) 通过抽象学习可以用较小的模型尺寸实现？

我们承认我们的工作中存在一些局限性。首先，用于测试我们方法的数据集不可能涵盖所有真实世界的推理场景。我们选择了最具代表性的领域，即小学数学，这是研究推理鲁棒性时一个常见且典型的领域，并且在我们的测试中主要使用英语。在未来的工作中，我们的方法的测试平台可以扩展到更多的领域，例如高中竞赛数学和常识（或事实）推理，并扩展到更多的语言。其次，我们的推理鲁棒性测试仅限于基于实例化和推论转移，这基于一个假设，即测试的扰动不会修改基础抽象，即抽象的数学推导。未来的工作可以将我们的研究扩展到对抽象的扰动，例如将“甲是  $M$  岁，乙比甲大  $N$  岁，乙多大？”( $M+N$ ) 改为“甲是  $M$  岁，乙比甲小  $N$  岁，乙多大？”( $M-N$ )，以测试对类似推理策略的泛化鲁棒性。此外，我们的方法是在调优完整的 LLMs 设置中进行测试的，这需要相当大的计算资源。更高效的模型训练方案，如 LoRA，可以在未来的工作中应用。最后，我们实验中的所有 LLMs 都使用贪婪解码来生成推理，这为未来的工作在更高级的解码策略上测试我们的方法留下了空间，例如自一致性解码。



我们的 AbstraL 框架提出采用 GRPO [29] 作为 RL 算法，在抽象推理任务上训练 LLMs（在监督微调的基础上），即基于输入的抽象问题  $\mathcal{X}^A$  生成抽象答案  $\mathcal{Y}^A$ 。

对于每个输入问题  $\mathcal{X}^A$ ，GRPO 从当前（旧策略）模型  $\pi_{\theta_{\text{old}}}$  中采样一组输出答案  $\{\mathcal{Y}_1^A, \mathcal{Y}_2^A, \dots, \mathcal{Y}_G^A\}$ ，并通过最大化以下目标来优化（策略）模型  $\pi_{\theta}$ ：

$$\frac{1}{G} \sum_{i=1}^G \left( \min \left( \frac{\pi_{\theta}(\mathcal{Y}_i^A | \mathcal{X}^A)}{\pi_{\theta_{\text{old}}}(\mathcal{Y}_i^A | \mathcal{X}^A)} R_i, \text{clip} \left( \frac{\pi_{\theta}(\mathcal{Y}_i^A | \mathcal{X}^A)}{\pi_{\theta_{\text{old}}}(\mathcal{Y}_i^A | \mathcal{X}^A)}, 1 - \varepsilon, 1 + \varepsilon \right) R_i \right) - \beta \mathbb{D}_{KL}(\pi_{\theta} || \pi_{\text{ref}}) \right) \quad (1)$$

其中  $\varepsilon$  和  $\beta$  是超参数。我们将参考策略  $\pi_{\text{ref}}$  设置为仅通过监督微调（SFT）训练的模型，用于计算 KL 散度正则化：

$$\mathbb{D}_{KL}(\pi_{\theta} || \pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(\mathcal{Y}_i^A | \mathcal{X}^A)}{\pi_{\theta}(\mathcal{Y}_i^A | \mathcal{X}^A)} - \log \frac{\pi_{\text{ref}}(\mathcal{Y}_i^A | \mathcal{X}^A)}{\pi_{\theta}(\mathcal{Y}_i^A | \mathcal{X}^A)} - 1 \quad (2)$$

$R_i$  是分配给从每个采样的输出答案  $\mathcal{Y}_i^A$  中检索到的抽象  $\tilde{\mathcal{A}}_i$  的组相对优势，它在我们提出的抽象奖励  $r_{\text{answer}}$  和  $r_{\text{symbolic}}$  上应用组归一化，这些奖励在 §3.2 中定义，分别参考输入条件  $\mathcal{C}$  和标准答案  $\text{Ans}$ ，以及从标准响应  $\mathcal{Y}^A$  中检索到的标准抽象  $\mathcal{A}$ ：

$$R_i = \frac{r_i - \text{mean}(\{r_1, r_2, \dots, r_G\})}{\text{std}(\{r_1, r_2, \dots, r_G\})}, \quad r_i = r_{\text{answer}}(\tilde{\mathcal{A}}_i, \mathcal{C}, \text{Ans}) + r_{\text{symbolic}}(\tilde{\mathcal{A}}_i, \mathcal{A}) \quad (3)$$

表格 4 和 5 分别展示了我们所有测试的 LLMs 在 GSM-Symbolic 和 GSM-Plus 数据集上的评估结果。所有 LLMs 的结果一致表明，AbstraL 在推广到实例化和推理转变时，能够有效提高推理的鲁棒性。对每个测试的 LLM，我们还在最佳和第二最佳结果之间进行了自举统计显著性测试 [37]，并在最佳结果显著优于其对应的第二最佳结果且显著性测试 p 值为  $< 0.05$  时进行突出显示（用 \*）。

## A 定性分析

表 6 和 7 展示了我们测试的最强大的长语言模型 Qwen2.5-Math-7B-Instruct 在 GSM-Symbolic 和 GSM-Plus 数据集上各自的两个数学推理示例，使用基线 CoT-8S 方法或我们的 AbstraL。面对相关条件的变化时，使用 AbstraL 的长语言模型执行的数学推导更为稳定（表 6），并且在处理插入的干扰条件时，实现了更具鲁棒性的推理链（表 7）。我们在相应的表格说明中包含了更详细的分析。

## B AbstraL 实施细节

本节包括我们如何实现 AbstraL 框架的更多细节。

对于 AbstraL 的第一个条件识别步骤，我们提示 Llama-3.3-70B-Instruct [12] 模型将输入问题  $\mathcal{X}$  中的数值替换为用方括号括起来的抽象符号，以创建抽象问题  $\mathcal{X}^A$ 。同时创建了一组条件  $\mathcal{C}$ ，用于记录替换情况，作为一系列方程的列表。表格 8 展示了用于任务演示的指令和少量样例。我们使用了 4 个 NVIDIA A100-SXM4 (80GB) GPUs 来运行基于 Llama-3.3-70B-Instruct 的条件识别步骤，处理所有训练和测试数据样本大约耗时 36 小时。

基于抽象问题  $\mathcal{X}^A$  和问题的黄金苏格拉底式 CoT 回答  $\mathcal{Y}$ ，我们还提示 Llama-3.3-70B-Instruct 将  $\mathcal{Y}$  重写为我们粒度分解的抽象推理（Granular）格式  $\mathcal{Y}^A$ ，该格式用于训练 LLMs 学习 AbstraL 中的抽象推理步骤。为了促进重写，我们采用了两步管道。首先，提示 Llama 模型用抽象符号替换  $\mathcal{Y}$  中的数值，方法是引用  $\mathcal{X}^A$  中的抽象符号作为输入值，或为推导出的输出值创建新的抽象符号。回复中的所有推导都应该用双尖括号 “<>>” 括起来。其次，基于第一步中重写的回复，提示 Llama 模型将回复（以苏格拉底式 CoT 格式）进一步重写为我们的 Granular 格式，同时保持回复中的所有抽象符号不变。表 9 和 10 展示了用于提示我们两步 Granular 训练数据构建的指令和少量示例。每个重写步骤中，使用了 4 张 NVIDIA A100-SXM4 (80GB) GPU 来运行 Llama 模型。大约需要 36 和 48 小时分别进行所有训练数据样本的第一步和第二步重写。

在每一步响应重写之后，我们通过验证推导的正确性来过滤 Llama 模型的输出，*i.e.*，我们使用正则表达式匹配提取出所有包含在 “<>>” 中的推导，并将它们（连同在条件识别步骤中生成的输入条件  $\mathcal{C}$ ）传入 SymPy<sup>9</sup> 求解器，以推导出最终答案（数值），检查是否与标准

<sup>9</sup><https://github.com/sympy/sympy>

Table 4: GSM-Symbolic 的完整评估结果。 $\Delta$  表示在 Vary Both 上的性能与 Origin 100 上的性能相比的下降百分比。每个模型的最佳结果加粗显示，其中  $\Delta$  值越低越好。多轮评估结果的标准偏差 (std) 在括号中显示，每个模型的最低 std 值为 underlined。具有 \* 的最佳（加粗）结果显著优于其相应的第二佳结果，显著性检验的 p 值为  $< 0.05$ 。

Model	Method	Vary Num.	Vary Name	Vary Both	Origin 100	$\Delta$ (%)
Llama-3.2-1B-Instruct	CoT-8S	0.3864 (0.030)	0.4250 (0.026)	0.3890 (0.038)	0.4800	18.96
	CoT-RL	0.4788 (0.031)	0.5342 (0.024)	0.4488 (0.038)	0.5700	21.26
	CoA	0.4534 (0.020)	0.4372 (0.026)	0.4242 (0.030)	0.4600	7.78
	AbstraL	0.5912 * ( <u>0.016</u> )	0.5838 * ( <u>0.023</u> )	0.5804 * ( <u>0.027</u> )	0.6000 *	3.27
Llama-3.2-3B-Instruct	CoT-8S	0.7218 (0.027)	0.7638 (0.027)	0.7114 (0.031)	0.8400 *	15.31
	CoT-RL	0.7056 (0.027)	0.7516 (0.023)	0.6898 (0.026)	0.8000	13.78
	CoA	0.6450 (0.019)	0.6802 (0.021)	0.6760 (0.026)	0.6800	0.59
	AbstraL	0.7960 * ( <u>0.014</u> )	0.7982 * ( <u>0.020</u> )	0.7946 * ( <u>0.023</u> )	0.8000	0.68
Llama-3.1-8B-Instruct	CoT-8S	0.8440 (0.026)	0.8746 (0.021)	0.8236 (0.032)	0.8700	5.33
	CoT-RL	0.7784 (0.029)	0.8710 ( <u>0.014</u> )	0.7540 (0.026)	0.8300	9.16
	CoA	0.7240 (0.019)	0.7086 (0.019)	0.6944 (0.025)	0.7200	3.56
	AbstraL	0.8686 * ( <u>0.013</u> )	0.8672 (0.018)	0.8620 * ( <u>0.023</u> )	0.8700	0.92
Qwen2.5-0.5B-Instruct	CoT-8S	0.3394 (0.039)	0.3724 ( <u>0.024</u> )	0.3398 (0.033)	0.3800	10.58
	CoT-RL	0.3192 (0.025)	0.3948 (0.032)	0.3228 (0.032)	0.3500	7.77
	CoA	0.2866 (0.025)	0.3060 (0.026)	0.2872 (0.026)	0.2900	0.97
	AbstraL	0.4396 * ( <u>0.015</u> )	0.4416 * (0.026)	0.4456 * ( <u>0.025</u> )	0.4400 *	-1.27
Qwen2.5-1.5B-Instruct	CoT-8S	0.5728 (0.032)	0.6416 (0.030)	0.5752 (0.033)	0.6600	12.85
	CoT-RL	0.5296 (0.037)	0.5830 (0.034)	0.5126 (0.034)	0.5600	8.46
	CoA	0.4680 (0.027)	0.4942 (0.029)	0.4656 (0.027)	0.5100	8.71
	AbstraL	0.6444 * ( <u>0.018</u> )	0.6414 ( <u>0.028</u> )	0.6416 * ( <u>0.025</u> )	0.6500	1.29
Qwen2.5-3B-Instruct	CoT-8S	0.7222 (0.037)	0.7820 (0.025)	0.7256 (0.027)	0.8200 *	11.51
	CoT-RL	0.7150 (0.036)	0.7706 (0.025)	0.6888 (0.038)	0.7900	12.81
	CoA	0.6424 (0.030)	0.6134 (0.030)	0.6234 (0.034)	0.6500	4.09
	AbstraL	0.7842 * ( <u>0.014</u> )	0.7852 ( <u>0.024</u> )	0.7834 * ( <u>0.024</u> )	0.7900	0.84
Qwen2.5-7B-Instruct	CoT-8S	0.8726 (0.026)	0.9230 ( <u>0.016</u> )	0.8740 (0.023)	0.9200 *	5.00
	CoT-RL	0.7770 (0.033)	0.8170 (0.024)	0.7928 (0.034)	0.8500	6.73
	CoA	0.7310 (0.023)	0.7408 (0.027)	0.7414 (0.029)	0.7600	2.45
	AbstraL	0.9022 * ( <u>0.014</u> )	0.9248 (0.017)	0.8834 * ( <u>0.019</u> )	0.8900	0.74
Qwen2.5-Math-7B-Instruct	CoT-8S	0.8956 (0.021)	0.9108 (0.018)	0.8766 (0.023)	0.9500	7.73
	CoT-RL	0.8942 (0.022)	0.9154 ( <u>0.012</u> )	0.8812 (0.021)	0.9600	8.21
	CoA	0.7122 (0.028)	0.6976 (0.031)	0.6970 (0.033)	0.7100	1.83
	AbstraL	0.9066 * ( <u>0.015</u> )	0.9014 (0.013)	0.9022 * ( <u>0.016</u> )	0.9100	0.86
Mathstral-7B-v0.1	CoT-8S	0.7876 (0.024)	0.8084 (0.018)	0.7604 (0.031)	0.8000	4.95
	CoT-RL	0.8082 (0.018)	0.7986 (0.021)	0.7688 (0.025)	0.7800	1.44
	CoA	0.7506 (0.031)	0.7740 (0.028)	0.7402 (0.027)	0.7500	1.31
	AbstraL	0.8100 * ( <u>0.012</u> )	0.8214 * ( <u>0.017</u> )	0.8228 * ( <u>0.019</u> )	0.8100	-1.58

答案匹配。我们将我们的两步重写应用于来自 OpenAI<sup>10</sup> 的 GSM8K [8] 训练集的苏格拉底式版本。经过第一步的重写和过滤，在 7473 个训练样本中有 6503 个样本被正确重写并保留，而经过第二步的重写和过滤，在 6503 个训练样本中有 6386 个样本被正确重写并作为最终训练数据保留。

在我们的 AbstraL 框架的抽象推理步骤中，LLMs 通过 SFT 和 RL 训练，以根据输入抽象问题  $\mathcal{X}^A$  生成抽象 (Granular) 答案  $\mathcal{Y}^A$  的任务。

对于 SFT，我们将批量大小设置为 8，使用 4 个 NVIDIA A100-SXM4 (80GB) GPU (*i.e.*，每个 GPU 上的批量大小为 2)，学习率设置为  $5e^{-6}$ ，使用带有  $\beta_1 = 0.9$ 、 $\beta_2 = 0.999$  和  $\epsilon = 1e^{-8}$  的 AdamW [18] 优化器。所有 LLM 在我们的 Granular 数据上用 SFT 训练了 2 个 epoch，用时不到 1 小时。

对于强化学习 (RL)，我们将正奖励 (正确的) 设为  $r_{correct} = 2.5$ ，并在我们的符号距离奖励  $r_{symbolic}(\bar{A}_i, \mathcal{A})$  中设定最大奖励  $r_{max} = 1.5$ 。GRPO 算法的超参数被设置为  $\beta = 0.04$  (KL 系数) 和  $\epsilon = 0.2$  (用于剪辑)，每组采样的世代数量 (*i.e.*，每给定的  $\mathcal{X}^A$  的  $\{\mathcal{Y}_1^A, \mathcal{Y}_2^A, \dots, \mathcal{Y}_G^A\}$ ) 被设置为  $G = 16$ ，采样的温度、 $\text{top}_p$  和  $\text{top}_k$  分别被设置为 0.9、1.0 和 50。我们的 RL 优化的学习率被设为  $5e^{-7}$ ，也使用了 AdamW 优化器，参数为  $\beta_1 = 0.9$ 、 $\beta_2 = 0.99$  和  $\epsilon = 1e^{-8}$ 。使用了余弦学习率调度器，预热比率设为 0.1，训练的权重衰减和梯度范数剪辑

<sup>10</sup><https://huggingface.co/datasets/openai/gsm8k/tree/main/socratic>

Table 5: GSM-Plus 的完整评估结果。每个模型的最佳结果用粗体表示。使用 \* 的最佳（粗体）结果明显优于其对应的第二佳结果，显著性检验的  $p$  值为  $< 0.05$ 。

Model	Method	Digit Ex.	Int-Dec-Fra	Num. Sub.	Rephrase	Distract	Original
Llama-3.2-1B-Instruct	CoT-8S	0.3889	0.2934	0.4321	0.5042	0.2790	0.4519
	CoT-RL	0.4064	0.3063	0.4572	0.5299	0.2646	0.5125
	CoA	0.3798	0.3237	0.3889	0.4139	0.2032	0.4261
	AbstraL	0.5641 *	0.5626 *	0.5641 *	0.5633 *	0.4359 *	0.5641 *
Llama-3.2-3B-Instruct	CoT-8S	0.7142	0.6277	0.7172	0.7877	0.6073	0.7953
	CoT-RL	0.6808	0.5406	0.7043	0.7718	0.5398	0.7763
	CoA	0.6224	0.5444	0.6126	0.6793	0.4177	0.6626
	AbstraL	0.7968 *	0.7945 *	0.7968 *	0.7923	0.6755 *	0.7968
Llama-3.1-8B-Instruct	CoT-8S	0.7938	0.7445	0.7854	0.8461	0.7400	0.8567
	CoT-RL	0.7240	0.5914	0.7187	0.8309	0.5466	0.8234
	CoA	0.7202	0.6398	0.7035	0.7657	0.5344	0.7497
	AbstraL	0.8506 *	0.8476 *	0.8506 *	0.8514	0.7854 *	0.8506
Qwen2.5-0.5B-Instruct	CoT-8S	0.3601	0.2866	0.3958	0.4359	0.2267	0.4238
	CoT-RL	0.3336	0.2373	0.3571	0.4079	0.1524	0.3798
	CoA	0.2745	0.2267	0.2782	0.3161	0.1266	0.3033
	AbstraL	0.4670 *	0.4663 *	0.4670 *	0.4625 *	0.3654 *	0.4670 *
Qwen2.5-1.5B-Instruct	CoT-8S	0.6096	0.5421	0.6194	0.6793	0.4488	0.6702
	CoT-RL	0.5527	0.4632	0.5754	0.6520	0.3950	0.6202
	CoA	0.4602	0.3700	0.4511	0.5186	0.2631	0.5019
	AbstraL	0.6778 *	0.6763 *	0.6778 *	0.6755	0.5777 *	0.6778
Qwen2.5-3B-Instruct	CoT-8S	0.7726	0.7074	0.7430	0.8218	0.6346	0.8120
	CoT-RL	0.7149	0.6353	0.7331	0.7817	0.5277	0.7726
	CoA	0.6232	0.5497	0.5989	0.6679	0.4200	0.6702
	AbstraL	0.8158 *	0.8128 *	0.8158 *	0.8249	0.7036 *	0.8158
Qwen2.5-7B-Instruct	CoT-8S	0.8400	0.8067	0.8324	0.8779	0.7938	0.8901
	CoT-RL	0.7582	0.6854	0.7369	0.8097	0.6331	0.8036
	CoA	0.7195	0.6528	0.6755	0.7657	0.5588	0.7597
	AbstraL	0.8825 *	0.8795 *	0.8825 *	0.8870 *	0.7953	0.8825
Qwen2.5-Math-7B-Instruct	CoT-8S	0.8552	0.8218	0.8453	0.9052	0.7627	0.9181
	CoT-RL	0.8757	0.8522	0.8506	0.9037	0.8150	0.9340 *
	CoA	0.7460	0.6907	0.7180	0.7642	0.5709	0.7809
	AbstraL	0.8916 *	0.8886 *	0.8916 *	0.8992	0.8226	0.8916
Mathstral-7B-v0.1	CoT-8S	0.7544	0.6892	0.7604	0.8029	0.6808	0.8074
	CoT-RL	0.7665	0.7111	0.7331	0.8089	0.5542	0.7953
	CoA	0.7149	0.6422	0.6831	0.7483	0.5246	0.7619
	AbstraL	0.8241 *	0.8211 *	0.8241 *	0.8247 *	0.7657 *	0.8241 *

值也都设为 0.1。我们使用 8 个 NVIDIA A100-SXM4 (80GB) GPU 来训练每个 LLM 的 RL。对于 7B 和 8B 的 LLM，生成的批量大小设置为每个 GPU 2，梯度累积步设置为 4，*i.e.*，当每 4 组  $G = 16$  代（2 批量大小乘以 8 个 GPU）采样时，执行策略梯度更新。对于其他较小的 LLM，生成的批量大小则设置为每个 GPU 8，梯度累积步设置为 1，此时每次策略梯度更新仍采样 4 组  $G = 16$  代（8 批量大小乘以 8 个 GPU）。所有的 LLM 使用 RL 在我们的 Granular 数据上训练了 8 个周期，这大约花费了 3 到 5 天。

为了构建基线方法 CoA [10] 的训练数据，我们提示相同的 Llama-3.3-70B-Instruct 模型重写黄金苏格拉底 CoT 响应  $\mathcal{Y}$ （也会提供原始输入问题  $\mathcal{X}$ ），其中仅将  $\mathcal{Y}$  中的输出数字替换为抽象符号。表 11 展示了用于 CoA 训练数据构建的指令和少数示例。遵循 CoA 方法，LLM 仅通过 SFT 训练以学习基于输入问题  $\mathcal{X}$  生成 CoA 抽象答案，使用与 AbstraL 的 SFT 相同的超参数。相同的符号工具（正则表达式匹配器和 SymPy 求解器）用于提取抽象推导并计算最终答案数字，无论是用于过滤重写的训练数据还是在推理时推导答案。

## References

- [1] Emmanuel Abbe, Samy Bengio, Aryo Lotfi, Colin Sandon, and Omid Saremi. How far can transformers reason? the globality barrier and inductive scratchpad. *Advances in Neural Information Processing Systems*, 37:27850–27895, 2024.



- [2] Qiming Bao, Gael Gendron, Alex Yuxuan Peng, Wanjun Zhong, Neset Tan, Yang Chen, Michael Witbrock, and Jiamou Liu. Assessing and enhancing the robustness of large language models with task structure variations for logical reasoning. *arXiv preprint arXiv:2310.09430*, 2023.
- [3] Enric Boix-Adserà, Omid Saremi, Emmanuel Abbe, Samy Bengio, Etai Littwin, and Joshua M. Susskind. When can transformers reason with abstract symbols? In *The Twelfth International Conference on Learning Representations*, 2024.
- [4] Guanyu Chen, Peiyang Wang, Tianren Zhang, and Feng Chen. Exploring the hidden reasoning process of large language models by misleading them. *arXiv preprint arXiv:2503.16401*, 2025.
- [5] Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025.
- [6] François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.
- [7] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- [8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [9] Wee Chung Gan and Hwee Tou Ng. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 6065–6075, 2019.
- [10] Silin Gao, Jane Dwivedi-Yu, Ping Yu, Xiaoqing Ellen Tan, Ramakanth Pasunuru, Olga Golovneva, Koustuv Sinha, Asli Celikyilmaz, Antoine Bosselut, and Tianlu Wang. Efficient tool use with chain-of-abstraction reasoning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2727–2743, 2025.
- [11] Gaël Gendron, Qiming Bao, Michael Witbrock, and Gillian Dobbie. Large language models are not strong abstract reasoners. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 6270–6278, 2024.
- [12] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [13] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [14] Ruixin Hong, Hongming Zhang, Xiaoman Pan, Dong Yu, and Changshui Zhang. Abstraction-of-thought makes language models better reasoners. *arXiv preprint arXiv:2406.12442*, 2024.
- [15] Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. Llm post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321*, 2025.
- [16] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.
- [17] Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2961–2984, 2024.

- [18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.
- [19] Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, 2022.
- [20] Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [21] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- [22] Kun Qian, Shunji Wan, Claudia Tang, Youzhi Wang, Xuanming Zhang, Maximillian Chen, and Zhou Yu. Varbench: Robust language model benchmarking through dynamic variable perturbation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16131–16161, 2024.
- [23] Yao Qiang, Subhrangshu Nandi, Ninareh Mehrabi, Greg Ver Steeg, Anoop Kumar, Anna Rumshisky, and Aram Galstyan. Prompt perturbation consistency learning for robust language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1357–1370, 2024.
- [24] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.
- [25] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.
- [26] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. *Advances in neural information processing systems*, 34:29935–29948, 2021.
- [27] Lorenza Saitta, Jean-Daniel Zucker, Lorenza Saitta, and Jean-Daniel Zucker. *Abstraction in Artificial Intelligence*. Springer, 2013.
- [28] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [29] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- [30] Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. Distilling reasoning capabilities into smaller language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7059–7073, 2023.
- [31] Saurabh Srivastava, Anto PV, Shashank Menon, Ajay Sukumar, Alan Philipose, Stevin Prince, Sooraj Thomas, et al. Functional benchmarks for robust evaluation of reasoning performance, and the reasoning gap. *arXiv preprint arXiv:2402.19450*, 2024.
- [32] Sowmya S Sundaram, Sairam Gurajada, Deepak Padmanabhan, Savitha Sam Abraham, and Marco Fisichella. Does a language model “understand” high school math? a survey of deep learning based word problem solvers. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 14(4):e1534, 2024.

- [33] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [34] Robert Wilensky. Planning and understanding: A computational approach to human reasoning. 1983.
- [35] Cheng Xu, Shuhao Guan, Derek Greene, M Kechadi, et al. Benchmark data contamination of large language models: A survey. *arXiv preprint arXiv:2406.04244*, 2024.
- [36] Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. Are large language models really good logical reasoners? a comprehensive evaluation and beyond. *IEEE Transactions on Knowledge and Data Engineering*, 2025.
- [37] Kuan-Man Xu. Using the bootstrap method for a statistical significance test of differences between summary histograms. *Monthly weather review*, 134(5):1442–1453, 2006.
- [38] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- [39] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.
- [40] Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E Gonzalez, and Ion Stoica. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*, 2023.
- [41] Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10210–10229, 2024.
- [42] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.
- [43] Fei Yu, Hongbo Zhang, Prayag Tiwari, and Benyou Wang. Natural language reasoning, a survey. *ACM Computing Surveys*, 56(12):1–39, 2024.
- [44] Tong Yu, Yongcheng Jing, Xikun Zhang, Wentao Jiang, Wenjie Wu, Yingjie Wang, Wenbin Hu, Bo Du, and Dacheng Tao. Benchmarking reasoning robustness in large language models. *arXiv preprint arXiv:2503.04550*, 2025.
- [45] Tianyang Zhong, Zhengliang Liu, Yi Pan, Yutong Zhang, Yifan Zhou, Shizhe Liang, Zihao Wu, Yanjun Lyu, Peng Shu, Xiaowei Yu, et al. Evaluation of openai o1: Opportunities and challenges of agi. *arXiv preprint arXiv:2409.18486*, 2024.
- [46] Yue Zhou, Yada Zhu, Diego Antognini, Yoon Kim, and Yang Zhang. Paraphrase and solve: Exploring and exploiting the impact of surface form on mathematical reasoning in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2793–2804, 2024.



Table 6: GSM-Symbolic 数据集上的数学推理示例。响应由 Qwen2.5-Math-7B-Instruct 使用 CoT-8S 或 AbstraL 方法生成。输入查询包括原始问题（来自 Origin 100）及其对应的变体（来自 Vary Both），其名称（或实体）和数字被更改为其他值，如由 橙色 单词所示。对于 Origin 100 问题，CoT-8S 和 AbstraL 都实现了正确的数学推导和答案，分别用 蓝色 和 棕色 着色。然而，当转移到 Vary Both 问题时，CoT-8S 对更改后的数字（7 x 39）进行错误计算，因此得出了错误答案，如 红色 数字所示。相比之下，AbstraL 由于使用对输入数字变化不变的抽象符号（带有符号求解器），保持了正确的数学推导。

---

GSM-Symbolic (Origin 100): 海梅 places 鸡蛋 on some trays. Each tray can hold 24 个鸡蛋. If he has 64 个鸡蛋 and 2 trays, how many 鸡蛋 won't he be able to place on the tray?

Gold Response: Jaime will be able to place a total of  $24 \times 2 = 48$  eggs. Therefore, there are  $64 - 48 = 16$  eggs that he won't be able to place on the tray. The answer is 16.

Response with CoT-8S: Let's think step by step. Each tray can hold 24 eggs, and Jaime has 2 trays, so the total capacity of the trays is 24 乘 2 等于 48 eggs. Jaime has 64 eggs, but only 48 can fit on the trays. Therefore, the number of eggs that won't be able to be placed on the tray is  $64 - 48 = 16$ . The final answer is 16.

Response with AbstraL:

(Abstract Question) Jaime places eggs on the tray. Each tray can hold [in0] eggs. If he has [in1] eggs and [控制 X<sub>MATHX</sub>\_x 的入口] trays, how many eggs won't he be able to place on the tray?

(Conditions) 输入 0=24 输入 1=64 in2=2

(Abstract Answer) Let's think about the sub-questions we need to answer.

Q1: How many eggs can Jaime place on the trays?

Q2: How many eggs won't Jaime be able to place on the tray?

Let's answer each sub-question one by one.

Q1: How many eggs can Jaime place on the trays? Each tray can hold [in0] eggs. Jaime has [in2] trays.

So, the total number of eggs that can be placed on the trays is  $<< \text{in0} * \text{in2} = \text{out0} >>$  eggs.

Q2: How many eggs won't Jaime be able to place on the tray? Jaime has [in1] eggs. The total number of eggs that can be placed on the trays is [out0] eggs. So, the number of eggs that won't be able to be placed on the tray is  $<< \text{in1} - \text{out0} = \text{out1} >>$  eggs. The final answer is out1.

(Abstraction)  $\text{in0} * \text{in2} = \text{out0}$  输入 1-输出 0= 输出 1

(Derived Final Answer) out0=48 out1=16

---

GSM-Symbolic (Vary Both): 阿拉什 places 饼干 on some trays. Each tray can hold 39 个饼干. If he has 302 个饼干 and 7 trays, how many 饼干 won't he be able to place on the tray?

Gold Response: Arash will be able to place a total of  $39 \times 7 = 273$  crackers. Therefore, there are  $302 - 273 = 29$  crackers that he won't be able to place on the tray. The answer is 29.

Response with CoT-8S: First, we need to determine how many crackers Arash can place on the 7 trays.

Since each tray can hold 39 crackers, the total number of crackers that can be placed on 7 trays is  $(7 \times 39)$ .

Calculating  $(7 \times 39)$ :  $[7 \text{ 乘以 } 39 = 7 \times (40 - 1) = 7 \times 40 - 7 \times 1 = 280 - 7 = 273]$ .

So, Arash can place 273 crackers on the 7 trays. He has 302 crackers in total. To find out how many crackers he won't be able to place, we subtract the number of crackers he can place from the total number of crackers:  $[302 - 273 = 29]$ . The final answer is 29. Arash won't be able to place 29 crackers on the trays.

Response with AbstraL:

(Abstract Question) Arash places crackers on some trays. Each tray can hold [在 0] crackers.

If he has [in1] crackers and [in2] trays, how many crackers won't he be able to place on the tray?

(Conditions) in0=39 in1=302 输入 2=7

(Abstract Answer) Let's think about the sub-questions we need to answer.

Q1: How many crackers can Arash place on the trays?

Q2: How many crackers won't Arash be able to place on the tray?

Let's answer each sub-question one by one.

Q1: How many crackers can Arash place on the trays? Each tray can hold [in0] crackers. Arash has [in2] trays.

So, the total number of crackers that can be placed on the trays is  $<< \text{in0} \text{ 乘以 } \text{in2} \text{ 等于 } \text{out0} >>$  crackers.

Q2: How many crackers won't Arash be able to place on the tray? Arash has [in1] crackers. The total number of crackers that can be placed on the trays is [out0]. So, the number of crackers that won't be able to be placed on the tray is  $<< \text{输入 1 输出 0} = \text{输出 1} >>$  crackers. The final answer is out1.

(Abstraction)  $\text{in0} * \text{in2} = \text{out0}$  内 1 出 0= 出 1

(Derived Final Answer) out0=273 out1=29

---

Table 7: GSM-Plus 数据集上的数学推理示例。响应由 Qwen2.5-Math-7B-Instruct 通过 CoT-8S 或 AbstrAL 方法生成。输入查询包括一个原始问题（来自 Original）及其对应的变体（来自 Distract），其中一个干扰条件被插入到问题中，如 紫色 句所示。对于原始问题，CoT-8S 和 AbstrAL 都能够得到正确的数学推导和答案，分别用 蓝色 和 棕色 标记。然而，当转到 Distract 问题时，CoT-8S 会采用错误的推理链条，其中干扰条件错误地被合并到答案的推导中，如 红色 句所示。相比之下，AbstrAL 在一个额外的子问题（Q2）中处理干扰条件，其数学推导（用 灰色 标记）不干扰正确答案的推导（用 棕色 标记）。这表明，我们在 Granular 训练数据格式中使用的问题解决的整体规划能够有效地缓解干扰项的影响，从而增强对抗抗性分布转移的推理鲁棒性。

---

GSM-Plus (Original): In a race with 240 Asians, 80 were Japanese, and the rest were Chinese. If the number of boys on the Chinese team was 60, how many girls were on the Chinese team?

Gold Response: If there were 80 Japanese, the number of Chinese was  $240-80=160$ . Since there were 60 boys in the Chinese team, there were  $160-60=100$  girls in the Chinese team. The answer is 100.

Response with CoT-8S: Let's think step by step. First, we need to determine the number of Chinese participants in the race. We know there are 240 Asians in total and 80 of them are Japanese. Therefore, the number of Chinese participants is:  $[240-80=160]$ . Next, we know that out of these 160 Chinese participants, 60 are boys. To find the number of girls, we subtract the number of boys from the total number of Chinese participants:  $[160-60=100]$ . So, the number of girls on the Chinese team is 100. The final answer is 100 .

Response with AbstrAL:

(Abstract Question) In a race with  $[in0]$  Asians,  $[在 1]$  were Japanese, and the rest were Chinese. If the number of boys on the Chinese team was  $[in2]$ , how many girls were on the Chinese team?

(Conditions)  $in0=240$   $in1=80$   $in2=60$

(Abstract Answer) Let's think about the sub-questions we need to answer. Q1: How many Chinese were there?

Q2: How many girls were on the Chinese team? Let's answer each sub-question one by one.

Q1: How many Chinese were there? The total number of Asians in the race is  $[in0]$ . There were  $[in1]$  Japanese among them. So, the number of Chinese was  $<< in0 减去 in1 等于 out0 >>$ .

Q2: How many girls were on the Chinese team? The total number of Chinese in the race is  $[out0]$ . The number of boys on the Chinese team was  $[in2]$ . So, the number of girls on the Chinese team was  $<< 输出 0-输入 2= 输出 1 >>$ . The final answer is  $out1$  .

(Abstraction)  $in0 扣除 in1 得到 out0$   $out0-in2=out1$

(Derived Final Answer)  $out0=160$   $out1=100$

---

GSM-Plus (Distract): In a race with 240 Asians, 80 were Japanese, and the rest were Chinese. 在日本人中, 有 30 个女孩和 50 名男孩。 If the number of boys on the Chinese team was 60, how many girls were on the Chinese team?

Gold Response: If there were 80 Japanese, the number of Chinese was  $240-80=160$ . Since there were 60 boys in the Chinese team, there were  $160-60=100$  girls in the Chinese team. The answer is 100.

Response with CoT-8S: Let's break this down step by step. 1.\*\*Total number of Asians in the race\*\*: 240. 2.\*\*Number of Japanese in the race\*\*: 80. 3.\*\*Number of Chinese in the race\*\*:  $[240-80=160]$ . 4.\*\*Number of boys among the Japanese\*\*: 50. 5.\*\*Number of girls among the Japanese\*\*: 30. 6.\*\*Number of boys on the Chinese team\*\*: 60. 7.\*\*Number of girls on the Chinese team\*\*: Since the total number of Chinese participants is 160, and we know the number of boys (60) and girls (30) among the Japanese, 我们可以通过从总数中减去日本队中的男孩和女孩的数量来找到中国队中的女孩数量 中国人数. First, calculate the total number of girls among the Japanese:  $[30 \text{ (girls among Japanese)} + 0 \text{ (girls among Chinese, since we don't have this information yet)} = 30]$ . Now, calculate the total number of boys among the Japanese and Chinese  $[50 \text{ (boys among Japanese)} + 60 \text{ (boys among Chinese)} = 110]$ . 从总人数中减去日籍男孩和女孩的人数 中文:  $[160-110=50]$ . Therefore, the number of girls on the Chinese team is 50. The final answer is 50 .

Response with AbstrAL:

(Abstract Question) In a race with  $[在 0]$  Asians,  $[in1]$  were Japanese, and the rest were Chinese. Among the Japanese, there are  $[英寸 2]$  girls and  $[在 3]$  boys. If the number of boys on the Chinese team was  $[in4]$ , how many girls were on the Chinese team?

(Conditions)  $in0=240$   $in1=80$   $in2=30$   $in3=50$   $in4=60$

(Abstract Answer) Let's think about the sub-questions we need to answer. Q1: How many Chinese were there? Q2: How many boys were there among the Chinese? Q3: How many girls were there among the Chinese? Let's answer each sub-question one by one.

Q1: How many Chinese were there? There were  $[in0]$  Asians in total.  $[in1]$  of them were Japanese.

So, the number of Chinese is  $<< in0-in1=out0 >>$ .

Q2: How many boys were there among the Chinese? The total number of boys among the Japanese is  $[in2] + [in3] =$

$<< in2+in3=out1 >>$ . The total number of boys among the Chinese is  $[in4]$ . So, the total number of boys among the Chinese is  $[in4]$ .

Q3: How many girls were there among the Chinese? The total number of Chinese is  $[out0]$ . The total number of boys among the Chinese is  $[in4]$ . So, the total number of girls among the Chinese is  $<< out0-in4=out2 >>$ . The final answer is  $out2$  .

(Abstraction)  $in0-输入 1= 输出 0$   $in2+in3=out1$   $out0-in4=out2$

(Derived Final Answer)  $out0=160$   $out1=80$   $输出 2=100$

---

Table 8: 在 AbstraL 中用于提示条件识别的指令和少量示例。

---

You are given a math word problem. Please replace the numerical values in the problem with abstract symbols, and enclose the abstract symbols in square brackets. Do not change any other words or add any other contents to the problem. Please also write equations to clarify which numerical value each abstract symbol is assigned to.

Example 1:

Input problem: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May.

How many clips did Natalia sell altogether in April and May?

Output problem: Natalia sold clips to [in0] of her friends in April, and then she sold [in1] as many clips in May.

How many clips did Natalia sell altogether in April and May?

Equations:  $\text{in0}=48$     $\text{in1}=1/2$

Example 2:

Input problem: The flowers cost \$ 9, the clay pot costs \$ 20 more than the flower, and the bag of soil costs \$ 2 less than the flower. How much does it cost to plant the flowers?

Output problem: The flowers cost \$ [in0], the clay pot costs \$ [in1] more than the flower, and the bag of soil costs \$ [in2] less than the flower. How much does it cost to plant the flowers?

Equations:  $\text{in0}=9$     $\text{in1}=20$     $\text{in2}=2$

Example 3:

Input problem: From March to August, Sam made \$ 460 doing 23 hours of yard work. However, from September to February, Sam was only able to work for 8 hours. If Sam is saving up to buy a video game console that costs \$ 600 and has already spent \$ 340 to fix his car, how many more hours does he need to work before he can buy the video game console?

Output problem: From March to August, Sam made \$ [in0] doing [in1] hours of yard work. However, from September to February, Sam was only able to work for [in2] hours. If Sam is saving up to buy a video game console that costs \$ [in3] and has already spent \$ [in4] to fix his car, how many more hours does he need to work before he can buy the video game console?

Equations:  $\text{in0}=460$     $\text{in1}=23$     $\text{in2}=8$     $\text{in3}=600$     $\text{in4}=340$

Example 4:

Input problem: Zhang is twice as old as Li. Li is 12 years old. Zhang's brother Jung is 2 years older than Zhang. How old is Jung?

Output problem: Zhang is [in0] times as old as Li. Li is [in1] years old. Zhang's brother Jung is [in2] years older than Zhang. How old is Jung?

Equations:  $\text{in0}=2$     $\text{in1}=12$     $\text{in2}=2$

Example 5:

Input problem: Of the 90 people on William's bus,  $3/5$  were Dutch. Of the  $1/2$  of the Dutch who were also American,  $1/3$  got window seats. What's the number of Dutch Americans who sat at the windows?

Output problem: Of the [in0] people on William's bus, [in1] were Dutch. Of the [in2] of the Dutch who were also American, [in3] got window seats. What's the number of Dutch Americans who sat at the windows?

Equations:  $\text{in0}=90$     $\text{in1}=3/5$     $\text{in2}=1/2$     $\text{in3}=1/3$

---



Table 9: 用于学习 AbstraL 中的抽象推理的 Granular 训练数据构建的第一步响应重写的说明和少量示例。

You are given a math word problem with input conditions and solution. Please rewrite the solution by replacing the numerical values in angle brackets with abstract symbols. If the numerical values are given in the conditions, replace them with the abstract symbols assigned to them in the square brackets, otherwise replace them with new abstract symbols. Please also remove the redundant calculations around the angle brackets. Do not add any other contents to the solution.

Example 1:

Problem: Natalia sold clips to [in0] of her friends in April, and then she sold [in1] as many clips in May.  
How many clips did Natalia sell altogether in April and May?  
Conditions: in0=48 in1=1/2  
Solution: How many clips did Natalia sell in May? \*\* Natalia sold  $48/2 = << 48/2=24 >> 24$  clips in May.  
How many clips did Natalia sell altogether in April and May? \*\* Natalia sold  $48+24 = << 48+24=72 >> 72$  clips altogether in April and May. The answer is 72.  
Rewrite solution: How many clips did Natalia sell in May? \*\* Natalia sold  $<< in0*in1=out0 >>$  clips in May.  
How many clips did Natalia sell altogether in April and May? \*\* Natalia sold  $<< in0+out0=out1 >>$  clips altogether in April and May. The answer is out1.

Example 2:

Problem: The flowers cost \$ [in0], the clay pot costs \$ [in1] more than the flower, and the bag of soil costs \$ [in2] less than the flower. How much does it cost to plant the flowers?  
Conditions: in0=9 in1=20 in2=2  
Solution: How much does the clay pot cost? \*\* The clay pot costs  $\$ 20 + \$ 9 = << 20+9=29 >> 29$ .  
How much does the bag of soil cost? \*\* The bag of soil costs  $\$ 9 - \$ 2 = << 9-2=7 >> 7$ .  
How much does it cost to plant the flowers? \*\* The cost to plant the flowers is  $\$ 9 + \$ 29 + \$ 7 = << 9+29+7=45 >> 45$ .  
The answer is 45.  
Rewrite solution: How much does the clay pot cost? \*\* The clay pot costs  $<< in1+in0=out0 >>$ .  
How much does the bag of soil cost? \*\* The bag of soil costs  $<< in0-in2=out1 >>$ .  
How much does it cost to plant the flowers? \*\* The cost to plant the flowers is  $<< in0+out0+out1=out2 >>$ .  
The answer is out2.

Example 3:

Problem: From March to August, Sam made \$ [in0] doing [in1] hours of yard work. However, from September to February, Sam was only able to work for [in2] hours. If Sam is saving up to buy a video game console that costs \$ [in3] and has already spent \$ [in4] to fix his car, how many more hours does he need to work before he can buy the video game console?  
Conditions: in0=460 in1=23 in2=8 in3=600 in4=340  
Solution: How much does Sam make per hour? \*\* Sam makes  $\$ 460 / 23 \text{ hrs} = << 460/23=20 >> 20/\text{hr}$ . How much did Sam make from September to February? \*\* From September to February, Sam made  $8 \text{ hrs} \times \$ 20/\text{hr} = << 8*20=160 >> 160$ .  
How much did Sam make from March to February? \*\* From March to February, Sam made a total of  $\$ 460 + \$ 160 = \$ 620$ .  
How much money did Sam have after fixing his car? \*\* After fixing his car, he was left with  $\$ 620 - \$ 340 = << 620-340=280 >> 280$ .  
How much money does Sam need to buy the video game console? \*\* Sam needs another  $\$ 600 - \$ 280 = << 600-280=320 >> 320$ .  
How many more hours does Sam need to work? \*\* Sam needs to work another  $\$ 320 / \$ 20/\text{hr} = << 320/20=16 >> 16$  hours.  
The answer is 16.  
Rewrite solution: How much does Sam make per hour? \*\* Sam makes  $<< in0/in1=out0 >> / \text{hr}$ . How much did Sam make from September to February? \*\* From September to February, Sam made  $<< in2*out0=out1 >>$ .  
How much did Sam make from March to February? \*\* From March to February, Sam made a total of  $<< in0+out1=out2 >>$ .  
How much money did Sam have after fixing his car? \*\* After fixing his car, he was left with  $<< out2-in4=out3 >>$ .  
How much money does Sam need to buy the video game console? \*\* Sam needs another  $<< in3-out3=out4 >>$ .  
How many more hours does Sam need to work? \*\* Sam needs to work another  $<< out4/out0=out5 >>$  hours.  
The answer is out5.

Example 4:

Problem: Zhang is [in0] times as old as Li. Li is [in1] years old. Zhang's brother Jung is [in2] years older than Zhang. How old is Jung?  
Conditions: in0=2 in1=12 in2=2  
Solution: How old is Zhang? \*\* Zhang is  $2 * 12 \text{ years old} = << 2*12=24 >> 24$  years old.  
How old is Jung? \*\* Jung is  $2 \text{ years} + 24 \text{ years} = << 2+24=26 >> 26$  years old.  
The answer is 26.  
Rewrite solution: How old is Zhang? \*\* Zhang is  $<< in0*in1=out0 >>$  years old.  
How old is Jung? \*\* Jung is  $<< in2+out0=out1 >>$  years old.  
The answer is out1.

Example 5:

Problem: Of the [in0] people on William's bus, [in1] were Dutch. Of the [in2] of the Dutch who were also American, [in3] got window seats. What's the number of Dutch Americans who sat at the windows?  
Conditions: in0=90 in1=3/5 in2=1/2 in3=1/3  
Solution: How many Dutch people were on the bus? \*\* On the bus, the number of Dutch people was  $3/5$  of the total number, a total of  $3/5*90 = << 3/5*90=54 >> 54$  people.  
How many Dutch Americans were on the bus? \*\* Out of the 54 people who were Dutch,  $1/2$  were Dutch Americans, a total of  $1/2*54 = << 1/2*54=27 >> 27$  people.  
How many Dutch Americans sat at the windows? \*\* If  $1/3$  of the passengers on the bus identifying as Dutch Americans sat at the windows, their number is  $1/3*27 = << 1/3*27=9 >> 9$ .  
The answer is 9.  
Rewrite solution: How many Dutch people were on the bus? \*\* On the bus, the number of Dutch people was [in1] of the total number, a total of  $<< in1*in0=out0 >>$  people.  
How many Dutch Americans were on the bus? \*\* Out of the [out0] people who were Dutch, [in2] were Dutch Americans, a total of  $<< in2*out0=out1 >>$  people.  
How many Dutch Americans sat at the windows? \*\* If [in3] of the passengers on the bus identifying as Dutch Americans sat at the windows, their number is  $<< in3*out1=out2 >>$ .  
The answer is out2.

Table 10: 用于学习 AbstrAL 中的抽象推理的 Granular 训练数据构建的第二个响应重写步骤的说明和少量示例。

---

You are given a math word problem with solution. The numerical values in the problem are replaced with abstract symbols and enclosed in square brackets. The calculations in the solution are also composed of abstract symbols and enclosed in double angle brackets. Please rewrite the solution by first listing all sub-questions, and then answering each sub-question one by one. Please list the relevant conditions before answering each sub-question. Please clarify the final answer at the end of the solution.

---

Example 1:

Problem: Natalia sold clips to [in0] of her friends in April, and then she sold [in1] as many clips in May.

How many clips did Natalia sell altogether in April and May?

Solution: How many clips did Natalia sell in May? \*\* Natalia sold  $\langle \langle in0 * in1 = out0 \rangle \rangle$  clips in May.

How many clips did Natalia sell altogether in April and May? \*\* Natalia sold  $\langle \langle in0 + out0 = out1 \rangle \rangle$  clips altogether in April and May. The answer is out1.

Rewrite solution: Let's think about the sub-questions we need to answer.

Q1: How many clips did Natalia sell in May?

Q2: How many clips did Natalia sell altogether in April and May?

Let's answer each sub-question one by one.

Q1: How many clips did Natalia sell in May? Natalia sold [in0] clips in April. She sold [in1] as many clips in May as she did in April. So she sold  $\langle \langle in0 * in1 = out0 \rangle \rangle$  clips in May.

Q2: How many clips did Natalia sell altogether in April and May? Natalia sold [in0] clips in April. She sold [out0] clips in May. So she sold  $\langle \langle in0 + out0 = out1 \rangle \rangle$  clips altogether in April and May.

The final answer is out1.

Example 2:

Problem: The flowers cost \$ [in0], the clay pot costs \$ [in1] more than the flower, and the bag of soil costs \$ [in2] less than the flower. How much does it cost to plant the flowers?

Solution: How much does the clay pot cost? \*\* The clay pot costs  $\langle \langle in1 + in0 = out0 \rangle \rangle$ .

How much does the bag of soil cost? \*\* The bag of soil costs  $\langle \langle in0 - in2 = out1 \rangle \rangle$ .

How much does it cost to plant the flowers? \*\* The cost to plant the flowers is  $\langle \langle in0 + out0 + out1 = out2 \rangle \rangle$ .

The answer is out2.

Rewrite solution: Let's think about the sub-questions we need to answer.

Q1: How much does the clay pot cost?

Q2: How much does the bag of soil cost?

Q3: How much does it cost to plant the flowers?

Let's answer each sub-question one by one.

Q1: How much does the clay pot cost? The flowers cost \$ [in0]. The clay pot costs \$ [in1] more than the flower.

So the clay pot costs  $\langle \langle in0 + in1 = out0 \rangle \rangle$ .

Q2: How much does the bag of soil cost? The flowers cost \$ [in0]. The bag of soil costs \$ [in2] less than the flower.

So the bag of soil costs  $\langle \langle in0 - in2 = out1 \rangle \rangle$ .

Q3: How much does it cost to plant the flowers? The flowers cost \$ [in0]. The clay pot costs \$ [out0].

The bag of soil costs \$ [out1]. So the cost to plant the flowers is  $\langle \langle in0 + out0 + out1 = out2 \rangle \rangle$ .

The final answer is out2.

Example 3:

Problem: From March to August, Sam made \$ [in0] doing [in1] hours of yard work. However, from September to February, Sam was only able to work for [in2] hours. If Sam is saving up to buy a video game console that costs \$ [in3] and has already spent \$ [in4] to fix his car, how many more hours does he need to work before he can buy the video game console?

Solution: How much does Sam make per hour? \*\* Sam makes  $\langle \langle in0 / in1 = out0 \rangle \rangle$  /hr. How much did Sam make from September to February? \*\* From September to February, Sam made  $\langle \langle in2 * out0 = out1 \rangle \rangle$ .

How much did Sam make from March to February? \*\* From March to February, Sam made a total of  $\langle \langle in0 + out1 = out2 \rangle \rangle$ .

How much money did Sam have after fixing his car? \*\* After fixing his car, he was left with  $\langle \langle out2 - in4 = out3 \rangle \rangle$ .

How much money does Sam need to buy the video game console? \*\* Sam needs another  $\langle \langle in3 - out3 = out4 \rangle \rangle$ .

How many more hours does Sam need to work? \*\* Sam needs to work another  $\langle \langle out4 / out0 = out5 \rangle \rangle$  hours.

The answer is out5.

Rewrite solution: Let's think about the sub-questions we need to answer.

Q1: How much does Sam make per hour?

Q2: How much did Sam make from September to February?

Q3: How much did Sam make from March to February?

Q4: How much money did Sam have after fixing his car?

Q5: How much money does Sam need to buy the video game console?

Q6: How many more hours does Sam need to work?

Let's answer each sub-question one by one.

Q1: How much does Sam make per hour? Sam made \$ [in0] doing [in1] hours of yard work.

So he makes  $\langle \langle in0 / in1 = out0 \rangle \rangle$  per hour.

Q2: How much did Sam make from September to February? From September to February, Sam worked for [in2] hours.

He makes \$ [out0] per hour. So from September to February, he made  $\langle \langle in2 * out0 = out1 \rangle \rangle$ .

Q3: How much did Sam make from March to February? From March to August, Sam made \$ [in0]. From September to February,

he made \$ [out1]. So from March to February, he made a total of  $\langle \langle in0 + out1 = out2 \rangle \rangle$ .

Q4: How much money did Sam have after fixing his car? Sam made a total of \$ [out2]. He spent \$ [in4] to fix his car.

So after fixing his car, he was left with  $\langle \langle out2 - in4 = out3 \rangle \rangle$ .

Q5: How much money does Sam need to buy the video game console? Sam was left with \$ [out3]. The video game console costs \$ [in3].

So he needs another  $\langle \langle in3 - out3 = out4 \rangle \rangle$ .

Q6: How many more hours does Sam need to work? Sam makes \$ [out0] per hour. He needs another \$ [out4].

So he needs to work another  $\langle \langle out4 / out0 = out5 \rangle \rangle$  hours.

The final answer is out5.

---

Table 11: CoA 训练数据构建的响应重写的说明和少量示例。

You are given a math word problem and solution. Please rewrite the solution by replacing the output values in angle brackets with abstract symbols. Please also remove the redundant calculations around the angle brackets. Do not add any other contents to the solution.

Example 1:

Problem: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May.

How many clips did Natalia sell altogether in April and May?

Solution: How many clips did Natalia sell in May? \*\* Natalia sold  $48/2 = << 48/2=24 >> 24$  clips in May.

How many clips did Natalia sell altogether in April and May? \*\* Natalia sold  $48+24 = << 48+24=72 >> 72$  clips altogether in April and May. The answer is 72.

Rewrite solution: How many clips did Natalia sell in May? \*\* Natalia sold  $<< 48/2=out0 >>$  clips in May.

How many clips did Natalia sell altogether in April and May? \*\* Natalia sold  $<< 48+out0=out1 >>$  clips altogether in April and May. The answer is out1.

Example 2:

Problem: The flowers cost \$ 9, the clay pot costs \$ 20 more than the flower, and the bag of soil costs

\$ 2 less than the flower. How much does it cost to plant the flowers?

Solution: How much does the clay pot cost? \*\* The clay pot costs  $\$ 20 + \$ 9 = \$ << 20+9=29 >> 29$ .

How much does the bag of soil cost? \*\* The bag of soil costs  $\$ 9 - \$ 2 = \$ << 9-2=7 >> 7$ .

How much does it cost to plant the flowers? \*\* The cost to plant the flowers is  $\$ 9 + \$ 29 + \$ 7 = \$ << 9+29+7=45 >> 45$ .

The answer is 45.

Rewrite solution: How much does the clay pot cost? \*\* The clay pot costs  $\$ << 20+9=out0 >>$ .

How much does the bag of soil cost? \*\* The bag of soil costs  $\$ << 9-2=out1 >>$ .

How much does it cost to plant the flowers? \*\* The cost to plant the flowers is  $\$ << 20+out0+out1=out2 >>$ .

The answer is out2.

Example 3:

Problem: From March to August, Sam made \$ 460 doing 23 hours of yard work. However, from September to February, Sam was only able to work for 8 hours. If Sam is saving up to buy a video game console that costs \$ 600 and has already spent \$ 340 to fix his car, how many more hours does he need to work before he can buy the video game console?

Solution: How much does Sam make per hour? \*\* Sam makes  $\$ 460 / 23 \text{ hrs} = \$ << 460/23=20 >> 20/\text{hr}$ . How much did Sam make from September to February? \*\* From September to February, Sam made  $8 \text{ hrs} \times \$ 20/\text{hr} = \$ << 8*20=160 >> 160$ .

How much did Sam make from March to February? \*\* From March to February, Sam made a total of  $\$ 460 + \$ 160 = \$ 620$ .

How much money did Sam have after fixing his car? \*\* After fixing his car, he was left with  $\$ 620 - \$ 340 = \$ << 620-340=280 >> 280$ .

How much money does Sam need to buy the video game console? \*\* Sam needs another  $\$ 600 - \$ 280 = \$ << 600-280=320 >> 320$ .

How many more hours does Sam need to work? \*\* Sam needs to work another  $\$ 320 / \$ 20/\text{hr} = << 320/20=16 >> 16$  hours.

The answer is 16.

Rewrite solution: How much does Sam make per hour? \*\* Sam makes  $\$ << 460/23=out0 >> / \text{hr}$ . How much did

Sam make from September to February? \*\* From September to February, Sam made  $\$ << 8*out0=out1 >>$ .

How much did Sam make from March to February? \*\* From March to February, Sam made a total of  $\$ << 460+out1=out2 >>$ .

How much money did Sam have after fixing his car? \*\* After fixing his car, he was left with  $\$ << out2-340=out3 >>$ .

How much money does Sam need to buy the video game console? \*\* Sam needs another  $\$ << 600-out3=out4 >>$ .

How many more hours does Sam need to work? \*\* Sam needs to work another  $<< out4/out0=out5 >>$  hours.

The answer is out5.

Example 4:

Problem: Zhang is twice as old as Li. Li is 12 years old. Zhang's brother Jung is 2 years older than Zhang. How old is Jung?

Solution: How old is Zhang? \*\* Zhang is  $2 * 12 \text{ years old} = << 2*12=24 >> 24$  years old.

How old is Jung? \*\* Jung is  $2 \text{ years} + 24 \text{ years} = << 2+24=26 >> 26$  years old.

The answer is 26.

Rewrite solution: How old is Zhang? \*\* Zhang is  $<< 2*12=out0 >>$  years old.

How old is Jung? \*\* Jung is  $<< 2+out0=out1 >>$  years old.

The answer is out1.

Example 5:

Problem: Of the 90 people on William's bus, 3/5 were Dutch. Of the 1/2 of the Dutch who were also

American, 1/3 got window seats. What's the number of Dutch Americans who sat at the windows?

Solution: How many Dutch people were on the bus? \*\* On the bus, the number of Dutch people was  $3/5$  of the total number, a total of  $3/5 * 90 = << 3/5*90=54 >> 54$  people.

How many Dutch Americans were on the bus? \*\* Out of the 54 people who were Dutch,  $1/2$  were Dutch Americans,

a total of  $1/2 * 54 = << 1/2*54=27 >> 27$  people.

How many Dutch Americans sat at the windows? \*\* If  $1/3$  of the passengers on the bus identifying as Dutch Americans sat at the windows, their number is  $1/3 * 27 = << 1/3*27=9 >> 9$

The answer is 9.

Rewrite solution: How many Dutch people were on the bus? \*\* On the bus, the number of Dutch people was  $3/5$  of the total number, a total of  $<< 3/5*90=out0 >>$  people.

How many Dutch Americans were on the bus? \*\* Out of the out0 people who were Dutch,  $1/2$  were Dutch Americans,

a total of  $<< 1/2*out0=out1 >>$  people.

How many Dutch Americans sat at the windows? \*\* If  $1/3$  of the passengers on the bus identifying as Dutch Americans sat at the windows, their number is  $<< 1/3*out1=out2 >>$

The answer is out2.