

通过问题空间映射将理解与推理解耦以进行小规模模型推理

Li Wang¹, Changhao Zhang², Zengqi Xiu¹, Kai Lu^{3,4}, Xin Yu⁵, Kui Zhang⁶, Wenjun Wu^{1,7,8*}

¹School of Artificial Intelligence, Beihang University, Beijing, China

²UCL Hawkes Institute and Department of Medical Physics and Biomedical Engineering, University College London, UK

³State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China

⁴School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

⁵School of Computer Science and Engineering, Beihang University, Beijing, China

⁶State Key Laboratory of Complex Critical Software Environment, Beihang University, Beijing, China

⁷Hangzhou International Innovation Institute, Beihang University, Hangzhou, China

⁸Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, Beijing, China

{ wangli_42, xiuzengqi, nlsdeyuxin, zhangkui, wwj09315 } @buaa.edu.cn, changhao.zhang, 24@ucl.ac.uk, lukai24@mails.ucas.ac.cn

Abstract

尽管大语言模型 (LLMs) 的推理能力最近有所进步, 提高小语言模型 (SLMs, 例如, $\leq 1.5B$) 的推理能力仍然具有挑战性。一个主要障碍在于自然语言的复杂性和多样性: 本质上等价的问题常常以多种不同的表层形式出现, 经常被冗余或令人分心的细节所掩盖。这给 SLMs 带来了双重负担: 它们首先必须从复杂的语言输入中提取核心问题, 然后基于这种理解执行推理。由此产生的巨大且嘈杂的问题空间妨碍了优化, 特别是对于能力有限的模型而言。为了解决这个问题, 我们提出了一种新的框架, 通过将自然语言问题映射到一个规范的问题空间 (一个语义上简化但富有表现力的领域), 将理解与推理解耦。这使得 SLMs 可以专注于标准化输入上的推理, 避免语言的多样性。在此框架内, 我们引入了 DURIT (通过迭代训练将理解与推理解耦), 这是一种三步算法, 迭代地: (1) 通过强化学习映射自然语言问题, (2) 通过自蒸馏对齐推理轨迹, 以及 (3) 在问题空间中训练推理策略。在整个过程中, 映射器和推理器在交替循环中共同训练。实验表明, DURIT 显著提高了 SLMs 在域内和跨域数学和逻辑推理任务上的性能。除了提高推理能力, DURIT 还提高了推理的鲁棒性, 验证了将理解与推理解耦是一种加强 SLMs 的有效策略。

引言

大型语言模型 (LLMs) (Yang et al. 2025a) 在推理能力方面展示了显著的进步 (Bi et al. 2025; Luo et al. 2025a; Wen et al. 2024)。然而, 目前大多数研究主要集中在相对较大的模型 (Guan et al. 2025; Li 2025; Shen et al. 2025) 上, 而小型语言模型 (SLMs, 例如, $\leq 1.5B$) 的推理能力尚未得到充分探索。尽管其能力有限, SLMs 在边缘部署场景和对时延敏感的应用中由于其紧凑的尺寸和快速推理能力而具有重要的潜力 (Sun et al. 2020; Xu et al. 2024)。尽管如此, 由于其参数容量有限, 增强其推理能力仍然是一个显著的挑战。

最近改善大语言模型 (LLM) 推理能力的努力集中在通过增强思维链 (CoT) (Wei et al. 2022), 使用诸

如基于搜索的推理 (Li 2025; Guan et al. 2025) 和错误纠正 (Ma et al. 2025; Yang et al. 2025b) 等技术。然而, 由于容量有限, 小语言模型 (SLMs) 难以生成复杂的推理痕迹, 使得这些方法效果欠佳。知识蒸馏 (KD) 是常用策略, 通过教师生成的痕迹 (例如, CoT) 或令牌级监管从较大的教师 LLM 中转移推理能力来改善 SLMs。然而, 教师和学生模型之间的不匹配在数据和教师选择上构成挑战。KD 严重依赖于多样的高质量数据 (Gu et al. 2025): 过于简单的例子可能导致对浅层模式的过拟合 (Shumailov et al. 2024), 而复杂的 CoT 痕迹可能超出 SLMs 的能力并阻碍学习 (Li et al. 2025)。教师-学生不匹配可能进一步降低性能 (Cho and Hariharan 2019; Chen et al. 2025), 突显了将高质量推理蒸馏到 SLMs 中的挑战。

与知识蒸馏 (KD) 不同, 强化学习 (RL) 使模型能够自主探索解决方案, 通常产生更强的泛化性能 (Chu et al. 2025; Huan et al. 2025)。DeepSeek-R1 的强劲表现 (Shao et al. 2024) 进一步强调了 RL 在增强大型语言模型 (LLM) 推理能力方面的潜力。然而, SLM 面临独特的挑战: 它们必须理解自然语言问题的语义复杂性, 并在能力有限的情况下执行多步骤推理。自然语言引起的庞大状态空间严重限制了 RL 的效率。即便是问题措辞上的表面变化也可能误导模型 (Mirzadeh et al. 2024; Liu et al. 2025a), 它们常常依赖于肤浅的启发式算法, 而非真正的理解。这表明模型可能无法抓住问题的本质, 容易被表层的语言变化分散注意力。相反, 人类一旦掌握问题的本质, 就能轻松地在不同的表面形式之间进行泛化。这种对比引出了一个关键问题: 模型如何才能获得这种基本的理解, 并以类似人类的方式进行泛化? 我们通过提出一种新的视角来解决这个问题——与其直接在高维、嘈杂的自然语言空间中进行推理, 我们首先将问题映射到一个低维的标准化问题空间中。这种转换减少了虚假的可变性, 通过将本质上相似的问题聚集到更具代表性和规范的形式中来限制搜索空间。因此, 它压缩了状态空间, 突出了问题的本质, 减轻了表层语言理解的负担, 从而提高了探索效率。重要的是,

*Corresponding author.

我们的方法与现有的基于 Chain-of-Thought (CoT) 的方法相互独立：问题空间转换充当前端标准化层，使更有效和鲁棒的下游推理成为可能。

在本文中，我们提出了一个通用框架，将自然语言问题映射到一个更抽象、低维度和语义规范化的问题空间，有效地减少了原始问题空间的复杂性。在这个空间内，模型可以更高效地学习和推理。我们通过一个新颖的三步交替训练算法具体化这个框架：(1) 使用强化学习和隐式模板训练问题空间映射器，将自然语言问题映射到标准化、低维度的形式；(2) 自蒸馏将这种映射能力转移到标准语言模型 (SLM)；(3) 通过强化学习训练推理模型在问题空间内操作。映射器和 SLM 以交替方式优化，推动推理能力的迭代改进。为了验证 DURIT 的有效性，我们使用 LLaMA (Grattafiori et al. 2024) 和 Qwen (Yang et al. 2025a) 系列的模型进行了全面的实证研究，参数规模从 0.5B 到 1.5B。即使仅在 GSM8K (Cobbe et al. 2021) 数学推理数据集上进行训练，DURIT 在一系列域内和域外数据集上，包括那些专注于数学和逻辑推理的数据集，取得了显著的提升，并展示了强大的泛化能力。与传统基于 CoT 的方法不同，DURIT 通过减少问题空间的复杂性引入了一种新的范式来改进推理。我们的主要贡献如下：

- 我们提出一个通用框架，将自然语言问题映射到一个标准化的低维空间，减少有效状态空间，改善探索和样本效率。
- 我们引入了一种三步交替训练算法——DURIT，该算法将理解与推理分离开来，并通过问题空间映射器和推理模型的迭代共同训练，逐步增强 SLM 的推理能力和稳健性。
- 实验表明，DURIT 在数学和逻辑推理任务上，无论是在域内还是域外设置中，均能提供显著的性能提升，即使是在训练数据有限的情况下。除了提高准确性，DURIT 还增强了推理的鲁棒性，这表明其对问题内在本质的理解更深刻，并且在不同形式的表述上具有更好的泛化能力。

相关工作

提示优化

提示优化通过改进 LLM 输入来提高测试时的性能。一些方法使用释义 (Yuan, Neubig, and Liu 2021; Deng et al. 2024)，而其他方法则应用 RL 以更有效地探索提示格式 (Deng et al. 2022; Zhang et al. 2022)。PReWrite (Kong et al. 2024) 通过 PPO (Schulman et al. 2017) 使用响应准确性作为奖励来训练 LLM，但由于基于 LLM 的提示生成而导致高推理成本。AbstRaL (Gao et al. 2025) 通过将问题抽象为符号形式并将推理委托给外部工具链来提高推理的鲁棒性。与之前的工作不同，我们的目标是消除对外部工具的依赖，并在自然语言空间内完全实现推理。为此，我们通过 RL 训练一个问题空间映射器，并将其转换为提炼到 SLM 中，从而提高推理性能和鲁棒性。

知识蒸馏

知识蒸馏将知识从大型教师模型转移到较小的学生模型，并可以分为离线和在线范式。离线知识蒸馏使用教师生成的数据。Std-CoT (Magister et al. 2023) 在 CoT 示例上微调学生模型，而 NesycD (Liao et al.

2025) 蒸馏一般能力并结合外部知识。在线知识蒸馏要求教师在推理过程中提供逐词级别的监督。Vanilla-KD (Muralidharan et al. 2024) 蒸馏隐藏状态和输出概率，BOND (Sessa et al. 2024) 采用基于模型最佳响应的自蒸馏，STaR (Zelikman et al. 2024) 通过正确的最终答案微调自生成的 CoT 轨迹以提高性能。与以往的研究不同，我们的方法专注于自蒸馏以在模型内部转移知识，使模型能够将其已学习的能力泛化到不熟悉的任务中。

大型语言模型推理的强化学习

强化学习已被证明在增强大型语言模型的能力方面是有效的。从人类反馈中进行的强化学习 (RLHF) (Bai et al. 2022; Ouyang et al. 2022) 现在是一种将模型输出与人类偏好对齐的标准方法。最近的工作，如 DeepSeek-R1 (Shao et al. 2024) 和 Kimi K1.5 (Team et al. 2025) 显示，像 GRPO 这样的技术可以显著提升推理能力，突显了使用可验证奖励 (RLVR) 的 RL 的潜力。在此基础上，许多研究提出了进一步的改进 (Yu et al. 2025; Team et al. 2025; Liu et al. 2025b)。然而，自然语言的广阔和复杂的状态空间对高效探索构成了主要挑战。为了解决这个问题，我们提出了一种问题空间映射，该映射将原始空间投射到一个低维、更有条理的表示中，从而提高 RL 的效率。

自然语言的内在复杂性对语言模型 (SLMs) 提出了双重挑战：理解微妙的语义差异和进行推理，这两者都受到模型能力限制的约束。为了解决这一问题，我们提出了一个将理解与推理相分离的通用框架。其核心是问题空间的概念——一种标准化的、低维度的表示方法，它在保留基本语义的同时抽象掉表面可变性。通过将基本相似的问题映射到相近的表示，问题空间减少了输入的复杂性，并为下游推理提供了更具可解释性和学习效率的界面。如附录 F 中所示，标准化复杂问题减轻了误解，并提高了推理的准确性。形式上，设 $\mathcal{Q}$ 表示自然语言问题的空间，设 $\mathcal{P} \subset \mathcal{L}$ 为从自然语言空间 $\mathcal{L}$ 中提取的有限集合的规范形式。我们定义一个映射 $f: \mathcal{Q} \rightarrow \mathcal{P}$ ，将每个问题 $q \in \mathcal{Q}$ 分配到一个规范表示 $p = f(q) \in \mathcal{P}$ 。 $\mathcal{P}$ 和 f 的构建受以下目标的指导：其中，参数为 θ 的 SLM 在映射输入上的准确率为 $\text{Acc}(f(q); \theta)$ ，而 $\mathcal{S}$ 是一组基本相似的问题对。这些约束条件鼓励状态压缩并在问题空间内实施标准化结构。

基于此公式，我们提出了一个统一的框架 (图 1)，该框架利用专门的问题空间映射器将自然语言问题投射到一种标准化的表示中。通过聚类基本相似的问题，此映射减少了探索空间，并在 SLM 训练过程中提高了样本和探索效率。随着模型在此空间中的推进，其解决更复杂问题的能力增强，逐渐改变潜在的问题分布。为了适应这种变化，我们的框架采用了一个迭代训练范式，在更新问题空间映射器和完善推理模型之间交替进行，使它们能够共同进化。减少问题空间的维度增强了探索能力并加速了收敛。为了分析这一效果，我们考虑了一种简化的上置信界设定，并通过以下定理显示，后悔界随着问题空间维度的减少而降低。

该证明详见附录 A。虽然 UCB 设置简化了 LLM 的情境，定理提供了关于如何通过降低问题空间的维度来改善探索的重要见解。具体而言，从 $\mathcal{Q}$ 到 $\mathcal{P}$ 的映射以 $\alpha = |\mathcal{P}|/|\mathcal{Q}| < 1$ 的比例压缩了空间，从而将遗憾界收紧了 $\sqrt{\alpha}$ 。这一结果支持了我们的核心动机：利用标准化的抽象可以使 SLM 的推理训练更为高效。

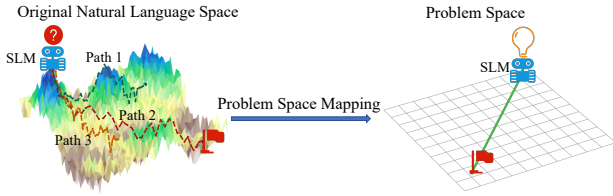


Figure 1: 我们的框架示例用于将理解与推理解耦。原始自然语言空间复杂且高维，使得探索变得困难；映射到标准化的低维问题空间可压缩状态空间，并促进更高效的探索。

方法

我们提出了通过迭代训练的理解与推理解耦 (DURIT)，设计旨在通过将问题理解与推理解耦来增强 SLMs 的推理能力。如图 2 所示，DURIT 由三个交替步骤组成：(1) 问题映射器训练：问题映射器 M 通过 RL 训练，在隐式模板的指导下，将原始自然语言问题映射到问题空间。(2) 自我蒸馏：转换能力通过自我蒸馏内化到推理 SLM R ，使其能够在推理时直接处理复杂问题，而无需依赖外部映射器 M 。(3) RL 训练：SLM R 进一步使用 RL 优化以提高其推理性能。这三个步骤重复交替进行，通过理解和推理的交替阶段逐步增强模型的推理能力。完整的伪代码在附录 C ?? 中提供。

步骤 I: 问题空间映射器训练

为了将理解与推理解耦，一个问题空间映射器 M 被实例化为一个大型语言模型 (LLM)，用于将自然语言问题映射到标准化的问题空间。虽然显式模板强制标准化，但它们劳动密集并可能阻碍小型语言模型 (SLM) 的理解。为了平衡标准化和灵活性，提出了一个隐式模板机制，使用一个代码簿柔性地引导 M 的输出风格。此映射旨在 1) 提高 SLM 对问题的理解能力，以及 2) 降低问题空间的复杂性。为了促进这一点，我们基于基本问题相似性使用 k 最近邻 (kNN) 对通过模型 M 编码的每个问题 Q_i 的表示 z_i 、描述及其答案进行聚类。由于没有真实标签存在，我们采用 GRPO (Shao et al. 2024) 来基于映射问题 Q'_i 的平均正确性 r_{acc} 对冻结的 SLM 的响应进行优化 M 。为了防止 M 直接解决问题，如果 Q'_i 包含在 Q_i 中不存在的解决方案特定术语 (例如，“答案值”这样的关键词)，我们应用作弊惩罚 r_{cheating} 。总奖励为：

$$r_i = r_{\text{acc}} + r_{\text{cheating}}. \quad (1)$$

然而，仅靠强化学习无法充分执行标准化。为了简化问题空间，引入了基于簇标签 t_i 的隐式模板。具体地，构建了一个包含 n 个隐式模板令牌 $\{T_1, \dots, T_n\}$ 及对应查询键 $\{k_1, \dots, k_n\}$ 的码本 C ，二者均随机初始化。在训练过程中，对于每个问题 Q_i ，选择模板令牌 T_{t_i} 并与原始输入 $x_i = [Q_i; T_{t_i}]$ 一起连接，以指导 M 生成映射后的问题 Q'_i 。为了鼓励 Q'_i 与其指定模板之间的对齐，我们基于 InfoNCE (He et al. 2020) 目标定义了一种模板相似性损失：

$$\mathcal{L}_{\text{template-sim}} = -\log \frac{\exp\left(\frac{\langle z_i, T_{t_i} \rangle}{\tau}\right)}{\sum_{j=1}^n \exp\left(\frac{\langle z_i, T_{t_j} \rangle}{\tau}\right)}, \quad (2)$$

其中 z_i 是映射问题 Q'_i 的归一化表示， τ 是温度超参数。在推断时，当 t_i 不可用时，通过输入问题嵌入 q_i (通过平均词嵌入来近似) 与学习的模板查询键之间的余弦相似性来选择最佳匹配的隐式模板。引入了一种关键相似性损失以促进键学习：

$$\mathcal{L}_{\text{key-sim}} = -\log \frac{\exp\left(\frac{\langle q_i, k_i \rangle}{\tau}\right)}{\sum_{j=1}^n \exp\left(\frac{\langle q_i, k_j \rangle}{\tau}\right)}, \quad (3)$$

从 q_i 的梯度被分离以防止对 M 训练的干扰，并且仅更新模板键。总体损失函数联合优化了映射策略和基于模板的约束：

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pg}} + \alpha_1 \mathcal{L}_{\text{key-sim}} + \alpha_2 \mathcal{L}_{\text{template-sim}}, \quad (4)$$

其中 $\mathcal{L}_{\text{pg}}$ 表示来自 GRPO 的策略梯度损失，而 α_1 、 α_2 是平衡不同损失的超参数。

在训练问题空间映射器 M 后，其转换能力通过自我蒸馏内化到 SLM 中。具体来说， M 将原始数据集 $\mathcal{D}_0$ 转换为规范化形式 $\mathcal{D}_1 = \{Q'_i = M(Q_i) \mid Q_i \in \mathcal{D}_0\}$ ，其中映射的问题被设计为促进 SLM 更容易推理。然后，我们使用 SLM 在每个 $Q'_i \in \mathcal{D}_1$ 上采样 N 的响应，并构建一个过滤数据集： $\mathcal{D}_2 = \{(Q_i, Q'_i, y_i) \mid Q_i \in \mathcal{D}_0, y_i = R(Q'_i), \text{answer}(y_i) = \text{True}\}$ ，其中 y_i 表示模型的响应， $\text{answer}(y_i)$ 评估其正确性。核心思想是鼓励 SLM 在 Q_i 上复制它在 Q'_i 上表现出的推理行为。为实现这一点，我们将 (Q'_i, y_i) 视为教师对，而 (Q_i, y_i) 视为相应的学生对。SLM 通过监督微调 L_{SFT} 和 KD L_{KD} 的组合损失进行训练：

$$p(x^k) = \frac{\exp(x^k/\tau)}{\sum_{j=1}^{|V|} \exp(x^j/\tau)}, \quad (5)$$

$$\mathcal{L}_i = \frac{1}{l} \sum_{k=1}^l [(1-\lambda)(-\log p_s(x_i^k)) + \lambda \text{KL}(p_t(x_i^k) \parallel p_s(x_i^k))], \quad (6)$$

其中 l 是序列长度， x_i^k 是 y_i 的 k -个令牌， p_s 和 p_t 分别是学生和教师对于前缀输入 Q_i 和 Q'_i 的 softmax 输出。参数 λ 在损失之间达到平衡。此设置允许学生在推理时内部化 M 而不访问它。

将转换能力蒸馏到 SLM 之后，该模型通过 RL 进行进一步训练，以便在原始问题空间中直接探索和推理，利用其内在化理解。具体而言，我们使用 GRPO 算法对原始训练数据集 $\mathcal{D}_0$ 上的 SLM 进行微调，以答案的正确性作为奖励信号。随着推理模型的改进，其解释和泛化能力也在演变，这可能会改变问题空间的最佳结构。为了适应这种变化，问题空间映射器 M 和推理模型 R 在一个迭代循环中接受训练，从而实现问题空间的持续完善和推理能力的逐步提升。

我们在所有实验中仅在 (Cobbe et al. 2021) 数据集上训练模型。为了评估，我们考虑域内 (IND) 和域外 (OOD) 设置。具体来说，我们使用 GSM8K-Platinum (Vendrow et al. 2025) 进行 IND 评估，并为了评估分布转移下的数学推理泛化，我们采用 MAWPS (Koncel-Kedziorski et al. 2016)、SVAMP (Patel, Bhattamishra, and Goyal 2021)、MATH500 (Hendrycks et al. 2021) 和 GAOKAO (Zhang et al. 2024) 数据集。此外，为了评估更广泛的推理能力，我们使用 LogiQA (Liu et al.

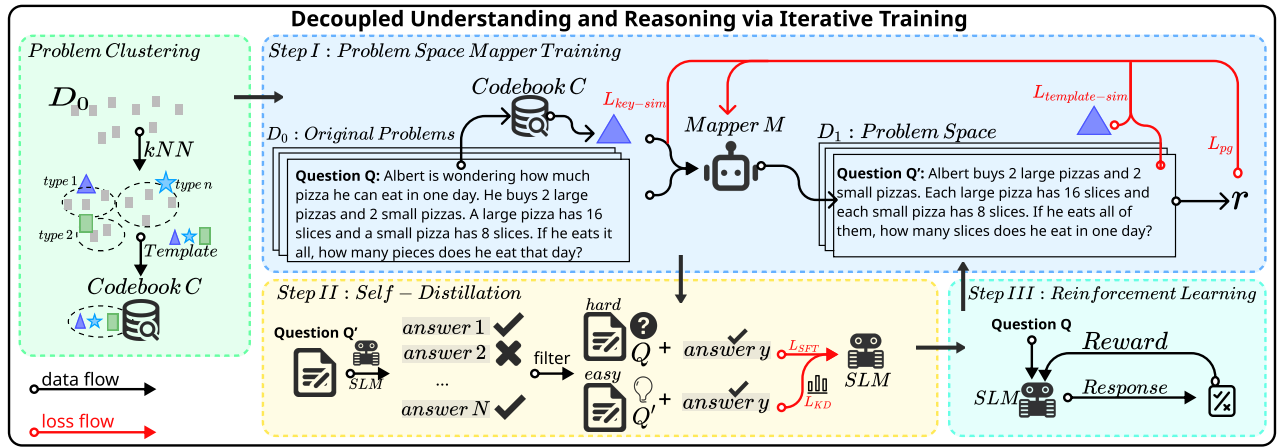


Figure 2: DURIT 方法的框架。经过基于 KNN 的聚类后，DURIT (1) 通过隐式映射压缩问题，(2) 将其提炼为 SLM，并且 (3) 通过具有交替共同训练的强化学习进行优化。

2020) 数据集对模型进行逻辑推理评估。这个全面的评估设置使我们能够系统地分析 DURIT 对 SLMs 跨多个领域总体推理能力的影响。

我们将我们的方法与四个类别中的几个代表性基线进行比较：(1) CoT 蒸馏：包括 Std-CoT (Magister et al. 2023)，并且类似于 DURIT 的第二阶段，我们额外引入了 STaR (Zelikman et al. 2024)，在每个问题中，我们使用 SLM 采样 N 个 CoT 响应，并根据最终答案过滤出正确的用于微调；(2) 提示优化：PRewrite (Kong et al. 2024)，其同样采用 RL 来优化提示；(3) 基于 RL 的方法：GRPO (Shao et al. 2024)；(4) 知识蒸馏：我们采用 Vanilla-KD (Muralidharan et al. 2024)，这需要教师 LM 的在线推理来获取令牌级输出分布。在我们的设置中，映射器模型作为教师。与之前的工作 (Sheng, Li, and Zeng 2025) 一致，我们采用答案准确性作为主要评估指标。更多基线的详细信息可在附录 B.3 ?? 中找到。

为了评估 DURIT 的泛化能力，我们测试了不同的基础模型，包括最近强指令跟随和推理导向的模型，如 Qwen2.5-0.5B-Instruct (Yang et al. 2025a) 和 Llama3.2-1B-Instruct (Grattafiori et al. 2024)。对于映射模型，我们在 Qwen 系列中使用 Qwen2.5-3B-Instruct，在 Llama 系列中使用 Llama3.2-3B-Instruct，以确保架构的一致性。代码簿包含 32 个隐式模板，模板查询键的损失系数为 $\alpha_1 = 1e-3$ ，令牌的损失系数为 $\alpha_2 = 1e-2$ 。训练分为三个步骤：步骤 I 运行 1 轮，步骤 II 运行 5 轮，步骤 III 运行 3 轮。所有实验均在 2 个 40GB 内存的 A100 GPU 上进行。对于推理，我们采用不使用 vLLM (Kwon et al. 2023) 加速的贪婪解码。附加的实现细节可以在附录 B.3 ?? 中找到。我们还在 Qwen2.5-1.5B-Instruct 模型上评估了我们的方法；详细结果和示例见附录 E.1 ?? 和 F。

如表 1 所示，DURIT 在 IND 和 OOD 基准上均优于所有基线。显著的是，即使仅在 GSM8K 数据集上训练，DURIT 也能在所有数据集上持续提供显著的性能提升。只需一次迭代，它在 Qwen2.5-0.5B-Instruct 和 Llama3.2-1B-Instruct 上分别实现了比最强基线方法高出 2.06 % 和 2.35 % 的平均准确率提升。重要的是，DURIT 在不依赖外部大模型进行 CoT 监督的情

况下实现了这些提升。相反，它充分利用模型自身的推理能力来探索、适应并转移先验知识。值得注意的是，DURIT 甚至胜过依赖更强教师模型如 DeepSeek-R1 的蒸馏方法。因为它完全在模型内部运行，DURIT 避免了额外的 API 成本和基础设施开销，提供了广泛的适用性和高成本效率。通过第二次迭代训练，DURIT 的推理能力进一步增强：即使继续使用 GSM8K 数据集，它在 Qwen2.5-0.5B-Instruct 上取得了 0.36 % 的平均准确率提升，在 Llama3.2-1B-Instruct 上取得了 0.69 % 的提升。使用不同数据集进行第二次迭代时会观察到更大的改进（详见后续部分 ??），这表明 DURIT 在各领域中的强泛化能力及其在减少推理获取的认知负荷方面的有效性。

通过将自然语言问题投射到一个更加本质且低维度的问题空间中，DURIT 专注于问题的基本语义。这种抽象减少了表面层次表达的变化，抑制了虚假或无关的提示，从而增强了推理的鲁棒性。为了验证这一说法，我们使用 Qwen2.5-0.5B-Instruct 和 LLaMA3.2-1B-Instruct 在 GSM-Symbolic 基准 (Mirzadeh et al. 2024) 上进行评估。由于原始数据集仅包含 100 个示例并且表现出高度方差，我们遵循 (Gao et al. 2025; Liu et al. 2025a) 并采用平均准确率的相对下降作为鲁棒性度量。结果报告在表格 2 中。DURIT 在所有方法中几乎达到了最小的准确率相对下降，这表明其推理增益伴随着显著增强的鲁棒性。其他模型规模的结果在附录 E.1 ?? 中提供。

为了评估不同迭代训练数据集对 DURIT 的影响，我们进行了第二次迭代，使用 GSM8K、MATH 和过滤版的 DeepScaleR，继第一次基于 GSM8K 的迭代之后。如图 ?? 所示，当训练数据不同于第一次迭代所用的数据时，第二次迭代带来了更显著的改进。这一发现突出了 DURIT 在解耦理解与推理方面的优势，使其能够有效利用互补的数据源。此外，使用更具多样性的数据集进行训练始终能提高整体性能并增强模型的推理能力。数据集的详细描述见附录 B.2。

为了评估不同模型作为映射器对 DURIT 性能的影响，我们将推理 SLM 固定为 Qwen2.5-0.5B-Instruct，并使用各种映射器模型执行一次 DURIT 更新。具体而言，我们尝试了 Qwen2.5-3B/1.5B/0.5B-Instruct 和

Methods	In-Domain	Out-of-Domain					Average
	gsm8k-platinum	MAWPS	SVAMP	MATH500	GAOKAO	LogiQA	
# Qwen2.5-0.5B-Instruct based							
Base (Yang et al. 2025a)	45.74	54.23	54.67	27.80	18.55	14.44	35.91
CoT-Dis (Magister et al. 2023)	44.67	55.77	58.33	18.80	12.90	30.41	36.81
STaR (Zelikman et al. 2024)	51.86	57.88	61.67	29.60	18.55	23.50	40.51
GRPO (Shao et al. 2024)	51.03	58.08	61.00	27.40	21.77	22.73	40.34
PRewrite (Kong et al. 2024)	47.23	56.73	57.00	29.80	19.35	23.96	39.01
Vanilla-KD (Muralidharan et al. 2024)	49.30	57.69	61.67	30.4	23.39	20.74	40.53
DURIT (ours, iter=1)	53.68	60.19	62.67	31.00	23.39	24.58	42.59
DURIT (ours, iter=2)	53.10	60.38	63.00	32.80	22.58	25.81	42.95
# Llama3.2-1B-Instruct based							
Base (Grattafiori et al. 2024)	30.52	5.77	20.67	22.60	12.10	1.54	15.53
CoT-Dis (Magister et al. 2023)	48.06	56.92	57.67	24.60	12.90	21.81	36.99
STaR (Zelikman et al. 2024)	36.31	52.50	54.33	20.00	16.94	8.45	31.42
GRPO (Shao et al. 2024)	48.39	59.23	57.67	26.40	16.13	4.45	35.38
PRewrite (Kong et al. 2024)	35.81	41.34	46.00	18.80	12.10	3.53	26.26
Vanilla-KD (Muralidharan et al. 2024)	42.35	64.23	62.67	22.40	16.13	7.99	35.96
DURIT (ours, iter=1)	50.37	59.62	64.33	26.00	14.52	21.20	39.34
DURIT (ours, iter=2)	52.36	62.31	66.00	27.60	12.10	19.82	40.03

Table 1: Qwen2.5-0.5B-Instruct 和 Llama3.2-1B-Instruct 模型在六个代表性基准测试中采用不同方法的性能 (%)。加粗和 underline 分别表示最佳和次佳结果。

Method	Qwen-0.5B			Llama-1B		
	Orig	Symb	$\Delta\%$	Orig	Symb	$\Delta\%$
Base	46.0	41.6	-9.6	21.0	16.0	-23.7
CoT-Dis	47.0	40.6	-13.7	51.0	38.3	-24.9
STaR	51.0	41.0	-19.7	33.0	27.1	-17.8
GRPO	50.0	42.9	-14.3	44.0	35.8	-18.6
PRewrite	48.0	42.0	-12.0	39.0	21.9	-43.8
Vanilla-KD	51.0	42.2	-17.2	42.0	33.4	-20.5
DURIT	48.0	42.6	<u>-11.3</u>	44.0	40.8	-7.2

Table 2: 在 Qwen2.5-0.5B-Instruct 和 Llama3.2-1B-Instruct 上不同方法的比较。DURIT 以单次迭代进行训练。Orig: 原始测试集; Symb: gsm-symbolic; $\Delta\%$: 从 Orig 到 Symb 的相对性能下降。每组结果中加粗和 underline 表示第一和第二最佳结果。

Llama3.2-3B-Instruct, 以评估模型规模和模型家族对 DURIT 的影响。鉴于 Qwen2.5-0.5B-Instruct 相对较弱的指令跟随能力, 我们用 Qwen2.5-3B-Instruct 生成的 200 个映射器数据对其进行热启动, 以改善其初始对齐行为。结果显示在图 ?? 中, 来自同一模型家族的映射器通常表现优于其他家族的映射器, 并且随着映射器模型大小的增加, 性能往往会略有改善。然而, 各映射器的总体性能差异仍然很小, 这表明 DURIT 对映射器的选择具有鲁棒性。即使使用像 Qwen2.5-0.5B-Instruct 这样轻量级的映射器, DURIT 也能在不依赖外部较大模型的情况下实现强大性能。

不同映射提示下的性能

为了评估映射器提示设计对 DURIT 性能的影响, 我们尝试了三种不同的提示公式 (见附录 B.3 ??), 使用 Qwen2.5-3B-Instruct 作为映射器, Qwen2.5-0.5B-Instruct 作为推理 SLM。每个配置在相同条件下进行了一个迭代的训练。如图 ?? 所示, 性能在不同提示之间略有变化: 更明确和标准化的提示往往能产生更好的结果。然而, 所有的提示变体都带来了强劲的性能, 表明 DURIT 在映射器提示设计的差异上具有鲁棒性。

我们使用 Qwen2.5-0.5B-Instruct 模型进行消融研究, 以评估 DURIT 中每个组件的贡献。对于步骤 I, 我们通过用 PRewrite (w/o tem) 替换映射器来评估隐式模板约束的影响, 同时保持 DURIT 的步骤 II 和步骤 III 中的后续过程不变。对于步骤 II, 我们通过去除自蒸馏并仅保留 SFT 损失 (w/o sd) 以及去除 sft 损失 (w/o sft) 来检验 SFT 的必要性。对于步骤 III, 我们研究去除 GRPO 训练的效果 (w/o grpo)。如表格 3 所示, 消融任何单一组件都会导致性能下降。w/o tem 破坏了问题空间的标准化, 导致表现不够紧凑, 以及较低的探索效率。w/o sd 对域内性能影响较小, 但显著削弱了域外泛化能力, 强调了自蒸馏在降低理解负担和增强鲁棒性方面的作用。w/o sft 可能对模型固有的推理机制产生过度破坏, 同时使其受限于映射问题中的偏见或不正确的推理模式, 可能导致性能进一步下降。最后, w/o grpo 始终降低准确性, 证实了在自蒸馏后强化推理的必要性。

为了可视化映射问题在 SLM 中的表示, 我们分析了 Qwen2.5-0.5B-Instruct 在 GSM8K-Platinum 上的最终隐藏层表示, 使用原始输入及其通过 PRewrite 和 DURIT 生成的映射版本。我们使用平均 k 近邻距离来

Variant	GSM	MW	SV	MT	GK	LQ	Avg.
DURIT	53.68	60.19	62.67	31.00	23.39	24.58	42.59
w/o tem	53.02	59.04	60.00	31.40	20.97	23.04	41.25
w/o sd	53.52	60.58	58.33	30.80	21.77	24.88	41.65
w/o sft	51.20	57.31	61.67	28.60	19.35	22.73	40.14
w/o grpo	49.30	57.69	61.67	30.40	23.39	20.74	40.53

Table 3: 对 Qwen2.5-0.5B-Instruct 在六个基准数据集上的消融研究。我们报告了在 GSM8K (简称 GSM)、MAWPS (MW)、SVAMP (SV)、MATH500 (MT)、GAOKAO (GK) 和 LogiQA (LQ) 上的准确率 (%)。所有变体都经过单次 DURIT 迭代训练。

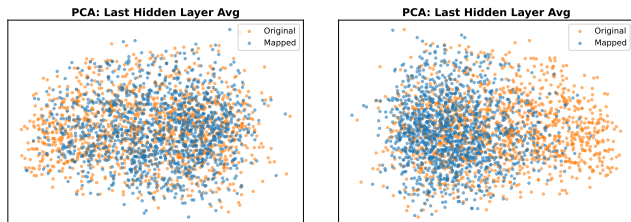


Figure 3:

(a) PRewrite 主成分分析

Figure 4:

(b) DURIT 主成分分析

Figure 5: 用 Qwen2.5-0.5B-Instruct 对不同方法的隐藏表示进行 PCA 可视化。

量化这些表示的局部紧凑性，如表 4 所报导的。此外，我们应用主成分分析 (PCA) 将高维隐藏状态投影到二维进行可视化，如图 5 所示。与原始和 PRewrite 映射的输入相比，DURIT 映射的输入在嵌入空间中形成了显著更紧凑的聚类。这表明 DURIT 映射有助于去除冗余或不相关的语言变异，有效地降低了问题空间的维度。因此，模型可能更好地捕捉到问题的本质，可能导致更有效的学习。

在这项工作中，我们提出了一个通用问题空间映射框架，并在此基础上实例化了一个具体算法 DURIT。DURIT 由三个关键步骤组成：(1) 通过增强学习和隐式模板指导训练的问题空间映射器，(2) 将映射能力内化到 SLM 中的自我蒸馏，以及 (3) 在缩减的问题空间内优化 SLM 的推理能力。通过交替训练映射器和 SLM，DURIT 实现了推理能力和鲁棒性的迭代改进。实验证明，DURIT 始终优于微调的基线，无论是域内还是域外的推理任务，表现出显著的改进，同时也具有增强的鲁棒性。

致谢

该工作得到了中国国家自然科学基金 (项目编号 62441617) 和北京未来区块链与隐私计算高精尖创新中心的部分支持。

References

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47: 235–256.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen,

Input	Original	PRewrite	DURIT
5NN Distance	75.16	73.68	68.59

Table 4: 在不同输入问题下，Qwen2.5-0.5B-Instruct 最终隐藏状态的平均 5-NN 距离。更低的值表示更紧密的局部聚类。

A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bi, Z.; Han, K.; Liu, C.; Tang, Y.; and Wang, Y. 2025. Forest-of-Thought: Scaling Test-Time Compute for Enhancing LLM Reasoning. *arXiv:2412.09078*.

Bubeck, S.; Cesa-Bianchi, N.; et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1): 1–122.

Chen, X.; Sun, Z.; Guo, W.; Zhang, M.; Chen, Y.; Sun, Y.; Su, H.; Pan, Y.; Klakow, D.; Li, W.; and Shen, X. 2025. Unveiling the Key Factors for Distilling Chain-of-Thought Reasoning. *arXiv:2502.18001*.

Cho, J. H.; and Hariharan, B. 2019. On the Efficacy of Knowledge Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Chu, T.; Zhai, Y.; Yang, J.; Tong, S.; Xie, S.; Schuurmans, D.; Le, Q. V.; Levine, S.; and Ma, Y. 2025. SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training. *arXiv:2501.17161*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv:2110.14168*.

Deng, M.; Wang, J.; Hsieh, C.-P.; Wang, Y.; Guo, H.; Shu, T.; Song, M.; Xing, E. P.; and Hu, Z. 2022. RL-Prompt: Optimizing Discrete Text Prompts with Reinforcement Learning. *arXiv:2205.12548*.

Deng, Y.; Zhang, W.; Chen, Z.; and Gu, Q. 2024. Rephrase and Respond: Let Large Language Models Ask Better Questions for Themselves. *arXiv:2311.04205*.

Gao, S.; Bosselut, A.; Bengio, S.; and Abbe, E. 2025. AbstrL: Augmenting LLMs’ Reasoning by Reinforcing Abstract Thinking. *arXiv:2506.07751*.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Gu, Y.; Zhou, H.; Meng, F.; Zhou, J.; and Huang, M. 2025. MiniPLM: Knowledge Distillation for Pre-Training Language Models. *arXiv:2410.17215*.

Guan, X.; Zhang, L. L.; Liu, Y.; Shang, N.; Sun, Y.; Zhu, Y.; Yang, F.; and Yang, M. 2025. rStar-Math:

- Small LLMs Can Master Math Reasoning with Self-Evolved Deep Thinking. arXiv:2501.04519.
- He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
- Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. arXiv:2103.03874.
- Huan, M.; Li, Y.; Zheng, T.; Xu, X.; Kim, S.; Du, M.; Poovendran, R.; Neubig, G.; and Yue, X. 2025. Does Math Reasoning Improve General LLM Capabilities? Understanding Transferability of LLM Reasoning. arXiv:2507.00432.
- Koncel-Kedziorski, R.; Roy, S.; Amini, A.; Kushman, N.; and Hajishirzi, H. 2016. MAWPS: A Math Word Problem Repository. In Knight, K.; Nenkova, A.; and Rambow, O., eds., Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1152–1157. San Diego, California: Association for Computational Linguistics.
- Kong, W.; Hombaiah, S.; Zhang, M.; Mei, Q.; and Bendersky, M. 2024. PRewrite: Prompt Rewriting with Reinforcement Learning. 594–601. Bangkok, Thailand: Association for Computational Linguistics.
- Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J.; Zhang, H.; and Stoica, I. 2023. Efficient memory management for large language model serving with pagedattention. In Proceedings of the 29th Symposium on Operating Systems Principles, 611–626.
- Li, Y. 2025. Policy Guided Tree Search for Enhanced LLM Reasoning. arXiv:2502.06813.
- Li, Y.; Yue, X.; Xu, Z.; Jiang, F.; Niu, L.; Lin, B. Y.; Ramasubramanian, B.; and Poovendran, R. 2025. Small Models Struggle to Learn from Strong Reasoners. arXiv:2502.12143.
- Liao, H.; He, S.; Xu, Y.; Zhang, Y.; Liu, K.; and Zhao, J. 2025. Neural-Symbolic Collaborative Distillation: Advancing Small Language Models for Complex Reasoning Tasks. Proceedings of the AAAI Conference on Artificial Intelligence, 39(23): 24567–24575.
- Liu, J.; Cui, L.; Liu, H.; Huang, D.; Wang, Y.; and Zhang, Y. 2020. LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning. arXiv:2007.08124.
- Liu, J.; Huang, Z.; Dai, W.; Cheng, C.; Wu, J.; Sha, J.; Li, S.; Liu, Q.; Wang, S.; and Chen, E. 2025a. CogMath: Assessing LLMs’ Authentic Mathematical Ability from a Human Cognitive Perspective. arXiv:2506.04481.
- Liu, Z.; Chen, C.; Li, W.; Qi, P.; Pang, T.; Du, C.; Lee, W. S.; and Lin, M. 2025b. Understanding R1-Zero-Like Training: A Critical Perspective. arXiv:2503.20783.
- Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; Tang, Y.; and Zhang, D. 2025a. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evolutionary Instruct. arXiv:2308.09583.
- Luo, M.; Tan, S.; Wong, J.; Shi, X.; Tang, W.; Roongta, M.; Cai, C.; Luo, J.; Zhang, T.; Li, E.; Popa, R. A.; and Stoica, I. 2025b. DeepScaleR: Surpassing O1-Preview with a 1.5B Model by Scaling RL. <https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2>. Notion Blog.
- Ma, R.; Wang, P.; Liu, C.; Liu, X.; Chen, J.; Zhang, B.; Zhou, X.; Du, N.; and Li, J. 2025. S²R: Teaching LLMs to Self-verify and Self-correct via Reinforcement Learning. arXiv:2502.12853.
- Magister, L. C.; Mallinson, J.; Adamek, J.; Malmi, E.; and Severyn, A. 2023. Teaching Small Language Models to Reason. 1773–1781. Toronto, Canada: Association for Computational Linguistics.
- Mirzadeh, I.; Alizadeh, K.; Shahrokhi, H.; Tuzel, O.; Bengio, S.; and Farajtabar, M. 2024. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models. arXiv:2410.05229.
- Muralidharan, S.; Turuvekere Sreenivas, S.; Joshi, R.; Chochowski, M.; Patwary, M.; Shoeybi, M.; Catanzaro, B.; Kautz, J.; and Molchanov, P. 2024. Compact Language Models via Pruning and Knowledge Distillation. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., Advances in Neural Information Processing Systems, volume 37, 41076–41102. Curran Associates, Inc.
- Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., Advances in Neural Information Processing Systems, volume 35, 27730–27744. Curran Associates, Inc.
- Patel, A.; Bhattamishra, S.; and Goyal, N. 2021. Are NLP Models really able to Solve Simple Math Word Problems? arXiv:2103.07191.
- Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.
- Sessa, P. G.; Dadashi, R.; Hussenot, L.; Ferret, J.; Vieillard, N.; Ramé, A.; Shariari, B.; Perrin, S.; Friesen, A.; Cideron, G.; et al. 2024. Bond: Aligning llms with best-of-n distillation. arXiv preprint arXiv:2407.14622.
- Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of

Mathematical Reasoning in Open Language Models. arXiv:2402.03300.

Shen, M.; Zeng, G.; Qi, Z.; Hong, Z.-W.; Chen, Z.; Lu, W.; Wornell, G.; Das, S.; Cox, D.; and Gan, C. 2025. Satori: Reinforcement Learning with Chain-of-Action-Thought Enhances LLM Reasoning via Autoregressive Search. arXiv:2502.02508.

Sheng, Y.; Li, L.; and Zeng, D. D. 2025. Learning Theorem Rationale for Improving the Mathematical Reasoning Capability of Large Language Models. Proceedings of the AAAI Conference on Artificial Intelligence, 39(14): 15151–15159.

Shumailov, I.; Shumaylov, Z.; Zhao, Y.; Papernot, N.; Anderson, R.; and Gal, Y. 2024. AI models collapse when trained on recursively generated data. Nature, 631(8022): 755–759.

Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; and Zhou, D. 2020. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. arXiv:2004.02984.

Team, K.; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C.; Liao, C.; et al. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. arXiv preprint arXiv:2501.12599.

Vendrow, J.; Vendrow, E.; Beery, S.; and Madry, A. 2025. Do Large Language Model Benchmarks Test Reliability? arXiv:2502.03461.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35: 24824–24837.

Wen, J.; Guan, J.; Wang, H.; Wu, W.; and Huang, M. 2024. Codeplan: Unlocking reasoning potential in large language models by scaling code-form planning. In The Thirteenth International Conference on Learning Representations.

Xu, J.; Li, Z.; Chen, W.; Wang, Q.; Gao, X.; Cai, Q.; and Ling, Z. 2024. On-Device Language Models: A Comprehensive Review. arXiv:2409.00088.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025a. Qwen3 technical report. arXiv preprint arXiv:2505.09388.

Yang, L.; Yu, Z.; Zhang, T.; Xu, M.; Gonzalez, J. E.; Cui, B.; and Yan, S. 2025b. SuperCorrect: Advancing Small LLM Reasoning with Thought Template Distillation and Self-Correction. arXiv:2410.09008.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Dai, W.; Fan, T.; Liu, G.; Liu, L.; et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476.

Yuan, W.; Neubig, G.; and Liu, P. 2021. BARTScore: Evaluating Generated Text as Text Generation. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., Advances in Neural Information Processing Systems, volume 34, 27263–27277. Curran Associates, Inc.

Zelikman, E.; Wu, Y.; Mu, J.; and Goodman, N. D. 2024. Star: Self-taught reasoner bootstrapping reasoning with reasoning. In Proc. the 36th International Conference on Neural Information Processing Systems, volume 1126.

Zhang, T.; Wang, X.; Zhou, D.; Schuurmans, D.; and Gonzalez, J. E. 2022. TEMPERA: Test-Time Prompting via Reinforcement Learning. arXiv:2211.11890.

Zhang, X.; Li, C.; Zong, Y.; Ying, Z.; He, L.; and Qiu, X. 2024. Evaluating the Performance of Large Language Models on GAOKAO Benchmark. arXiv:2305.12474.

附录 A: 定理 1 的证明

Theorem 1. 设 $\mathcal{Q}$ 为自然语言问题的有限集合，被视为不同的状态 s ，并设 A 表示候选响应的集合。在每一轮 $t \in \{1, \dots, T\}$ 中，SLM 观察一个问题 $s_t \in \mathcal{Q}$ ，选择一个动作 $a_t \in A$ ，并获得奖励 $r(s_t, a_t)$ 。假设通过在情境赌博机设置中使用按状态的上置信界 (UCB) 算法进行学习。那么，经过 T 轮后的总后悔值被限制在

$$R_T = O\left(\sqrt{|\mathcal{Q}| \cdot |A| \cdot T \cdot \ln T}\right),$$

，其中 $|\cdot|$ 是集合的元素数量。

Proof. 我们将每个自然语言问题 $x \in \mathcal{Q}$ 视为一个独特的状态 s ，并将候选响应集定义为动作空间 A 。在每一轮 $t \in \{1, \dots, T\}$ 中，代理观察一个状态 $s_t \in \mathcal{Q}$ ，选择一个动作 $a_t \in A$ ，并接收一个奖励 $r_t = r(s_t, a_t) \in [0, 1]$ ，其期望值为 $\mu(s_t, a_t)$ 。令 $\mu^*(s) = \max_{a \in A} \mu(s, a)$ 为状态 s 的最优期望奖励。总遗憾定义为：第 1 步：按状态-动作对分解遗憾。令 $\Delta(s, a) = \mu^*(s) - \mu(s, a)$ 表示在状态 s 下动作 a 的非最优性差距。令 $N_T(s, a)$ 表示在状态 s 下选择动作 a 的次数，直到第 T 轮。则总遗憾表达为：其中 $a^*(s) = \arg \max_{a \in A} \mu(s, a)$ 。第 2 步：对单个状态限制遗憾。固定 $s \in \mathcal{Q}$ ，并令 $T_s = \sum_{a \in A} N_T(s, a)$ 表示状态 s 出现的次数。当处于状态 s 时，代理面临一个标准的多臂老虎机问题，具有 $|A|$ 个臂。根据 UCB 算法，对状态 s 的遗憾被限制为 (Auer, Cesa-Bianchi, and Fischer 2002; Bubeck, Cesa-Bianchi et al. 2012)：由于 $T_s \leq T$ ，我们有 $\ln T_s \leq \ln T$ ，因此：第 3 步：对所有状态求和。总遗憾是所有状态的遗憾之和：令 $C_1 > 0$ 为一个常数，使得对于所有 $s \in \mathcal{Q}$ ， $R_s \leq C_1 \sqrt{|A| \cdot T_s \cdot \ln T}$ 成立。则：第 4 步：应用柯西-施瓦兹不等式。为了限制和 $\sum_{s \in \mathcal{Q}} \sqrt{T_s}$ ，我们使用柯西-施瓦兹不等式：由于 $\sum_{s \in \mathcal{Q}} T_s = T$ ，这简化为：第 5 步：最终界限。代入回 R_T 的表达式中，我们得到：因此，总遗憾被界定为：□

附录 B: 实验设置

我们在表 ?? 中总结了训练和评估数据集集中的样本数量。GSM8K (Cobbe et al. 2021) 数据集用作主要的训练集。GSM8K-1 到 GSM8K-3 指的是在 DURIT 的步骤二中获得的过滤子集，而 GSM8K-4 包含由 DeepSeek-R1 (Shao et al. 2024) 生成的 CoT 示例。GSM8K-5 到 GSM8K-7 对应于不同基本模型为 STaR 生成的 CoT 数据。所有其他数据集仅用于图 ?? 中提示的评估。

附录 B.2: 迭代训练实验的数据集

迭代训练实验中使用的数据集描述如下。在第二次迭代中，我们使用了来自 GSM8K 数据集的完整 7,473 个样本和来自 MATH 数据集的完整 7,500 个样本 (Hendrycks et al. 2021)。对于 DeepScaleR (Luo et al. 2025b)，为了控制难度水平，我们首先使用温度 0.7 生成每个实例的 8 个响应，并使用 vLLM (Kwon et al. 2023) 加速解码。然后，我们舍弃了表现出极端难度的样本，具体而言，就是那些在采样响应中平均准确率为 1.0 或 0.0 的样本。在剩余的样本中，我们随机选择了 7,500 个样本进行训练，以确保所有三个实验设置使用的数据集大小相等。

所有实验均使用 VeRL 框架实现，并使用 Python 3.10 和 PyTorch 2.6。我们使用 DeepSeek-R1-0528 在 GSM8K 训练数据集上生成 CoT 推理轨迹，这些轨迹由类似于 (Liao et al. 2025) 的提示引导，并如图 6 所示。然后在我们的 CoT-Dis 设置中蒸馏这些轨迹以监督 SLM 训练。对于 STaR，我们遵循与 DURIT 相同的设置，通过使用 SLM 采样 8 个答案并在过滤后只提炼正确的 CoT 轨迹。对于 Vanilla-KD (Muralidharan et al. 2024)，我们使用来自 DURIT 的映射模型作为知识蒸馏的教师。所有蒸馏方法训练 5 个周期，与 DURIT 步骤 II 一致。我们将最大序列长度设置为 2536，学习率为 $1e-5$ ，KL 损失系数为 0.001，批量大小为 4。对于 GRPO (Shao et al. 2024)，我们对每个查询采样 8 个响应，将批量大小设置为 16，学习率为 $1e-6$ ，最大响应长度为 1024，并训练 3 个周期。对于 PRewrite (Kong et al. 2024)，我们使用 GRPO 算法进行提示优化，在优化过程中使用 DURIT 映射模型作为翻译器。模型训练 1 个周期，与 DURIT 步骤 I 一致。对于 PRewrite，我们通过评估 20 个提示变体（包括原始的 PRewrite 提示）进行提示微调，并选择表现最佳的一个，它与 DURIT 映射器使用的提示相一致（如图 7 所示）。对于 DURIT，我们采用相同的提示（见图 7），将选择参数 λ 设置为对 Qwen2.5-0.5B-Instruct 模型 (Yang et al. 2025a) 为 0.2，对于 Llama3.2-1B-Instruct 模型 (Grattafiori et al. 2024) 为 0.05。对于 Qwen2.5-1.5B-Instruct 模型，我们在 DURIT 中将自蒸馏学习率设置为 $1e-6$ ，并将 λ 设置为 0.01。为了评估 DURIT 对提示变化的鲁棒性，我们另外评估了两个备选提示设计，如图 8 和 9 所示。所有其他训练超参数与相应的基线相同，以确保公平比较，并将随机种子设置为 1。所有模型均使用 AdamW 优化器进行优化，设置 $\beta_1 = 0.9$ ， $\beta_2 = 0.95$ ，和权重衰减为 0.01。

我们在算法 1 中展示了 DURIT 的伪代码。

附录 D: 局限性

我们所提出的将理解与推理解耦的方法为增强大型语言模型 (LLMs) 的推理能力提供了一种新的视角。然而，承认我们研究的局限性同样重要。由于硬件限制，我们没有对更大的模型（例如超过 3B 参数的模型）进行实验。然而，所提出的方法在扩展到更大的模型方面展现出很大的潜力，因为解耦机制和问题空间维度的减少可以共同提高探索、学习效率和推理的鲁棒性。尽管我们的方法由于需要专用映射器而引入了额外的训练开销，其核心优势在于通过模型自身的能力来增强推理和鲁棒性——而无需依赖外部强模型。相比于蒸馏等方法，我们的方法在应用于强大模型方面更加广泛，避免

了 API 访问的高成本以及需要匹配良好的教师与大规模训练数据的限制。这带来了更好的可扩展性和泛化能力。

为了进一步验证我们方法的有效性，我们在 Qwen2.5-1.5B-Instruct 模型上进行了实验，结果如表 5 所示。经过一次迭代后，DURIT 相较于最强的极端基线平均准确率提升了 1.01%，特别是在 MATH500 和 GAOKAO 数据集上有显著的提升，分别达到 53.60% 和 32.26%。第二次迭代带来了进一步的改进，证明了该方法的扩展性和稳定性。虽然由于 GSM8K 数据集相对简单，提供的额外收益有限，但在训练更复杂的数据集时，DURIT 展示出了更大的潜力，突显了其通过明确将理解与推理解耦来提升探索效率的能力。此外，如表 6 所示，DURIT 在推理方面表现出很强的鲁棒性：在 gsm-symbolic (Mirzadeh et al. 2024) 数据集上，它的准确率甚至高于原始的 GSM8K-100 子集，并在符号扰动下减少性能下降方面超越了所有极端基线。这些结果证实，DURIT 通过在一个更抽象和规范的问题空间中进行推理，有效地减轻了虚假相关的影响，从而在推理准确性和鲁棒性上均有一致的提升。

为了评估 DURIT 在超出数学领域的泛化能力，我们使用 Qwen2.5-1.5B-Instruct 模型在逻辑推理数据集 LogiQA 上进行了额外的训练。由于 LogiQA 由选择题组成，我们引入了来自 GSM8K 的填空题子集，以缓解潜在的特定格式遗忘，并保持模型在不同答案格式下的泛化能力。具体来说，我们通过从 LogiQA 中随机抽取 4000 个实例和在 GSM8K 中抽取 2000 个实例，构建了一个包含 6000 个例子的混合训练集。如表 7 所示，基于 DeepSeek-R1 的 CoT-Dis 方法表现明显较差，主要是由于在 LogiQA 数据集上生成的过于复杂的 CoT 推理。这些复杂的推理过程对于 SLM 来说难以学习和泛化，最终削弱了学生模型在任务间的推理能力。这个观察揭示了传统知识蒸馏方法的潜在局限性，即当教师输出与学生的学习能力不匹配时。然而，尽管 DURIT 主要是在逻辑推理数据上训练的，它仍能实现最佳的整体准确率，一次和两次迭代后在 LogiQA 数据集上分别超出最强基线 +0.43% 和 +0.62% 平均值，以及 +1.23%，并在大多数基准测试中，包括那些专注于数学推理的基准测试中持续提供高准确率。此外，如表 8 所示，DURIT 显示出优越的鲁棒性，甚至在 gsm-symbolic 数据集上达到更高的准确率，这突显了通过提议的理解和推理分离来增强其推理能力和鲁棒性。这些结果证实了 DURIT 在提高一般推理性能方面的有效性和领域可转移性。

为了研究超参数对算法性能的影响，我们使用 Qwen2.5-0.5B-Instruct 模型对蒸馏系数 λ 进行深入分析。实验结果如图 10 所示。我们观察到，当 λ 过小时，训练目标被监督微调 (SFT) 损失主导，这可能过度限制模型在强化学习 (RL) 阶段的探索能力。随着 λ 增加，性能提高并在 $\lambda = 0.2$ 达到峰值。然而，进一步增加 λ 导致性能下降，可能是由于自蒸馏的过度影响。在这种情况下，模型可能过度依赖于对转换问题 Q' 的不准确解释，引入有害噪声，不仅降低学习效果，还可能扰乱模型原有的内部推理模式。总体而言，性能在广泛的 λ 值范围内（从 0.1 到 0.5）保持相对稳定， $\lambda = 0.2$ 成为最有效的选择。

为了进一步验证 DURIT 在提高训练效率方面的有效性，我们将 DURIT（第一轮的第二步）的训练奖励与 Qwen2.5-0.5B-Instruct 和 Llama3.2-1B-Instruct 模

Generate CoTs

You are an expert assistant teacher specializing in math problems. For each task, first provide a detailed step-by-step reasoning process and then give the final answer enclosed in `\boxed{}`.

Question: {QUESTION}.

Answer: Let's think step by step and output the final answer within `\boxed{}`.

Figure 6: 用于指导 DeepSeek-R1-0528 生成 CoT 推理轨迹的提示。

Prompt Design for the Problem Space Mapper M

You are a professional exam editor. Your task is to rephrase exam questions to make them clearer and easier to understand without changing their meaning. Focus only on improving clarity, conciseness, and consistency in phrasing. Do not solve the question, do not explain it, and do not suggest or imply any answers. You may simplify wording and remove unnecessary background information, but you must retain all details, conditions, and context necessary to solve the problem. Output only the rewritten question as a single clear paragraph.

Figure 7: 问题空间映射器 M 的提示。

Design Variant of Prompt 2 for the Problem Space Mapper M

Rewrite the following questions by rephrasing them without altering their original meaning. Preserve all details relevant to solving the problem, including any conditions, constraints, or contextual information. Use consistent language for questions with the same meaning. Output only the rewritten questions.

Figure 8: 问题空间映射器 M 的探索性提示 2。

Design Variant of Prompt 3 for the Problem Space Mapper M

You are a professional exam question editor. Your task is to rewrite exam questions to improve clarity, conciseness, and consistency, without altering their original meaning. Preserve all essential information, conditions, and context needed to solve the question, but remove any redundant or overly complex wording. Do not solve, explain, or provide hints for the question. Only output the revised version as a single clear and well-structured paragraph.

Figure 9: 问题空间映射器 M 的探索性提示 3。

Methods	In-Domain	Out-of-Domain					Average
	gsm8k-platinum	MAWPS	SVAMP	MATH500	GAOKAO	LogiQA	
# Qwen2.5-1.5B-Instruct based							
Base (Yang et al. 2025a)	72.46	57.88	73.00	46.40	25.00	40.86	52.60
CoT-Dis (Magister et al. 2023)	71.63	59.42	85.67	42.40	20.97	37.48	52.93
STaR (Zelikman et al. 2024)	<u>76.34</u>	63.27	80.67	49.40	26.61	<u>42.24</u>	56.42
GRPO (Shao et al. 2024)	76.43	63.85	83.67	50.20	28.23	39.32	56.95
PRewrite (Kong et al. 2024)	70.55	54.42	80.00	45.60	20.16	31.03	50.29
Vanilla-KD (Muralidharan et al. 2024)	76.18	62.69	82.00	51.80	24.19	36.56	55.57
DURIT (ours, iter=1)	<u>76.34</u>	<u>63.46</u>	81.67	53.60	32.26	40.40	<u>57.96</u>
DURIT (ours, iter=2)	76.01	62.88	<u>84.00</u>	<u>52.20</u>	<u>30.65</u>	42.40	58.02

Table 5: 在六个代表性基准上评估，GSM8K 数据集中训练的 Qwen2.5-1.5B-Instruct 模型的性能 (%)。加粗和 underline 分别表示最佳和次佳结果。

Method	Qwen-1.5B		
	Orig	Symb	$\Delta\%$
Base	67.00	63.82	-4.75
CoT-Dis	71.00	68.24	-3.89
STaR	79.00	67.10	-15.06
GRPO	65.00	66.50	+2.31
PRewrite	70.00	63.82	-8.83
Vanilla-KD	73.00	68.34	-6.38
DURIT	64.00	66.68	+4.19

Table 6: 基于在 GSM8K 数据集上训练的 Qwen2.5-1.5B-Instruct，对不同方法的比较。DURIT 使用单次迭代进行训练。Orig: 原始的 GSM8K-100 子集; Symb: gsm-symbolic; $\Delta\%$: 从 Orig 到 Symb 的相对性能下降。每组中的最佳和次佳结果分别用粗体和 underline 标出。

型上的 GRPO 基线进行比较，如图 11 和 12 所示。对于这两种模型，DURIT 只在一小部分训练步骤后就达到了与收敛的 GRPO 基线相同的奖励水平，大大减少了训练时间并提高了样本效率。此外，DURIT 在收敛后获得了更高的最终奖励，这表明它不仅能够通过将理解与推理解耦和压缩有效状态空间来加速强化学习，而且还能提高收敛质量。这些结果也为定理 ?? 提供了实证支持。

为了比较所有方法的训练开销，我们报告了在一块 80GB 内存的 NVIDIA A800 GPU 上测量的总运行时间，使用 Qwen2.5-0.5B-Instruct 作为基础模型，结果如表 9 所示。对于 CoT-Dis，主要瓶颈在于从一个强大的教师模型中收集 CoT 数据。在 DeepSeek-R1 的情况下，由于 API 响应相对较慢，我们使用三个并发终端，但仍然需要 60 小时来完成数据收集，并且增加了额外的 API 使用成本。对于 STaR，从基础模型生成 CoT 输出大约需要 2 小时，总运行时间为 3.2 小时。对于 GRPO，训练在 3 个 epoch 内收敛，总计需要 6.2 小时。PRewrite 仅在 0.23 个 epoch 的训练 (1.7 小时) 内收敛；然而，由于需要一个问题重写器作为预处理步骤，它在推理时引入了额外的开销。Vanilla-KD 花费 2.4 小

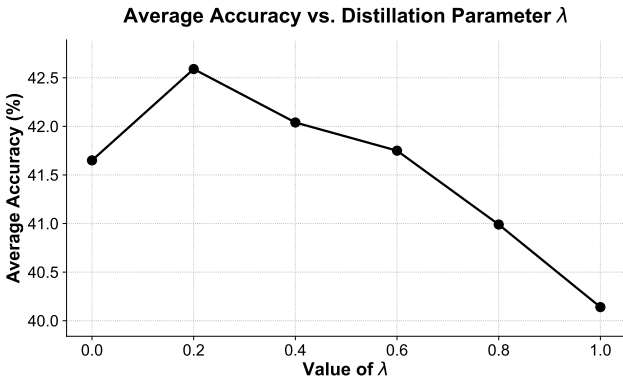


Figure 10: Qwen2.5-0.5B-Instruct 在不同蒸馏系数 λ 下六个数据集的平均准确率。

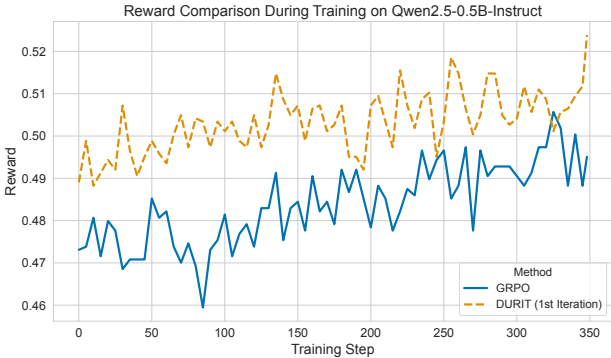


Figure 11: Qwen2.5-0.5B-Instruct 在 DURIT-步骤 III 与 GRPO 基线之间的训练奖励比较。

Methods	In-Domain		Out-of-Domain				Average
	gsm8k-platinum	LogiQA	SVAMP	MATH500	GAOKAO	MAWPS	
# Qwen2.5-1.5B-Instruct based							
Base (Yang et al. 2025a)	72.46	40.86	73.00	46.40	25.00	57.88	52.60
CoT-Dis (Magister et al. 2023)	69.73	30.26	74.33	33.20	21.77	55.58	47.48
STaR (Zelikman et al. 2024)	74.44	<u>43.93</u>	<u>84.00</u>	49.40	25.00	62.69	56.58
GRPO (Shao et al. 2024)	75.85	43.63	83.67	50.20	<u>26.61</u>	62.69	57.11
PRewrite (Kong et al. 2024)	70.14	31.64	79.00	46.60	23.39	55.38	51.03
Vanilla-KD (Muralidharan et al. 2024)	<u>75.93</u>	39.01	82.00	<u>51.60</u>	27.42	63.65	56.60
DURIT (ours, iter=1)	76.84	43.47	<u>84.00</u>	50.60	27.42	<u>62.88</u>	<u>57.54</u>
DURIT (ours, iter=2)	75.43	45.16	85.00	53.00	<u>26.61</u>	61.15	57.73

Table 7: 在六个具有代表性的基准测试中，对在 GSM8K + LogiQA 混合数据集上训练的 Qwen2.5-1.5B-Instruct 模型的性能 (%) 进行评估。加粗和 underline 分别表示最佳和次佳结果。

Method	Qwen-1.5B		
	Orig	Symb	$\Delta\%$
Base	67.00	63.82	<u>-4.75</u>
CoT-Dis	70.00	55.20	-21.14
STaR	73.00	66.70	-8.63
GRPO	70.00	66.04	-5.66
PRewrite	68.00	63.60	-6.47
Vanilla-KD	76.00	68.42	-9.97
DURIT	64.00	65.42	+2.22

Table 8: 基于在 GSM8K + LogiQA 混合数据集上训练的 Qwen2.5-1.5B-Instruct 的不同方法比较。DURIT 使用单次迭代训练。Orig: 原始 GSM8K-100 子集; Symb: gsm-符号化; $\Delta\%$: 相对于 Orig 到 Symb 的性能下降。加粗和 underline 表示每组中的最佳和次佳结果。

时从教师模型收集 CoT 响应，并在 2.3 小时内完成训练，总计 4.7 小时。对于 DURIT，步骤 I 在 1.7 小时内 (0.23 个 epoch) 训练问题空间映射器到收敛。步骤 II 在 2.8 小时内执行问题映射并收集基础模型 CoT 响应，然后进行 1.2 小时的自蒸馏。步骤 III 涉及 6.2 小时的 GRPO 训练，总运行时间为 11.9 小时。值得注意的是，DURIT 在没有步骤 III 的情况下 (即没有 grpo) 已经实现了最先进的性能，如主论文中的消融结果所示，总训练成本仅为 5.7 小时。此外，当 Step III 强化学习仅在数据的 10 % 上进行训练时，该模型在奖励方面超过了完整的 GRPO 基线 (见图 11)，总训练时间只比 GRPO 长 0.1 小时。这些结果强调了 DURIT 通过将理解与推理分离并有效压缩问题空间，提高了探索效率和收敛速度。这为增强 LLM 的推理能力和鲁棒性提供了新的视角。

附录 F：案例研究

原始问题和转换后问题的示例，以及 Qwen2.5-0.5B-Instruct 和 Llama 3.2-1B-Instruct 基础模型的相应回答，分别展示在图 13 和图 14 中。在转换之前，模型误解了原始问题中描述的折扣机制，导致得到错误的答案。经过问题空间映射器 M 的转换后，问题被重新表

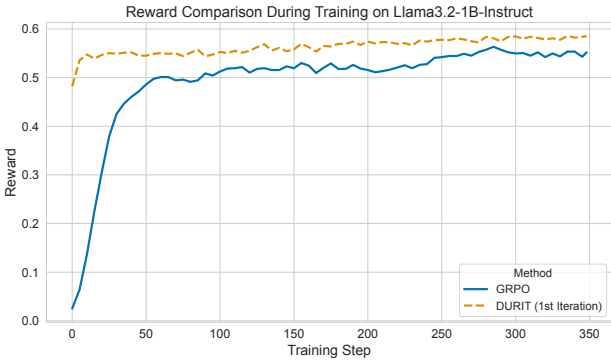


Figure 12: Llama3.2-1B-Instruct 在 DURIT-step III 和 GRPO 基线之间的训练奖励比较。

述为更简洁和标准化的形式，有效地降低了模型的理解负担。这使更集中的推理成为可能，最终导致了正确的解决方案。

图 15、图 16、图 17 和图 18 提供了来自 GSM8K-Platinum (Vendrow et al. 2025)、GAOKAO (Zhang et al. 2024)、SVAMP (Patel, Bhattamishra, and Goyal 2021) 和 MATH500 (Hendrycks et al. 2021) 数据集的示例，突显了我们的 DURIT 方法带来的显著推理增强。在第一个案例中 (图 15)，基础模型因复杂的题目细节而产生混乱，采用了错误的逆向推理方法。相反，通过 DURIT 训练的模型正确解读了问题，采用了强有力的方程求解策略，并利用 RL 激活的自我纠正机制在解题中途纠正错误，最终得到正确结论。在第二个案例中 (图 16)，基础模型依赖于错误的公式，导致答案冗长且错误。然而，DURIT 训练的模型展示了优越的推理能力，提供了既简洁又正确的解决方案。在第三个案例中 (图 17)，基础模型被关于公交容量的干扰信息误导，结果计算错误。相反，DURIT 训练的模型准确理解了公交容量所施加的限制，执行了正确的条件推理，得到正确答案。在第四个案例中 (图 18)，基础模型错误地将 9 当作 3 的立方，导致错误的结论。相反，DURIT 训练的模型采用了正确且更为简洁的推理路径，得出正确答案。

Instance-Level Comparison of Qwen2.5-0.5B-Instruct's Answers Before and After Problem Mapping

Original Problem:

A concert ticket costs \$40. Mr. Benson bought 12 tickets and received a 5% discount for every ticket bought that exceeds 10. How much did Mr. Benson pay in all? Let's think step by step and Only output the final choice within $\boxed{}$.

Model' response:

To determine how much Mr. Benson paid in total for the concert tickets, we need to follow these steps:

1. Calculate the number of tickets he bought that qualify for the discount.
2. Determine the cost per ticket after the discount.
3. Calculate the total cost for all the tickets.

First, let's find out how many tickets Mr. Benson bought that qualify for the discount. Since each ticket is priced at \$40 and he bought 12 tickets, but only 10 exceed the discount threshold (which is $12 - 10 = 2$), he qualifies for a 5% discount on the first 10 tickets.

The cost for the first 10 tickets is:

$$\backslash[10 \times 40 = 400 \text{ dollars} \backslash]$$

Since he qualifies for a 5% discount on the first 10 tickets, the discount amount is:

$$\backslash[400 \times 0.05 = 20 \text{ dollars} \backslash]$$

So, the cost for the first 10 tickets is reduced by \$20, leaving him with:

$$\backslash[400 - 20 = 380 \text{ dollars} \backslash]$$

Next, we calculate the cost for the remaining 2 tickets. The cost for each of these tickets is:

$$\backslash[40 \times 0.95 = 38 \text{ dollars} \backslash]$$

Therefore, the total cost for all 12 tickets is:

$$\backslash[380 + 38 = 418 \text{ dollars} \backslash]$$

Thus, the total amount Mr. Benson paid is:

$$\backslash[\backslashboxed{418} \backslash]$$

Mapped Problem:

A concert ticket costs \$40. Mr. Benson bought 12 tickets and received a 5% discount on each ticket beyond the 10th ticket. How much did Mr. Benson pay in total? Let's think step by step and Only output the final choice within $\boxed{}$.

Model's Response:

To determine how much Mr. Benson paid for the concert tickets, we need to follow these steps:

1. Calculate the cost of the first 10 tickets.
2. Determine the number of additional tickets (beyond the first 10) that receive the discount.
3. Calculate the discounted price for the additional tickets.
4. Add the cost of the first 10 tickets to the discounted price of the additional tickets.

Let's start with the first step:

The cost of the first 10 tickets is:

$$\backslash[10 \times \$40 = \$400 \backslash]$$

Next, we calculate the number of additional tickets (beyond the first 10) that receive the discount. Since there are 12 tickets in total and 10 of them are already included in the initial purchase, the number of additional tickets is:

$$\backslash[12 - 10 = 2 \backslash]$$

Each of these additional tickets receives a 5% discount, so the discount per ticket is:

$$\backslash[5\% \times \$40 = 0.05 \times \$40 = \$2 \backslash]$$

Therefore, the cost of each additional ticket after the discount is:

$$\backslash[\$40 - \$2 = \$38 \backslash]$$

So, the total cost for the additional 2 tickets is:

$$\backslash[2 \times \$38 = \$76 \backslash]$$

Finally, we add the cost of the first 10 tickets to the discounted price of the additional tickets:

$$\backslash[\$400 + \$76 = \$476 \backslash]$$

Thus, the total amount Mr. Benson paid is:

$$\backslash[\backslashboxed{476} \backslash]$$

Figure 13: 对此前后问题空间映射进行说明性比较, Qwen2.5-0.5B-Instruct 的回应, 红色 突出显示不正确的推理, 绿色 表示正确的推理, 蓝色 标示关键区别。

Instance-Level Comparison of Llama 3.2-1B-Instruct's Answers Before and After Problem Mapping

Original Problem:

Grandma Jones baked 5 apple pies for the fireman's luncheon. She cut each pie into 8 pieces and set the five pies out on the buffet table for the guests to serve themselves. At the end of the evening, after the guests had taken and eaten their pieces of pie, there were 14 pieces of pie remaining. How many pieces were taken by the guests? Let's think step by step and output the final answer within `\boxed{}`.

Model' response:

Step 1: Determine the total number of pies initially set out

Grandma Jones set 5 pies out on the buffet table.

Step 2: Calculate the number of pieces taken by the guests

Since each pie was cut into 8 pieces, we need to find out how many pieces were taken by the guests. To do this, we need to divide the total number of pies by the number of pieces per pie.

Step 3: Perform the division

To find the number of pieces taken by the guests, we divide the total number of pies (5) by the number of pieces per pie (8).

$5 \div 8 = 0.625$

Step 4: Round down to the nearest whole number

Since we can't have a fraction of a pie, we need to round down to the nearest whole number.

0.625 rounded down is 0.

The final answer is: `\boxed{0}`

Mapped Problem:

Grandma Jones baked 5 apple pies, cutting each into 8 pieces, and placed them on the buffet table for guests to serve themselves. After the guests finished, 14 pieces of pie remained. How many pieces of pie did the guests take in total? Let's think step by step and output the final answer within `\boxed{}`.

Model's Response:

Step 1: Calculate the total number of pie pieces initially

Grandma Jones baked 5 apple pies, and each pie was cut into 8 pieces. To find the total number of pieces, we multiply the number of pies by the number of pieces per pie: $5 * 8 = 40$.

Step 2: Calculate the number of pieces taken by the guests

After the guests finished, 14 pieces of pie remained. To find the number of pieces taken by the guests, we subtract the number of remaining pieces from the total number of pieces initially: $40 - 14 = 26$.

The final answer is: `\boxed{26}`

Figure 14: 在问题空间映射之前和之后，Llama 3.2-1B-Instruct 对问题的回应的对比示例，使用 红色 高亮显示错误的推理，使用 绿色 指示正确的推理，并使用 蓝色 标记关键差异。

Instance-Level Comparison of Qwen2.5-0.5B-Instruct on the GSM8K-platinum Dataset

Problem:

Melanie is a door-to-door saleswoman. She sold a third of her vacuum cleaners at the green house, 2 more to the red house, and half of what was left at the orange house. If Melanie has 5 vacuum cleaners left, how many did she start with? Let's think step by step and Only output the final choice within $\boxed{}$.

Base Model' response:

To determine how many vacuum cleaners Melanie started with, we need to work backwards from the information given about the remaining vacuum cleaners.

1. **Identify the number of vacuum cleaners left after the first sale:**

Melanie had 5 vacuum cleaners left after selling some at the green house.

2. **Determine how many vacuum cleaners were sold at the green house:**

Melanie sold a third of her vacuum cleaners at the green house. Therefore, the number of vacuum cleaners sold at the green house is:

$$\frac{5}{3} = 1 \text{ (since } 5 \div 3 = 1.67 \text{)}$$

This means that $(5 - 1 = 4)$ vacuum cleaners were sold at the green house.

3. **Calculate the number of vacuum cleaners left after the second sale:**

Melanie then sold 2 more vacuum cleaners to the red house. So, the number of vacuum cleaners left after this sale is:

$$4 - 2 = 2$$

4. **Determine how many vacuum cleaners were sold at the red house:**

Melanie sold half of the remaining vacuum cleaners at the orange house. The number of vacuum cleaners sold at the orange house is:

$$\frac{2}{2} = 1$$

This means that $(2 - 1 = 1)$ vacuum cleaner was sold at the orange house.

5. **Calculate the total number of vacuum cleaners sold:**

Adding up all the vacuum cleaners sold, we get:

$$4 + 2 + 1 = 7$$

Therefore, Melanie started with $\boxed{7}$ vacuum cleaners.

DURIT Model's Response:

To determine how many vacuum cleaners Melanie started with, we need to work backwards through the transactions and account for the remaining vacuum cleaners.

1. Let's denote the total number of vacuum cleaners Melanie started with as x .

2. Melanie sold a third of her vacuum cleaners at the green house:

$$\text{Vacuum cleaners sold at the green house} = \frac{x}{3}$$

After selling at the green house, Melanie has:

$$x - \frac{x}{3} = \frac{2x}{3}$$

3. Melanie sold 2 more vacuum cleaners at the red house:

$$\text{Vacuum cleaners sold at the red house} = \frac{2x}{3} - 2$$

After selling at the red house, Melanie has:

$$\frac{2x}{3} - 2 - \frac{2x}{3} = -2$$

This is incorrect, so let's re-evaluate the problem. The correct approach is to consider the remaining vacuum cleaners after the first two sales.

4. Melanie sold half of the remaining vacuum cleaners at the orange house:

$$\text{Vacuum cleaners sold at the orange house} = \frac{1}{2} \left(\frac{2x}{3} - 2 \right) = \frac{x}{3} - 1$$

After selling at the orange house, Melanie has:

$$\left(\frac{2x}{3} - 2 \right) - \left(\frac{x}{3} - 1 \right) = \frac{2x}{3} - 2 - \frac{x}{3} + 1 = \frac{x}{3} - 1$$

According to the problem, Melanie has 5 vacuum cleaners left:

$$\frac{x}{3} - 1 = 5$$

Solving for x : $\frac{x}{3} = 6 \implies x = 18$

Therefore, Melanie started with $\boxed{18}$ vacuum cleaners.

Figure 15: 对 GSM8K-platinum 数据集上的 Qwen2.5-0.5B-Instruct 进行实例级比较, 对比基础模型和 DRUIT 模型的响应, 红色 表示错误推理, 绿色 表示正确推理。

Problem:

Given the hyperbola $\Gamma: x^2 - \frac{y^2}{b^2} = 1$ ($b > 0$) with left and right vertices A_1, A_2 . If the eccentricity $e=2$, find the value of b . Let's think step by step and Only output the final choice within $\boxed{}$.

Base Model' response:

To determine the value of b for the given hyperbola $\Gamma: x^2 - \frac{y^2}{b^2} = 1$ with an eccentricity $e = 2$, we need to follow these steps:

1. **Identify the coordinates of the vertices:**

The vertices of the hyperbola $x^2 - \frac{y^2}{b^2} = 1$ are located at $(\pm c, 0)$. Here, $c = ae$, where a is the distance from the center to each vertex along the x-axis.

2. **Calculate a :**

Given that the eccentricity $e = 2$, we have: $e = \sqrt{\frac{a^2}{c}} = 2$

Squaring both sides, we get: $\left(\sqrt{\frac{a^2}{c}}\right)^2 = 4 \implies \frac{a^2}{c} = 4 \implies a^2 = 4c$

3. **Determine the relationship between a and b :**

Since $a^2 = 4c$, we can express c in terms of a : $c = \frac{a^2}{4}$

4. **Find the coordinates of the vertices:**

The vertices are at $(\pm c, 0)$. Substituting $c = \frac{a^2}{4}$ into this equation, we get:

$$c = \frac{a^2}{4}$$

Therefore, the coordinates of the vertices are:

$$A_1 = \left(\frac{a^2}{4}, 0\right)$$

$$A_2 = \left(-\frac{a^2}{4}, 0\right)$$

5. **Use the standard form of the hyperbola:**

For the hyperbola $x^2 - \frac{y^2}{b^2} = 1$, the vertices are at $(\pm a, 0)$. Thus, we have: $a = \frac{a^2}{4}$

6. **Solve for a :**

From the equation $a = \frac{a^2}{4}$, we multiply both sides by 4 to clear the fraction: $4a = a^2$

Rearrange the equation to form a quadratic equation: $a^2 - 4a = 0$

Factor out a : $a(a - 4) = 0$

This gives us two solutions: $a = 0$ or $a = 4$

Since $a > 0$ (as it represents a distance), we discard $a = 0$.

7. **Determine the value of b :**

Using the relationship $a^2 = 4c$ and substituting $a = 4$:

$$4^2 = 4c \implies 16 = 4c \implies c = 4$$

Now, substitute $c = 4$ back into the expression for a : $a = \frac{a^2}{4} = \frac{16}{4} = 4$

Therefore, the value of b is $\boxed{4}$.

DURIT Model's Response:

To find the value of b for the hyperbola $\Gamma: x^2 - \frac{y^2}{b^2} = 1$ with an eccentricity $e = 2$, we need to follow these steps:

1. **Identify the vertices of the hyperbola:**

The standard form of the hyperbola is $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$. Here, $a^2 = 1$, so $a = 1$. The vertices of the hyperbola are at $(\pm a, 0)$, which are $(\pm 1, 0)$.

2. **Use the eccentricity formula:**

The eccentricity e of a hyperbola is given by $e = \sqrt{1 + \frac{b^2}{a^2}}$. We are given that $e = 2$. Substituting the known values, we get: $2 = \sqrt{1 + \frac{b^2}{1^2}} = \sqrt{1 + b^2}$

Squaring both sides of the equation, we obtain: $4 = 1 + b^2$

Solving for b^2 , we get: $b^2 = 3$

3. **Conclusion:**

The value of b is $\sqrt{3}$.

Therefore, the value of b is $\boxed{\sqrt{3}}$.

Figure 16: 在 GAOKAO 数据集上对 Qwen2.5-0.5B-Instruct 进行实例级比较，比对基础模型和 DRUIT 模型的响应，其中 红色 表示推理错误，绿色 表示推理正确。

Instance-Level Comparison of Llama3.2-1B-Instruct on the SVAMP Dataset

Problem:

There were 12 people on the bus. At the next stop 4 more people got on the bus. Each bus can not have more than 36 people. How many people are there on the bus now? Let's think step by step and Only output the final choice within \boxed{ }.

Base Model' response:

Step 1: Identify the initial number of people on the bus

There were initially 12 people on the bus.

Step 2: Calculate the number of people on the bus after 4 more people got on 4 people got on the bus, so we need to add 4 to the initial number of people. $12 + 4 = 16$.

Step 3: Check if the total number of people exceeds the maximum allowed capacity

The bus can have a maximum of 36 people. Since 16 people are already on the bus, we need to subtract 16 from 36 to find the final number of people on the bus. $36 - 16 = 20$.

The final answer is: $\boxed{20}$

DURIT Model's Response:

Step 1: Identify the initial number of people on the bus

There were initially 12 people on the bus.

Step 2: Calculate the number of people on the bus after 4 more people got on 4 people got on the bus, so we need to add 4 to the initial number of people. $12 + 4 = 16$.

Step 3: Check if the new total exceeds the maximum capacity of the bus

The bus can only have 36 people, so we need to check if 16 is less than 36. Since 16 is less than 36, the new total is valid.

The final answer is: $\boxed{16}$

Figure 17: 在 SVAMP 数据集上进行 Llama3.2-1B-Instruct 的实例级比较，比较基础模型和 DURIT 模型的响应，其中 红色 表示错误推理，绿色 表示正确推理。

Method	Data Processing	Training	Total Time
CoT-Dis	60.0	1.6	61.6
STaR	2.0	1.2	3.2
GRPO	0	6.2	6.2
PRewrite	0	1.7	1.7
Vanilla-KD	2.4	2.3	4.7
DURIT	2.8	9.1	11.9
DURIT w/o grpo	2.8	2.9	5.7
DURIT grpo 10 %	2.8	3.5	6.3

Table 9: 使用 Qwen2.5-0.5B-Instruct 模型对不同方法的计算时间估算（以小时为单位）。

Instance-Level Comparison of Qwen2.5-1.5B-Instruct on the MATH500 Dataset

Problem:

What is the smallest positive perfect cube that can be written as the sum of three consecutive integers? Let's think step by step and Only output the final choice within $\boxed{}$.

Base Model' response:

Step 1: Understand the problem

We need to find the smallest positive perfect cube that can be expressed as the sum of three consecutive integers.

Step 2: Define the equation

Let's denote the first integer as n . The sum of three consecutive integers can be written as $n + (n+1) + (n+2)$. We need to find the smallest perfect cube that satisfies this equation.

Step 3: Express the sum as a perfect cube

The sum of three consecutive integers can be expressed as $n + (n+1) + (n+2) = 3n + 3$.

Step 4: Factor out 3

We can factor out 3 from the equation: $3(n + 1)$

Step 5: Find the smallest perfect cube

We need to find the smallest perfect cube that can be expressed as $3(n + 1)$. Let's start with $n = 1$: $3(1 + 1) = 3(2) = 6$, which is not a perfect cube.

Step 6: Increment n and check

Let's increment n and check if the resulting perfect cube is a perfect cube. For $n = 2$: $3(2 + 1) = 3(3) = 9$, which is a perfect cube (3^2).

Step 7: Verify the result

9 is indeed a perfect cube (3^2), so we have found the smallest positive perfect cube that can be expressed as the sum of three consecutive integers.

The final answer is: $\boxed{9}$

DURIT Model's Response:

To find the smallest positive perfect cube that can be written as the sum of three consecutive integers, we start by expressing the sum of three consecutive integers in a general form. Let the three consecutive integers be $(n-1)$, (n) , and $(n+1)$. Their sum is: $(n-1) + n + (n+1) = 3n$

We need this sum to be a perfect cube. Therefore, we need $(3n)$ to be a perfect cube. Let's denote this perfect cube by (k^3) , where (k) is an integer. Then we have: $(3n = k^3)$

This implies that (k^3) must be divisible by 3, and since 3 is a prime number, (k) must be divisible by 3. Let $(k = 3m)$ for some integer (m) .

Substituting $(k = 3m)$ into the equation, we get: $(3n = (3m)^3 = 27m^3)$

Solving for (n) , we get: $(n = 9m^3)$

We need (n) to be a positive integer, so (m) must be a positive integer. The smallest positive integer for (m) is 1. Substituting $(m = 1)$ into the equation for (n) , we get: $(n = 9 \cdot 1^3 = 9)$

Thus, the three consecutive integers are $(9-1 = 8)$, (9) , and $(9+1 = 10)$. Their sum is: $(8 + 9 + 10 = 27)$

Since 27 is a perfect cube ((3^3)), the smallest positive perfect cube that can be written as the sum of three consecutive integers is: $\boxed{27}$

Figure 18: 在 MATH500 数据集上进行的 Qwen2.5-1.5B-Instruct 实例级别比较，对比基础模型和 DRUIT 模型的响应，其中 红色 表示错误推理，绿色 表示正确推理。

Algorithm 1: DURIT: 通过迭代训练实现理解与推理的解耦

Require: Dataset D_0 , Pretrained SLM R , Mapper M , Codebook C

Randomly initialize template tokens $\{T_1, \dots, T_n\}$ and query keys $\{k_1, \dots, k_n\}$

for iteration = 1 to N do

Step I: Problem Mapper Training

Cluster D_0 into n groups via kNN over problem representations z_i

for each question $Q_i \in D_0$ do

Select template token T_{t_i} based on cluster label t_i

Construct mapper input $x_i = [Q_i; T_{t_i}]$

Generate mapped question $Q'_i = M(x_i)$

Compute reward $r_i = r_{\text{acc}} + r_{\text{cheating}}$

Compute template similarity loss $\mathcal{L}_{\text{template-sim}}$

Compute key similarity loss $\mathcal{L}_{\text{key-sim}}$

Update M and C using total loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pg}} + \alpha_1 \mathcal{L}_{\text{key-sim}} + \alpha_2 \mathcal{L}_{\text{template-sim}}$$

end for

Step II: Self-Distillation Training

Generate normalized dataset $\mathcal{D}_1 = \{Q'_i = M(Q_i) | Q_i \in D_0\}$

for each $Q'_i \in \mathcal{D}_1$ do

Sample N responses y using frozen SLM $R(Q'_i)$

if answer (y_i) is correct then

Let $x_s = [Q_i, y_i]$ (student), $x_t = [Q'_i, y_i]$ (teacher), and add (x_s, x_t) to $\mathcal{D}_2$

end if

end for

Train R on $\mathcal{D}_2$ using the following loss:

$$\mathcal{L}_i = \frac{1}{l} \sum_{k=1}^l [(1 - \lambda)(-\log p_s(x_i^k)) + \lambda \text{KL}(p_t(x_i^k) \| p_s(x_i^k))]$$

Step III: Reinforcement Learning Training

Fine-tune R via GRPO on the original dataset D_0
end for
