

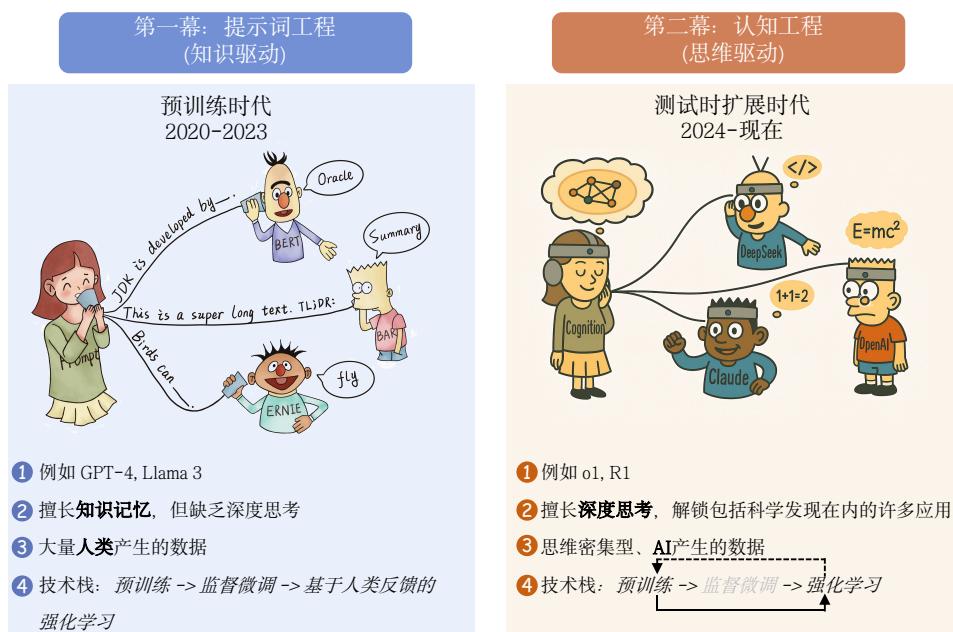
# 生成式 AI 第二幕： 测试时扩展技术驱动认知工程

夏世杰<sup>1,2,3</sup> 覃奕玮<sup>3</sup> 李学峰<sup>1,2,3</sup> 马琰<sup>3</sup> 樊润泽<sup>3</sup>  
 陈奕融<sup>3</sup> 邹皓阳<sup>1,2,3</sup> 周凡<sup>1,2,3</sup> 胡祥坤<sup>2,3</sup> 金嘉禾<sup>1,2,3</sup>  
 何彦衡<sup>1,2,3</sup> 叶懿芯<sup>1,2,3</sup> 刘一秀<sup>1,2,3</sup> 刘鹏飞<sup>1,2,3\*</sup>

<sup>1</sup> 上海交通大学, <sup>2</sup> 上海创智学院, <sup>3</sup> 生成式人工智能研究实验室 (GAIR)

## Abstract

第一代大语言模型——可称为生成式 AI 的“第一幕”(2020-2023 年)——通过海量参数与数据规模取得了惊人成就，却在知识时效性、浅层推理与受限认知过程方面存在根本局限。这一时期，提示工程成为人类与 AI 交互的主要界面，实现了自然语言层级的对话沟通。如今我们正迎来“第二幕”(2024 年至今)的崛起，模型通过测试时扩展技术，正从（潜空间中的）知识检索系统蜕变为思维构建引擎。这一新范式通过语言化的思维，建立了人机之间思维层级的连接。本文阐明了认知工程的概念基础，并阐释为何当前是其发展的关键契机。我们通过系统教程与优化实现方案，对这些前沿方法进行结构化拆解，推动认知工程的民主化进程，让每位从业者都能参与 AI 第二幕的演进。相关测试时扩展技术的论文合集将持续更新于 GitHub 仓库。



## 本文对哪些读者有帮助？

- 研究员: 提供推动领域发展的开放性问题和设计挑战 (参见 §6节、§10节)。
- 学生与初学者: 认知工程教程与代码示例 (参见 §9节)。
- 教育工作者: 结构化教学资源及测试时扩展的指导方案 (参见 §4节、§5节)。
- 投资者与决策者: 通过“第一/二幕”框架获得强化视野, 提供深度的认知洞察 (参见 §1节)。

\* 通讯作者

## 三大扩展阶段

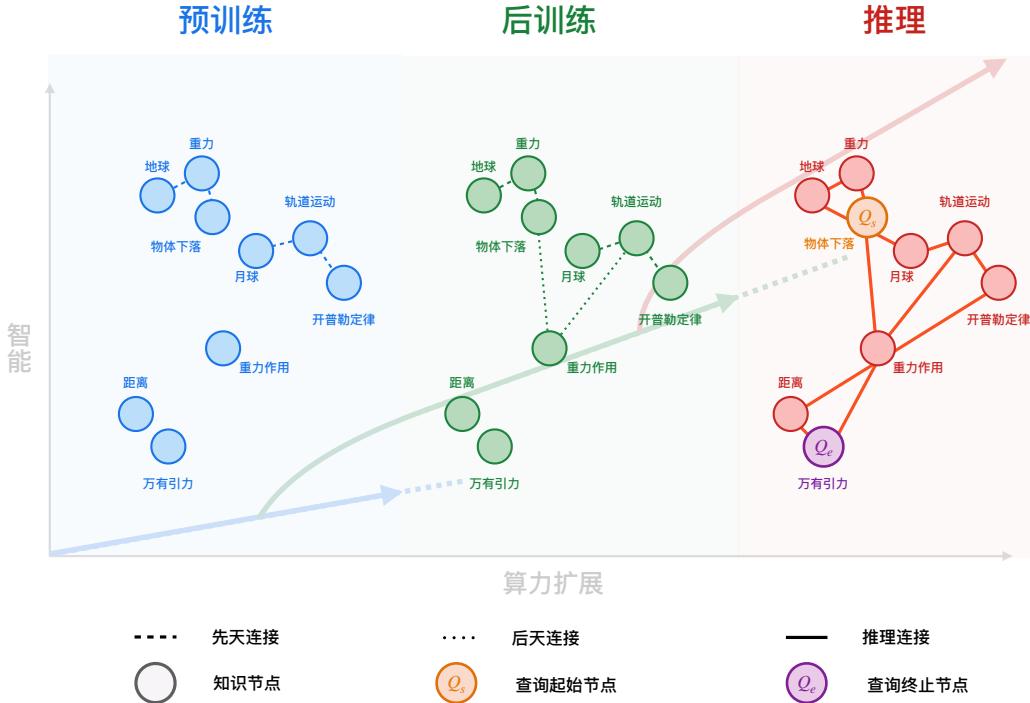


图 1: 知识表示的演进过程可分为三个阶段。预训练扩展（蓝色）形成了由基础物理概念构成的孤立知识岛，其间仅存在少量固有联系。后训练扩展（绿色）通过在相关概念间建立更复杂的学习连接，使这些知识岛变得更加紧密。测试时扩展（红色）则通过延长计算时间，在原先互不关联的概念间形成动态推理路径，从而实现了跨知识领域的多步推理能力。测试时扩展的关键作用在于，它能够在预训练和常规后训练后仍然孤立的知识岛之间架起桥梁，连接远距离的知识节点。

在展开正文论述之前，我们首先提出一个关于三大扩展阶段如何塑造模型认知能力的理论框架，并着重强调测试时扩展在这一过程中的核心地位。

**阶段一：预训练扩展——知识岛的雏形** 这是智能涌现的基础阶段，通过增加模型规模和训练数据量，模型获得了基础的知识获取能力。在预训练扩展过程中，我们观察“知识岛”的形成——即各个物理概念以松散关联的方式聚集成专业领域集群。这些概念虽已存在，但彼此间的联系有限，主要表现为图中蓝色虚线所示的内在关联。此时模型虽然掌握了这些物理概念，但尚未建立起强有力的连接，这限制了其在知识网络中进行复杂推理的能力。

**阶段二：后训练扩展——知识的密集连接** 后训练扩展阶段展示了微调如何使知识表示更加紧密。相同的物理概念通过学习连接（绿色虚线）形成了更丰富的关联关系。随着后训练在原先孤立的概念间建立起联系，物理知识网络变得更加完整。我们看到不同物理原理之间的连接适度增加，使得更复杂的关联成为可能。然而，这些连接主要集中在密切相关概念之间，仍缺乏建立远距离知识节点间多步推理路径的能力。

**阶段三：测试时扩展——认知通路的构建** 最终阶段代表了测试时扩展（或称“长思考”）带来的范式突破。这一创新方法使模型能够在原先连接薄弱的物理概念间建立强有力的推理路径（红色实线）。如图所示，从特定起始节点  $Q_s$  出发的查询，现在可以沿着知识网络中的复杂推理路径，最终在目标节点  $Q_e$  处获得全面解答。通过延长推理时的计算时间，模型能够在其物理知识表示中

---

探索更深层次的搜索空间，实现概念间的多步推理连接。测试时阶段展现出对引力原理的完整理解，使万有引力等基础概念能够与轨道运动、自由落体等具体应用建立有意义的联系。

**结论** 这一演进过程清晰地展示了计算扩展如何直接影响物理领域的知识表示和推理能力。预训练扩展奠定了物理知识基础，后训练扩展完善了相关概念间的连接，而唯有测试时扩展才能实现跨领域的复杂推理——这正是高级科学思维的特征。这一进展从根本上重新定义了 AI 的发展方向：不再仅仅是积累更多数据或参数，而是发展认知能力，使模型能够通过系统性推理驾驭物理原理的全部复杂性。测试时扩展代表着这一进程中的关键突破，使模型从单纯的知识存储库，转变为能够通过深度思考获得科学洞见的智能系统——这与人类专家通过持续思考解决复杂物理问题的认知过程高度相似。这一进展重新定义了人工智能的发展范式：不再局限于数据或参数的简单累积，而是致力于培养模型运用基本原理驾驭人类知识复杂性的能力。测试时扩展正是这一进程的前沿所在，它使 AI 系统能够像人类专家那样，通过深度思考获得对复杂问题的深刻认知。

## 实践者路线图：如何将测试时扩展技术应用到您的领域中？

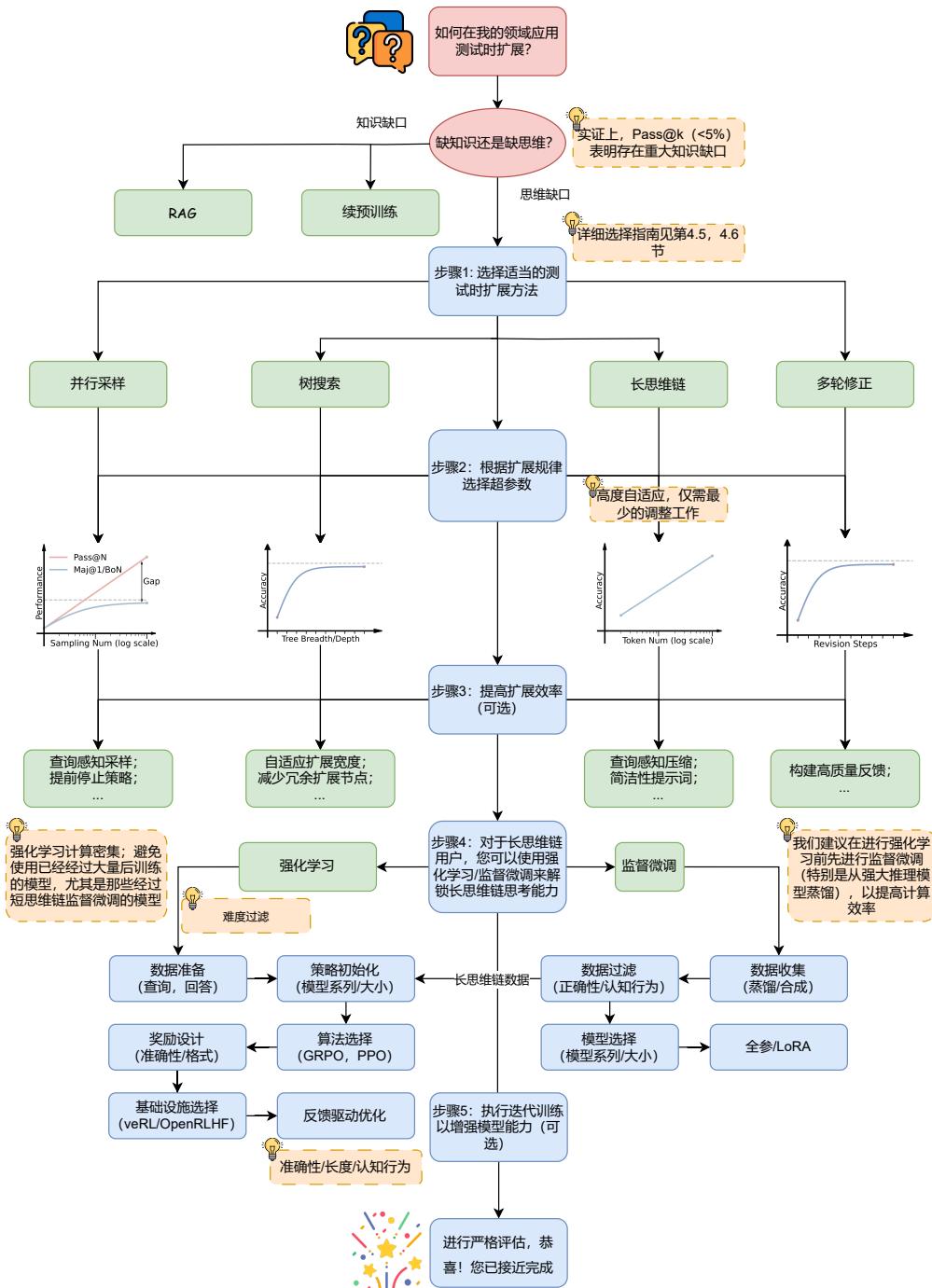


图 2: 在特定领域应用测试时扩展技术的工作流程。更多详细信息，请参阅主论文。

## 目录

<b>1 引言</b>	<b>7</b>
<b>2 认知工程的定义</b>	<b>8</b>
2.1 认知 . . . . .	8
2.2 工程方法论 . . . . .	8
2.3 认知工程 . . . . .	8
<b>3 为何是现在——技术基础</b>	<b>9</b>
3.1 认知工程的必要性 . . . . .	9
3.2 三大支柱 . . . . .	10
3.2.1 知识基础 . . . . .	10
3.2.2 测试时扩展技术 . . . . .	10
3.2.3 自训练技术 . . . . .	10
3.3 从理论到实践：前行之路 . . . . .	11
<b>4 方法——第一部分：测试时扩展方法</b>	<b>11</b>
4.1 并行采样 . . . . .	11
4.1.1 核心组件 . . . . .	11
4.1.2 扩展规律 . . . . .	13
4.1.3 提高扩展效率 . . . . .	13
4.2 树搜索方法 . . . . .	15
4.2.1 核心组件 . . . . .	15
4.2.2 扩展规律 . . . . .	17
4.2.3 提升扩展效率 . . . . .	17
4.3 多轮修正方法 . . . . .	18
4.3.1 核心组件 . . . . .	18
4.3.2 扩展规律 . . . . .	18
4.3.3 提升扩展效率 . . . . .	18
4.4 长思维链 . . . . .	19
4.4.1 核心组件 . . . . .	19
4.4.2 扩展规律 . . . . .	19
4.4.3 提升扩展效率 . . . . .	20
4.5 测试时扩展方法的比较 . . . . .	21
4.6 测试时扩展方法的集成 . . . . .	22
<b>5 方法——第二部分：测试时扩展的训练策略</b>	<b>25</b>
5.1 扩展强化学习 . . . . .	25
5.1.1 训练算法 . . . . .	25
5.1.2 奖励函数 . . . . .	26
5.1.3 策略模型选择 . . . . .	29
5.1.4 训练数据构建 . . . . .	29
5.1.5 多阶段训练 . . . . .	30
5.2 监督微调 . . . . .	30
5.3 迭代自强化学习 . . . . .	33
<b>6 进展如何——迄今的应用</b>	<b>34</b>
6.1 数学 . . . . .	35
6.2 代码 . . . . .	36
6.3 多模态 . . . . .	37
6.4 智能体 . . . . .	38

6.5 具身智能 . . . . .	39
6.6 安全对齐 . . . . .	41
6.7 检索增强生成 . . . . .	42
6.8 评估 . . . . .	43
<b>7 那又怎样？——从规模化到认知智能</b>	<b>44</b>
7.1 数据工程 2.0：认知数据工程 . . . . .	44
7.2 奖励与环境工程 . . . . .	45
7.2.1 奖励模型设计 . . . . .	45
7.2.2 认知环境设计 . . . . .	45
7.3 人机认知伙伴关系 . . . . .	45
7.4 研究加速 . . . . .	46
<b>8 基础设施</b>	<b>46</b>
8.1 强化学习 . . . . .	46
8.2 蒙特卡洛树搜索 . . . . .	47
<b>9 教程</b>	<b>48</b>
9.1 准备工作 . . . . .	48
9.2 启动 RL 训练 . . . . .	48
9.3 通过代码分析理解 RL 算法 . . . . .	49
9.4 结果 . . . . .	51
<b>10 未来方向</b>	<b>51</b>
<b>11 结论</b>	<b>53</b>

## 1 引言

近年来，以 GPT (OpenAI, 2023)、LLaMA (Meta, 2024, 2023) 和 Claude (Anthropic, 2024a) 为代表的大语言模型 (LLMs) 通过大规模预训练和微调，已成为强大的知识管理工具。这些模型在海量人类文本数据上训练，能够有效组织和系统化人类积累的知识。基于预训练数据、计算资源和模型参数的扩展范式 (Kaplan et al., 2020)，这些系统展现出自然语言对话、信息检索、内容生成和多领域问题解答等能力 (Zhao et al., 2023b; Wang et al., 2024g; Zheng et al., 2023a)。我们将这一代大语言模型称为生成式人工智能的“第一幕”，它带来了人机交互方式的根本变革。这一阶段的基石是“提示词工程”——通过精心设计的输入引导模型产生预期输出的技术 (Liu et al., 2021; Sahoo et al., 2024)。这项创新首次实现了用自然语言与人工智能交流，大幅降低了交互门槛。第一幕的核心目标是通过更大规模的训练数据来收集和组织现有知识。然而，这些模型仍存在明显局限：(i) **知识滞后**：主要学习训练数据中的高频信息，对新兴概念理解有限 (Huang et al., 2025b); (ii) **推理浅层**：虽能进行基本逻辑推理，但难以处理需要多步深度推理的问题 (Zhang et al., 2024d; Mirzadeh et al., 2024; Kambhampati, 2024); (iii) **思维局限**：面对新颖或开放性问题时缺乏类人思维的深度 (Wu et al., 2024d)。这些限制使得第一幕模型主要适用于知识检索和简单推理任务，距离通用人工智能 (AGI) 仍有很大差距。正如仅靠知识不足以发展人类智能，单纯积累信息也无法让 AI 系统接近人类智能水平——它们还需要发展深度思考和推理能力 (Newell et al., 1959, 1972)。

近期，人工智能领域正经历一场深刻的范式转变。以“测试时扩展”为核心的新技术 (OpenAI, 2024; DeepSeek-AI et al., 2025; Snell et al., 2024) 正在重新定义大语言模型的能力边界，开启了生成式人工智能的第二幕——**认知工程**。

认知工程是通过超越传统预训练方法的测试时扩展范式，系统性构建人工智能思维能力的方法论。它融合人类认知模式提炼和 AI 自主发现 (如强化学习)，有意识地培育人工系统的深度认知能力。

认知工程的核心在于两个关键概念的结合：“认知”在此语境下不仅指知识获取，更强调深度推理能力——包括复杂逻辑推演、深思熟虑、概念关联和新见解生成。它涵盖元认知过程 (Metcalfe and Shimamura, 1994)，使系统不仅能理解“是什么”，更能理解“为什么”和“如何做”，这正是人类智能的精髓。“工程”则代表一种建构性方法。它突破简单规模扩展的局限，通过训练方法（如强化学习）和推理优化（如延长计算时间）的有针对性干预，主动塑造认知能力。

认知工程标志着大语言模型开发的全方位技术转型。在推理层面，从设计提示词模板获取知识，转向构建测试时扩展策略以深度探索知识空间。这一转变要求对扩展策略的组成、特性和效率进行严谨分析，凸显结构化工程方法的重要性。在训练层面，认知工程将计算资源从知识导向的预训练，转向通过人类认知数据学习和强化学习等技术来发展深度思维。这标志着第二幕中认知交流的双向性：不仅人类可以教导 AI 处理复杂问题，AI 也能通过强化学习自主发现新的认知模式。例如，类似 AlphaGo 著名的“第 37 步”时刻 (Silver et al., 2016)，AI 展现出超越人类直觉却卓有成效的思考方式。这些 AI 发现的认知策略有望拓展人类认知边界，开辟新的研究路径。这种双向认知交流预示着我们正步入智能共生的新时代。

本文将深入探讨认知工程的定义内涵、技术基础与应用前景。首先，我们将明确认知工程的概念要义 (§2)，并阐释当前发展阶段的关键性 (§3)；随后详细解析测试时扩展的技术原理 (§4) 及其多样化训练策略 (§5)；进而探讨认知工程为 AI 研究带来的系统性变革及现有应用成果 (§6)；最后从技术实现层面延伸，分析其深层影响 (§7)，探讨支撑基础设施 (§8)，并展望认知工程的未来发展方向 (§10)。文中还附有测试时扩展的实践教程与代码示例 (§9)。通过系统性的论述，我们试图勾勒生成式人工智能第二幕的发展轮廓，为学界与业界从业者提供这一新范式下的思考框架与实践指南。

## 2 认知工程的定义

“认知工程”这一术语代表了人工智能发展过程中的重大范式转变。为理解这一新兴领域的本质，我们可以借助 DIKW（数据-信息-知识-智慧）金字塔理论 (Zeleny, 1987; Ackoff, 1989) 作为概念框架，探究认知工程如何实现从知识到智慧的跨越。

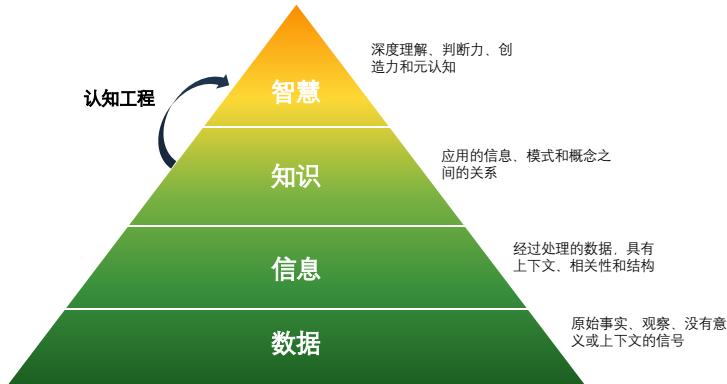


图 3: DIKW 金字塔及其与认知工程范式的关系

### 2.1 认知

DIKW 理论将认知过程描述为层级式转化：从原始数据到情境化信息，再到可应用知识，最终形成深刻智慧。这一框架为理解认知工程提供了深刻洞见。数据层面是缺乏内在意义的原始事实和观察；信息层面是经过处理和组织、具有上下文结构的数据；知识层面表现为对信息的理解和应用，包括对规则、模式和关系的掌握；而智慧层面则体现为对知识的深刻领悟，涉及判断力、创造力和元认知能力。传统 AI 系统主要停留在数据和信息层面，第一代大语言模型在知识层面取得重大突破，而认知工程则是向智慧层面迈进的关键步骤。

在心理学和认知科学中，认知指生物体获取和处理信息、形成知识并应用于问题解决的复杂心智过程 (Von Eckardt, 1995; Núñez et al., 2019)。但认知工程所追求的并非这种基础认知能力，而是 DIKW 所描述的智慧层面的认知——理解深层原理、进行创造性思维和展现判断力的能力。这种深度认知不仅关乎“知道是什么”（即知识），更涉及“知道为什么”和“知道怎么做”（即智慧）。认知的特征在于深度思考能力——能够进行多层次、复杂推理并探索多种解决路径，以及元认知能力——能够反思自身的思维过程。它包含跨领域连接知识以产生新见解的创造性推理，将既有模式应用于新情境的认知适应性，以及从具体实例中提取高阶原则的概念抽象能力。这些能力共同构成了人类智能的核心，是人类在科学发现和技术创新中不断进步的基础。

### 2.2 工程方法论

工程作为一种方法论，本质上是设计、构建和优化系统以解决特定问题的组织化方法。在 DIKW 框架下，工程方法可视为人类有意识地引导系统从数据层面向智慧层面攀升的过程。在认知工程中，这种有意识构建的强调体现在对训练方法和推理优化的针对性干预上，而非单纯依赖规模扩展。

### 2.3 认知工程

结合认知与工程的概念，并从 DIKW 理论视角出发，我们可以更深刻地定义认知工程：

认知工程是一种系统化方法论，通过特定设计模式、训练策略和计算资源分配，构建和优化 AI 系统从 DIKW 金字塔的知识层面向智慧层面跃迁的能力。它使 AI 系统能够进行深度思考、复杂推理和创造性问题解决，展现出类人的智慧层面认知特征。

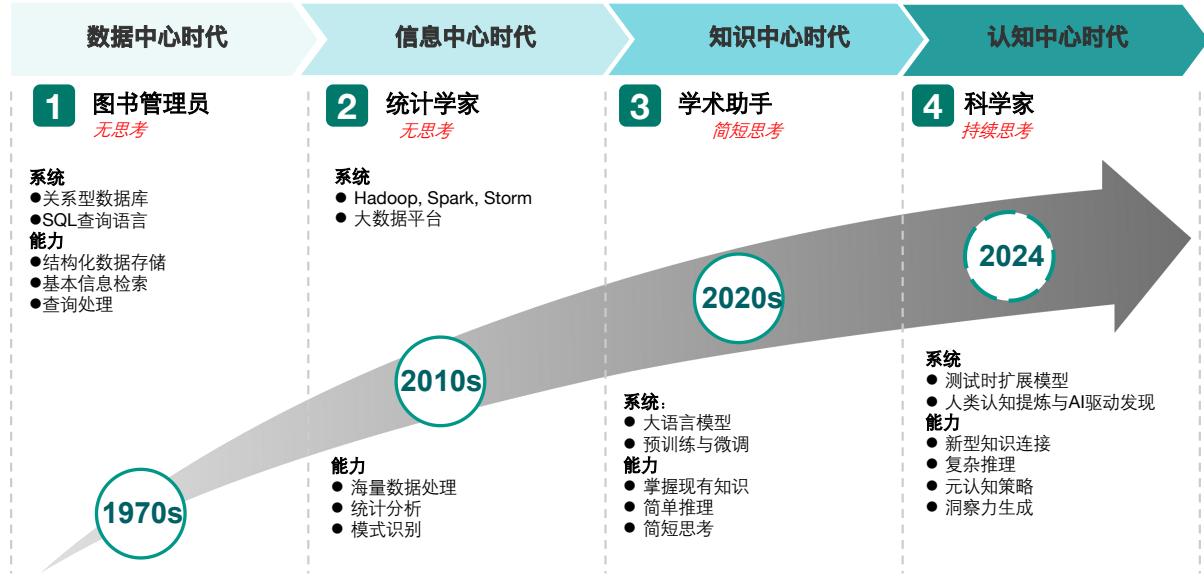


图 4: 人工智能工程范式的演进

认知工程与传统 AI 开发方法的关键区别在于其方法论特征：

- 从涌现到构建：**传统 AI 发展路径主要依赖通过增加数据规模、模型参数和训练计算量自然“涌现”能力，主要在 DIKW 的数据和信息层面累积；认知工程则采用更具主动性的构建方法，有意识地设计促进知识向智慧转化的机制。
- 从行为模仿到思维模仿：**传统模型主要通过模仿人类输出行为来学习，停留在 DIKW 的知识层面；认知工程则聚焦模仿人类思维过程，直接针对智慧层面的认知特征。
- 从静态知识到动态智慧：**传统模型训练后能力相对固定，认知工程则强调 AI 系统在推理过程中的动态思维能力，能根据问题复杂度调整思考深度和资源分配。
- 从知识检索到知识创造：**传统模型主要检索和组合既有知识，认知工程则旨在使 AI 系统通过深度思考产生新见解和发现，实现 DIKW 智慧层面的创造性特征。

在 DIKW 框架内，我们可以清晰区分 AI 发展的两个阶段：第一阶段主要解决了信息向知识转化的问题，而第二阶段开始系统探索从知识向智慧的飞跃。如果说传统大语言模型训练方式是“自然式”的——通过海量资源和时间投入让能力自然涌现，那么认知工程则是“文明式”的——通过有规划的设计和干预来塑造 AI 系统的思维能力，正如人类文明有意识地培养智慧。

图 4 展示了人工智能的发展也可理解为不同工程范式的演进历程，每个范式都代表了能力和应用的根本转变。这一进程完美体现了从“数据到智慧”的自然旅程 (Zeleny, 1987; Ackoff, 1989)，每个阶段都在前一范式基础上拓展边界。认知工程是通向 AGI 道路上的关键阶段，因为它解决了深度思维能力这一根本需求，而这种能力无法仅通过知识积累自然涌现。

### 3 为何是现在——技术基础

#### 3.1 认知工程的必要性

认知工程的兴起并非偶然，而是对 AI 发展在 DIKW 金字塔中遭遇“智慧鸿沟”的直接回应。尽管在知识检索、内容生成和基础推理方面取得显著进展，大语言模型在智慧层面仍存在明显缺陷：

**复杂推理的局限性** 当前模型在需要多步深度推理的问题上表现欠佳 (Zhang et al., 2024d; Mirzadeh et al., 2024; Kambhampati, 2024)。即使最先进的模型也难以可靠完成数学证明、复杂科学问题求解或多维分析 (Yang et al., 2024b; Rein et al., 2023)。这些任务要求模型将问题分解为子问题、探索多种推理路径并进行深度逻辑分析——仅靠扩大预训练数据无法获得这些能力。

**知识更新与创造的挑战** 预训练模型的知识在训练结束时固化，无法自动适应新发展。更重要的是，它们难以产生真正原创的洞见或发现——科学发现的本质不在于理解已知事实，而在于提出新假设、设计实验方法并从结果得出新结论。这种知识创造能力需要超越简单的知识检索和模式识别。

**提升的应用需求** 随着 AI 应用从简单任务扩展到复杂决策、科学的研究和创造性工作，对系统智慧层级能力的要求也随之提高 (OpenAI, 2025b,a)。用户不再满足基于统计模式的答案（知识层面），他们期待 AI 能提供深思熟虑的分析、多视角考量与创新见解（智慧层面）。

## 3.2 三大支柱

认知工程在此特定时刻兴起，得益于多项技术突破的同步成熟。这些突破共同创造了必要条件，使 AI 得以从知识管理迈向深度认知能力。认知工程的崛起建立在三大关键技术支柱之上：

### 3.2.1 知识基础

认知工程的第一个基础是大语言模型获取知识方式的根本变革。现代基础模型不仅实现了训练数据量的指数级增长（如 Llama 2 的 2 万亿 token 训练规模 (Meta, 2023)），更重要的是发生了质的转变。预训练数据已从简单网络爬取文本发展为精心构建的知识语料库 (Shao et al., 2024; Zhou et al., 2024b; Wang et al., 2023d; Yang et al., 2024a)，整合了科学文献与技术文档、数学教材与习题集、多语言编程代码库以及专业领域结构化知识，形成了远比以往丰富的知识生态。这一全面的知识基础是认知工程的必要前提——没有这些内嵌的丰富知识，模型将缺乏深度思考所需的原材料。

### 3.2.2 测试时扩展技术

认知工程的第二个关键支柱是对推理阶段计算资源分配方式的根本重构——我们称之为“测试时扩展”。传统推理方法受限于固定输出长度和单次生成范式。近期一系列技术突破显著扩展了模型的推理能力：思维链提示 (CoT) (Wei et al., 2022) 引导模型像人类解题那样逐步推理；树状搜索 (Yao et al., 2023a; Hao et al., 2023; Feng et al., 2023) 允许同时探索多条推理路径而非局限于单一思路；自我修正与验证技术 (DeepSeek-AI et al., 2025; Kumar et al., 2024; Qu et al., 2024) 进一步强化这些能力，使模型能评估自身推理、识别潜在错误并改进方法——模拟人类元认知过程。这些创新共同提供了可理解为“认知工作空间”的环境，让模型能系统探索其知识——如同人类需要草稿纸解决复杂问题或需要时间“深入思考”。

### 3.2.3 自训练技术

认知工程的第三支柱是先进的自训练方法。仅通过专家人类认知数据来开发模型的复杂认知能力存在固有扩展限制。自训练技术不仅提供了激发认知能力的替代路径，更通过 AI 自主发现策略创造了超人类性能的可能性。如 DeepSeek-R1 (DeepSeek-AI et al., 2025) 及后续研究 (Gandhi et al., 2025; Yu et al., 2025a) 所示，使用可验证奖励的强化学习训练使模型能掌握包括反思、回溯和验证在内的复杂认知行为。通过这一过程，模型学会根据问题难度动态分配计算资源，有效内化测试时扩展技术。此外，对测试时扩展方法生成的推理轨迹进行迭代自训练可实现持续改进 (Zelikman et al., 2022; Feng et al., 2023; Xiong et al., 2025)，使 AI 系统逐步提升问题解决能力。

### 3.3 从理论到实践：前行之路

理论基础已经就位，但将其转化为实践需要探索复杂的具体方法与技术路径。当前实现认知工程最直接且前景广阔实践路径是测试时扩展技术——这类方法通过优化模型在推理时的计算资源分配来实现深度推理。这些技术构成了认知工程理论承诺与现实应用之间的实践桥梁。通过理解和完善这些技术，我们才能系统构建真正“会思考”而非仅“会预测”的AI系统。下一节我们将深入探讨实现测试时扩展的具体机制，分析不同方法如何应对扩展和深化AI推理过程的根本挑战。这番探索不仅将揭示这些方法的技术细节，更将展现其认知意义及其在更广阔认知工程范式中的角色。

## 4 方法——第一部分：测试时扩展方法

给定查询 $q$ 和生成器 $g$ ，测试时扩展方法可以抽象为一种搜索策略 $M$ ，该策略指导生成器 $g$ 找到最优响应：

$$y \sim M(\cdot|q, g, \phi) \quad (1)$$

其中， $\phi$ 表示任何额外的输入，例如评分函数 $v$ （也称为价值函数、奖励模型或验证器<sup>1</sup>）以及策略的超参数。

**扩展规律** 对于任何测试时扩展方法，都存在相应的扩展维度 $\lambda$ （在 $\phi$ 内），这些维度直接决定了推理过程中的计算成本。方法 $M$ 的扩展规律描述了 $\lambda$ 与性能之间的关系。

**扩展效率** 给定计算预算<sup>2</sup>  $C$ ，我们定义一个抽象函数 $f : C \times M \rightarrow \mathbb{R}$ ，该函数将计算预算 $C$ 和测试时扩展方法 $M$ 映射到性能。扩展效率衡量了相对于计算预算的性能：

$$\text{效率} = \frac{f(C, M)}{C} \quad (2)$$

提高方法 $M$ 效率的高层策略可以分为以下两类：<sup>3</sup> 1) 优化单个测试时扩展方法：这包括在计算预算约束下仔细选择和调整组件，或利用额外的训练时计算来专门优化测试时扩展的模型；2) 结合多种测试时扩展方法：这包括同时结合多种方法或根据不同的上下文选择合适的测试时扩展方法。

在接下来的部分中，我们将研究四种主要的测试时扩展方法：并行采样（§4.1）、树搜索（§4.2）、多轮修正（§4.3）和长链推理（§4.4）。对于每种测试时扩展方法，我们将涵盖构建方法、扩展规律以及如何从个体优化角度提高扩展效率。此外，我们将在多个维度上比较这些测试时扩展方法（§4.5），并讨论如何有效地结合它们以提升性能（§4.6）。

### 4.1 并行采样

#### 4.1.1 核心组件

并行采样算法从生成器中为同一查询独立采样一组候选响应 $\mathcal{Y} = \{y_i\}_{i=1}^N$ ，其中 $N$ 是采样数量，并从这些响应中选择目标响应或答案。这种方法可以概念化为在知识空间中进行全局搜索（Snell et al., 2024）。选择方法如下：

- **F1: 最佳 N 采样 (BoN)**。该方法使用评分函数 $v$ 评估每个响应，并选择得分最高的响应：

$$y^* = \arg \max_{\tilde{y} \in \mathcal{Y}} v(\tilde{y}) \quad (3)$$

<sup>1</sup>这些术语在上下文中略有差别，我们为每种方法选择最合适术语。

<sup>2</sup>可以通过FLOPs、运行时间、令牌数量等来衡量。

<sup>3</sup>在本节中，我们不考虑模型压缩技术（如模型量化）或基础设施方面的推理加速，因为它们与方法设计正交。

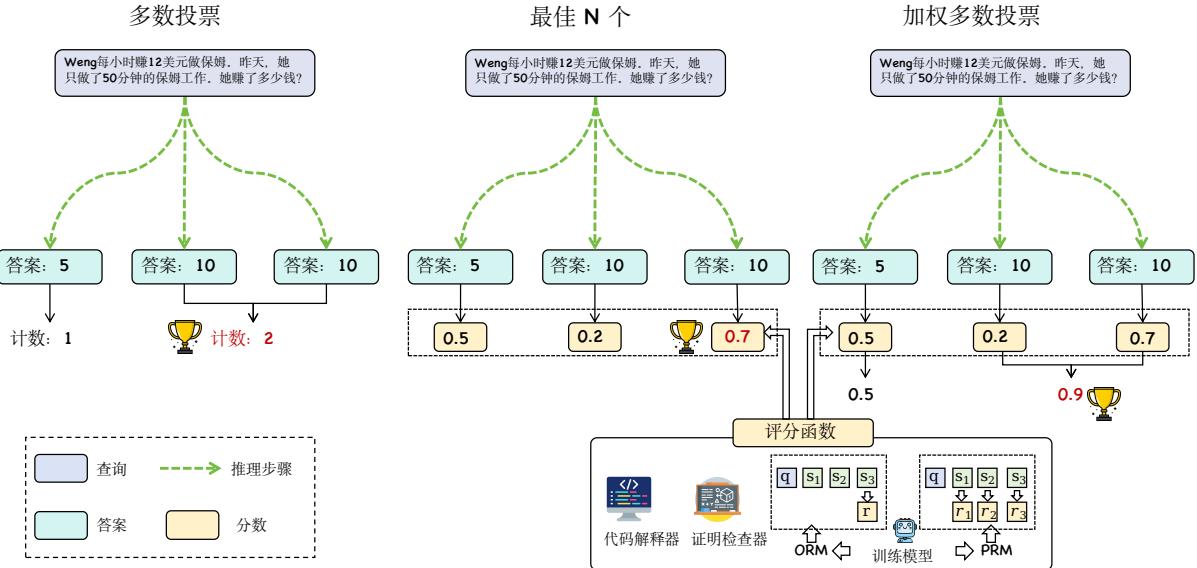


图 5: 并行采样选择方法的示意图：最佳 N 采样 (F1)、多数投票 (F2) 和组合策略 (F3)。

评分函数  $v$  可以是直接验证响应有效性的外部工具，例如代码解释器 (Li et al., 2022; Chen et al., 2023a) 或数学证明检查器 (Brown et al., 2024)。对于缺乏验证工具的任务， $v$  可以是一个专门训练的模型。例如，Cobbe et al. (2021) 训练结果奖励模型 (ORM) 来对整个响应进行评分，而 Lightman et al. (2023); Uesato et al. (2022) 训练过程奖励模型 (PRM) 对响应中的每一步进行评分，并应用聚合函数确定整体响应得分。Self-Certainty (Kang et al., 2025) 通过利用生成器固有的概率分布进行评分，消除了对额外奖励模型的需求。

- **F2: 多数投票。** 多数投票（或自一致性 (Wang et al., 2023c)）从候选答案中选择出现频率最高的答案：

$$y^* = \arg \max_{\tilde{y} \in \mathcal{Y}} \sum_{\hat{y} \in \mathcal{Y}} g(\tilde{y}, \hat{y}) \quad (4)$$

$$g(\tilde{y}, \hat{y}) = \begin{cases} 1 & \text{如果 } \tilde{y} \text{ 与 } \hat{y} \text{ 等价,} \\ 0 & \text{否则,} \end{cases} \quad (5)$$

其中  $g$  是一个自动评分函数，首先从响应中提取答案并检查等价性。虽然这种方法轻量级，但需要容易进行答案等价比较的要求限制了其在开放任务中的适用性。Universal Self-Consistency (Chen et al., 2023b) 使用大语言模型本身在多个候选答案中选择最一致的答案，尽管模型的有限上下文窗口大小对于大采样数量仍然存在挑战。

- **F3: 结合投票和评分策略。** 评分策略可以帮助选择目标低频响应，但高度依赖于评分函数的可靠性，而投票策略则提供了更强的鲁棒性，但上限更为固定。这种组合方法利用了两种方法的优势，以实现更鲁棒的选择 (Sun et al., 2024b)。例如，加权多数投票 (Uesato et al., 2022; Liu et al., 2023d) 根据每个簇中得分的总和对答案簇重新排序，并选择得分最高的答案簇：

$$y^* = \arg \max_{\tilde{y} \in \mathcal{Y}} \sum_{\hat{y} \in \mathcal{Y}} g(\tilde{y}, \hat{y}) v(\hat{y}) \quad (6)$$

图 5 展示了这些选择方法。

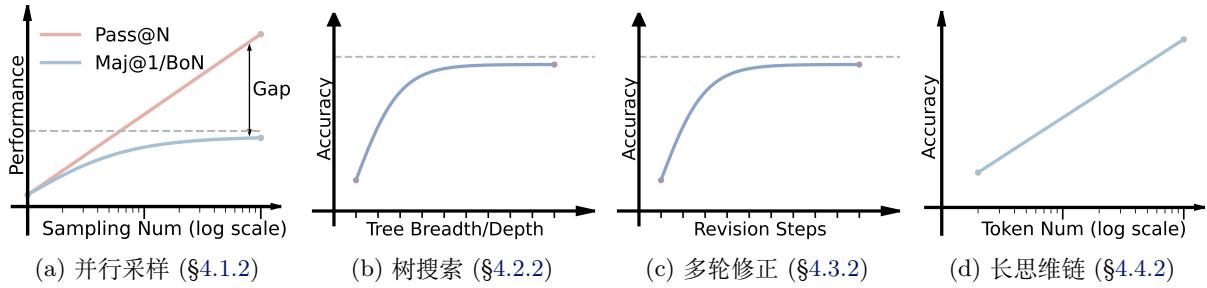


图 6: 每种测试时扩展方法的扩展维度与性能之间的关系。

#### 4.1.2 扩展规律

并行采样的主要扩展维度是采样数量  $N$ 。我们研究了  $N$  与各种性能指标之间的关系。具体来说，我们关注两类指标：Pass@ $N$ ，表示在  $N$  个候选响应中至少生成一个正确响应的概率；以及 Maj@1 或 BoN 等指标，它们衡量并行采样的实际性能。

**$N$  与 Pass@ $N$  之间的单调增长关系** Brown et al. (2024) 研究了不同模型和任务中  $N$  与 Pass@ $N$  之间的关系。Pass@ $N$  随着采样数量的增加而稳步增长。此外，两者之间的关系通常是线性对数关系，如图 6a 所示，类似于训练时扩展规律 (Kaplan et al., 2020)。

**扩展 Pass@ $N$  并不直接转化为实际性能提升** 尽管 Pass@ $N$  随着采样数量的增加而持续改进是令人鼓舞的，但该指标与实际性能之间仍存在差距。这种差距存在的原因有几个。首先，只有当存在适当的工具从样本集中选择正确响应时，性能改进才能实现。然而，对于大多数任务来说，完美的验证器并不存在。正如 Brown et al. (2024) 所观察到的，当使用 ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024d) 作为评分模型时，Pass@ $N$  与实际指标（如 Maj@1 或 BoN）之间会出现显著差异（见图 6a）。其次，验证器本身可能被欺骗。代码可能通过单元测试，但在额外的测试用例中失败 (Stroebel et al., 2024)，或者数学解决方案可能通过错误的推理得出正确答案 (Xia et al., 2024)，从而导致假阳性问题。Stroebel et al. (2024) 观察到，在代码任务中，假阳性率随着 Pass@1 准确率的降低而增加，并得出结论：即使计算资源无限，这也对基于重采样的推理扩展的准确率设置了上限。对于实际应用方法（如多数投票或评分方法），性能往往会饱和 (Brown et al., 2024; Wu et al., 2024c)，甚至可能随着采样数量的增加而下降 (Chen et al., 2024c)，这是由于不完美的验证器导致的。

#### 4.1.3 提高扩展效率

提高并行采样扩展效率的策略如下：

**查询感知采样** 对所有查询应用固定的采样数量并不是最优的，因为困难问题需要更多采样，而简单问题则需要较少采样。这类方法根据查询的难度自适应地调整采样数量以提高采样效率。Chen et al. (2024c) 根据模型的不确定性将查询分为简单和困难情况，并相应应用不同的采样数量。DSC (Wang et al., 2024f) 提示模型对查询难度进行排名，并根据排名分配采样数量。

**提前停止策略** 该方法在采样过程中估计响应的质量，并通过利用先验知识或模型估计决定何时提前停止采样。它包括在小窗口大小内观察到答案相同或符合预定义分布时终止采样 (Aggarwal et al., 2023; Li et al., 2024e; Wan et al., 2024)，或者训练生成器本身估计响应的置信度，并在观察到高置信度响应时停止 (Huang et al., 2025a)。此外，Speculative Rejection (Sun et al., 2024a) 和 ST-BoN (Wang et al., 2025e) 提出并行采样响应，并在响应具有低奖励模型得分或自估计一致性得分时停止解码以提高效率。

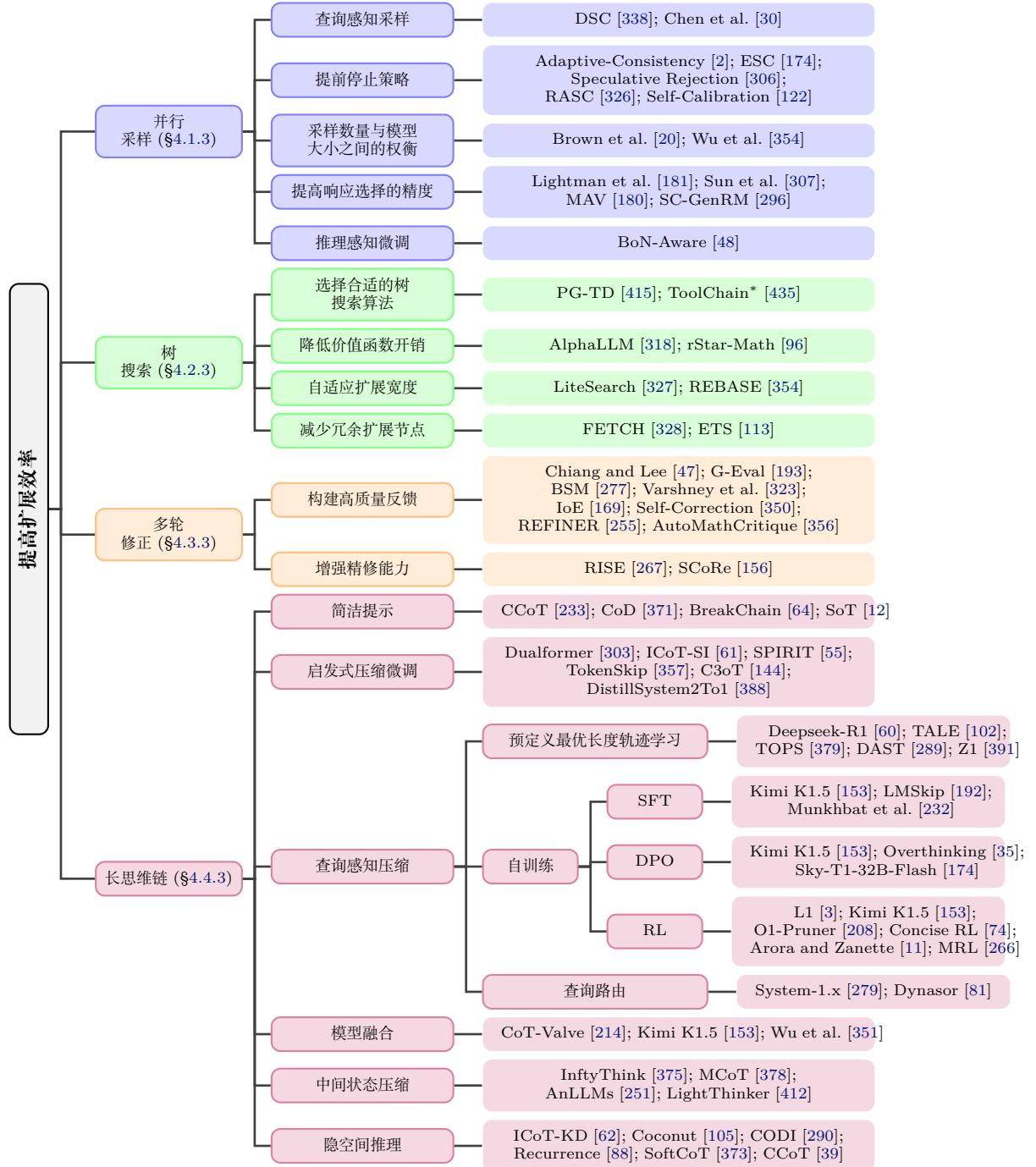


图 7: 提高扩展效率的方法。

**采样数量与模型大小之间的权衡** 在固定的推理计算预算下，考虑到不同模型大小的计算成本不同，使用较大模型和较少采样与较小模型和较多采样之间存在权衡。Brown et al. (2024) 观察到，较大模型在代码任务中表现更好，而较小模型在数学任务中更有效。Wu et al. (2024c) 进一步发现，对于数学任务，虽然较小模型是最优的，但随着推理计算的增加，它们的性能也会更早饱和。

**提高响应选择的精度** 考虑到有效选择机制的重要性，一些工作专注于提高响应选择的精度。Lightman et al. (2023) 发现 PRM 在 BoN 设置中优于 ORM。Sun et al. (2024b) 发现，当使用大采样数量时，加权多数投票的性能优于多数投票或 BoN。MAV (Lifshitz et al., 2025) 使用多个验证器评估响应质量，并在生成器和验证器的总计算预算较高时，比单个验证器实现更好的性能。

## 4.2 树搜索方法

表 1: 树搜索相关工作的系统梳理。本表仅包含推理阶段方法，结合训练策略的工作将在 §5.3 讨论。在状态评估器列中，**E1** 表示自评估，**E2** 表示专用训练模型，**E3** 表示动作似然，**E4** 表示自治分数，**E5** 表示推演评估。

工作	应用领域	搜索空间	价值函数	搜索算法
Pangu (Gu et al., 2023)	知识库问答	步骤级	E2	束搜索
PG-TD (Zhang et al., 2023)	代码生成	词元级	E5	MCTS
ToT (Yao et al., 2023a)	24 点游戏/写作/填字游戏	步骤级	E1	广度优先/深度优先
GuidedDecoding (Xie et al., 2023)		步骤级	E1	束搜索
RAP (Hao et al., 2023)	逻辑推理	步骤级	E1/E3/E4	MCTS
PPO-MCTS (Liu et al., 2023b)	对齐任务	词元级	E2	MCTS
LATS (Zhou et al., 2023)	编程/推理	步骤级	E1/E4/E5	MCTS
ToolChain* (Zhuang et al., 2023)	工具调用/推理	步骤级	E1/E3/E4	A* 算法
MindStar (Kang et al., 2024a)	数学推理	步骤级	E2	束搜索
Q* (Wang et al., 2024b)	数学/代码	步骤级	E2/E5	A* 算法
LiteSearch (Wang et al., 2024a)	数学推理	步骤级	E2	束搜索
MCTS <sub>r</sub> (Zhang et al., 2024b)	数学推理	解级	E1	MCTS
REBASE (Wu et al., 2024c)	数学推理	步骤级	E2	束搜索
SearchAgent (Koh et al., 2024)	网页代理	步骤级	E1	A* 算法
rStar (Qi et al., 2024)	数学推理	步骤级	E4/E5	MCTS
PLANSEARCH (Wang et al., 2024c)	代码生成	步骤级	-	束搜索
RethinkMCTS (Li et al., 2024d)	代码生成	步骤级	E1/E5	MCTS
SC-MCTS* (Gao et al., 2024b)	积木世界	步骤级	E1/E3	MCTS
LLaMA-Berry (Zhang et al., 2024c)	数学推理	解级	E2	MCTS
ETS (Hooper et al., 2025)	数学推理	步骤级	E2	束搜索

**推理感知微调** Chow et al. (2024) 克服了 BoN 采样中不可微的 argmax 运算符，并开发了 BoN-Aware 微调以直接优化并行采样性能。

## 4.2 树搜索方法

### 4.2.1 核心组件

树搜索方法将问题建模为树结构上的搜索过程。在特定树搜索算法的引导下，生成器在搜索空间  $S$  中进行探索，评估不同解题路径的价值。该框架能显著增强模型的有序规划能力。以下对各组件进行详细说明。

**搜索空间** 搜索空间定义了树节点的粒度，直接影响搜索效率。其分类如下：

- **S1：词元级。** 词元级搜索能提升候选解的最优性，但由于粒度过细会导致高昂计算成本。该方法适用于对单个词元错误容忍度低的场景。PG-TD (Zhang et al., 2023) 在代码任务中采用蒙特卡洛树搜索 (MCTS) 进行词元级搜索，因为代码的细微改动可能导致错误。PPO-MCTS (Liu et al., 2023b) 通过词元级搜索提升回答的有用性和无害性。
- **S2：步骤级。** 步骤级搜索平衡了粒度与效率，是最常用的方法。”步骤”的定义因任务而异：可以是推理问题解中的句子 (Yao et al., 2023a; Xie et al., 2023; Hao et al., 2023)、模拟世界中的动作 (Gu et al., 2023; Zhuang et al., 2023)、代码行 (Wang et al., 2024c)，或是提出的计划/假设 (Yao et al., 2023a; Wang et al., 2024e, 2023b)。
- **S3：解级。** 解级搜索将树节点扩展视为对完整解的更新 (如批评与修订 (Zhang et al., 2024b,c))。该方法与后文的多轮修正框架存在重叠，我们视其为两种方法的融合。

**价值函数** 价值函数用于评估候选节点的价值以进行剪枝或利用。常用构建方法包括：

- **E1：自评估。** 通过精心设计的提示词直接要求生成器评估节点价值。ToT (Yao et al., 2023a) 提出独立评估节点或跨节点投票。Xie et al. (2023) 采用选择题形式的提示词以校准预测。
- **E2：专用训练模型。** 为降低评估噪声，使用专训的大语言模型进行评估 (Gu et al., 2023; Kang et al., 2024a)。包括过程奖励模型 (PRM) 和词元级价值函数。PRM 训练方式有：1) 人工标注：Lightman et al. (2023); Uesato et al. (2022) 雇佣标注员标记步骤正确性，成本高昂且难

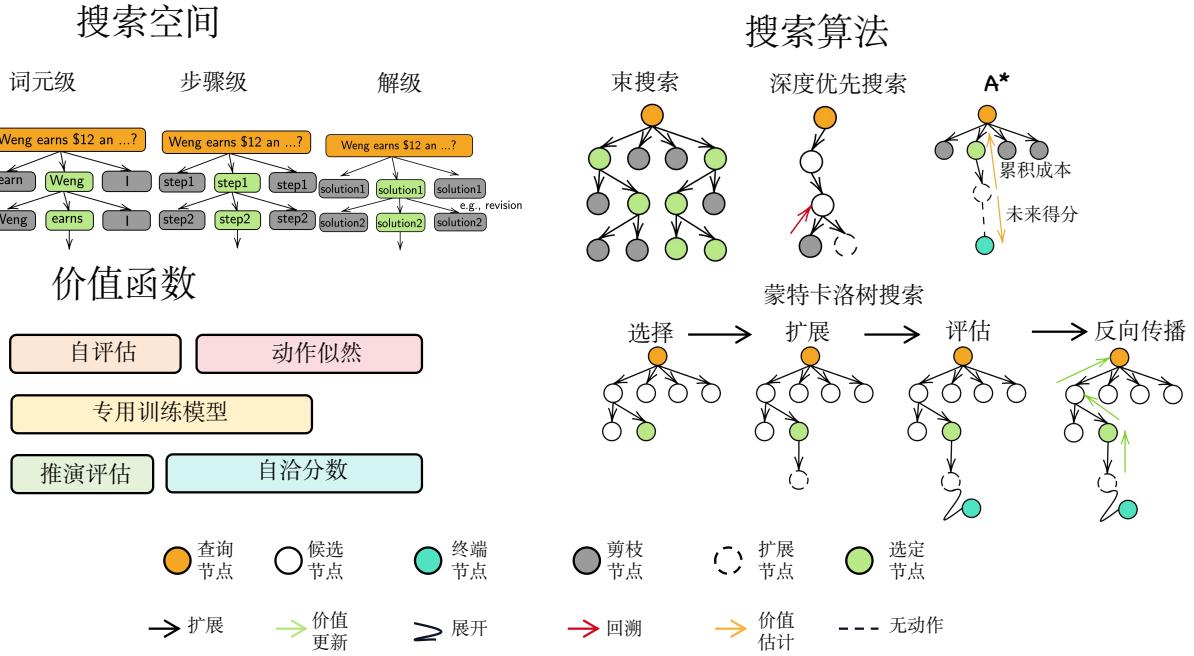


图 8: 树搜索核心组件示意图

以避免噪声; 2) 蒙特卡洛采样: Wang et al. (2023a, 2024i); Havrilla et al. (2024); Luo et al. (2024) 通过从当前步骤展开多个完成路径来估计正确率。OmegaPRM (Luo et al., 2024) 采用二分定位错误并将数据收集融入搜索过程; 3) 从 ORM 推导 PRM: Lu et al. (2024b) 利用 ORM 通过检测置信度变化自动生成过程标注。Yuan et al. (2024a) 理论证明 PRM 可通过奖励参数化从 ORM 导出。词元级价值函数可直接来自训练后阶段 (Liu et al., 2023b) 或蒙特卡洛采样数据训练 (Lee et al., 2024)。

- **E3: 动作似然。** 利用生成器执行特定动作 (即树节点) 的概率估计节点价值 (Hao et al., 2023; Gao et al., 2024b)。
- **E4: 自治分数。** 中间节点的出现频率可反映模型置信度 (Qi et al., 2024; Zhuang et al., 2023; Zhou et al., 2023)。LATS (Zhou et al., 2023) 结合自生成分数与自治分数。rStar (Qi et al., 2024) 将自治分数作为终端节点奖励。
- **E5: 推演评估。** 在 MCTS 等算法中, 中间节点价值可通过推演估计 (Qi et al., 2024)。终端状态奖励可来自代码解释器等外部工具或前述评估方法。

**搜索算法** 搜索算法定义树节点的操作规则, 具体实现包括:

- **A1: 广度优先搜索 (BFS)。** 常用变体包括束搜索和 A\* 算法。束搜索每层生成  $k$  个候选节点, 根据节点价值选择前  $m$  个 (Gu et al., 2023; Yao et al., 2023a; Xie et al., 2023)。A\* 算法 (Hart et al., 1968) 计算累计成本与未来分数之和, 选择最小值节点。ToolChain\* (Zhuang et al., 2023) 依赖启发式函数, Q\* (Wang et al., 2024b) 利用 PRM 和推演方法估计。
- **A2: 深度优先搜索 (DFS)。** DFS 优先探索最有希望的节点直至不满足条件或达到终局, 再回溯探索替代路径 (Yao et al., 2023a)。Long (2023) 在数独解题中实现 DFS, 通过检查器模块验证部分解的有效性。

### 4.3 多轮修正方法

表 2: 多轮修正工作的系统梳理。微调列标注方法是否需要额外训练，自反馈与自精修列中的✓ 表示与初始生成器共享参数但采用不同角色提示。

工作	反馈源		精修方式		微调
	自反馈	外部	自精修	外部	
Self-Correction (Welleck et al., 2023)	✗	✗	✗	专训模型	✓
Self-refine (Madaan et al., 2023)	✓	✗	✓	✗	✗
Reflexion (Shinn et al., 2023)	✓	游戏环境/解释器/真值	✓	✗	✗
RCI (Kim et al., 2023)	✓	真值标签	✓	✗	✗
Self-Debug (Chen et al., 2023c)	✓	解释器	✓	✗	✗
Baldur (First et al., 2023)	✗	证明检查器	✗	专训模型	✓
REFINER (Paul et al., 2024)	✗	专训模型	✗	专训模型	✓
LLM-Debate (Du et al., 2023)	✓	✗	✓	✗	✗
MAD (Liang et al., 2023)	✓	✗	✓	✗	✗
CRITIC (Gou et al., 2023)	✓	搜索引擎/解释器	✓	✗	✗
CoVe (Dhuliawala et al., 2023)	✓	✗	✓	✗	✗
RISE (Qu et al., 2024)	✗	✗	✓	✗	✓
IHR (Qiu et al., 2023)	✗	解释器	✓	✗	✗
SCoRe (Kumar et al., 2024)	✗	✗	✓	✗	✓
AutoMathCritique (Xi et al., 2024)	✗	专训模型	✗	专训模型	✓
DARS (Li et al., 2025d)	✗	专训模型	✗	专训模型	✓

- **A3: 蒙特卡洛树搜索 (MCTS)。** 系列工作将 MCTS 算法应用于提升 LLM 规划能力 (Zhang et al., 2023; Hao et al., 2023; Liu et al., 2023b)。其核心流程包括选择、扩展、评估和回溯 (见图 8)。选择阶段通过 UCT/PUCT 等算法平衡探索与利用；扩展阶段基于当前状态添加子节点；评估阶段采用推演或直接评估；回溯阶段沿路径更新节点值与访问次数。

表 1 按上述分类体系对树搜索相关工作进行了梳理。

#### 4.2.2 扩展规律

实证研究表明，通过增加树搜索的宽度和深度可进一步提升性能，包括增加 MCTS 的推演次数 (Zhang et al., 2023; Liu et al., 2023b; Zhang et al., 2024b; Qi et al., 2024)、束搜索的束宽 (Yao et al., 2023a; Xie et al., 2023) 以及 A\* 算法的步数限制 (Zhuang et al., 2023)。Snell et al. (2024) 发现性能最终会饱和，可能因模型难以生成多样化节点所致。此外，Kang et al. (2024a) 发现增大 PRM 模型规模可提升性能，说明通过额外训练时间或测试时计算提升价值函数可靠性的重要性。

#### 4.2.3 提升扩展效率

提升树搜索扩展效率的策略包括：

**选择合适的树搜索算法** 不同算法的特性使其适用于不同任务。PG-TD (Zhang et al., 2023) 在代码生成任务中验证 MCTS 优于束搜索；ToolChain\* (Zhuang et al., 2023) 证明 A\* 算法在 API 调用任务中时效性更优。

**降低价值函数开销** ALPHALLM (Tian et al., 2024) 采用小语言模型作为快速推演策略；rStar-Math (Guan et al., 2025b) 通过两阶段训练替代推演的价值函数。

**自适应扩展宽度** LiteSearch (Wang et al., 2024a) 根据节点价值与深度动态分配扩展宽度；RE-BASE (Wu et al., 2024c) 通过定义轨迹收集需求实现动态分配。

**减少冗余扩展节点** FETCH (Wang et al., 2025a) 和 ETS (Hooper et al., 2025) 通过文本嵌入聚类合并语义相似节点。

### 4.3 多轮修正方法

#### 4.3.1 核心组件

多轮修正旨在通过迭代修订提升响应质量。其系统由初始生成器  $g_0$ 、反馈模型  $f$  和精修模型  $g$  构成：

$$y^0 \sim g_0(y|x) \quad (7)$$

$$z^t \sim f(z|x, y^{(<t)}, z^{(<t)}) \quad (8)$$

$$y^t \sim g(y|x, y^{(<t)}, z^{(\leq t)}) \quad (9)$$

其中  $x$  表示查询， $y^t$  表示第  $t$  步响应， $z^t$  表示第  $t$  步反馈。系统在满足停止条件时输出最终响应。反馈生成阶段可省略，直接精修初始响应 (Welleck et al., 2023; Kamoi et al., 2024)。

该方法模拟人类反思与精修的认知过程，其核心设计在于构建可靠的反馈信号与精修模型。反馈源可分为：

- **F1：自反馈。** 初始生成器  $g_0$  与反馈模型可共享同一语言模型。例如 Self-Debug (Chen et al., 2023c) 指导  $g_0$  逐行解释代码并生成执行轨迹作为反馈；Self-Refine (Madaan et al., 2023) 通过反思式提示生成反馈。还可采用“多智能体辩论”技术，通过角色扮演激发发散思维 (Du et al., 2023; Liang et al., 2023; Khan et al., 2024)。
- **F2：外部反馈。** 包括：1) 外部工具（代码解释器 (Chen et al., 2023c; Gou et al., 2023)、证明检查器 (First et al., 2023) 等）；2) 外部知识 (Gou et al., 2023; Zhao et al., 2023a)；3) 真值标签（如数学题标准答案 (Shinn et al., 2023)）；4) 专训模型 (Paul et al., 2024; Xi et al., 2024)。

精修模型实现方式与反馈模型类似，包括自精修 (Madaan et al., 2023; Shinn et al., 2023) 或专训模型 (Welleck et al., 2023)。表 2 按此分类体系梳理了相关工作。

研究表明，可靠的外部反馈可使多轮修正显著提升任务表现 (Kamoi et al., 2024)。但在缺乏外部反馈的内在自修正场景下，大模型往往难以生成可靠反馈，尤其在规划 (Valmeekam et al., 2023; Stechly et al., 2023) 和推理 (Huang et al., 2023b; Tyen et al., 2023) 任务中收效甚微，且可能放大自我偏见 (Xu et al., 2024c)。

#### 4.3.2 扩展规律

在具备可靠反馈的场景中，增加修正步数可持续提升性能直至饱和 (Welleck et al., 2023; Madaan et al., 2023)。而内在自修正场景下，增加步数可能损害性能 (Huang et al., 2023b)。通过额外训练提升自修正能力可突破此限制 (Qu et al., 2024)，修正步数的扩展上限可超越训练时的步数设定 (Snell et al., 2024)，表明训练阶段的计算投入能提升测试时扩展潜力。

#### 4.3.3 提升扩展效率

需从反馈质量与精修能力两方面提升效率：

**构建高质量反馈** 1) 采用基于人类评价标准的无参考评估指标 (Chiang and Lee, 2023)；2) 任务分解策略 (Saha et al., 2024a)；3) 置信度估计技术 (Varshney et al., 2023; Li et al., 2024c)；4) 专训反馈模型 (Paul et al., 2024)。

**增强精修能力** RISE (Qu et al., 2024) 通过合成多轮修正数据微调模型；SCoRe (Kumar et al., 2024) 设计多轮强化学习方案解决行为坍塌问题。

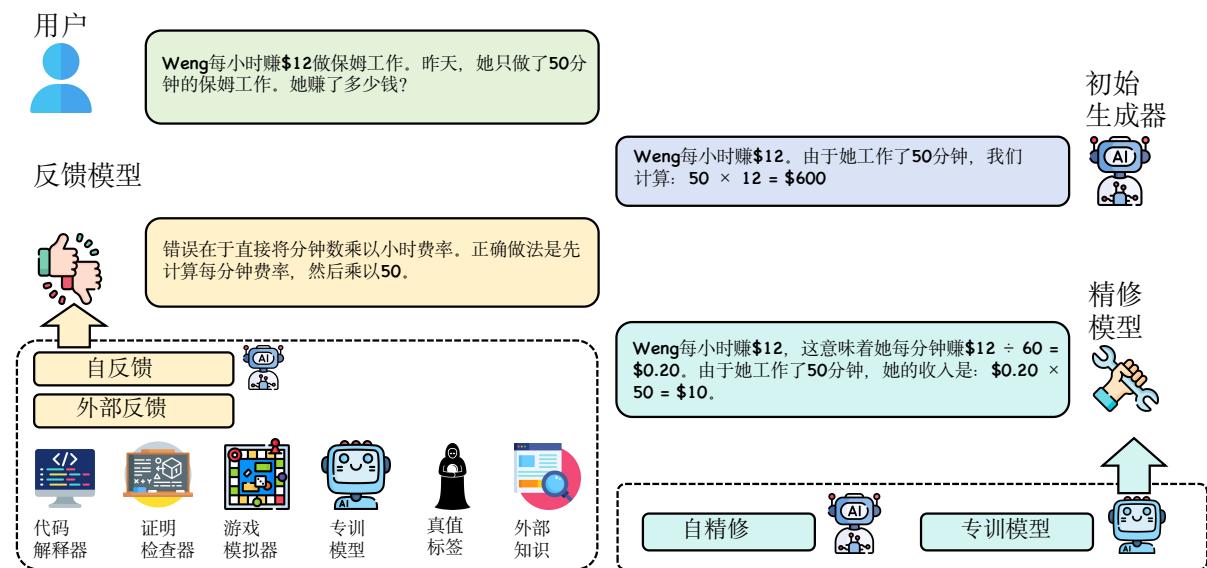


图 9: 多轮修正核心组件示意图

## 4.4 长思维链

### 4.4.1 核心组件

思维链提示 (CoT) 技术 (Wei et al., 2022; Nye et al., 2021) 通过引导模型生成人类可理解的问题求解过程说明，能有效提升模型的表征复杂度 (Merrill and Sabharwal, 2023; Nowak et al., 2024)，并显著增强其在推理任务中的表现 (Wei et al., 2022)。当前 ChatGPT、Llama 3.1 等模型在遇到推理问题时默认采用思维链模式 (Sprague et al., 2024)。尽管应用广泛，传统思维链的推理过程通常呈现浅层线性特征，暴露出复杂认知能力的局限 (Kambhampati, 2024; Chen et al., 2025e)。近期 OpenAI o1 (OpenAI, 2024) 和 Deepseek R1 (DeepSeek-AI et al., 2025) 等模型将传统思维链升级为长思维链，融合了更复杂的思维模式和更长的响应内容。以下是长思维链特有而传统思维链中较少出现的认知模式：

- 反思:** 模型发展出元认知能力 (Metcalfe and Shimamura, 1994)，可评估自身回答的正确性与合理性。例如当检测到潜在问题时，模型会通过输出“wait”暂停推理。
- 回溯:** 当发现回答存在错误时，模型能返回先前步骤进行修正。这种能力对数独、密码破解等长程规划问题至关重要，模型需在多种可能性中寻找最优解，而初始方案并不保证正确性。
- 验证:** 模型学会对单个步骤和完整解决方案进行复核，增强问题求解的稳健性。
- 发散思维:** 当意识到当前方案无法解决问题或导致明显错误时，模型能运用发散思维探索替代方案，常通过“alternatively”等过渡短语体现。
- 内隐思考:** 模型能生成超出显式解题步骤的人类式思维过程。这种细粒度推理能力可提升整体表现 (Wu et al., 2024a)。

### 4.4.2 扩展规律

早期研究证明延长推理步骤能显著提升大语言模型的推理能力 (Jin et al., 2024)。针对长思维链模型，最新研究发现 token 数量与模型性能呈正相关。虽然未明确描述 token 控制方法，但 OpenAI (2024) 和 DeepSeek-AI et al. (2025) 均发现性能随 token 数量呈对数线性增长。Hou et al. (2025) 与 Muennighoff et al. (2025) 通过响应后处理或解码技术调控 token 数量，进一步揭示了响应长度

与性能的正相关关系。具体而言, Hou et al. (2025) 从起始位置截断不同长度响应, 建议使用摘要模型提取最终答案; Muennighoff et al. (2025) 则开发了通过添加/抑制思维终止标记来控制 token 数量的预算强制技术。

尽管这些研究为 token 数量与性能的正相关提供了充分证据, 关于长响应有效性的争论仍然存在。争论主要源于观察到短响应比长响应具有更高准确率 (Zeng et al., 2025b; Ballon et al., 2025)。这种现象可能源于模型对高风险难题分配更多预算, 或长响应方案比短响应方案更迂回从而导致更高失败率 (Fatemi et al., 2025)。

#### 4.4.3 提升扩展效率

尽管长思维链赋予模型深度思考能力, 但可能导致过度思考问题。例如模型可能对” $2+3=5$ ”这类简单问题生成数百 token (Chen et al., 2024d), 在早期得出正确答案后仍进行冗余推理。此外, 基于思维链的方法在语言空间运行, 对所有 token 分配相似计算资源, 这种均质分配对维持文本连贯性的低需求 token 和关键推理 token 而言是次优的 (Hao et al., 2024b)。以下详述解决这些问题的技术:<sup>4</sup>

**简洁提示法** 通过提示直接要求模型将响应 token 限制在特定数量 (Nayab et al., 2024; Xu et al., 2025b) 或仅捕获关键信息 (Ding et al., 2024; Aytes et al., 2025)。虽实现简单, 但仅适用于简单任务, 且大语言模型无法严格遵循 token 限制 (Muennighoff et al., 2025; Aggarwal and Welleck, 2025)。

**启发式压缩微调法** 先使用启发式方法压缩思维链响应再进行微调。压缩技术包括: 直接删除中间步骤 (Su et al., 2024; Deng et al., 2024); 通过困惑度评估 token 重要性 (Cui et al., 2025b) 或专用模型保留关键 token (Xia et al., 2025); 利用 GPT-4 等高级模型重构思维链序列, 保留关键信息并消除冗余 (Kang et al., 2024b)。该方法效果高度依赖启发式压缩设计, 泛化性受限。例如 C3oT (Kang et al., 2024b) 发现仅训练 GPT-4 压缩数据会显著降低任务表现, 需同时加入原始未压缩数据。

**查询感知压缩法** 响应长度限制因查询类型而异 (Lee et al., 2025; Arora and Zanette, 2025), 难题需更多 token 而简单问题需更少。该方法通过查询感知方式逼近限制, 在保持模型计算资源分配适应性的同时提升 token 效率。具体方法包括:

- **预定义最优长度轨迹学习:** 先显式确定最优长度再进行训练。最优长度参考可基于任务类型 (如 DeepSeek-R1 (DeepSeek-AI et al., 2025) 对推理任务收集长思维链训练, 对非推理任务则收集短链或无链响应); 也可基于搜索 (Han et al., 2024; Yang et al., 2025a)、提示 (Han et al., 2024) 或采样估计查询难度 (Shen et al., 2025b)。选定最优长度轨迹可用于监督微调或偏好优化训练。
- **自训练法:** 不预设最优长度, 而是从策略模型 rollout 轨迹出发, 通过自训练激励模型在保持准确性的同时减少 token。训练方法包括: 1) 监督微调: 对每个问题生成多个响应, 选择较短的正确响应进行训练 (Kimi et al., 2025; Munkhbat et al., 2025; Liu et al., 2024d); 2) 偏好优化: 用长链模型生成多个响应样本, 选择较短的正确解作为正例, 较长响应作为负例构成偏好对数据。Chen et al. (2024d) 发现选择包含两次正确求解尝试的响应作为正例效果最佳; Sky-T1-32B-Flash (Li et al., 2024e) 采用多种偏好数据构造方法以避免准确率下降; 3) 强化学习: 在奖励函数中添加长度惩罚 (Aggarwal and Welleck, 2025; Kimi et al., 2025; Luo et al., 2025a; Arora and Zanette, 2025) 或设计中间步骤稠密奖励 (Qu et al., 2025)。例如 L1 (Aggarwal and

<sup>4</sup>需说明部分研究针对传统思维链而非长思维链, 但考虑到其易推广性仍予收录

表 3: 不同测试时扩展方法的比较。灰色部分 表示该模型是可选的或可以与其他模型共享参数。这些特征的描述针对标准版本。

方法	所需模型	可控性	适应性	无训练	兼容性
并行采样	生成器 评分函数	粗粒度	不支持	✓	完全
树搜索	生成器 价值函数	粗粒度	部分支持	✓	完全
多轮修正	初始生成器 反馈模型 精修模型	粗粒度	部分支持	✓	完全
长思维链	长思维链模型	不支持	支持	✗	完全

Welleck, 2025) 在 RL 奖励函数中加入长度控制因子；MRL (Qu et al., 2025) 则通过进度测量实现稠密奖励最大化。

- **查询路由法：**将查询分类为难题或易题，分别交由不同类型模型 (Saha et al., 2024b) 或同一模型的不同计算预算处理 (Fu et al., 2024b)。例如 System-1.x (Saha et al., 2024b) 训练控制器将规划问题分解为子目标，按难度分配给系统 1 或系统 2 规划器处理。

**模型融合法** 将长链模型与短链模型参数合并为新模型，无需额外训练。CoT-Valve (Ma et al., 2025b) 通过调节两模型参数权重实现可变长度。

**中间状态压缩法** 针对 Transformer 架构中 KV 缓存随上下文长度线性增长的问题，该方法将中间步骤压缩为更短形式（如摘要 (Yan et al., 2025)、子问题 (Yang et al., 2024c) 或特殊标记 (Pang et al., 2024a; Zhang et al., 2025a)），通过特定训练和推理策略降低内存开销。

**潜空间推理法** 将推理从语言空间切换到潜空间等其他空间，可能突破语言限制提升 token 效率。可通过微调现有模型获得该能力 (Deng et al., 2023; Hao et al., 2024b; Shen et al., 2025c)，或开发支持隐式潜空间推理的新架构 (Geiping et al., 2025) 实现。

## 4.5 测试时扩展方法的比较

对于不同的测试时扩展方法，我们在表 3 中总结了它们的特征。具体来说，我们关注以下几个方面：

**性能** 在相同的计算预算下，最佳的测试时扩展方法是什么？由于每种方法中的各种实现方式以及确保公平比较的困难，建立测试时扩展方法的绝对排名是具有挑战性的。就性能上限而言，长思维链方法在基于传统大语言模型 (LLM) 的测试时扩展方法中表现最为出色，尤其是在解决奥林匹克级别的问题时 (OpenAI, 2024; DeepSeek-AI et al., 2025)。此外，不同的测试时扩展方法在不同难度的问题和不同的计算约束下表现出不同的优势。例如，Snell et al. (2024) 通过实验证明，在有限的计算预算下，束搜索在复杂问题上表现优异，而在更多的计算资源可用时，BoN 采样在简单问题上表现更好。这些互补的优势为集成方法创造了机会，我们将在后续部分讨论这些方法。

**认知行为** 哪种测试时扩展方法具有与人类最相似的认知行为？长思维链与其他方法相比，展现出最多认知行为，包括反思、回溯、发散思维等。更重要的是，它在生成过程中统一了这些认知行为，从而具有更大的灵活性。像树搜索和多轮修正这样的方法依赖于外部的树搜索算法或预定义的多轮修正框架来赋予模型规划或反思的认知能力，这限制了它们对特定问题的适应性。

**适应性** 测试时扩展方法能否为不同的查询分配不同的计算资源？测试时扩展方法的适应性程度取决于其停止条件。在并行采样方法中，标准实现为所有查询分配相同的采样次数，导致缺乏适应性。对于树搜索和多轮修正方法，存在不同的情况。一种变体在达到预定义的超参数（例如校正次数、树深度）或答案时停止 (Yao et al., 2023a; Kang et al., 2024a; Snell et al., 2024)，因此框架本身并未提供额外的适应性。另一类方法在停止条件中引入了验证器，例如要求输出的质量分数超过给定的阈值 (Wang et al., 2024a; Welleck et al., 2023)，这基于这些验证器的可靠性引入了适应性。例如，LiteSearch (Wang et al., 2024a) 观察到，树搜索算法在停止条件包括验证器值的情况下，为更困难的问题分配了更多的计算资源。对于长思维链方法，停止条件是隐式的，并内在于生成过程中。最近的研究观察到，长思维链模型在面对更具挑战性的问题时生成了更长的响应 (Zeng et al., 2025b)。从泛化的角度来看，长思维链是差异化分配计算资源最有前途的方法。

**可控性** 在给定的计算预算下，它能否在指定的约束内运行？对于具有外部可控扩展维度（例如采样次数、树深度、修正步骤）的测试时扩展方法，可以通过根据经验估计将计算预算映射到这些超参数的具体数量来实现粗粒度的可控性 (Welleck et al., 2024)。对于长思维链，虽然将响应直接截断到特定数量可以确保不超过计算预算，但由此产生的不完整响应会显著损害性能，因此将标准长思维链视为具有可控性的方法是不切实际的。为了解决这一限制，S1 (Muennighoff et al., 2025) 通过实现思维结束标记分隔符来实现响应长度的控制，而 L1 (Aggarwal and Welleck, 2025) 开发了一种强化学习算法，以实现对令牌数量的精确控制，与 S1 相比具有更高的令牌效率。

**简洁性** 测试时扩展方法的组件是否易于实现？不包括长思维链的方法通常需要额外的角色，例如评估器来指导搜索过程，以及多个过程来推导最终解决方案。考虑到为大多数任务部署高质量评估器的额外成本，这可能会阻碍它们的实际应用。相比之下，长思维链方法不需要多个组件，并且易于实现。

**无训练** 测试时扩展方法是否需要额外的训练？不包括长思维链的方法可以直接与传统的大语言模型一起操作，而长思维链能力需要通过额外的训练来激发。值得注意的是，额外的训练可以帮助提高各种方法的扩展效率，例如增强模型的自我校正能力 (Kumar et al., 2024; Qu et al., 2024)，或应用推理感知的微调以提高计算利用率 (Chow et al., 2024; Yu et al., 2025c)。

**兼容性** 该方法能否与其他测试时扩展方法集成？正如我们将在 §4.6 中讨论的那样，所有方法都可以相互兼容。其中，并行采样最容易与其他方法兼容，因为实现多次采样相对简单。总的来说，长思维链测试时扩展方法以其简洁性、适应性、更高的性能上限和更复杂的认知行为优于其他方法，但它需要通过额外的训练来激发。此外，这些测试时扩展方法的兼容性和优势使得综合利用它们以实现更好的性能变得有益，而不是专注于单一方法。

## 4.6 测试时扩展方法的集成

集成方法旨在综合利用多种测试时扩展方法，而不是将计算资源分配给单一方法，从而可能实现比单个方法更优的性能。这些方法包括同时结合多种方法或根据不同的上下文选择适当的测试时扩展方法。图 10 展示了关于集成方法的工作组织。

**将并行采样与其他方法结合** 并行采样的简单性使其易于与其他测试时扩展方法兼容：

- **树搜索。** 相比于沿着单一树进行搜索，平行采样可以通过将初始集束扩展为多个独立的子树并独立搜索这些子树来增强树搜索的多样性 (Beeching et al., 2024; Bi et al., 2024)。实验结果表明，尤其是在大计算预算下，树搜索性能得到提升。此外，树搜索算法也可以帮助加速并行采样。例如，TreeBON (Qiu et al., 2024) 通过使用树搜索在早期阶段修剪低质量响应，减少了 BoN 的计算开销。

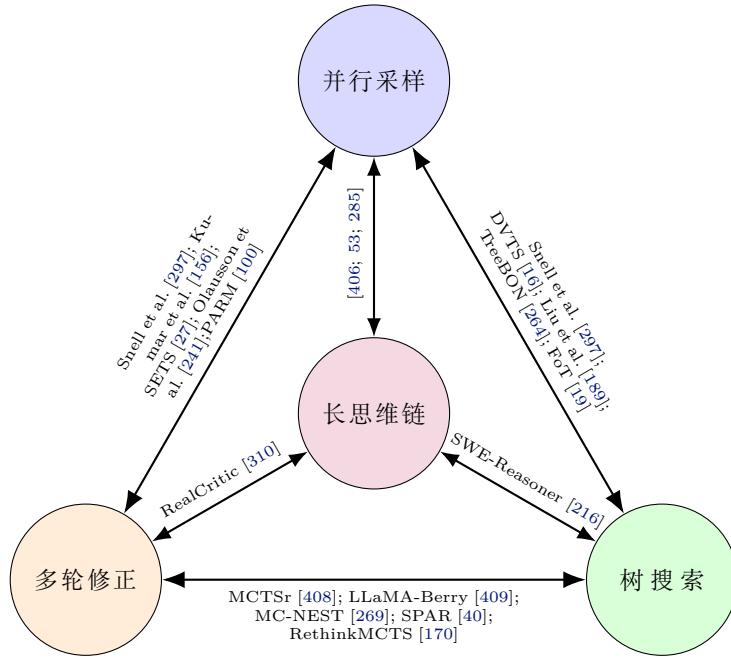


图 10: 测试时扩展方法的集成 (§4.6)。实线表示两个方法之间的组合工作。

- **多轮修正**。虽然并行采样通过并行生成独立响应来进行全局搜索，但多轮修正则是对初始响应进行局部搜索 (Snell et al., 2024)。这种互补性表明，将两种方法结合可以获得更好的性能。Kumar et al. (2024); Chen et al. (2025b) 表明，将部分计算预算分配给初始响应的自我校正，而不是仅仅增加采样次数，可以实现更高的标记效率。Olaussion et al. (2023) 证明，在混合方案中，将更多的采样预算用于生成多样化的初始候选集比进行广泛的校正更为优化。
- **长思维链**。将长思维链与并行采样方法（如多数投票）结合是直接的。最近的研究通过考虑长思维链中的过度思考现象优化了多数投票策略 (Zeng et al., 2025b; Cuadron et al., 2025)。具体而言，高过度思考与数学任务或代理环境中的性能下降相关。因此，将衡量过度思考程度的指标与投票策略结合，可以优于多数投票和单一高计算成本的响应生成 (Zeng et al., 2025b; Cuadron et al., 2025)。此外，Setlur et al. (2025) 比较了通过预算强制扩展长思维链长度 (Muennighoff et al., 2025) 与将计算用于并行采样生成较短响应的性能，发现后者在计算上更为优化。

**将树搜索与多轮修正结合** 这类工作通过将修订行为视为更新响应的动作，将批判和修订纳入树搜索算法中，无论是在解决方案层面 (Zhang et al., 2024b,c; Rabby et al., 2024; Cheng et al., 2024a) 还是在步骤层面 (Li et al., 2024d)。这种方法丰富了树搜索的扩展行为，并帮助实现比简单地顺序修订响应更好的性能 (Li et al., 2024d; Cheng et al., 2024a)。

**将长思维链与树搜索或多轮修正结合** 长思维链的内容隐含着分支搜索过程或自我校正 (Xiang et al., 2025)。因此，它可以被视为一种内部化这两种方法的方式。对于与多轮修正的结合，Tang et al. (2025) 表明，o1-mini 从自我校正中受益，而传统 LLM 表现较差，这表明长思维链模型具有强大的内在自我校正能力。对于树搜索，未来的研究应分析如何在长思考过程中定义搜索空间。

**自适应选择测试时扩展方法** 对不同测试时扩展方法性能相对于各种因素的实证分析可以帮助推导出基于自适应选择的最佳测试时扩展方法。Snell et al. (2024) 发现，多轮修正方法更适合于简单查询，而一定比例的并行采样和多轮修正则适合于困难查询。此外，他们确定束搜索对更难的问题更有效，而最佳 N 采样对较简单的问题更有效。这些发现基于查询难度分类器指导了最佳的测

## 4.6 测试时扩展方法的集成

表 4: 强化学习扩展工作总结。对于训练算法, ‘PMD’ 表示策略镜像下降方法。REINFORCE\* 表示 REINFORCE 类方法。对于奖励类型, 和 分别代表基于规则和基于模型的奖励, 而 和 分别代表结果奖励和过程奖励。‘#D’ 表示查询数据集大小。‘MS’ 表示多阶段训练策略, 包括长思维链冷启动 (LCS)、迭代延长策略 (IL) 和课程采样策略 (CSS)。在准确率 (Acc.) 和长度 (Len.) 图表中, 对于呈现多个数据的工作, 我们展示其常见模式。“Cog.” 表示响应中是否包含表示认知行为的词, 如“等待”。

工作	算法 (\$\S{5.1.1}\$)	奖励 (\$\S{5.1.2}\$)	系列 (\$\S{5.1.3}\$)	大小 (\$\S{5.1.3}\$)	#D (\$\S{5.1.4}\$)	MS (\$\S{5.1.5}\$)	准确率	长度	认知
Eurus-2-7B-PRIME (Cui et al., 2025a)	REINFORCE*			7B	150K			-	-
Deepseek-R1-Zero (DeepSeek-AI et al., 2025)	GRPO			671B	-				
Kimi k1.5 (Kimi et al., 2025)	PMD			-	-	LCS CSS			-
SimpleRL-Zero (Zeng et al., 2025a)	PPO			7B	8K				
SimpleRL (Zeng et al., 2025a)	PPO			7B	8K	LCS			-
STILL-3-ZERO-32B (Chen et al., 2025g)	GRPO			32B	90K	IL			
Sea AI Lab (Liu et al., 2025f)	PPO			1.5B	8K				
DeepScaleR-1.5B-Preview (Luo et al., 2025c)	GRPO			1.5B	40K	IL			-
T1 (Hou et al., 2025)	REINFORCE*			14B	30K	LCS			
DAPO (Yu et al., 2025a)	GRPO			32B	17K				
LIMR (Li et al., 2025f)	GRPO			7B	1.4K				-
Open-Reasoner-Zero (Hu et al., 2025)	PPO			7B 32B	57K				
Logic-RL (Xie et al., 2025)	REINFORCE*			7B	5K				

试时扩展策略。[Liu et al. \(2025c\)](#) 分析了模型大小与测试时扩展方法之间的关系, 以推导出最优的扩展策略。

## 5 方法——第二部分：测试时扩展的训练策略

正如在 §4.5 中讨论的那样，长思维链（long CoT）与其他测试时扩展策略相比，展示了更高的性能上限和更复杂的认知行为，尽管它需要额外的训练。在本节中，我们将探讨通过两种主要方法来激发模型的长思维链能力：强化学习（§5.1）和监督微调（§5.2）。此外，我们还将讨论如何有效地将测试时扩展技术与迭代训练方法结合，以实现自我改进（§5.3）。

### 5.1 扩展强化学习

最近的研究表明，通过在线强化学习并使用基于规则的奖励来训练大型语言模型（LLMs），在数学和代码等任务中可以显著增强其推理能力（DeepSeek-AI et al., 2025; Kimi et al., 2025）。在训练过程中，模型自主学会掌握长思维链的测试时扩展方法，以解决复杂问题，并展示出自我反思和自我修正等认知行为。这种现象被称为 RL 扩展现象<sup>5</sup>或“顿悟时刻”。我们系统总结了最近的工作，如表 4 所示。此外，表 6 展示了基于近期研究的 RL 扩展训练中常见挑战的解决方案。在接下来的章节中，我们将详细讨论每个组件的设计考虑。

#### 5.1.1 训练算法

**REINFORCE** REINFORCE (Sutton et al., 1999) 算法是强化学习中的一种基础策略梯度方法，通过梯度上升直接优化策略的期望回报。该算法通过最小化以下损失来优化策略模型  $\pi_\theta$ ：

$$\mathcal{L}_{\text{REINFORCE}}(\theta) = -\mathbb{E}_{\tau \sim \pi_\theta} \left[ \sum_{t=1}^T G_t \nabla_\theta \log \pi_\theta(a_t | s_t) \right] \quad (10)$$

其中  $G_t$  是从时间步  $t$  开始的折扣累积奖励。尽管 REINFORCE 简单，但其梯度估计的方差较高。

**近端策略优化 (PPO)** 对于 PPO 算法 (Schulman et al., 2017)，它通过最小化以下损失来优化策略模型：

$$\mathcal{L}_{\text{PPO}}(\theta) = -\mathbb{E}_{q \sim P(Q), o \sim \pi_{\theta_{\text{old}}}(O|q)} \frac{1}{|O|} \sum_{t=1}^{|O|} \min \left( \frac{\pi_\theta(o_t | q, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t | q, o_{<t})} A_t, \text{clip}(\theta) A_t \right) \quad (11)$$

$$\text{clip}(\theta) = \text{clip} \left( \frac{\pi_\theta(o_t | q, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t | q, o_{<t})}, 1 - \varepsilon, 1 + \varepsilon \right) \quad (12)$$

其中  $\pi_\theta$  和  $\pi_{\theta_{\text{old}}}$  分别是当前和旧的策略模型， $q, o$  是采样的问题和输出。 $\text{clip}(\theta)$  函数约束策略更新以确保训练稳定性。 $A_t$  是通过应用 GAE (Schulman et al., 2016) 基于奖励  $r_{\geq t}$  和学习到的价值函数  $V_\psi$  计算的优势值。KL 惩罚可以添加到奖励函数中：

$$r_t = r_\varphi(q, o_{\leq t}) - \beta \log \frac{\pi_\theta(o_t | q, o_{<t})}{\pi_{\text{ref}}(o_t | q, o_{<t})} \quad (13)$$

其中  $r_\varphi$  是奖励模型， $\pi_{\text{ref}}$  是参考模型（初始 SFT 模型）， $\beta$  是 KL 惩罚系数。

**组相对策略优化 (GRPO)** GRPO 算法 (Shao et al., 2024) 直接使用多个并行采样响应的平均奖励作为基线，消除了 PPO 中额外的价值函数近似的需要。具体来说，对于每个问题  $q$ ，GRPO 从旧策略  $\pi_{\theta_{\text{old}}}$  中采样一组输出  $\{o_1, o_2, \dots, o_G\}$ ，然后通过最小化以下损失来优化策略模型  $\pi_\theta$ ：

<sup>5</sup>在本文中，我们使用“RL 扩展”来描述这一系列工作。

表 5: 不同训练算法的比较。对于计算开销, 红色表示模型需要被更新, 蓝色表示模型需要执行推理。

算法	计算开销				强化学习扩展现象
	策略模型	奖励模型	评论模型	参考模型	
REINFORCE	🔥 *	*	✗	✗	✗
PPO	🔥 *	*	🔥 *	*	✓
GRPO	🔥 *	*	✗	*	✓
REINFORCE++	🔥 *	*	✗	*	✓

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\mathbb{E}_{q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(O|q)} \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{|O_i|} \sum_{t=1}^{|O_i|} \left[ \min \left( \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})} A_{i,t}, \text{clip}(\theta) \hat{A}_{i,t} \right) - \beta \mathbb{D}_{\text{KL}}[\pi_\theta || \pi_{\text{ref}}] \right] \right] \quad (14)$$

$$\text{clip}(\theta) = \text{clip} \left( \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})}, 1 - \varepsilon, 1 + \varepsilon \right) \quad (15)$$

$$\mathbb{D}_{\text{KL}}[\pi_\theta || \pi_{\text{ref}}] = \frac{\pi_{\text{ref}}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} - \log \frac{\pi_{\text{ref}}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} - 1 \quad (16)$$

其中  $\varepsilon$  和  $\beta$  是超参数,  $\hat{A}_{i,t}$  是使用组内输出对应的奖励计算的优势值。

**REINFORCE++** REINFORCE++ (Hu, 2025) 是经典 REINFORCE 算法的变体, 集成了 PPO 的关键优化技术, 同时消除了对批评网络的需求。该算法通过以下增强措施来解决 REINFORCE 的局限性:

- 实现了基于 token 的 KL 散度惩罚, 以防止策略偏离初始模型太远。
- 采用了 PPO 的剪裁机制, 以约束策略更新并保持训练稳定性。
- 引入了小批量更新, 以提高训练效率和收敛速度。
- 通过全面的奖励归一化和剪裁来稳定训练, 减轻异常值的影响并将奖励值约束在预定义范围内。
- 使用 z-score 归一化进行优势归一化, 以确保梯度稳定并防止训练期间发散。

**不同算法的比较** 我们在表 5 中总结了不同训练算法的特点。关于计算成本, PPO 的计算成本最高, 需要加载四个模型, 其中策略模型和评论模型需要同时进行推理和训练。GRPO 和 REINFORCE++ 消除了对评论模型的需求, 并且比 REINFORCE 具有更高的训练稳定性 (Hu, 2025)。关于性能, 除 REINFORCE 外, 所有算法都展示了 RL 扩展现象。对于具体性能比较, Hou et al. (2024) 发现在 RLHF 设置中 PPO 和 GRPO 的性能相似, 而 Xie et al. (2025) 观察到, 在基于规则的奖励设置下, PPO 和 REINFORCE++ 在合成逻辑谜题中的性能优于 GRPO。需要进行更严格的大规模研究以全面评估这些算法的性能。

### 5.1.2 奖励函数

奖励类型可以根据其来源和粒度分类如下:

## 5.1 扩展强化学习

表 6: 近期研究中解决强化学习扩展训练的常见问题的方法汇总

待解决问题	方法概述	观测现象	相关研究
训练算法			
传统 PPO 算法在 LLM 训练中的计算效率低下	<b>GRPO (Group Relative Policy Optimization):</b> 通过使用来自同一提示的多个输出的平均奖励作为优势计算的基线，消除了对单独价值模型的需求。	性能比较表明该方法在保持与传统 PPO 相当的效果的同时提高了计算效率，特别适用于奖励通常具有比较性质的 LLM 奖励建模。	GRPO [286]
长篇推理中的 token 效率低下和过度思考问题	<b>Dr.GRPO (Doctor GRPO):</b> 通过移除响应长度归一化和奖励标准化，解决 GRPO 中的优化偏差，实现无偏的策略梯度估计。	实验结果显示显著改善的 token 效率和更好控制的响应长度，有效缓解过度思考问题。	Dr.GRPO [200]
长思维链推理中不同回答长度导致的训练不稳定	<b>DAPO (Decouple Clip and Dynamic Sampling Policy Optimization):</b> 实现 token 级策略梯度计算，使更长序列能够适当影响梯度更新，而不受单个回答长度的限制。	比较分析显示更稳定的训练动态、更健康的熵管理和更好的质量模式识别，特别适合处理变长序列。	DAPO [389]
由于刚性约束导致的策略探索受限	<b>PGP (Group Policy Gradient):</b> 通过移除参考模型和策略约束，同时通过组级奖励归一化维持稳定性，简化策略梯度方法。	比较实验展示了增强的探索能力和降低的计算需求，提供更灵活的策略更新。	PGP [50]
重复或狭窄的推理模式	辅助熵奖励：将一个额外熵项纳入 RL 损失函数，以鼓励 token 多样性并防止确定性回答模式。	实验结果显示在不牺牲解决方案准确性的情况下，生成了更多样化和具有创造性的推理路径。	T1 [116]
固定参考模型的局限性	在线策略 KL 归一化：结合 KL 归一化与参考模型的指数移动平均 (EMA) 更新。	动态参考模型更新允许更有效的 RL 扩展，同时保持稳定的训练动态。	T1 [116]
价值模型与强先验策略模型的不对齐	价值预训练对齐：实现价值模型的专门预训练阶段，确保在 RL 开始前与强先验策略模型对齐。	两阶段收敛模式有利于初始对齐后的关键知识注入，防止长链思考任务中输出长度崩溃。	VC-PPO [397], VAPO [399]
价值与策略优化之间的方差-偏差需求冲突	<b>Decoupled-GAE (解耦广义优势估计):</b> 分离价值函数和策略优化的 $\lambda$ 参数，允许无偏价值估计的同时为策略更新保持方差降低的好处。	数学分析和实验结果证明了改进的收敛性，且没有引入额外偏差，对长链思考任务中的轨迹级奖励特别有效。	VC-PPO [397], VAPO [399]
约束策略优化中的有限探索	<b>KL 散度移除:</b> 消除约束策略与参考模型发散的 KL 惩罚项，使推理策略能够更自由地探索。	实验揭示在移除扩展推理训练期间策略分布偏移约束时的显著性能提升。	Open-Reasoner-Zero [120], DAPO [389]
RL 系统中的过早确定性行为	<b>Clip-Higher 策略:</b> 解耦 PPO 中的下限和上限裁剪范围，特别促进低概率 token 的探索，同时保持稳定性。	非对称裁剪阈值能有效对抗熵崩溃并在整个扩展训练中保持策略多样性。	DAPO [389]
后期训练中的无效梯度信号	动态采样：实现自适应采样方法，过滤掉准确率为 0 或 1 的问题回答对，以确保有效的梯度信号。	比较训练曲线证明尽管有额外的过采样计算开销，仍加速收敛至目标性能。	DAPO [389], Bae et al. [14]
长度截断样本的噪声奖励信号	过长样本过滤：掩蔽超出最大长度的截断样本的损失贡献，防止对本应合理的推理进行不当惩罚。	消融研究突显了从长度截断样本中移除噪声奖励信号后训练稳定性的显著改进。	DAPO [389]
可变长度序列间的不一致优势估计	<b>Length-Adaptive GAE:</b> 根据序列长度动态调整 GAE 中的 $\lambda$ 参数，确保短输出和长输出均衡的 TD-error 影响。	实证测试揭示了跨不同长度序列的更平衡优势估计和改进的训练稳定性，对长思维链推理特别有益。	VAPO [399]
奖励设计			
推理任务中未控制的思维链长度	余弦长度奖励：应用基于余弦的奖励塑形，优先考虑更短的正确思维链，同时惩罚短的不正确答案。	跨多样化推理任务的评估揭示了稳定的思维链长度和保持的性能。	Demysitify [387]
确定性推理任务中的奖励欺骗	准确度 + 格式奖励：结合答案正确性验证与结构化格式要求，强制在特定标签内进行明确推理。	基于规则的奖励系统比神经替代方案展示出更强的抗奖励欺骗能力，同时简化训练流程。	DeepSeek-R1 [60], SimpleRL [405], T1 [116], Logic-RL [360], STILL-3 [38]
多语言环境中的语言混合问题	语言一致性激励：根据目标语言词汇在思维链中的比例计算奖励，以减轻语言混合问题。	用户研究表明，尽管在多语言环境中存在轻微性能权衡，但可读性得到增强。	DeepSeek-R1 [60]
模型过度思考和冗长	过度思考长度惩罚：实现加权奖励机制，在保持正确性的同时惩罚过长响应，以应对模型过度思考。	逐步引入的长度惩罚有利于更高效的推理。	KIMI-K1.5 [153], DAPO [389]
细微领域中不准确的奖励建模	链式思考奖励模型：通过在最终正确性判断前加入明确的步骤推理过程增强奖励建模，特别适用于具有细微评估标准的领域。	人工验证确认链式思考奖励模型相比不含推理步骤的经典奖励模型有显著提升。	KIMI-K1.5 [153]
训练数据			
资源受限的 RL 训练环境	<b>高影响力样本选择:</b> 基于学习影响力测量优先考虑训练样本。	结果显示在维持性能的同时显著减少所需训练数据。	LIMR [172]
使用从网络提取的噪声数据训练	<b>噪声减少过滤:</b> 采用过滤机制去除从网络提取的噪声数据。	经过过滤的数据集在分布外任务上展示出改进的泛化能力。	Demysitify [387]
多阶段训练			
直接 RL 方法中的可读性和推理能力差	<b>冷启动进阶:</b> 实施分阶段训练方法，从高质量思维链数据微调开始，再过渡到大规模强化学习。	具有冷启动初始化的模型比直接 RL 方法展示出更强的可读性和推理能力。	DeepSeek-R1 [60], T1 [116], DeepscaleR [211], STILL-3 [38]
训练中难度各异的问题的处理低效	策略性采样：结合从简单到复杂问题的课程式进阶与模型表现最弱的困难案例优先处理。	目标采样方法展示了更快的收敛和训练中计算资源的更高效利用。	KIMI K1.5 [153]
长链推理中上下文的低效使用	<b>渐进式上下文扩展:</b> 实施多阶段训练方法。在模型性能开始在每个级别达到稳定状态时逐步增加上下文窗口大小。	分阶段上下文窗口扩展相比固定最大上下文训练在计算效率和最终性能指标上都表现出显著改进。	DepscaleR [211]
复杂推理问题上的性能差距	<b>目标性退火:</b> 通过线性衰减学习率对特定挖掘的困难问题实施最终退火训练阶段，以进一步提高推理能力。	在不影响一般能力的情况下提高复杂推理任务的性能指标。	Open-Reasoner-Zero [120]

- 基于模型的奖励：在传统的 RLHF (Ouyang et al., 2022) 设置中，从人类偏好数据中学习一个显式的奖励模型，并在 RL 训练中指导优化过程。通过直接在人类偏好数据上训练，可以

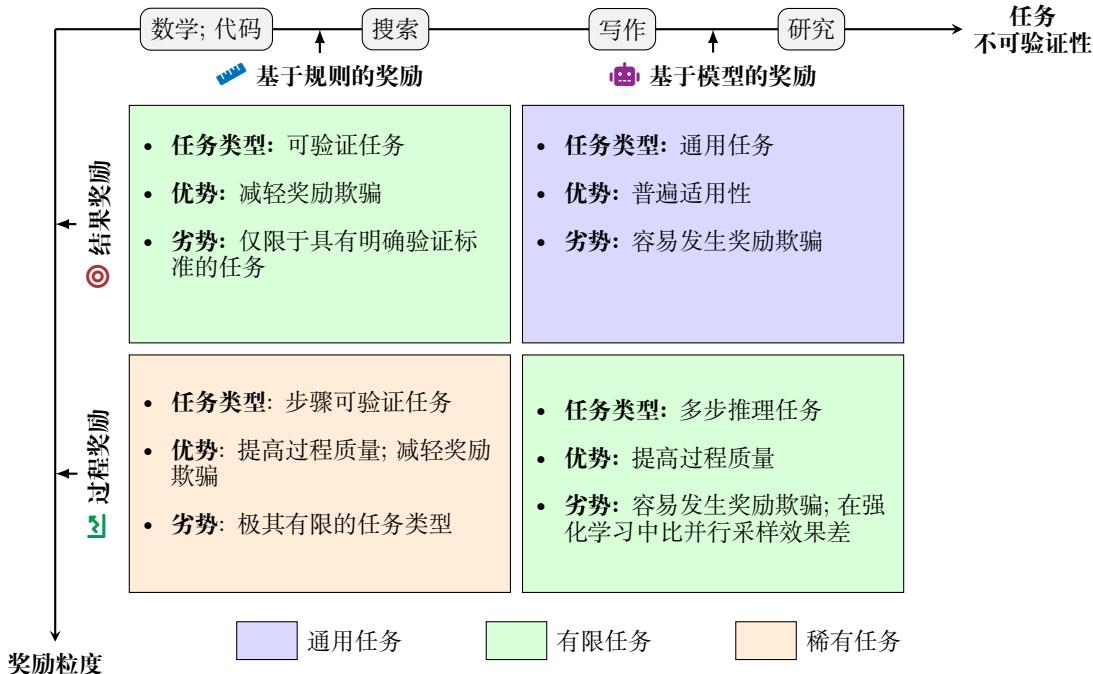


图 11: 不同奖励类型的比较。颜色表示适用的任务范围。

省略显式的奖励模型，形成隐式的奖励模型 (Rafailov et al., 2023)。

- **基于规则的奖励**: 术语“基于规则”表示奖励是明确定义的，并且可以通过明确的规则确定，有时也称为可验证的奖励。例如，对于有标准答案的数学问题或带有单元测试的代码任务，响应的正确性可以轻松验证，从而用于构建奖励。这可以进一步扩展到包括响应格式或语言一致性。即使使用专门模型自动验证答案等价性 (Chen et al., 2024b; Kimi et al., 2025)，只要模型的性能与理想的规则验证紧密匹配，我们仍然将其归为基于规则的奖励。
- **结果奖励**: 在一般情况下，基于规则的奖励或基于模型的奖励仅给予响应的最后一个 token，称为“结果奖励”。
- **过程奖励**: 在多步推理任务中，结果奖励可能不足以监督策略模型并帮助避免解决方案中的逻辑错误 (Shao et al., 2024; Lightman et al., 2023)。这需要对每个步骤进行更细粒度的奖励，称为“过程奖励”，通常以基于模型的方式计算。我们在 §4.2.1 中详细讨论了过程奖励模型的构建。除了构建过程奖励模型，最近的工作还探索了其他方法以实现更准确的信用分配。例如，Kazemnejad et al. (2024) 用无偏的蒙特卡罗估计替换 PPO 算法中的价值网络。Hwang et al. (2024) 和 Setlur et al. (2024) 引入了基于 MC 的方法来检测推理链中的关键错误，并将其用作 DPO 中的临时机制。

图 11展示了不同奖励类型的比较。我们在下面详细讨论。

**基于规则的奖励 vs. 基于模型的奖励:** 基于模型的奖励可以应用于一般任务，但也容易导致奖励黑客问题。构建偏好数据以学习奖励模型来代理人类偏好的流程可以应用于一般任务，因此被广泛采用。然而，已经观察到奖励在训练过程中是一个不完美的代理。有两种主要的解释这一现象 (Rafailov et al., 2024): 1) OOD 鲁棒性: 奖励函数不断使用未见过的模型样本进行查询，这些样本可能是分布外的；2) 奖励错误指定: 学到的奖励函数可能表现出虚假相关性，导致它们偏好非预期的行为。这些问题导致奖励过度优化问题，在训练过程中，代理奖励分数单调增加，而目标奖励分数

## 5.1 扩展强化学习

---

会饱和然后下降 (Gao et al., 2023)。尽管通过增加规模或训练数据来提高奖励模型的能力 (Ouyang et al., 2022; Hou et al., 2024) 或迭代重新训练奖励模型以改善其对策略模型的监督 (Shao et al., 2024) 可以缓解这一问题, 但这种现象仍然存在并阻碍大规模 RL 的成功 (DeepSeek-AI et al., 2025)。

**结果奖励 vs. 过程奖励:** 细粒度的过程奖励可能有助于提高 RL 性能, 但也会引入奖励黑客问题。实证结果表明, 与仅使用结果奖励相比, 过程奖励可以帮助提高 RL 性能 (Cui et al., 2025a; Shao et al., 2024)。然而, 它仍然面临几个挑战: 1) 高质量过程奖励的构建需要大量的人工; 2) 不完美的过程奖励模型容易被黑客攻击。例如, Gao et al. (2024a) 发现重复正确但不必要的推理步骤可以从过程奖励模型中获得高奖励。尽管这些问题可以通过奖励精炼来解决, 但它使 RL 流程复杂化; 3) 过程奖励在 RL 训练中显示的改进不如在并行采样设置中显著。在并行采样设置中, 实证结果表明过程奖励模型在响应选择方面显著优于结果奖励模型 (Lightman et al., 2023; Wang et al., 2023a)。然而, 在 RL 设置中, 增益并不那么明显 (Gao et al., 2024b; Cui et al., 2025a; Shao et al., 2024)。

**基于规则奖励的优化** 用于引发长思维链推理的基于规则的奖励主要由正确性奖励和特定标签的格式奖励组成。虽然这种方法已被证明足以用于 RL 扩展, 但由于其对准确性的狭隘关注, 可能导致内容对齐问题。这种方法主要引发两个问题。首先, 它可能导致可读性差和语言使用不一致。Deepseek-R1 (DeepSeek-AI et al., 2025) 通过在 RL 训练期间引入语言一致性奖励来解决这些挑战。其次, 这种方法可能导致响应过长, 引发过度思考问题。为了解决这个问题, Kimi k1.5 (Kimi et al., 2025) 在后期训练阶段实施了长度惩罚, 而 T1 (Hou et al., 2025) 则惩罚了超出上下文窗口大小或包含重复 n-gram 的响应。

RL 在可验证任务中的成功展示了强大奖励信号的重要性。随着更多关于 RL 扩展的研究加强了其理论和实证基础以促进实施, 它将 RL 训练过程解耦为两个不同的步骤: 首先定义可验证的奖励, 然后进行 RL 训练, 如 OpenAI 的强化微调服务部分实现的那样。<sup>6</sup> Search-R1 (Jin et al., 2025) 利用一个简单的结果奖励函数来验证最终答案的正确性, 并进行 RL 训练, 成功赋予 LLMs 在逐步推理过程中自主生成搜索查询的能力, 展示了 RL 在数学和代码之外的力量。对于未来在开放科学问题等领域的工作, 构建可靠的奖励信号仍然是一个开放的挑战, 并提供了显著的创新潜力。

### 5.1.3 策略模型选择

策略模型是成功 RL 训练的前提。选择标准可以基于以下方面:

**模型家族** 如表 4 所示, 大多数 RL 扩展工作使用 Qwen2.5 作为基础模型。最近的研究表明, Qwen2.5 在应用 RL 之前的问题解决过程中展示了验证和修正等认知行为 (Gandhi et al., 2025; Liu et al., 2025g,f), 尽管模型无法有效使用它们。这表明模型的预训练知识已经包含了这些思维模式。Gandhi et al. (2025) 基于 Qwen-2.5-3B 在 Countdown 游戏中表现出显著增益而 Llama-3.2-3B 在相同的 RL 训练条件下迅速达到平台期的观察, 深入研究了这一现象。当 Llama 被富含这些行为的合成推理轨迹或认知行为增强数据预训练时, 它在 RL 期间表现出显著改进, 与 Qwen 的性能轨迹匹配。这突出了在进行 RL 之前对包含认知行为的语料进行预训练的重要性。

**模型大小** 虽然传统的 RLHF 设置显示较大的模型从强化学习优化中获得的收益较少 (Gao et al., 2023; Hou et al., 2024), 但 RL 扩展设置表明较大的模型实现了更高的 token 效率, 从而获得更好的性能 (Kimi et al., 2025)。在没有长思维链冷启动的情况下, 在 7B 或更小的模型中复制 DeepSeek-R1-Zero (671B) 的扩展行为的有限成功进一步表明, 模型大小显著影响扩展行为。

### 5.1.4 训练数据构建

训练数据的质量和数量显著影响 RL 的效率和上限。

---

<sup>6</sup><https://openai.com/form/rft-research-program/>

**数据质量** 消除不需要进一步训练的简单查询有助于节省 RL 作为后训练技术的必要计算成本，其中查询难度可以通过从策略模型中多次采样以计算正确答案的成功率来估计 (Kimi et al., 2025; Chen et al., 2025g)。同样，移除当前模型缺乏基本能力解决的问题也有益 (Chen et al., 2025g)。从训练角度来看，模型始终正确或错误回答的查询引入了梯度下降问题。DAPO (Yu et al., 2025a) 提出了一种动态采样策略，过度采样并过滤掉准确率为 1 和 0 的提示，观察到显著的性能提升，这可以被视为一种在线难度控制方法。

**数据数量** 在传统的 RLHF 设置中，扩展提示数量不会导致显著的性能提升 (Hou et al., 2024)。然而，这一结论不适用于 RL 扩展场景。Open-Reasoner-Zero (Hu et al., 2025) 调查了 7.5K MATH 训练集和他们策划的 57K 提示集之间的性能差异，发现较大的集合会导致准确性和响应长度的持续扩展，而较小的集合则达到平台期。同样，DeepSeek-R1-Zero 使用他们的大规模策划数据集观察到持续的性能改进 (DeepSeek-AI et al., 2025)。

### 5.1.5 多阶段训练

通过采用以下多阶段训练策略，可以提高训练效率：

**长思维链冷启动** 在 RL 训练之前对长思维链数据进行微调可以促进后续的 RL 改进 (Yeo et al., 2025)，并缓解 RL 训练期间的早期不稳定问题 (DeepSeek-AI et al., 2025)。此外，提高长思维链的质量显著放大了 RL 增益 (Yeo et al., 2025)。此外，Li (2025) 通过将稀疏更新和自适应终止机制纳入 SFT 损失函数中，展示了改进的性能，这有助于在训练后保持响应多样性。

**迭代延长策略** DeepScaleR-1.5B-Preview (Luo et al., 2025c) 最初将上下文窗口大小限制为 8K，在此期间模型生成较短的响应，同时训练奖励增加。当模型响应开始延长时，上下文窗口大小扩展到 16K，随后扩展到 24K (见表 4 中的“DeepScaleR-1.5B-Preview”行)。此策略在减少计算成本的同时指导控制响应长度的扩展。

**课程采样策略** 当在初始训练阶段将有限的计算预算分配给非常具有挑战性的问题时，这通常会产生少量正确样本，导致训练效率较低。为了解决这一限制，课程采样策略从训练简单任务开始，然后逐步推进到更复杂的任务。Kimi K1.5 (Kimi et al., 2025) 报告了通过实施这一课程采样策略提高了性能，利用了其包含年级和难度标签的训练数据集。同样，logic-RL (Xie et al., 2025) 检查了这一方法的效用，但发现它在逻辑谜题任务中的改进并不显著，得出结论认为需要平衡阶段性训练的复杂性与潜在性能增益。

## 5.2 监督微调

近期研究表明，通过简单的监督微调方法，可以在类似数据上激发出长思维链的测试时扩展行为 (Muennighoff et al., 2025; Ye et al., 2025)。相较于基于强化学习 (RL) 的方法，这一方法因其更简单的训练过程和更高的数据效率而显得尤为有前景。表 7 展示了长思维链资源的整理。我们详细阐述了基于监督微调方法的核心设计考虑如下：

**训练数据来源** 数据来源可以分为基于合成数据或从现有长思维链模型中进行蒸馏两种。轨迹合成方法包括直接从树或图搜索过程中收集轨迹 (Lehnert et al., 2024; Gandhi et al., 2024; Ye et al., 2024)，例如在路径规划或形式逻辑问题中。然而，由于任务和认知行为的局限性，其广泛应用受到限制。另一类工作首先使用测试时扩展方法（如树搜索过程和多轮修正）解决问题，然后将搜索历史转化为详尽的探索轨迹 (Zhao et al., 2024; Ma et al., 2025a)。例如，Journey Learning (Qin et al., 2024) 提出先引导 LLMs 使用树搜索解决问题，然后使用另一个 LLM 将回溯或评估步骤转化为自然语言以形成轨迹。然而，这些轨迹中逻辑连贯性和多样性的缺乏限制了其表现。Xi et al. (2024)

表 7: 长思维链资源组织。

工作	应用	类型	来源	数量	模态	链接
O1 Journey-Part 1 [261]	数学	合成	GPT-4o	0.3K	文本	🔗 🧠
Marco-o1 [425]	推理	合成	Qwen2-7B-Instruct	10K	文本	🔗
STILL-2 [228]	数学, 代码, 科学, 谜题	蒸馏	DeepSeek-R1-Lite-Preview QwQ-32B-preview	5K	文本	🔗 🧠
RedStar-math [370]	数学	蒸馏	QwQ-32B-preview	4K	文本	🧠
RedStar-code [370]	代码	蒸馏	QwQ-32B-preview	16K	文本	🧠
RedStar-multimodal [370]	数学	蒸馏	QwQ-32B-preview	12K	视觉; 文本	🧠
S1K [231]	数学, 科学, 代码	蒸馏	Gemini Flash Thinking	1K	文本	🔗 🧠
S1K-1.1 [231]	数学, 科学, 代码	蒸馏	DeepSeek R1	1K	文本	🔗 🧠
LIMO [386]	数学	蒸馏	DeepSeek R1 DeepSeekR1-Distill-Qwen-32B	0.8K	文本	🔗 🧠
OpenThoughts-114k [314]	数学, 代码, 科学, 谜题	蒸馏	DeepSeek R1	114K	文本	🔗 🧠
OpenR1-Math-220k [72]	数学	蒸馏	DeepSeek R1	220K	文本	🔗 🧠
OpenThoughts2-1M [314]	数学, 代码, 科学, 谜题	蒸馏	DeepSeek R1	1M	文本	🔗 🧠
CodeForces-CoTs [314]	代码	蒸馏	DeepSeek R1	47K	文本	🔗 🧠
Sky-T1-17k [166]	数学, 代码, 科学, 谜题	蒸馏	QwQ-32B-Preview	17K	文本	🔗 🧠
$S^2R$ [213]	数学	合成	Qwen2.5-Math-7B	3K	文本	🔗 🧠
R1-Onevision [380]	科学, 数学, 通用	蒸馏	DeepSeek R1	155K	视觉; 文本	🔗 🧠
OpenO1-SFT [313]	数学, 代码	合成	-	77K	文本	🔗 🧠
Medical-o1 [28]	医学	蒸馏	Deepseek R1	25K	文本	🔗 🧠
O1 Journey-Part 3 [129]	医学	蒸馏	o1-preview	0.5K	文本	🔗 🧠
SCP-116K [206]	数学, 科学	蒸馏	Deepseek R1	116K	文本	🔗 🧠
open-r1-multimodal [71]	数学	蒸馏	GPT-4o	8K	视觉; 文本	🔗 🧠
Vision-R1-cold [127]	科学, 数学, 通用	蒸馏	Deepseek R1	200K	视觉; 文本	🔗 🧠
MMMU-Reasoning-Distill-Validation [230]	科学, 数学, 通用	蒸馏	Deepseek R1	0.8K	视觉; 文本	🔗
Clevr-CoGenT [29]	视觉计数	蒸馏	Deepseek R1	37.8K	视觉; 文本	🔗 🧠
VL-Thinking[26]	科学, 数学, 通用	蒸馏	Deepseek R1	158K	视觉; 文本	🔗 🧠
Video-R1[75]	视频	蒸馏	Qwen2.5-VL-72B	158K	视觉; 文本	🔗 🧠
Embodied-Reasoner [416]	具身 AI	合成	GPT-4o	9K	视觉; 文本	🔗 🧠
OpenCodeReasoning [4]	代码	蒸馏	DeepSeek R1	736K	文本	🧠
SafeChain [135]	安全	蒸馏	Deepseek R1	40K	文本	🔗 🧠
KodCode [374]	代码	蒸馏	DeepSeek R1	2.8K	文本	🔗 🧠

将多轮修正过程转化为自对话数据，但由于 LLMs 在批判性思维上的局限性，该方法在应对复杂问题时表现不佳。与合成方法的复杂性相比，蒸馏方法直接从开源的长思维链模型（如 Deepseek R1 或 QwQ (Ye et al., 2025; Muenennighoff et al., 2025; Li et al., 2025c)）中提取轨迹。尽管这种方法成本效益高且表现优于合成方法，但蒸馏的本质使得学生模型难以超越其教师模型 (Huang et al., 2024)。

**训练数据质量** 长思维链数据的质量显著决定了其在激发模型推理能力方面的效果。这包括查询质量和响应质量。对于查询而言，它们应该对基础模型具有挑战性，并涵盖多个领域。LIMO (Ye et al., 2025) 采用了多阶段过滤过程，保留了即使对最先进的推理模型也具有挑战性的查询。S1 (Muen-

**Algorithm 1:** 迭代自强化学习

---

**Input:** 原始训练集  $\mathcal{D}_0 = \{(x_i, y_i)\}$ , 原始查询集  $\mathcal{D}_0^{\text{query}} = \{x_i\}$ ,  $M_0$ : 初始策略模型,  $T$ : 迭代次数,  $R_0$ : 初始奖励模型,  $\text{Sample}(\cdot)$ : 策略采样函数,  $\text{Synthesize}_{\text{query}}(\cdot)$ : 查询合成函数,  $\text{Update}_{\text{policy}}(\cdot)$ : 策略更新函数,  $\text{Update}_{\text{reward}}(\cdot)$ : 奖励模型更新函数,  $\text{Filter}(\cdot)$ : 过滤函数

```

for  $t = 0$  to  $T - 1$  do
     $\mathcal{D}_t^{\text{query}} = \text{Synthesize}_{\text{query}}(\mathcal{D}_{t-1}^{\text{query}}, M_t)$ 
     $\mathcal{D}_t = \{(x^i, \{y_j^i\}_{j=1}^{N_i})|_{i=1}^{N_t} \text{ 满足 } x^i \sim \mathcal{D}_t^{\text{query}}, \{y_j^i\}_{j=1}^{N_i} \sim \text{Sample}(M_t, x^i)\}$ 
    使用奖励模型  $R_t$  标注  $\mathcal{D}_t$ 
     $\mathcal{D}_t^{\text{filter}} = \text{Filter}(\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_t)$ 
     $M_{t+1} = \text{Update}_{\text{policy}}(M_t, \mathcal{D}_t^{\text{filter}})$ 
     $R_{t+1} = \text{Update}_{\text{reward}}(R_t, \mathcal{D}_t^{\text{filter}})$ 
end

```

---

nighoff et al., 2025) 根据模型表现和推理轨迹长度维护了难度较高的查询, 同时覆盖了多样化的主题。对于响应, 可以通过答案检查器和代码解释器进行后期过滤。例如, 带有验证器的 OpenThoughts-114k (Team, 2025a) 在性能上优于未验证的 OpenThoughts-Unverified-173k, 且验证器的精度影响了表现。关于响应内容, Li et al. (2025c) 通过扰动实验发现, 全局推理结构比局部内容细节更为重要。

**训练数据量** 实验结果表明, 相对于计算成本, 增加数据量并未带来预期的性能提升。S1 (Muen-nighoff et al., 2025) 比较了精心挑选的 1k 数据集与 59k 完整数据集, 发现性能提升有限。类似地, OpenThoughts-114k (Team, 2025a) 与精心挑选的 1k 数据集之间的性能差距也支持了这一观察。LIMO (Ye et al., 2025) 将这一现象归因于“少即是多”推理假设, 该假设认为训练数据主要用于激发模型固有的复杂推理能力, 而非教授新知识。

**训练方法** 是否可以通过参数高效的微调方法 (如 LoRA (Hu et al., 2022)) 学习自我修正和回溯能力仍在探索中。Li et al. (2025c) 比较了 LoRA 微调与全参数微调的表现, 发现两者表现接近。这一观察与 Ye et al. (2024) 的结论相矛盾, 后者认为 LoRA 微调无法学习自我修正模式, 尽管其实验仅在 GPT2-small 的合成数据集上进行。更多研究应验证 LoRA 微调的有效性。

**基础模型** 不同基础模型在长思维链微调后的性能提升差异显著 (Li et al., 2025c)。Li et al. (2025h) 发现小模型 ( $\leq 3B$  参数) 并未从长思维链推理中一致受益, 反而在微调于较短的推理链时表现更好。他们将此归因于小模型领域知识的局限性, 并展示了具备更多领域知识的模型表现优于不具备的模型。未来工作应定量分析基础模型特征与性能之间的关系。

尽管与基于 RL 的方法相比, 基于监督微调的方法更易实现且成本效益更高, 但其存在一些潜在局限性。首先, 监督微调方法在激发长思维链推理能力方面的成功很大程度上依赖于通过 RL 训练的现有开源长思维链模型, 凸显了对教师模型的依赖。这一特点表明, 监督微调与 RL 方法应结合使用以提高数据效率。例如, 在 Deepseek-R1 的训练过程中, 采用了多阶段训练方法, 交替进行 RL 和监督微调训练, 充分发挥了两种方法的优势。其次, 监督微调方法常被批评为记忆固定模式而非实现真正的泛化 (Chu et al., 2025a; Mirzadeh et al., 2024; Zhang et al., 2024d)。尽管实验结果表明这种批评并不总是成立, 因为在小数据监督微调后, 模型在其他主题和领域中仍能提升表现 (Ye et al., 2025), 但未来研究应仔细分析监督微调训练步骤与泛化能力之间的关系。

表 8: 迭代自强化学习工作的组织。IT 表示是否涉及迭代训练。在采样下，查询表示是否合成新查询，响应表示采样方法。对于评分列，GT 表示真实标签，CI 表示代码解释器，MV 表示多数投票，RM 表示基于模型的奖励模型或以 LLM 作为评判。在选择与更新下，算法表示策略模型的训练算法，模型表示每轮训练的模型（原始表示原始模型，当前表示迭代中的当前模型）。数据表示训练数据的来源（当前表示当前轮次的数据，原始表示原始数据，之前表示所有之前轮次的数据）。RM 表示奖励模型是否在过程中更新。

工作	IT	采样		评分	选择与更新			
		查询	响应		算法	模型	数据	RM
STaR [404]	✓	✗	并行采样	GT	SFT	原始	当前	✗
P3 [101]	✓	✓	并行采样	CI	SFT	当前	当前	✗
LMSI [123]	✗	✓	并行采样	MV	SFT	当前	当前	✗
RAFT [66]	✓	✗	并行采样	RM	SFT	当前	当前	✗
RFT [398]	✗	✗	并行采样	GT	SFT	原始	当前；原始	✗
ReST [97]	✓	✗	并行采样	RM	SFT	当前	当前；之前	✗
ReST <sup>EM</sup> [295]	✓	✗	并行采样	GT	SFT	原始	当前	✗
Self-Rewarding [396]	✓	✓	并行采样	RM	DPO	当前	当前	✓
V-STaR [114]	✓	✗	并行采样	GT	SFT	原始	当前；之前	✓
IRPO [252]	✓	✗	并行采样	GT	DPO	当前	当前	✗
Qwen2.5-MATH [376]	✓	✗	并行采样	GT; RM	SFT	-	-	✓
Process-SelfRewarding [414]	✓	✗	并行采样	RM	DPO	当前	当前	✓
TS-LLM [77]	✓	✗	MCTS	GT	SFT	当前	当前	✓
ALPHALLM [318]	✓	✗	MCTS	RM	SFT	当前	当前	✓
AlphaMath [25]	✓	✗	MCTS	GT	SFT	当前	当前	✓
ReST-MCTS* [407]	✓	✗	MCTS	GT	SFT	当前	当前	✓
MCTS-IPL [362]	✓	✗	MCTS	GT; RM	DPO	当前	当前	✗
CPL [336]	✓	✗	MCTS	GT; RM	SFT; Step-APO	当前	当前	✓
SRA-MCTS [368]	✗	✗	MCTS	RM		当前	当前	✓
rStar-Math [96]	✓	✗	MCTS	GT	SFT	当前	当前	✓
Xiong et al. [366]	✗	✗	多轮修正	GT	SFT	当前	当前	✓
$\mu$ CODE [132]	✓	✗	多轮修正	RM	SFT	当前	当前	✓
SPaR [40]	✓	✗	集成	RM	DPO	当前	当前	✓
SWE-Reasoner [216]	✗	✗	长链推理	GT	SFT	当前	当前	✗

### 5.3 迭代自强化学习

通过测试时扩展方法生成的轨迹可以通过离线方法（如监督微调或 DPO）优化策略模型，从而实现自我改进。该框架作为一个自强化循环运行，其中首先生成数据，然后利用数据进行学习，之后用其增强后的迭代版本替换原始策略模型。我们将这一训练范式称为**迭代自强化学习 (ISRL)**。算法 1 提供了该算法的概述，表 8 展示了相关工作的整理。算法的核心步骤如下：

**采样** 首先，使用受控采样函数从策略模型中采样响应，该函数可以通过上述测试时扩展方法实现，包括并行采样 (Zelikman et al., 2022; Gulcehre et al., 2023; Dong et al., 2023)、树搜索 (Feng et al., 2023; Tian et al., 2024; Zhang et al., 2024a) 以及多轮修正 (Xiong et al., 2025; Jain et al., 2025)。为了增强候选响应的多样性和质量，STaR (Zelikman et al., 2022) 在模型无法独立解决问题时提供正确答案作为提示以引导生成推理过程。ReST-MCTS\* (Zhang et al., 2024a) 通过实验证明，步骤级树搜索优于并行采样，因为步骤级搜索提高了中间推理步骤的质量。此外，其他研究探索了增加查询多样性的方法，例如使用少样本提示合成新问题 (Haluptzok et al., 2023; Yuan et al., 2024c)。

**评分** 采样后的响应可以通过以下方法进行评分：1) 对于数学和代码等任务，生成的解决方案可以通过与真实答案对比进行验证 (Zelikman et al., 2022) 或通过单元测试验证 (Huang et al., 2023a)；2) 对于一般任务，可以使用现成的奖励模型对响应进行评分 (Dong et al., 2023)，或者策略模型本身可以作为评分者 (Yuan et al., 2024c)；3) 对于树搜索算法，采样过程中的伴随分数有助于选择正确的解决方案或构建偏好对 (Guan et al., 2025b; Zhang et al., 2024f; Xie et al., 2024b)；4) 多数投票。当真实答案不可用时，多数投票策略可以帮助确定正确答案 (Huang et al., 2023a)。

**选择与更新** 带有分数的响应池进一步被选择和利用以更新策略模型，并可选地更新奖励模型。策略模型可以通过监督微调 (Zelikman et al., 2022; Gulcehre et al., 2023; Singh et al., 2023) 或 DPO (Yuan et al., 2024c; Pang et al., 2024b) 进行更新。奖励模型也可以更新，例如在树搜索过程

## 6. 进展如何——迄今的应用

表 9: 测试时扩展技术在不同领域的应用。在**长思维链**一列中，斜体表示传统链式思考工作，黄色表示使用强化学习技术，紫色表示使用监督微调技术。绿色表示将测试时扩展技术与迭代训练相结合，即论文中的迭代自强化学习。

应用领域	并行采样	树搜索	多轮修正	长思维链
数学	Self-Consistency [339]; ORM [52]; PRM80K [181]; Math-shepherd [334]; STaR [404]; V-STaR [114]; IRPO [252]; Process-Self-Rewarding [414]	CPT-f [258]; HTPS [158]; BFS-Prover [365]; ToT [383]; MindStar [143]; RAP [104]; Q* [329]; Self-Evaluation Guided [363]; TS-LLM [77]; LiteSearch [327]; ALPHALLM [318]; AlphaMath [25]; MCTS [408]; ReST-MCTS* [407]; REBASE [354]; rStar-Math [96]; LLaMA-Berry [409]	RISE [267]; SCoRe [156]; AutoMathCritique [356]	Openai o1 [313]; O1 Journey-Part1 [261]; O1 Journey-Part2 [128]; STILL-2 [228]; T1 [116]; DeepseekR1 [60]; Kimi k1.5 [153]; SimpleRL [405]; S1 [231]; LIMO [386]; Demystifying [387]; LIMR [172]; DeepScaleR [211]; QwQ [316]; DAPO [389]; Eurus-2.7B-PRIME [54]; STILL-3 [38]; Open-Reasoner-Zero [120]; VAPO [399]; Open-RS [56]
代码	MFR-EXEC [292]; CodeT [24]; S* [165]; AlphaCode [176]; AlphaCode2 [311]	PG-TD [415]; o1-Coder [419]; Q* [329]; SWE-Reasoner [216]; PLANSEARCH [330]; RethinkMCTS [170]; SRA-MCTS [368]	Reflexion [293]; Self-Debug [37]; CRITIC [89]; IHR [265]; Self-Repair [241]; STOP [403]	Deepseek-R1 [60]; Kimi k1.5 [153]; SWE-RL [348]; SWE-Gym [249]; OpenAI o1 [243]; QwQ-preview [315]; QwQ [316]; ToRL [173]; OpenCodeReasoning [4]; SWE-Reasoner [216]; DeepCoder [210]; Seed-Thinking-v1.5 [283]
多模态	URSA [212]; PARM [100]	VisVIM [367]; LLaVA-CoT [369]; Mulberry [381]; LlamaV-o1 [317]; Video-T1 [183]	R <sup>3</sup> V [41]; Insight-V [67]; PARM [100]; GoT [73]	AMmoTH-VL [98]; Virgo [68]; QVQ-72B-Preview [268]; Open-R1-Multimodal [71]; R1-Multimodal-Journey [220]; R1V [29]; LMM-R1 [256]; VLM-R1 [288]; R1-Video [337]; R1-Onevision [380]; MM-Eureka [221]; Vision-R1 [127]; VisualThinker-R1-Zero [432]; MAYE [215]; Visual-RFT [288]; Seg-Zero [196]; vsGRPO [179]; Video-R1 [75]; MVoT [164]; Kimi-VL [154]; Kimi k1.5 [153]; O3/O4-mini [246; 247]
智能体	-	Agent Q [259]; ToT [383]; SearchAgent [155]	Reflexion [293]; Agent Q [259]; Agent-Eval-Refine [250]	ReAct [384]; Deep Research [245]; SWE-RL [348]; Operator [244]; UI-TRARS [263]; PC Agent [108]; DeepResearcher [428]; SWEET-RL [433]; Claude 3.7 Sonnet [9]
具身智能	-	-	Inner Monologue [126]; REFLECT [197]; KnowNo [274]	Embodied-CoT [226]; CoA [168]; SpatialCoT [195]; RAD [51]; Cosmos-Reasoner [13]; Gemini Robotics [312]; CoT-VLA [421]; Embodied-Reasoner [416]
安全对齐	SelfCheckGPT [219]; SRG [332]	STAIR [418]; C-MCTS [254]; HaluSearch [42]; InferenceGuard [133]; ARGS [150]	MART [87]; Combat Adv. Attacks [45]; Improve Factuality [69]; Multi-expert Prompting [203]; DebateGPT [304]	Deliberate Alignment [94]; SafeChain [135]; Chain-of-Verification [63]; MoTE [198]
检索增强生成	CoRAG [333]	AirRAG [76]; CoRAG [333]	-	IterDRAG [400]; Plan*RAG [324]; DeepRAG [95]; Search-o1 [171]; AirRAG [76]; Auto-RAG [390]; CoRAG [333]; Search-R1 [137]; R1-Searcher [298]; ReSearch [32]
评估	CCE [413]	MCTS-Judge [343]	ChatEval [23]; ScaleEval [44]	FactScore [227]; Factool [43]; RefChecker [121]; RAGChecker [275]; Agent-as-a-Judge [436]; EvalPlanner [278]; Kim et al. [152]

中根据 rollout 生成的标签更新过程奖励模型 (Feng et al., 2023; Zhang et al., 2024a)，或者根据生成的正负样本训练结果奖励模型 (Hosseini et al., 2024; Yang et al., 2024a)。例如，V-star (Hosseini et al., 2024) 训练一个 ORM 以利用采样过程中生成的负样本，并在推理时用于重新排序。在更新过程中，用于下一代数据生成的策略模型可以从初始模型或当前迭代的模型中进行微调。训练数据可以来源于多个来源：当前轮次、初始数据集或所有先前轮次的累积数据。

尽管 ISRL 在离线方法赋能下的数据效率和无需专家演示的优势方面颇具前景，但实验结果表明，经过 4-5 次迭代后，改进率往往趋于平稳甚至略有下降 (Wu et al., 2024b)。这一现象与 RL 扩展方法形成对比，后者中模型性能单调提升。两个算法之间的主要区别在于，ISRL 更接近离策略采样，其中每次迭代中训练数据的收集和利用是分开处理的。正如 Tajwar et al. (2024) 所展示的，更高程度的在策略采样能带来更好的表现。这一观点得到了 Shao et al. (2024) 的观察支持，即在线拒绝采样微调 (RFT) 在训练早期阶段与 RFT 表现相当，但在后期阶段获得显著优势。此外，对于使用监督微调进行策略更新的算法，它们并未利用负梯度来抑制某些响应。实验结果表明，在策略更新中包含负梯度相较于仅使用正梯度能显著提升性能，尤其是在推理任务中 (Kimi et al., 2025)。

## 6 进展如何——迄今的应用

在本节中，我们将考察测试时扩展驱动的认知工程给人工智能研究带来的系统性变化以及已经出现的应用。

## 6.1 数学

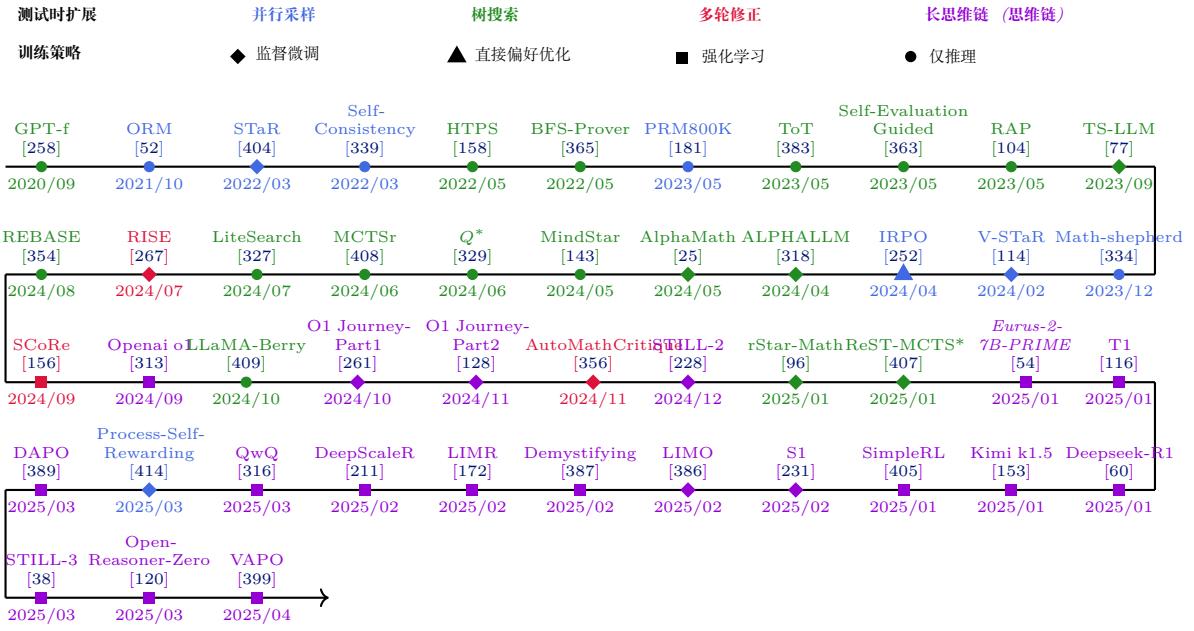


图 12: 在数学领域中应用测试时扩展方法的相关工作时间线。

数学推理对解决复杂问题和做出明智决策至关重要 (Hendrycks et al., 2021; Xia et al., 2024)。数学人工智能 (AI4Math) 研究沿着两条互补的路径发展: 自然语言推理专注于具有可验证答案的问题, 而形式语言推理利用像 Lean (De Moura et al., 2015) 和 Isabelle (Nipkow et al., 2002) 这样的形式系统进行自动形式定理证明。对于具有可验证答案的问题, 易于验证的特性使构建搜索和学习的反馈信号变得可靠, 促进了测试时扩展方法的广泛应用以增强推理能力。这些包括并行采样 (Cobbe et al., 2021; Wang et al., 2023c)、树搜索 (Feng et al., 2023; Chen et al., 2024a; Hao et al., 2023)、多轮修正 (Kumar et al., 2024; Qu et al., 2024) 和长思维链 (DeepSeek-AI et al., 2025; OpenAI, 2024)。值得注意的是, 在长思维链的支持下, DeepSeek-R1 在美国邀请数学考试 (AIME) 上获得了 79.8 分, 显著超越了没有长思维链的传统模型, 接近有竞争力的人类表现。对于形式语言推理, 形式系统使推理过程可验证, 并为树搜索 (Polu and Sutskever, 2020; Lample et al., 2022; Xin et al., 2024, 2025) 或多轮修正 (First et al., 2023) 提供信号。像 AlphaProof (AlphaProof and teams, 2024) 和 AlphaGeometry (Trinh et al., 2024) 这样的突破性系统证明, 将神经网络与形式方法和证明检查器结合可以实现前所未有的数学推理能力。

尽管取得了这些成功, 仍有改进的空间。在自然语言推理中, 虽然训练数据积累很多, 但严格验证推理过程正确性的难度意味着大语言模型生成的解决方案可能包含逻辑错误或中间步骤缺乏严谨性 (Lightman et al., 2023; Xia et al., 2024)。对于形式语言推理, 虽然它确保了推理过程的可验证性, 但与自然语言相比, 训练数据的缺乏限制了其发展。未来的工作可以专注于统一形式语言和自然语言的优势, 以开发更健壮的模型。此外, 虽然具有强大推理和认知能力的大语言模型在考试问题和竞赛任务上取得了进展, 但在数学研究等更高级领域的应用仍相对未被探索 (Yang et al., 2024b)。这不仅需要增强模型能力, 还需要新型评估框架来评估这些能力。

## 数学领域的未来方向

- 统一形式语言和自然语言的优势，开发更加健壮的推理模型，结合形式系统的可验证性与自然语言中可获得的丰富训练数据。
- 扩展具有强大推理能力的大语言模型的应用范围，从考试问题扩展到数学研究等更高级领域，开发新型评估框架来评估这些更高水平的能力。

## 6.2 代码



图 13: 在代码领域中应用测试时扩展方法的相关工作时间线。

语言模型编程能力的迅速发展——以 Codex (Chen et al., 2021) 和 AlphaCode (Li et al., 2022; Team, 2024a) 为代表——正在重塑软件开发流程，并显著提升开发效率。已有多项研究指出，代码能力有助于提升模型智能水平 (Fu and Khot, 2022; Ma et al., 2023; Shao et al., 2024)。此外，编程能力如今已成为通用基础模型的核心组成部分 (DeepSeek-AI, 2024)，进一步凸显其在现代模型开发中的关键地位。

此前的代码生成与代码合成研究表明，引入执行时的验证机制能够为训练阶段与测试阶段同时提供可扩展且可验证的信号，为后续技术发展奠定了基础 (Le et al., 2022; Chen et al., 2023a; Zhu et al., 2024)。此外，直接提示大语言模型自我反思、调试并生成测试用例，是另一种提升测试阶段表现的可扩展方法 (Shinn et al., 2023; Chen et al., 2023c)，这不仅能在特定下游任务中提升模型性能，还能推动训练策略的优化 (Gu et al., 2024a)。这些工作的探索被认为对更强推理能力模型的发展至关重要，例如 OpenAI o1 系列模型便在 SWE-bench 等顶级编程基准上取得了最先进的性能 (Jimenez et al., 2024)，甚至在 Codeforces<sup>7</sup> 等具有人类竞技水平的平台上也展现出色表现。这一点也体现在 o1 与 o3 系列模型在 2024 年国际信息学奥林匹克竞赛 (IOI) 中获得了金牌的成绩，表明其在结合强化学习与具有人类认知对齐的引导方式下，能够有效应对复杂编程任务 (El-Kishky et al., 2025)。

尽管如此，代码领域仍存在关键挑战。首先，尽管代码可执行性有助于验证，但直接执行存在安全风险，因而需要鲁棒的沙盒机制以确保安全 (Hui et al., 2024; Liu et al., 2024c)。这也要求配套基础设施的开发，以支持这些安全机制的实际部署。其次，频繁的反思行为——常被称为“过度思考”——在某些任务中可能导致性能下降，例如在 Aider 等代码智能体基准中已被观察到。<sup>8</sup> 第三，基于执行的反馈的可靠性仍是一个悬而未决的问题。尽管 DeepSeek-R1 (DeepSeek-AI et al., 2025) 依赖执行结果作为反馈信号以获得更优性能，但通过单元测试的代码在加入额外测试后仍可

<sup>7</sup><https://codeforces.com><sup>8</sup><https://aider.chat/>

能失败，导致假阳性 (Stroebel et al., 2024)，这凸显了基于执行的评估方式的根本局限。此外，还需进一步研究如何使模型更好地对齐于真实世界中的编程任务，正如 SWE-Arena (team swe arena, 2025) 和 Copilot-Arena (Chi et al., 2025) 所强调的那样。这些挑战甚至可能与多模态理解和智能体能力等更广泛的问题交叉，后续章节也将对这些挑战展开进一步讨论。

### 代码领域的未来方向

- 超越竞赛级编程任务，拓展至真实世界的软件开发，如自动化代码调试及仓库级优化。
- 扩展对更多编程语言的支持，并持续学习新发布的库，以实现专家级编程开发能力。

## 6.3 多模态

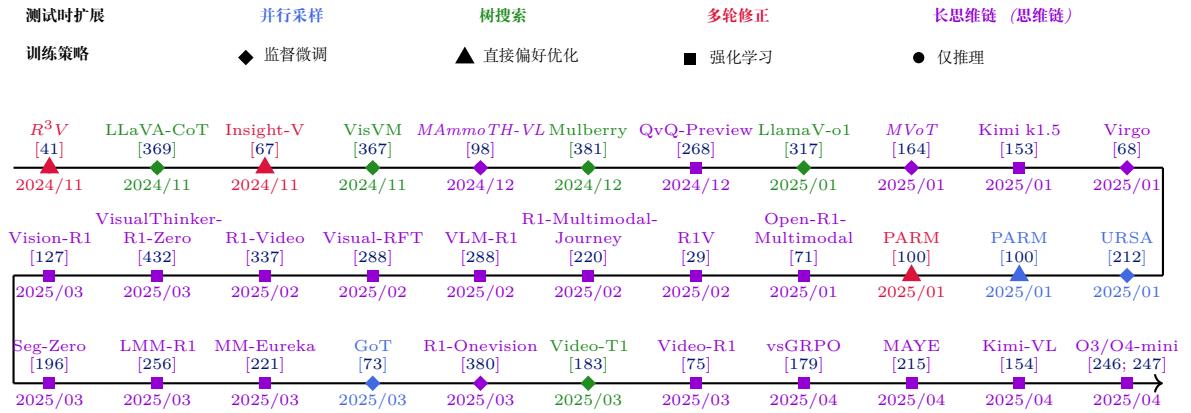


图 14: 在多模态领域中应用测试时扩展方法的相关工作时间线。

**多模态理解的测试时扩展** 视觉语言模型 (VLMs) (Zhang et al., 2024e) 在多模态理解和生成文本输出的任务中表现出巨大潜力。由于任务输出为文本，VLM 的测试时扩展技术可以直接从 LLM 借鉴。

DeepSeek R1 是这一领域的重要里程碑，它首次完整地展示了强化学习 (RL) 训练在 LLM 测试时扩展中的有效性，并有效地将 VLM 领域的研究划分为 R1 前和 R1 后的两个阶段。在 R1 之前，研究主要依赖长链推理蒸馏 (Guo et al., 2024; Du et al., 2025; Xu et al., 2024b)、树搜索 (Xu et al., 2024b; Thawakar et al., 2025; Yao et al., 2024a; Xiyao et al., 2024)、多轮修正 (Cheng et al., 2024b; Dong et al., 2024) 和并行采样 (Luo et al., 2025d)。例如，LLaVA-CoT (Xu et al., 2024b) 和 LlamaV-O1 (Thawakar et al., 2025) 通过明确分解推理步骤 (如问题总结、图像内容回顾及推理执行) 增强模型的推理能力。同样地，Mulberry (Yao et al., 2024a) 和 LLaVA-CoT (Xu et al., 2024b) 使用波束搜索和蒙特卡罗树搜索 (MCTS) 等搜索算法拓展推理空间。MAmmoTH-VL (Guo et al., 2024) 和 Virgo (Du et al., 2025) 通过从更大的 VLM 蒸馏推理链，提高了小型模型的输出长度，在视觉推理任务中显著提升表现。同时期的研究 QVQ-72B-Preview (Qwen, 2024) 和 K1.5 (Kimi et al., 2025) 展示了强化学习在 VLM 领域的巨大潜力。

在 R1 之后，研究界愈加倾向于利用强化学习训练激发 VLM 的测试时扩展能力。近期的研究主要分为两个方向：(1) 加强多模态推理的深度，进一步推动 VLM 在视觉问题求解方面的极限，例如 open-r1-multimodal (EvolvingLMMs-Lab, 2025)、MM-Eureka (Meng et al., 2025b)、LMM-R1 (Peng et al., 2025)、Vision-R1 (Huang et al., 2025c)、VisualThinker-R1-Zero (Zhou et al., 2025a) 等；(2) 扩大多模态任务的广度，验证强化学习训练在各种视觉中心任务 (如视觉计数 (Chen et al., 2025c)、检测 (Liu et al., 2025i; Shen et al., 2025a)、分割 (Liu et al., 2025e) 等) 中的有效性。

尽管成果显著，基于长链推理的 VLM 测试时扩展仍面临挑战。首先，与 LLM 不同，VLM 的输入指令涉及视觉和文本等多种模态。许多直接复制 LLM 扩展方法到 VLM 的研究未能在 VLM 的回应中纳入对视觉输入的反思，忽略了多模态理解任务的独特特性。未来的研究应更好地整合不同模态输入，解决非文本输入导致的幻觉问题 (Liu et al., 2024b)，同时利用多模态输入的协同效应提升扩展的有效性。其次，直接在基础模型上进行训练仍然存在困难。与 LLM 不同，VLM 缺乏强大的基础模型，因为它们的训练主要集中在通过图像-文本对和指令调优进行模态对齐 (Liu et al., 2023a, 2024a; Li et al., 2024a)，而缺少大量通用多模态语料的预训练，这需要巨大的计算资源和数据资源。

**多模态生成的测试时扩展** 近期 Gemini 的图文交错生成和 GPT-4o 的图像编辑能力的突破，再次激发了研究界对多模态生成的广泛兴趣。不同于多模态理解任务（输出为文本，可直接借鉴 LLM 测试时扩展技术），多模态生成任务涉及非文本输出（如图像、视频或音频），因此需要新的方法以释放测试时扩展在此领域的潜力。

最近一些研究探索了这一前沿领域。Li et al. (2025a) 通过多模态思维可视化 (MVoT) 框架增强了 MLLM 的空间推理能力，采用长链推理策略进行测试时扩展。PARM (Guo et al., 2025b) 和 MINT (Wang et al., 2025d) 通过多轮修正促进多模态生成，通过反复反思和修改相关文本提示以提升图像质量。GoT (Fang et al., 2025) 引入生成链式思维 (Generation Chain-of-Thought) 框架，释放 MLLM 在视觉生成和编辑方面的推理能力，采用长链推理策略实现测试时扩展。此外，Video-T1 (Liu et al., 2025a) 将输入模态扩展到视频生成，通过推理时的树搜索方法提升视频生成质量。

展望未来，将测试时扩展应用于多模态生成是一个极具前景但尚未被充分探索的领域。待解决问题包括确定最有效的模型架构 (Chameleon, 2024; Zhou et al., 2024a; Chen et al., 2025f)、为非文本输出设计适合的预训练和指令调优策略、提高系统的计算效率等。此外，若要将强化学习延伸到多模态生成领域，则需要解决如何定义合适的奖励信号以及构建适用于非文本模态的强化学习基础设施等关键问题。

多模态领域的未来方向

- 更加关注将测试时扩展应用到以视觉为中心的任务（如分类和检测）以及图像和视频等非文本输出。
  - 整合更多模态（如音频），开发更强大的原生多模态基础模型，以释放测试时扩展的潜力。

#### 6.4 智能体



图 15: 在智能体领域中应用测试时扩展方法的相关工作时间线。

LLM 智能体是一种自主系统，它利用 LLMs 作为其认知核心，通过动作执行在动态环境中自动完成复杂任务 (Sumers et al., 2023)。在先前研究的基础上，智能体的目标正在从特定的预定义工作流程转向处理更开放的任务，这些任务需要在复杂环境中进行大规模决策，例如软件工程 (Jimenez et al., 2024)、深度研究 (OpenAI, 2025b) 和计算机使用 (Anthropic, 2024b; OpenAI, 2025a)。此类任务通常需要长期规划，通过与环境的多步交互来完成，从而导致大量的测试时计算。例如，OpenAI DeepResearch (OpenAI, 2025b) 完成一项研究任务需要 5–30 分钟，而 CUA (OpenAI, 2025a) 可能需要数百步才能完成一项计算机使用任务，并表现出明显的测试时扩展行为。

有效执行多步任务的先决条件是在每一步中增强决策能力。这需要模型具备高级推理能力，以便根据历史轨迹和当前环境观察执行验证、回溯和反思，从而使行动与长期目标保持一致。许多方法采用了测试时扩展策略来优化每步决策质量。例如，ReAct (Yao et al., 2023b) 在动作选择过程中引入了 CoT 推理。Reflexion (Shinn et al., 2023) 进一步推进了这一范式，通过整合先前步骤的显式反馈信号，实现自我修正。最近，深度研究强调了在单步决策中扩展推理过程的潜力。在 OpenAI o3 的优化版本的支持下，它在将在线资源综合为全面报告方面达到了研究分析员的水平，并在具有挑战性的“人类最后的考试”基准测试中取得了 26.6% 的准确率 (Phan et al., 2025)，显著超越了之前的 SOTA 模型。

在训练策略方面，许多方法将历史轨迹引入 SFT 训练样本中，以帮助模型学会处理多步历史，并在每一步中整合思维过程，以实现 CoT 能力 (He et al., 2024)。此外，UI-TARS (Qin et al., 2025) 通过 DPO 方法解决了 SFT 方法仅利用修正步骤的局限性，从而提高了智能体的错误修正和后反思能力。尽管 OpenAI Deep Research 的具体实现尚不明确，但最近的工作应用强化学习对深度研究智能体进行端到端训练 (Zheng et al., 2025)。

尽管有一些生产就绪的实现，例如 GitHub Copilot，但大多数智能体系统仍局限于概念验证演示，而非大规模稳健部署。主要障碍包括通用模型能力不足、提示工程主导而非专门的智能体训练，以及长轨迹的上下文窗口限制——特别是对于视觉观察。此外，与代码或数学任务不同，许多智能体任务缺乏定义良好的外部验证器，这使得在强化学习框架中提供可靠的奖励变得具有挑战性。此外，长 CoT 推理的参与也引入了“推理-行动困境”，这要求模型在积极参与环境和内部推理需求之间仔细权衡，凸显了开发在应用于智能体任务时仍能有效基于环境背景的推理模型的重要性 (Cuadron et al., 2025)。

### 智能体的未来方向

- 开发支持多样化工具使用的稳健执行环境，并构建可扩展的评估框架，共同为强化学习在智能体训练中的应用铺平道路。
- 进一步探索动作扩展作为新的扩展维度——通过增加与环境的交互步骤来扩展智能体的能力。

## 6.5 具身智能



图 16: 在具身智能领域中应用测试时扩展方法的相关工作时间线。

具身智能是推动通用人工智能发展的关键要素，其通过建立认知表征与物理世界交互的基础性联系，为智能系统奠定了重要基础。该技术通过使机器人具备环境感知与物体操控能力，能够执行现实世界中的复杂任务，这需要系统具备高阶认知与推理能力。认知工程学的应用进一步强化了具身人工智能系统的认知与推理能力水平。典型的具身人工智能系统采用分层式架构设计，包含两个既相互独立又有机衔接的运作阶段：高层任务规划与底层控制策略执行 (Ahn et al., 2022)。

首先，高层规划是指为机器人创建可执行子任务序列的过程，使其能够在所处环境中实现特定目标。该过程基于机器人当前状态与行为预测结果进行决策，旨在优化动态复杂环境中的效率、安全性和目标达成度。这通常需要具备推理能力的高级认知功能。Huang et al. (2022) 率先将内心独白作为反馈机制以增强推理能力，通过内部对话实现多轮自我修正。基于此，Liu et al. (2023e) 利用大语言模型显式地生成对错误的解释，并据此优化推理流程，从而显著提升规划与问题解决能力。Ren et al. (2023) 提出通过量化基于大语言模型的规划不确定性，在超过预设阈值时触发辅助请求，实现基于不确定性的校准的推理增强。受 o1 系统架构启发 (OpenAI, 2024)，Liu et al. (2025d) 采用空间坐标对齐与空间定位技术的思维链，促进长序列思维生成，进而提升模型规划性能。此外，Chai et al. (2024) 提出基于 Q 学习的创新方法，使模型能做出最优决策。受 Deepseek-R1 架构影响 (DeepSeek-AI et al., 2025)，Azzolini et al. (2025) 开发了专为增强物理环境推理能力设计的新型视觉语言模型。该模型整合 Mamba、MLP 与 Transformer 的混合架构，结合视觉预训练、物理世界监督微调及物理世界强化学习，使其能有效处理视频输入与语言指令，生成长推理序列后预测下一个动作。与此同时，Zhang et al. (2025d) 构建了 9.3K 合成数据集，利用 GPT-4o 生成“观察-思考-行动”轨迹，为长程动作预测任务提供了详细的推理链。

其次，低级控制策略的核心目标是将任务转化为可执行的动作，例如可以在七自由度机械臂上执行的动作。当前主流方法主要依托大模型技术，尤其是基于视觉-语言-动作 (Vision-Language-Action, VLA) 框架的研究。这类模型使用轨迹数据来微调已预训练的视觉-语言模型，生成可执行的动作序列。值得注意的是，推理能力在此过程中起着关键作用。例如，当执行“将苹果置于 A 盘、香蕉置于 B 盘”的指令时，模型需首先完成水果类别的语义识别，而非简单依赖先前学习形成的“肌肉记忆”。具身思维链 (Michał et al., 2024) 通过整合外部模型或算法的先验知识，构建了连接感知信息与任务目标的推理链条，实验证明该方法能显著提升系统性能。相较之下，Zhao et al. (2025) 提出的视觉思维链框架，显式地将视觉推理过程嵌入到 VLA 模型，通过自回归方式先预测未来图像帧作为视觉上的子目标，再据此生成最终的动作序列。Zhang et al. (2024h) 开发了机器人策略学习的偏好对齐方法，使 VLA 模型不仅能从成功轨迹中学习，还能有效利用失败轨迹的反馈信息。此外，Li et al. (2024b) 通过融合多模态机器人功能的表征信息，增强了模型在测试阶段的推理泛化能力，并利用生成式推理机制提升长程推理能力。然而，基于监督微调的方法高度依赖高质量轨迹数据集，这在机器人领域面临着采集成本高昂且数据稀缺的挑战。更关键的是，由于分布偏移问题，现有数据集难以确保 VLA 模型与现实物理环境的适配。针对这一局限，Guo et al. (2025a) 提出了一种融合在线强化学习与监督学习的交替训练范式，显著提升了 VLA 模型的泛化性能。与此同时，Clark et al. (2025) 利用大规模人类视频数据来增强推理能力：首先通过 Gemini 模型解析人类视频来生成推理步骤，再结合采用有限机器人数据训练出来的模型将抽象推理映射为底层动作，然后利用这些视频数据获取到的可执行动作持续提升模型的推理水平。值得一提的是，Team et al. (2025) 基于 Gemini 2.0 视觉语言模型开发了 Gemini Robotics-ER 系统，该模型展现出卓越的具身推理能力。研究团队进一步拓展该工作，构建了整合机器人动作数据的 VLA 模型 Gemini Robotics，实现了跨任务跨形态的高频精密控制、强泛化能力与快速自适应能力。

尽管已有大量研究通过引导生成较长的思维链来显著提升了模型性能，但在具身智能领域仍存在较大的研究空间。首先，尽管纯强化学习在大型语言模型中已被证明能够通过自主探索促进自我

反思能力 (DeepSeek-AI et al., 2025)，但在具身智能中尚未建立起行之有效的方法。其次，现有框架通常将高层规划与低层策略相分离。然而，更为有效的范式应当将规划与执行融为一体，通过内部推理与迭代反馈实现持续迭代优化。

### 具身智能领域的未来方向

- 构建统一的框架，将高层规划与低层策略执行整合在一个独立的模型中，使智能体在与物理环境交互的过程中，能够通过内部推理反思和实时执行反馈不断优化其模型行为。
- 建立细粒度的评估框架，系统性地分离并评估具身智能体在思维、规划与执行等环节的性能，从而更深入地揭示各个组成部分对整体性能的具体贡献。

## 6.6 安全对齐

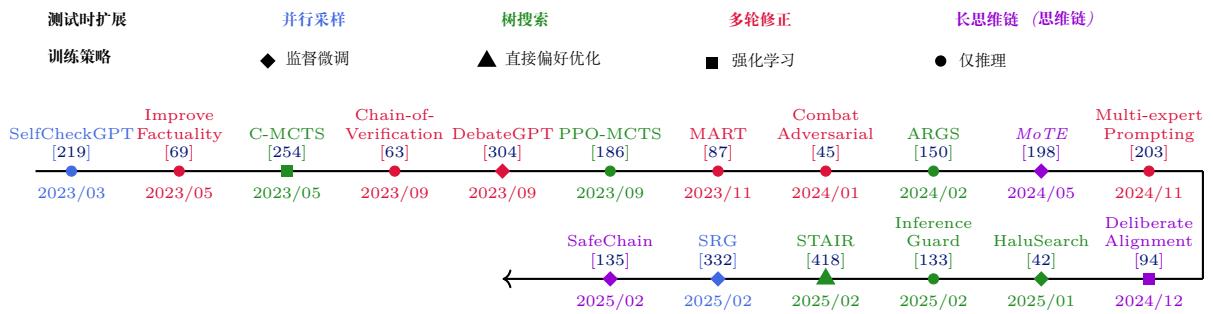


图 17: 在安全对齐领域中应用测试时扩展方法的相关工作时间线。

由于测试时扩展的出现，使得 AI 系统在进行复杂的、长时推理方面取得了显著进展，对 AI 安全具有重要意义 (Bengio, 2023; Park et al., 2024)。这一进展带来了双重影响：一方面，能够解决高度复杂问题的 AI 系统，有望帮助识别并应对新兴的安全挑战，甚至能够主动发现和缓解那些人类可能忽视的风险，从而更全面地探索边缘案例和潜在漏洞 (OpenAI, 2024)。但另一方面，这类系统也可能做出超出人类认知能力范围的决策，采取与人类价值观不一致的策略，在长时间、难以控制的交互中引发灾难性后果 (Hendrycks et al., 2023)。下文将探讨在大语言模型中，测试时思维扩展如何帮助识别和缓解安全风险，例如幻觉生成、越狱、对抗性攻击等问题。

近年来，越来越多的研究开始探索并行采样作为提升大语言模型安全性推理的一种机制，尤其适用于无法重新训练模型或无法访问模型参数的场景。其核心思想在于：对同一输入进行多次采样生成响应，有助于更可靠地评估事实性、不确定性以及与安全目标的一致性。SelfCheckGPT (Manakul et al., 2023) 提出了一种零资源的幻觉检测方法，利用采样来识别不同生成结果之间的事实不一致，基于的前提是假如模型知识可靠，其生成结果应具有一致性。类似地，Lin et al. (2024) 从语义离散度的角度对这一原理进行了形式化，证明通过采样结果的多样性来估计不确定性，可以在仅有黑盒访问权限的情况下，可靠地预测模型输出的可信度。尽管上述方法都聚焦于采样框架，SRG (Wang et al., 2025b) 则提出通过预设的安全策略，将显式推理步骤注入模型中。在使用 Best-of-N 采样进行评估的实验中，该方法在应对分布外 (OOD) 攻击方面展现出了更好的泛化性能。这些方法共同展示了测试时扩展的一种强大范式：在推理过程中通过多个输出的操作来增强模型的对齐性、鲁棒性和事实可靠性，而无需更改模型权重本身。因此，并行采样不仅是一种有效的潜在知识挖掘策略，也是一条脱离昂贵训练流程、实现安全性提升的可扩展路径，为大模型在现实世界中的可信部署提供了重要方向。

除了并行采样策略之外，基于树搜索的方法也为测试时安全性提供了一种更具结构化的路径，

## 6.7 检索增强生成

通过有意识地探索替代输出或推理路径，实现更安全的模型行为控制。例如，InferenceGuard (Ji et al., 2025) 和 ARGs (Khanov et al., 2024) 等技术将生成过程建模为一个受奖励约束的搜索过程，借助限制性马尔可夫决策过程 (constrained Markov decision process) 在大语言模型的潜在空间或学习到的奖励模型中，引导生成结果朝向安全对齐的方向。其他方法如 HaluSearch (Cheng et al., 2025) 则通过显式地搜索中间思维步骤，并使用自我评估或特定规则对其进行打分，来提升事实可靠性和推理鲁棒性（例如判断模型何时应“慢思考”，何时应回退至快速生成模式）。而以规划为核心的策略，如 C-MCTS (Parthasarathy et al., 2023) 和 STAIR (Zhang et al., 2025e)，则将安全评估器或内省机制整合进 MCTS 中，以规避决策过程中可能出现的不安全行为序列。这些方法通过支持回溯和显式展示可解释的中间决策步骤，显著提升了测试时对齐能力，在模型规模不断扩展的背景下，为维护安全性提供了关键优势。

在上述基础上，近期研究进一步探索了在推理过程中增加交互步骤（如多轮纠错或多智能体协作）来提升模型的安全性与鲁棒性。多个多智能体交互框架 (Du et al., 2023; Ge et al., 2024; Chern et al., 2024b; Long et al., 2024) 已被证明能够有效减少有害输出。通过模型之间的协同批评与修正，这些框架能够识别并缓解潜在的安全风险，例如减少幻觉、有害内容以及对抗性攻击。此外，通过结构化辩论来鼓励发散性思维 (Liang et al., 2023)，可以促进更细致入微的推理与交叉验证，从而提高生成内容的可靠性。近期也有研究利用更长的思维链推理来提升模型在测试时的安全性，通过延长模型在推理过程中的思考深度，增强其审慎能力。SafeChain (Jiang et al., 2025) 提出了一个包含长文本安全对齐推理的训练数据集，实验证明，在此基础上微调的大模型不仅能够保持高水平的推理能力，还能提升拒答率、减少有害内容的输出。Deliberative Alignment (Guan et al., 2024) 则通过训练模型在作答前显式查阅并推理安全策略，实现多步链式思维。这类“先思考、后回答”的模型在推理深度提升的同时也变得更安全，表明链式思维的长度与测试时对齐能力呈正相关。类似地，Chain-of-Verification (Dhuliawala et al., 2023) 提出通过引导模型生成并回答关于自身输出的验证问题，将推理过程转化为一个多阶段、自我审核的链式流程。随着模型能力增强、推理链加长，该方法在事实安全性上表现更为可靠。最后，MoTE (Liu et al., 2024e) 将安全推理过程划分为多个专门的链式思维阶段（如问题分析、指引、回答、检查），并为每个阶段分配专家模块。这种模块化方法能充分利用更大模型和更长推理链的优势，实现更具可扩展性和可解释性的自我对齐。

这些研究共同表明，在推理过程中有策略地增加交互与推理步骤，为实现更安全、更稳健的模型行为提供了一条可扩展的路径，而无需重新训练模型。

### 安全对齐领域的未来方向

- 探讨测试时扩展方法如何与现有的对齐策略（如 RLHF 和基于过程的监督）相结合，以确保它们相辅相成，并了解它们如何共同提升模型的安全性。
- 研究测试时扩展方法在多大程度上能够提升模型对现实世界的安全挑战（包括欺骗、对抗性攻击和其他分布外威胁）的泛化能力。
- 进行严格测试，以评估长链思维模型在面对各种越狱行为和攻击时的鲁棒性。这包括分析在测试时增加计算量是否会意外引入新漏洞，或是缓解已有安全问题。

## 6.7 检索增强生成

检索增强生成 (RAG) 系统通过整合外部知识源来提升大型语言模型的能力，使其回应更加符合事实且有依据。尽管这些系统效果显著，但在需要跨多文档进行复杂推理时往往效果不佳。测试时扩展已成为一种很有前景的方法，通过在推理过程中合理地增加计算资源，可以显著增强 RAG 系统的推理能力。



图 18: 在检索增强生成领域中应用测试时扩展方法的相关工作时间线。

Yue et al. (2024) 提出了 IterDRAG，它将复杂问题分解为一个个子查询，并进行迭代式的搜索和推理过程来构建全面的回答。这项研究表明，RAG 性能与有效上下文长度之间存在近乎线性的关系，为 RAG 系统建立了明确的测试时扩展法则。多项后续研究进一步验证了这种工作流程的有效性 (Verma et al., 2024; Guan et al., 2025a; Li et al., 2025e; Feng et al., 2025b; Yu et al., 2024b; Wang et al., 2025c)。

除了基于提示的智能体外，研究人员还通过微调大型语言模型来端到端地融合推理与搜素能力。收集训练数据的一种方法是通过拒绝采样合成推理和搜索轨迹，然后通过监督微调 (Wang et al., 2025c; Yu et al., 2024b) 或偏好微调 (Guan et al., 2025a) 对模型进行训练。近期，强化学习已被应用于 RAG 的端到端训练，如 R1-Searcher (Song et al., 2025)、Search-R1 (Jin et al., 2025) 和 ReSearch (Chen et al., 2025d) 等研究。这些方法使模型能够通过试错学习更高效的搜索策略，而非模仿人类设计的搜索模式。

然而，这些工作存在一个显著的局限性：它们主要专注于开放域问答任务，并奖励机制依赖于为简短、事实性回答设计的规则式的奖励。这种方法可能难以很好地推广到更复杂的推理任务，特别是那些需要在回复中详细解释的任务。

#### 检索增强生成领域的未来方向

- 构建更为精妙的强化学习奖励机制，使其能够有效评估长篇内容的生成质量，而非仅限于简短的事实性回答。
- 打造专门的评估体系，能够区分并精准测量 RAG 系统中内部推理能力与外部检索功能各自的价值和对回答的贡献。

## 6.8 评估



图 19: 在评估领域中应用测试时扩展方法的相关工作时间线。

LLM-as-a-Judge (Zheng et al., 2023a; Gu et al., 2024b) 这一范式已经彻底改变了语言模型的评估方式，使评估从基于规则的指标（如 BLEU 和 ROUGE）转向更接近人类判断的评估。近期研究表明，在推理阶段分配额外的计算资源能显著提升评估质量。这些方法包括：细粒度评估，即将 LLM 的回应分解并逐步检验 (Chern et al., 2023; Min et al., 2023; Hu et al., 2024b; Ru et al., 2024)、将 LLM 评估器的思维链结构化为明确的规划与执行阶段 (Saha et al., 2025)、采用多智能体

系统提供中间反馈 (Zhuge et al., 2024)，以及利用并行采样的群体回应进行成对比较 (Zhang et al., 2025b)。此外，MCTS-Judge (Wang et al., 2025f) 将蒙特卡洛树搜索应用于代码评估，在评估时探索不同的评估视角，证明了这种方法的可扩展性——增加搜索深度和采样次数能持续提高准确性。Kim et al. (2025) 还观察到，生成更多推理内容能够提升长思维链模型的评估性能。

随着人工智能向更贴近现实世界的任务发展（如由多个子任务组成的软件开发），当前的评估框架越来越聚焦于复杂智能体和工作流程 (Jimenez et al., 2024; Xie et al., 2024a)。未来研究方向应当着重提高对这些复杂且长期任务的评估可靠性，这需平衡测试时扩展的收益与评估速度。

### 评估领域的未来方向

- 提升对复杂且长周期任务的评估可靠性。
- 将评估框架和强化学习训练中的奖励设计进行整合，充分释放强化学习的潜力。

## 7 那又怎样？——从规模化到认知智能

认知工程标志着人工智能领域的一个根本性范式转变。这一转变远不止于技术实现，它对我们如何开发人工智能系统、重新构想人机协作以及开展科学研究都产生了深远的影响。

### 7.1 数据工程 2.0：认知数据工程

传统人工智能主要关注知识获取——训练系统学习人类思维的成果。然而，认知工程要求一种根本性的不同：从思维成果转向思维过程本身。这一转变催生了一门新学科——**认知数据工程**，它彻底改变了我们对有价值训练数据的理解。认知数据来源于三个不同但互补的来源，每个来源都为开发过程带来了独特的优势和挑战：

**来源 1：人类认知投射** 尽管目前缺乏直接捕捉人类思维过程的脑机接口，我们仍可以通过物理世界中的投射来获取人类认知：

- **直接记录的产物。** 专家问题解决过程的视频记录、出声思考记录以及详细的研究日志，捕捉了认知过程的展开。这些记录不仅保留了解决方案，还保留了专家思维中的混乱现实——错误的开始、修改和突破。
- **工具介导的认知痕迹。** 复杂的认知活动在专用工具中留下了痕迹——实验室笔记本、协作白板会议、软件开发中的版本控制系统，以及科学论文通过草稿和修订的逐步完善。这些工具作为代理，使隐含的认知过程变得显性和可观察。
- **前沿专业知识提取。** 最有价值的认知模式通常存在于领域前沿专家的头脑中。这些模式需要精心设计的提取方法——专门的访谈技术、定制的问题场景和高质量的互动，将隐性知识提炼为显性的推理轨迹。

**来源 2：AI 生成的认知** 通过适当的奖励机制和复杂的强化学习方法，AI 系统现在可以在环境中独立生成有价值的认知数据或轨迹：

- **环境与奖励的协同作用。** 当提供设计良好的环境、适当的奖励函数和强大的初始化模型时，AI 系统可以通过扩展探索发现新的认知策略。这些策略可能与人类方法大不相同，但能达到同等或更好的效果——类似于 AlphaGo 著名的“第 37 手”，最初让人类专家感到困惑，但最终证明非常有效。
- **自我对抗与对抗性发现。** 系统可以通过与自己竞争或面对越来越复杂的场景，生成越来越复杂的认知数据，开发出仅靠模仿人类例子无法出现的推理策略。

- **认知发现中的规模化效应。**随着计算资源的增加，AI 系统可以探索由于生物限制（如记忆、注意力跨度或处理速度）而无法为人类所及的认知路径——可能在从数学到药物设计的各个领域中发现新的问题解决方法。

**来源 3：人机协作生成** 最有前景的或许是通过人机伙伴关系共同创造认知数据：

- **轨迹采样与人工过滤。** AI 代理可以生成多样化的解决路径，然后由人类专家评估和提炼，结合机器生成的多样性和人类对质量和相关性的判断。
- **人工种子与 AI 扩展。**人类专家可以提供复杂领域中的初始推理示例，然后 AI 系统进行**认知完成**（即扩展、系统化变化和完成）——创建比仅靠人工标注更大的训练数据集 (He et al., 2024)。
- **迭代优化循环。**人工和 AI 的贡献可以在渐进循环中交替进行，每一方都在对方工作的基础上进行增强——人工提供创造性飞跃或概念重构，AI 提供系统化的探索和边缘案例。

这种认知数据建立了一类全新的数字资源，有可能推动 AI 能力超越仅靠自然数据收集或合成生成所能达到的水平。由此产生的认知数据存储库很可能变得与大规模计算资源一样具有战略价值，成为决定 AI 进步领导地位的关键因素。

## 7.2 奖励与环境工程

向认知工程的转变从根本上改变了我们设计 AI 系统运行环境和引导其发展的奖励信号的方式。

### 7.2.1 奖励模型设计

认知工程中的一个关键趋势是随着任务复杂度的增加，奖励验证的难度也随之增加。数学奥林匹克问题代表了相对简单的验证——证明要么逻辑上成立，要么不成立。超越这一点，像 Deep Research (OpenAI, 2025b) 和 PaperBench (Starace et al., 2025) 这样的任务需要复杂的规划，同时产生的输出难以进行客观评估。科学发现和文学创作占据了验证复杂度的前沿，需要创造性和独特的见解，其价值评估无法完全客观化。这些领域涉及审美判断、原创性考虑和跨文化和视角的上下文相关性。为了应对这些挑战，我们提出了两种互补的方法：**基于参考的评估**：一种受文本摘要和机器翻译评估方法启发的方法，将输出与高质量范例进行比较，利用人类判断能力 (Bhandari et al., 2020)，而不需要正式的质量定义 (Qin et al., 2023; Yuan et al., 2021; Zheng et al., 2025)。**基于标准的评估**：建立结构化框架，将主观判断分解为具体组成部分和特定标准 (Fu et al., 2024a; Yuan et al., 2024b)。这些方法共同在简单正确性指标失效的领域中创建了强大的奖励信号。

### 7.2.2 认知环境设计

AI 认知发展的环境跨越了复杂度光谱——从纯文本交互 (Song et al., 2025) 到代码解释 (Li et al., 2025g)、浏览器环境 (Zheng et al., 2025)、完整计算机系统访问 (Anthropic, 2024b)，以及物理世界交互。在这个框架内，专门的认知模拟器开发了特定领域的推理：模仿假设-实验-分析循环的科学发现环境；要求法规解释和先例应用的法律推理竞技场；在不确定性下要求鉴别诊断的医学模拟器。对抗性框架通过结构化辩论平台、红队模拟和苏格拉底式对话加强了认知能力。这些环境应该形成认知课程，系统地开发从基础到前沿的能力。

## 7.3 人机认知伙伴关系

测试时扩展和认知工程的出现从根本上改变了人类与人工智能之间的关系，创造了超越传统工具-用户动态的合作可能性。这种伙伴关系代表了一种真正新型的认知生态系统，具有深远的影响。

**双向认知交换** 人类通过专家示范、元认知指导和价值对齐的推理示例传递思维策略，塑造 AI 系统处理问题的方式，而 AI 系统则相互照亮盲点，扩展策略库，并提供新颖的概念框架，增强人类思维。这些伙伴关系利用了每种智能类型的独特优势，创造了一种真正的认知互补性，超越了 **AI 作为工具** 和 **AI 作为替代** 的范式。

**认知放大** AI 系统通过管理细节、保持一致性和在整个复杂推理过程中保存上下文，扩展了人类的工作记忆，同时通过并行假设评估、穷尽推理链和反事实场景生成，超越了人类的探索广度。这种伙伴关系还通过推理可视化、假设浮现和推理链验证，使思维过程外部化，克服了人类认知的基本限制。

**新的交互范式** 共享工作空间允许人类和 AI 使用多种格式（图表、文本、符号）一起可视化问题，并实时评论彼此的工作，创建了他们思维过程的记录历史。通信变得自适应——AI 解释根据用户需求调整，仅在必要时揭示更多细节，并使用用户熟悉的概念。工具不仅设计用于解决问题，还用于相互改进：人类可以要求 AI 提供更清晰的解释，提供有针对性的反馈，双方都可以审查已完成的工作，从成功和错误中学习。这些方法将思维从孤立的活动转变为真正的协作过程，在这个过程中，想法不断被分享、提炼和共同构建。这种人类-AI 认知伙伴关系的演变远不止是 AI 助手的渐进式改进——它代表了一种全新的智力关系，有可能显著增强人类集体解决问题的能力。最深远的影响可能出现在那些问题超出个体人类认知能力，但需要上下文理解和价值对齐的领域中，而纯人工方法在这些领域中往往难以实现。

## 7.4 研究加速

将认知工程应用于科学研究，有望从根本上改变跨学科发现的步伐和性质。通过将人类科学思维与机器推理能力相结合，我们正站在知识创造可能前所未有的加速门槛上。传统科学发展缓慢，因为人类只能生成和测试有限的假设。认知系统通过系统地映射知识缺口、识别未探索的连接，并同时生成数千种可能的解释，克服了这些认知限制 (Lu et al., 2024a)。它们在大规模数据集中检测可能预示着突破机会的异常，并确定哪些实验最能有效地测试理论。这些能力极大地扩展了科学家可以提出的问题以及他们找到答案的速度。此外，专门研究创造了孤立的知识孤岛，阻碍了进步。认知工程通过连接跨学科、跨时间段和跨信息格式的发现，弥合了这些鸿沟。它识别了看似无关现象之间的相似性，并翻译专业术语以促进跨领域协作。这种整合促进了混合学科的出现，并允许洞察力在领域之间转移，将孤立的知识岛屿转变为连贯的科学网络，在这里，思想自由流动。通过认知工程加速科学研究，其意义远不止于效率提升。通过消除假设生成、文献整合、实验设计和理论提炼中的**认知瓶颈**，这些方法可能使人类能够以前所未有的速度和规模应对紧迫的挑战——从气候变化到疾病预防再到可持续发展。更深刻的是，通过民主化科学发现的参与，认知工程可能有助于实现全球人类智力的全部创造潜力，将多样化的视角带到我们最重要的问题上。

## 8 基础设施

本节，我们简要讨论下认知工程中的两个重要技术，强化学习和蒙特卡洛树搜索的基础设施。除此之外，长文本生成的加速也非常重要，我们建议感兴趣的读者可以参考该综述 (Liu et al., 2025b)。

### 8.1 强化学习

以 PPO 算法为例，一个经典的 RL 工作流包含两个主要的步骤：

- **生成 (Rollout):** 动作模型 (Actor Model) 基于预先准备好的提示，通过自回归解码生成提示的回答。这个过程需要使用到 Actor 模型的推理引擎。虽然传统的训练后段支持自回归生成，

但他们的速度通常很慢。为了解决这个问题，现代的 RL 框架通常会使用一个独立的推理引擎来 Rollout，比如 vLLM (Kwon et al., 2023) 和 SGLang (Zheng et al., 2023b).

- **模型更新:** 当有了生成的序列后，会首先计算动作模型在序列上的对数概率，价值模型 (Critic Model/Value Model) 的价值估计，参考模型 (Reference Model) 的对数概率，还有奖励函数。然后，这些值会被用于计算一些 KL 散度，回报 (Return)，优势 (Advantage)。最后，这些值会用于计算出动作模型和价值模型的损失 (Loss) 的值，并更新两者。模型更新的后端通常是 DeepSpeed (Rasley et al., 2020) 和 FSDP (Zhao et al., 2023c) 等。

一些流行的 RL 训练框架包括 DeepSpeed-Chat, NeMo-Aligner (Shen et al., 2024), OpenRLHF (Hu et al., 2024a) 和 veRL (Sheng et al., 2024). 其中，OpenRLHF 和 veRL 更新相对频繁。由于在涉及到众多的模型以及工作流程相当复杂，现有的 RL 框架通常采用不同的资源分类和进程调度策略。比如 OpenRLHF 可以为不同的模块分配不同的资源组，保证每一个模块独占其资源组。veRL 采用一个共享资源的方法，动态的将不在工作的模块资源即使释放，分配给其他正在工作的模块。

尽管这些开源框架简化了强化学习训练流程，但大规模的强化学习训练仍然面临重大挑战。比如，Yeo et al. (2025) 在尝试训练 32B 的模型时，发现需要的 GPU 数量过多。此外，研究者在训练过程中还发现 GPU 利用率偏低的问题，这一现象在长链式思维 (Long CoT) 场景中尤为突出，因为序列长度差异较大使得生成时间主要与最长序列有关。这凸显了针对此类场景优化强化学习框架的迫切需求。

## 8.2 蒙特卡洛树搜索

对于 MCTS 的基础设施，前面提到的几项工作已经发布了他们的代码，为在大语言模型中应用 MCTS 提供了基础 (Hao et al., 2024a, 2023; Chen et al., 2024a; Feng et al., 2023)。虽然这些代码库促进了后续研究，但它们通常缺乏考虑硬件和软件优化的加速策略，这限制了大规模 MCTS 的部署。我们概述关键加速策略如下。

**推测解码** 推测解码采用小型草稿模型顺序生成 token，由更大的目标模型验证这些 token，这在加速展开速度方面得到广泛实施 (Gao et al., 2024b; Wang et al., 2024h)。SEED (Wang et al., 2024h) 实现了预定推测解码，同时高效管理运行时速度和 GPU 内存使用。该框架利用轮次预定策略，使用先来先服务队列管理执行流程，控制目标模型的验证而不发生冲突。SC-MCTS (Gao et al., 2024b) 利用推测解码将 MCTS 推理速度平均提高 52

**KV 缓存管理** 大语言模型推理通常受内存带宽限制 (Hooper et al., 2024)。在树搜索中，每个唯一轨迹都需要单独的 KV 缓存状态，创造了重大的内存瓶颈。DEFT (Yao et al., 2024b) 引入了高效的内核实现，计算带有树结构 KV 共享的注意力。Hydragen (Juravsky et al., 2024) 和 vLLM (Kwon et al., 2023) 提供了对共享前缀工作负载的支持，有效消除了 KV 缓存重复。SGLang (Zheng et al., 2023b) 实现了 Radix Attention，存储并动态引用重用的 KV 缓存段。此外，ETS (Hooper et al.,

表 10: OpenRLHF 和 veRL 的基本信息与功能. 其中混合引擎指的是将各种训练和推理放在同一个资源组上，其相比于将不同的后端放在不同的资源组上具有更高的资源利用率。

RL 框架	支持的算法	混合引擎	训练后端	推理引擎	序列并行	多模态	
OpenRLHF	PPO, RLOO, FORCE++	GRPO, REIN-	支持	Deepspeed	vLLM, HF Transformers	Ring Attention	支持
veRL	PPO, RLOO, FORCE++, DAPO, PRIME	GRPO, REIN-	支持	Megatron-LM, FSDP	vLLM, SGLang, HF Transformers	DeepSpeed Ulysses	支持

2025) 采用线性规划成本模型，通过惩罚节点保留来鼓励 KV 缓存共享，同时融入语义覆盖参数以保持保留轨迹之间的多样性。

**并行处理** 树扩展和模拟阶段的加速可以通过并行处理技术实现。然而，路径之间的频繁切换使并行化变得复杂。Ding et al. (2025) 通过在生成阶段实现细粒度缓存管理和对齐，为任意路径开发了灵活且自适应的并行系统。该系统根据实时 GPU 内存可用性调整处理的并行路径数量，优化资源利用。

## 9 教程

本章节，我们提供了一个使用强化学习解锁大语言模型长思维链能力的教程。

### 9.1 准备工作

在开始正式训练之前，一些基础的准备虽然简单，但是对实验的成功很有必要。我们首先介绍下代码框架，使用的基础模型，训练数据集还有算法。

👉 **理解 RL 框架** 为了使用的便利性，稳定性和效率，本教程使用 veRL 作为框架。其中与算法有关的核心代码的组织方式如下：

- **trainer/** 包含核心的训练代码，比如训练控制流程，优势估计函数，损失函数等
- **utils/** 包括数据读取，模型保存以及其他工具
- **workers/** 定义了各种 workers，包括动作模型，价值模型，奖励模型，生成框架等
- **protocol.py** 定义了 veRL 中用于在不同 worker 之间进行数据交换的数据结构。

👉 **选择基础模型** 基础模型的选择对于实验结果有着决定性的影响。我们推荐使用 Qwen2.5-1.5B, Qwen2.5-3B 和 Qwen2.5-7B. 我们强烈推荐使用这些模型的 Base 版本而非 SFT 版本，因为 Base 版本因为没有在具体任务上拟合过，通常具有更高的多样性，进而具有更强的探索能力，使得他们在 RL 过程中可以发现更加多样化的解题策略。

👉 **准备数据集** 数据集的质量对于训练也很重要。我们提供了一个包含 NuminaMath, DeepScaleR 和 MATH 中数学问题的数据集。其难度适中，既能够挑战模型，也不会过于困难使得模型完全无法学习。在这些数据集上训练可以让模型在性能和解题长度上都稳定增长。

👉 **选择算法** 考虑到 GRPO 的有效性，在教程中主要使用 GRPO 算法。

### 9.2 启动 RL 训练

读者可以按照如下的命令来启动 Qwen2.5-1.5B 上的 RL 训练。

```

1 git clone git@github.com:GAIR-NLP/simple_tts.git simple_tts
2 cd simple_tts
3 # Create the conda environment
4 conda create -n verl python==3.10
5 conda activate verl
6 pip install -r requirements.txt
7 pip3 install vllm==0.7.3
8 pip3 install flash-attn --no-build-isolation
9 # launch training
10 bash examples/simple_tts.sh

```

如果希望使用其他模型，可以替换 `examples/simple_tts.sh` 文件中 `policy_path` 的值。我们提供了一个已经被验证有效的超参数，但也可以探索其他超参数。此外，你可以依据 `data/train/` 中的文件结构，自行构建训练集和测试集，并在 `examples/simple_tts.sh` 修改为响应的路径。

### 9.3 通过代码分析理解 RL 算法

本小节，让我们从代码实现来理解 RL 的算法。

**启动脚本** veRL 的启动脚本中包含了一些核心的算法参数：

```

1 python3 -m verl.trainer.main_ppo \
2     algorithm.adv_estimator=grpo \
3     data.train_batch_size=$rollout_batch_size \
4     actor_rollout_ref.actor.ppo_mini_batch_size=$mini_batch_size \
5     actor_rollout_ref.actor.kl_loss_coef=$kl_loss_coef \
6     actor_rollout_ref.actor.entropy_coeff=$entropy_coeff \
7     actor_rollout_ref.rollout.temperature=$temperature \
8     actor_rollout_ref.rollout.n=$n_samples_per_prompts

```

关键参数：

- `algorithm.adv_estimator`: 优势估计方法 (GRPO vs PPO/REINFORCE)
- `train_batch_size`: 每个采样批次中提示的数目 Number of prompts per sampling batch
- `mini_batch_size`: 每个更新批次中提示的数目
- `n_samples_per_prompts`: 为每个提示生成的回答的数目
- `temperature`: 控制生成的随机性
- `kl_loss_coef` and `entropy_coeff`: KL 损失和熵正则的系数

**算法流程分析** 在 `verl.trainer.ppo.ray_trainer` 中的 `RayPPOTrainer` 的 `fit` 方法是整个强化学习算法的控制流程。以下是简化的核心过程：

```

1 # Loop through the training set for total_epochs times
2 for epoch in range(self.config.trainer.total_epochs):
3     # Take train_batch_size prompts from the dataset
4     for batch_dict in self.train_dataloader:
5         # Prompts are organized into specific data structures
6         batch: DataProto = DataProto.from_single_dict(batch_dict)
7         # Rollout process, using inference engines to generate answers for prompts
8         gen_batch_output = self.actor_rollout_wg.generate_sequences(gen_batch)
9
10        # Performs forward propagation to obtain log probability of old policy
11        old_log_prob = self.actor_rollout_wg.compute_log_prob(batch)
12
13        # Perform forward propagation to get log probability of ref model
14        ref_log_prob = self.ref_policy_wg.compute_ref_log_prob(batch)
15
16        # Calculate value with critic model
17        # But our algorithm GRPO doesn't require critic

```

```

18     values = self.critic_wg.compute_values(batch)
19
20     # Use reward function to get reward for each sequence
21     reward_tensor = self.reward_fn(batch)
22
23     # Calculate the advantage value for each sequence based on the reward
24     batch = compute_advantage(batch, adv_estimator, gamma, lam, num_repeat)
25
26     # Update model using calculated advantage values
27     critic_output = self.critic_wg.update_critic(batch)
28     actor_output = self.actor_rollout_wg.update_actor(batch)

```

**👉 数据加载和提示模版** 在 `verl.utils.dataset.rl_dataset` 中的数据集类用于读取和处理数据。每条数据包含一个问题和一个答案。在数据集类中，问题会与特定的模板结合以生成提示 (prompt)。我们采用的模板是：

### Prompt

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. \nUser: You must put your answer inside \boxed{} and Your final answer will be extracted automatically by the \boxed{} tag.\nprompt\nAssistant:

该模板鼓励模型首先进行推理，然后提供最终答案，并用 `\boxed{}` 标记答案，以便后续评估。

**👉 奖励设计** 在 `verl.utils.reward_score.math_verifier` 中的 `compute_score` 方法展示了如何根据模型的解答和标准答案来计算奖励：

```

1 def compute_score(solution_str, ground_truth, reward_type) -> float:
2     return correctness_score_default(solution_str, ground_truth)
3
4 def correctness_score_default(response, gt):
5     # Use regular expressions to extract boxed content from the answer
6     pred = boxed_pattern.findall(response)[-1][-1]
7     # Judge whether the answer is correct
8     return 1.0 if is_equiv(pred, gt) else -1.0
9
10 def is_equiv(str1, str2, verbose=False):
11     # Parse and verify whether two answers are mathematically equivalent
12     return verify(parse(str1), parse(str2))

```

奖励函数十分直接，正确的答案 1 分而错误的答案-1 分。

**👉 GRPO 优势估计** GRPO 优势估计的实现在 `verl.trainer.ppo.core_alogs` 中：

```

1 # This implementation only considers outcome supervision, i.e., the reward is a scalar
2 def compute_grpo_outcome_advantage(token_level_rewards: torch.Tensor, eos_mask, index,
3     → epsilon: float = 1e-6):
4         for idx in id2score:

```

```

4         id2mean[idx] = torch.mean(torch.tensor(id2score[idx]))
5         id2std[idx] = torch.std(torch.tensor([id2score[idx]]))
6         # GRPO advantage estimation
7         for i in range(bsz):
8             scores[i] = (scores[i] - id2mean[index[i]]) / (id2std[index[i]] + epsilon)
9             # Expand scalar advantage to sequence length and mask with eos_mask
10            scores = scores.unsqueeze(-1).tile([1, response_length]) * eos_mask
11
12        return scores, scores

```

👉 动作模型损失函数 最后，以下是动作模型损失函数的计算方法，该方法同样在 `verl.trainer.ppo.core_alsogs` 中

```

1 def compute_policy_loss(old_log_prob, log_prob, advantages, eos_mask, cliprange):
2     # Calculate the log probability difference between new and old policies
3     negative_approx_kl = log_prob - old_log_prob
4     # Calculate probability ratio
5     ratio = torch.exp(negative_approx_kl)
6     # Calculate KL divergence
7     ppo_kl = verl_F.masked_mean(-negative_approx_kl, eos_mask)
8     # Original policy gradient loss
9     pg_losses = -advantages * ratio
10    # Clipped policy gradient loss
11    pg_losses2 = -advantages * torch.clamp(ratio, 1.0 - cliprange, 1.0 + cliprange)
12    # Take the larger of the two as the final loss
13    pg_loss = verl_F.masked_mean(torch.max(pg_losses, pg_losses2), eos_mask)
14    # Calculate clipping ratio
15    pg_clipfrac = verl_F.masked_mean(torch.gt(pg_losses2, pg_losses).float(), eos_mask)
16

```

## 9.4 结果

如图 20 所示，我们观察到以下方面的显著改进：

- 准确性：模型在解决数学问题时的准确性显著提高。
- 响应长度：模型生成的解题过程变得更加详细。

我们还建议通过例子分析以及跟踪响应中的特定关键词，分析模型是否表现出反思和自我纠正等高级认知能力。

## 10 未来方向

尽管我们已经在各个应用领域中概述了特定领域的未来方向，但认知工程也面临一些跨领域的基本挑战。在本节中，我们将识别这些关键的跨领域未来方向，这些方向可能会显著加速整个认知工程领域的发展。

**新架构** 基于 Transformer 的架构由于其线性内存扩展和在生成过程中的内存限制，面临着根本性的局限。这对于生成上下文时的测试时间扩展构成了关键约束。尽管在提高扩展效率部分描述的方法可以缓解这一问题，但更根本的解决方案需要探索新的架构。有前景的替代方案包括状态空间模型，如 Mamba (Gu and Dao, 2023; Dao and Gu, 2024)，它提供了线性时间复杂度的序列建模；

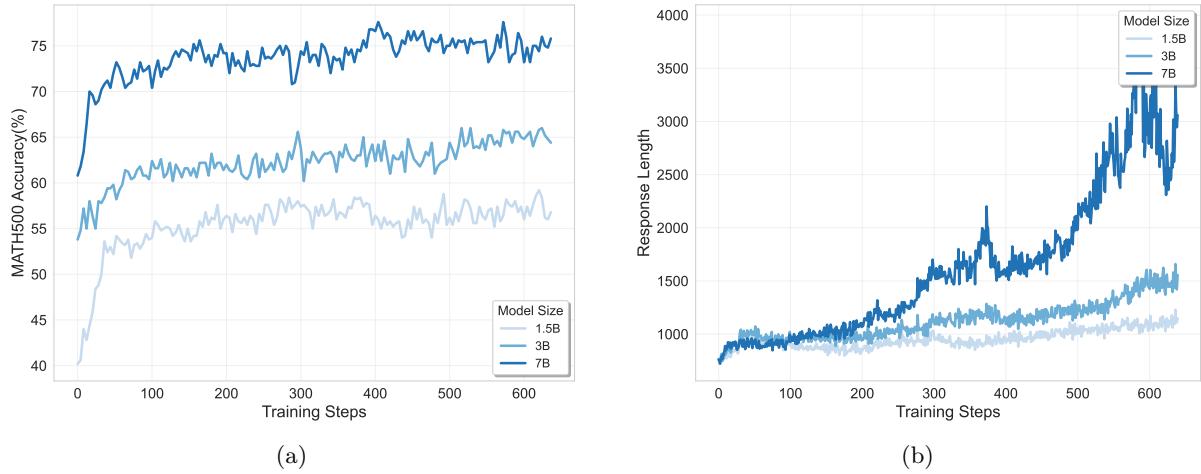


图 20: 在 Qwen-Base 模型上 RL 训练过程中，MATH500 上准确率和回答长度的变化趋势。

线性 Transformer，它减少了二次注意力瓶颈 (Katharopoulos et al., 2020)；甚至语言扩散模型 (Nie et al., 2025)。这种架构转型需要跨多个维度进行全面的系统工程努力：开发新架构的健壮理论框架，为高效训练和推理构建基础设施支持，以及基于这些架构创建大规模预训练基础模型。这些架构与认知工程技术的集成可能会显著增强认知能力和计算效率。

**认知数据预训练** 当前的预训练数据主要由人类书写的文本组成，但缺乏背后的潜在思维过程。在包含认知行为的数据上进行 RL 扩展的成功表明，包括人类思维过程具有潜力 (Gandhi et al., 2025; Liu et al., 2025g,f)。最近的研究表明，除了显式文本外，纳入隐藏思维过程也有益处 (Zelikman et al., 2024a; Jiang et al., 2024; Ruan et al., 2025)，尽管这些研究仍局限于小规模实验。未来的工作应侧重于通过利用现有推理模型推断潜在思维等技术获取大规模人类认知数据，并研究在这种数据上进行预训练如何有益于测试时扩展方法。

**强化学习** 尽管我们已经基于最新研究考察了 RL 扩展的通用设计原则，但该领域仍处于早期发展阶段，具有显著潜力来解锁 AI 的认知能力。鉴于 RL 的复杂组件和需要调优的众多超参数，未来的研究在得出结论时应采用更严谨的方法，考虑所有这些元素 (Jordan et al., 2024; Hochlehnert et al., 2025)。此外，可复现的开源工作目前仅限于小模型和数据集，这限制了可能结论的范围。大规模实验需要基础设施改进和算法优化，以便让计算资源有限的研究者也能进行。此外，当前的 RL 扩展主要集中在可验证的任务上，如数学和代码。扩展到更广泛的领域需要深入研究奖励欺骗现象，并建立奖励可靠性与 RL 扩展之间更清晰的关系。这一进展不仅需要实证研究，还需要理论分析。

**评估** 作为一种工程方法，认知工程依赖于迭代反馈和增强，这需要超越简单基准性能指标的评估方法。尽管一些工作已经开始关注认知行为变化，但这些努力通常依赖于匹配特定词语（如“wait”）或案例研究 (DeepSeek-AI et al., 2025; Gandhi et al., 2025)，未能完全捕捉认知行为的质量或推理过程的深度。未来的工作应开发全面的评估框架，不仅评估任务表现，还评估认知过程的质量、效率和通用性。这包括创建推理深度、回溯效率、验证质量和元认知意识等指标。此外，能够适应不断发展的认知能力的动态评估协议将更好地捕捉这一快速发展领域的进展。开发这些评估工具需要 AI 研究人员、认知科学家和领域专家之间的跨学科合作，以确保它们准确反映与人类类似推理最相关的认知维度。

**科学发现** 认知工程为 AI 系统作为科学发现伙伴开辟了前所未有的可能性。测试时间扩展方法展示了通过延长推理时间连接不同知识领域并生成创新见解的潜力。未来的工作应探索如何优化这些认知能力，特别是针对科学发现任务，如假设生成、实验设计和理论形成。这将需要开发专门的提

## 11. 结论

---

示技术或训练方法，鼓励在科学严谨的前提下进行创造性探索。此外，研究应探讨如何将特定领域的科学工具和实验平台与推理模型集成，使它们不仅能生成假设，还能设计并可能执行实验以验证这些假设 (Lu et al., 2024a)。

## 11 结论

认知工程代表了人工智能发展的一次范式转变，从根本上将我们的方法从知识积累转变为系统化开发思维能力。生成式人工智能的第二阶段利用测试时扩展方法以及专门的训练策略，使模型能够进行深度思考、复杂推理和创造性问题解决。从数学到多模态理解等多个领域的实践表明，这些能力已经在模型性能上带来了显著提升。认知工程的出现不仅标志着技术的进步，也开启了人类思维与人工智能之间新型关系的序幕——一种基于深度理解和认知交换的共生关系，在这种关系中，人类与人工智能能够相互赋能，共同探索认知的新前沿。

## 致谢

我们要感谢柳奕昕对本工作提出的建设性意见。同时，我们也感谢 Ethan Chern 在项目早期阶段的参与和支持。

## References

- [1] Russell L Ackoff. 1989. From data to wisdom. *Journal of applied systems analysis*, 16(1):3–9.
- [2] Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 2023. Let’s sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12375–12396, Singapore. Association for Computational Linguistics.
- [3] Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning.
- [4] Wasi Uddin Ahmad, Sean Narendhiran, Somshubra Majumdar, Aleksander Ficek, Siddhartha Jain, Jocelyn Huang, Vahid Noroozi, and Boris Ginsburg. 2025. Opencodereasoning: Advancing data distillation for competitive coding. *ArXiv preprint*, abs/2504.01943.
- [5] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *ArXiv preprint*, abs/2204.01691.
- [6] AlphaProof and AlphaGeometry teams. 2024. AI achieves silver-medal standard solving international mathematical olympiad problems. <https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/>.
- [7] Anthropic. 2024a. The claude 3 model family: Opus, sonnet, haiku. Technical Report.
- [8] Anthropic. 2024b. Introducing computer use, a new claude 3.5 sonnet, and claude 3.5 haiku. anthropic.com.
- [9] Anthropic. 2025. Introducing deep research. anthropic.com.
- [10] team swe arena. 2025. Swe arena: An open evaluation platform for automated software engineering.
- [11] Daman Arora and Andrea Zanette. 2025. Training language models to reason efficiently.
- [12] Simon A. Aytes, Jinheon Baek, and Sung Ju Hwang. 2025. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching.
- [13] Alisson Azzolini, Hannah Brandon, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, Francesco Ferroni, Rama Govindaraju, et al. 2025. Cosmos-reason1: From physical common sense to embodied reasoning. *ArXiv preprint*, abs/2503.15558.
- [14] Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. 2025. Online difficulty filtering for reasoning oriented reinforcement learning.
- [15] Marthe Ballon, Andres Algaba, and Vincent Ginis. 2025. The relationship between reasoning and performance in large language models – o3 (mini) thinks harder, not longer.
- [16] Edward Beeching, Lewis Tunstall, and Sasha Rush. 2024. Scaling test-time compute with open models.
- [17] Yoshua Bengio. 2023. Faq on catastrophic ai risks.
- [18] Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online. Association for Computational Linguistics.
- [19] Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. 2024. Forest-of-thought: Scaling test-time compute for enhancing llm reasoning.
- [20] Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling.
- [21] Jiajun Chai, Sicheng Li, Yuqian Fu, Dongbin Zhao, and Yuanheng Zhu. 2024. Empowering llm agents with zero-shot optimal decision-making through q-learning. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*.

- [22] Team Chameleon. 2024. **Chameleon**: Mixed-modal early-fusion foundation models. *ArXiv preprint*, abs/2405.09818.
- [23] Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. **Chateval**: Towards better llm-based evaluators through multi-agent debate. *ArXiv preprint*, abs/2308.07201.
- [24] Bei Chen, Fengji Zhang, Anh Nguyen, Daoguang Zan, Zeqi Lin, Jian-Guang Lou, and Weizhu Chen. 2023a. **Codet**: Code generation with generated tests. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- [25] Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024a. Alphamath almost zero: Process supervision without process.
- [26] Hardy Chen, Haoqin Tu, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. 2025a. Vl-thinking: An r1-derived visual instruction tuning dataset for thinkable llms. <https://github.com/UCSC-VLAA/VL-Thinking>.
- [27] Jiefeng Chen, Jie Ren, Xinyun Chen, Chengrun Yang, Ruoxi Sun, and Sercan Ö Arik. 2025b. Sets: Leveraging self-verification and self-correction for improved test-time scaling.
- [28] Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. 2024b. **Huatuogpt-o1**, towards medical complex reasoning with llms.
- [29] Liang Chen, Lei Li, Haozhe Zhao, Yifan Song, and Vinci. 2025c. R1-v: Reinforcing super generalization ability in vision-language models with less than \$3. <https://github.com/Deep-Agent/R1-V>.
- [30] Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. 2024c. Are more llm calls all you need? towards scaling laws of compound inference systems.
- [31] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *ArXiv preprint*, abs/2107.03374.
- [32] Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025d. **Research**: Learning to reason with search for llms via reinforcement learning.
- [33] Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025e. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models.
- [34] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. 2025f. **Janus-pro**: Unified multimodal understanding and generation with data and model scaling. *ArXiv preprint*, abs/2501.17811.
- [35] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuseng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2024d. Do not think that much for 2+3=? on the overthinking of o1-like llms.
- [36] Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023b. Universal self-consistency for large language model generation.
- [37] Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023c. Teaching large language models to self-debug.
- [38] Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, Lei Fang, Zhongyuan Wang, and Ji-Rong Wen. 2025g. An empirical study on eliciting and improving r1-like reasoning models.
- [39] Jeffrey Cheng and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations.
- [40] Jiale Cheng, Xiao Liu, Cunxiang Wang, Xiaotao Gu, Yida Lu, Dan Zhang, Yuxiao Dong, Jie Tang, Hongning Wang, and Minlie Huang. 2024a. **Spar**: Self-play with tree-search refinement to improve instruction-following in large language models.

- [41] Kanzhi Cheng, Yantao Li, Fangzhi Xu, Jianbing Zhang, Hao Zhou, and Yang Liu. 2024b. Vision-language models can self-improve reasoning via reflection. *ArXiv preprint*, abs/2411.00855.
- [42] Xiaoxue Cheng, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2025. Think more, hallucinate less: Mitigating hallucinations via dual process of fast and slow thinking. *ArXiv preprint*, abs/2501.01306.
- [43] I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. Factool: Factuality detection in generative ai – a tool augmented framework for multi-task and multi-domain scenarios.
- [44] Steffi Chern, Ethan Chern, Graham Neubig, and Pengfei Liu. 2024a. Can large language models be trusted for evaluation? scalable meta-evaluation of llms as evaluators via agent debate. *ArXiv preprint*, abs/2401.16788.
- [45] Steffi Chern, Zhen Fan, and Andy Liu. 2024b. Combating adversarial attacks with multi-agent debate. *ArXiv preprint*, abs/2401.05998.
- [46] Wayne Chi, Valerie Chen, Anastasios Nikolas Angelopoulos, Wei-Lin Chiang, Aditya Mittal, Naman Jain, Tianjun Zhang, Ion Stoica, Chris Donahue, and Ameet Talwalkar. 2025. Copilot arena: A platform for code llm evaluation in the wild. *ArXiv preprint*, abs/2502.09328.
- [47] Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.
- [48] Yinlam Chow, Guy Tennenholz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Sridhar Thiagarajan, Craig Boutilier, Rishabh Agarwal, Aviral Kumar, and Aleksandra Faust. 2024. Inference-aware fine-tuning for best-of-n sampling in large language models.
- [49] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025a. Sft memorizes, rl generalizes: A comparative study of foundation model post-training.
- [50] Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. 2025b. Gpg: A simple and strong reinforcement learning baseline for model reasoning.
- [51] Jaden Clark, Suvir Mirchandani, Dorsa Sadigh, and Suneel Belkhale. 2025. Action-free reasoning for policy generalization. *ArXiv preprint*, abs/2502.03729.
- [52] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.
- [53] Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, Nicholas Thumiger, Aditya Desai, Ion Stoica, Ana Klimovic, Graham Neubig, and Joseph E. Gonzalez. 2025. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks.
- [54] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. 2025a. Process reinforcement through implicit rewards.
- [55] Yingqian Cui, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, Suhang Wang, Yue Xing, Jiliang Tang, and Qi He. 2025b. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models.
- [56] Quy-Anh Dang and Chris Ngo. 2025. Reinforcement learning for reasoning in small llms: What works and what doesn't.
- [57] Tri Dao and Albert Gu. 2024. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality.
- [58] Leonardo De Moura, Soonho Kong, Jeremy Avigad, Floris Van Doorn, and Jakob von Raumer. 2015. The lean theorem prover (system description). In *Automated Deduction-CADE-25: 25th International Conference on Automated Deduction, Berlin, Germany, August 1-7, 2015, Proceedings* 25, pages 378–388. Springer.

- [59] DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model.
- [60] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhua Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyu Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.
- [61] Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024. From explicit cot to implicit cot: Learning to internalize cot step by step.
- [62] Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. 2023. Implicit chain of thought reasoning via knowledge distillation.
- [63] Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models.
- [64] Mengru Ding, Hanmeng Liu, Zhizhang Fu, Jian Song, Wenbo Xie, and Yue Zhang. 2024. Break the chain: Large language models can be shortcut reasoners.
- [65] Yifu Ding, Wentao Jiang, Shunyu Liu, Yongcheng Jing, Jinyang Guo, Yingjie Wang, Jing Zhang, Zengmao Wang, Ziwei Liu, Bo Du, Xianglong Liu, and Dacheng Tao. 2025. Dynamic parallel tree search for efficient llm reasoning.
- [66] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment.
- [67] Yuhao Dong, Zuyan Liu, Hai-Long Sun, Jingkang Yang, Winston Hu, Yongming Rao, and Ziwei Liu. 2024. Insight-v: Exploring long-chain visual reasoning with multimodal large language models. *ArXiv preprint*, abs/2411.14432.
- [68] Yifan Du, Zikang Liu, Yifan Li, Wayne Xin Zhao, Yuqi Huo, Bingning Wang, Weipeng Chen, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. 2025. Virgo: A preliminary exploration on reproducing o1-like mllm. *ArXiv preprint*, abs/2501.01904.
- [69] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate.
- [70] Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, et al. 2025. Competitive programming with large reasoning models. *ArXiv preprint*, abs/2502.06807.

- [71] EvolvingLMMs-Lab. 2025. open-r1-multimodal: A fork to add multimodal model training to open-r1. <https://github.com/EvolvingLMMs-Lab/open-r1-multimodal>.
- [72] Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.
- [73] Rongyao Fang, Chengqi Duan, Kun Wang, Linjiang Huang, Hao Li, Shilin Yan, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, et al. 2025. Got: Unleashing reasoning capability of multimodal large language model for visual generation and editing. *ArXiv preprint*, abs/2503.10639.
- [74] Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. 2025. Concise reasoning via reinforcement learning.
- [75] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Benyou Wang, and Xiangyu Yue. 2025a. Video-r1: Reinforcing video reasoning in mllms. *ArXiv preprint*, abs/2503.21776.
- [76] Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Jingyi Song, and Hao Wang. 2025b. Airrag: Activating intrinsic reasoning for retrieval augmented generation using tree-based search.
- [77] Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2023. Alphazero-like tree-search can guide large language model decoding and training.
- [78] Emily First, Markus N. Rabe, Talia Ringer, and Yuriy Brun. 2023. Baldur: Whole-proof generation and repair with large language models.
- [79] Hao Fu, Yao; Peng and Tushar Khot. 2022. How does gpt obtain its ability? tracing emergent abilities of language models to their sources. *Yao Fu's Notion*.
- [80] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024a. GPTScore: Evaluate as you desire. 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 6556–6576, Mexico City, Mexico. Association for Computational Linguistics.
- [81] Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Aurick Qiao, and Hao Zhang. 2024b. Efficiently serving llm reasoning programs with certainindex.
- [82] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars.
- [83] Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D. Goodman. 2024. Stream of search (sos): Learning to search in language.
- [84] Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. 2024a. On designing effective rl reward at training time for llm reasoning.
- [85] Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR.
- [86] Zitian Gao, Boye Niu, Xuzheng He, Haotian Xu, Hongzhang Liu, Aiwei Liu, Xuming Hu, and Lijie Wen. 2024b. Interpretable contrastive monte carlo tree search reasoning.
- [87] Suyu Ge, Chunting Zhou, Rui Hou, Madian Khabsa, Yi-Chia Wang, Qifan Wang, Jiawei Han, and Yuning Mao. 2024. MART: Improving LLM safety with multi-round automatic red-teaming. 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 1927–1937, Mexico City, Mexico. Association for Computational Linguistics.
- [88] Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. 2025. Scaling up test-time compute with latent reasoning: A recurrent depth approach.
- [89] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing.
- [90] Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces.
- [91] Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I Wang. 2024a. Cruxeval: A benchmark for code reasoning, understanding and execution. *ArXiv preprint*, abs/2401.03065.

- [92] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2024b. *A survey on llm-as-a-judge*.
- [93] Yu Gu, Xiang Deng, and Yu Su. 2023. *Don't generate, discriminate: A proposal for grounding language models to real-world environments*. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4928–4949, Toronto, Canada. Association for Computational Linguistics.
- [94] Melody Y. Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, Hyung Won Chung, Sam Toyer, Johannes Heidecke, Alex Beutel, and Amelia Glaese. 2024. *Deliberative alignment: Reasoning enables safer language models*. *ArXiv preprint*, abs/2412.16339.
- [95] Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. 2025a. *Deeprag: Thinking to retrieval step by step for large language models*.
- [96] Xinyu Guan, Li Lyra Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025b. *rstar-math: Small llms can master math reasoning with self-evolved deep thinking*.
- [97] Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. *Reinforced self-training (rest) for language modeling*. *ArXiv preprint*, abs/2308.08998.
- [98] Jarvis Guo, Tuney Zheng, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Yizhi Li, Graham Neubig, Wenhua Chen, and Xiang Yue. 2024. *Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale*. *ArXiv preprint*, abs/2412.05237.
- [99] Yanjiang Guo, Jianke Zhang, Xiaoyu Chen, Xiang Ji, Yen-Jen Wang, Yucheng Hu, and Jianyu Chen. 2025a. *Improving vision-language-action model with online reinforcement learning*. *ArXiv preprint*, abs/2501.16664.
- [100] Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Peng Gao, Hongsheng Li, and Pheng-Ann Heng. 2025b. *Can we generate images with cot? let's verify and reinforce image generation step by step*. *ArXiv preprint*, abs/2501.13926.
- [101] Patrick Halupczok, Matthew Bowers, and Adam Tauman Kalai. 2023. *Language models can teach themselves to program better*. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- [102] Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2024. *Token-budget-aware llm reasoning*.
- [103] Shibo Hao, Yi Gu, Haotian Luo, Tianyang Liu, Xiyan Shao, Xinyuan Wang, Shuhua Xie, Haodi Ma, Adithya Samavedhi, Qiyue Gao, Zhen Wang, and Zhiting Hu. 2024a. *Llm reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models*.
- [104] Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. *Reasoning with language model is planning with world model*. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173, Singapore. Association for Computational Linguistics.
- [105] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024b. *Training large language models to reason in a continuous latent space*.
- [106] Peter E Hart, Nils J Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107.
- [107] Alex Havrilla, Sharath Raparthy, Christoforus Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Raileanu. 2024. *Glore: When, where, and how to improve llm reasoning via global and local refinements*.
- [108] Yanheng He, Jiahe Jin, Shijie Xia, Jiadi Su, Runze Fan, Haoyang Zou, Xiangkun Hu, and Pengfei Liu. 2024. *Pc agent: While you sleep, ai works – a cognitive journey into digital world*.
- [109] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. *Measuring mathematical problem solving with the math dataset*.

- [110] Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. 2023. An overview of catastrophic AI risks. *ArXiv preprint*, abs/2306.12001.
- [111] Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. 2025. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility.
- [112] Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Monishwaran Maheswaran, June Paik, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. 2024. Squeezed attention: Accelerating long context length llm inference.
- [113] Coleman Hooper, Sehoon Kim, Suhong Moon, Kerem Dilmen, Monishwaran Maheswaran, Nicholas Lee, Michael W. Mahoney, Sophia Shao, Kurt Keutzer, and Amir Gholami. 2025. Ets: Efficient tree search for inference-time scaling.
- [114] Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-star: Training verifiers for self-taught reasoners.
- [115] Zhenyu Hou, Pengfan Du, Yilin Niu, Zhengxiao Du, Aohan Zeng, Xiao Liu, Minlie Huang, Hongning Wang, Jie Tang, and Yuxiao Dong. 2024. Does rlhf scale? exploring the impacts from data, model, and method.
- [116] Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. 2025. Advancing language model reasoning through reinforcement learning and inference scaling.
- [117] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- [118] Jian Hu. 2025. Reinforce++: A simple and efficient approach for aligning large language models.
- [119] Jian Hu, Xibin Wu, Zilin Zhu, Weixun Wang, Dehao Zhang, Yu Cao, et al. 2024a. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *ArXiv preprint*, abs/2405.11143.
- [120] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, and Heung-Yeung Shum Xiangyu Zhang. 2025. Open-reasoner-zero: An open source approach to scaling reinforcement learning on the base model. <https://github.com/Open-Reasoner-Zero/Open-Reasoner-Zero>.
- [121] Xiangkun Hu, Dongyu Ru, Lin Qiu, Qipeng Guo, Tianhang Zhang, Yang Xu, Yun Luo, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024b. Knowledge-centric hallucination detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6953–6975, Miami, Florida, USA. Association for Computational Linguistics.
- [122] Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. 2025a. Efficient test-time scaling via self-calibration.
- [123] Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023a. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.
- [124] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023b. Large language models cannot self-correct reasoning yet.
- [125] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025b. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.
- [126] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. 2022. Inner monologue: Embodied reasoning through planning with language models.
- [127] Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Yao Hu, and Shaohui Lin. 2025c. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *ArXiv preprint*, abs/2503.06749.

- [128] Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. 2024. [O1 replication journey –part 2: Surpassing o1-preview through simple distillation](#). *Github*.
- [129] Zhongzhen Huang, Gui Geng, Shengyi Hua, Zhen Huang, Haoyang Zou, Shaoting Zhang, Pengfei Liu, and Xiaofan Zhang. 2025d. [O1 replication journey – part 3: Inference-time scaling for medical reasoning](#).
- [130] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. 2024. [Qwen2. 5-coder technical report](#). *ArXiv preprint*, abs/2409.12186.
- [131] Hyeonbin Hwang, Doyoung Kim, Seungone Kim, Seonghyeon Ye, and Minjoon Seo. 2024. [Self-explore: Enhancing mathematical reasoning in language models with fine-grained rewards](#).
- [132] Arnav Kumar Jain, Gonzalo Gonzalez-Pumariega, Wayne Chen, Alexander M Rush, Wenting Zhao, and Sanjiban Choudhury. 2025. [Multi-turn code generation through single-step rewards](#).
- [133] Xiaotong Ji, Shyam Sundhar Ramesh, Matthieu Zimmer, Ilija Bogunovic, Jun Wang, and Haitham Bou-Ammar. 2025. [Almost surely safe alignment of large language models at inference-time](#). *ArXiv preprint*, abs/2502.01208.
- [134] Dongwei Jiang, Guoxuan Wang, Yining Lu, Andrew Wang, Jingyu Zhang, Chuyu Liu, Benjamin Van Durme, and Daniel Khashabi. 2024. [Rationalyst: Pre-training process-supervision for improving reasoning](#).
- [135] Fengqing Jiang, Zhangchen Xu, Yuetai Li, Luyao Niu, Zhen Xiang, Bo Li, Bill Yuchen Lin, and Radha Poovendran. 2025. [Safechain: Safety of language models with long chain-of-thought reasoning capabilities](#).
- [136] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. 2024. [SWE-bench: Can language models resolve real-world github issues?](#) In *The Twelfth International Conference on Learning Representations*.
- [137] Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. [Search-r1: Training llms to reason and leverage search engines with reinforcement learning](#).
- [138] Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. [The impact of reasoning step length on large language models](#).
- [139] Scott M. Jordan, Adam White, Bruno Castro da Silva, Martha White, and Philip S. Thomas. 2024. [Position: Benchmarking is limited in reinforcement learning research](#).
- [140] Jordan Juravsky, Bradley Brown, Ryan Ehrlich, Daniel Y. Fu, Christopher Ré, and Azalia Mirhosseini. 2024. [Hydragen: High-throughput llm inference with shared prefixes](#).
- [141] Subbarao Kambhampati. 2024. [Can large language models reason and plan?](#) *Annals of the New York Academy of Sciences*, 1534(1):15–18.
- [142] Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. [When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs](#). *Transactions of the Association for Computational Linguistics*, 12:1417–1440.
- [143] Jikun Kang, Xin Zhe Li, Xi Chen, Amirreza Kazemi, Qianyi Sun, Boxing Chen, Dong Li, Xu He, Quan He, Feng Wen, Jianye Hao, and Jun Yao. 2024a. [Mindstar: Enhancing math reasoning in pre-trained llms at inference time](#).
- [144] Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2024b. [C3ot: Generating shorter chain-of-thought without compromising effectiveness](#).
- [145] Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. [Scalable best-of-n selection for large language models via self-certainty](#).
- [146] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling laws for neural language models](#).

- [147] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. *Transformers are rnns: Fast autoregressive transformers with linear attention*. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5156–5165. PMLR.
- [148] Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. 2024. *Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment*.
- [149] Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. 2024. *Debating with more persuasive llms leads to more truthful answers*.
- [150] Maxim Khanov, Jirayu Burapachheep, and Yixuan Li. 2024. *ARGS: alignment as reward-guided search*. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- [151] Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2023. *Language models can solve computer tasks*. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- [152] Seungone Kim, Ian Wu, Jinu Lee, Xiang Yue, Seongyun Lee, Mingyeong Moon, Kiril Gashteovski, Carolin Lawrence, Julia Hockenmaier, Graham Neubig, and Sean Welleck. 2025. *Scaling evaluation-time compute with reasoning models as process evaluators*.
- [153] Team Kimi, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. 2025. *Kimi k1. 5: Scaling reinforcement learning with llms*. *ArXiv preprint*, abs/2501.12599.
- [154] Kimi Team. 2025. *Kimi-vl technical report*. <https://github.com/MoonshotAI/Kimi-VL/blob/main/Kimi-VL.pdf>.
- [155] Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. 2024. *Tree search for language model agents*.
- [156] Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Srivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2024. *Training language models to self-correct via reinforcement learning*.
- [157] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- [158] Guillaume Lample, Timothée Lacroix, Marie-Anne Lachaux, Aurélien Rodriguez, Amaury Hayat, Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. 2022. *Hypertree proof search for neural theorem proving*. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- [159] Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu-Hong Hoi. 2022. *Coderl: Mastering code generation through pretrained models and deep reinforcement learning*. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- [160] Ayeong Lee, Ethan Che, and Tianyi Peng. 2025. *How well do llms compress their own chain-of-thought? a token complexity approach*.
- [161] Jung Hyun Lee, June Yong Yang, Byeongho Heo, Dongyoon Han, and Kang Min Yoo. 2024. *Token-supervised value models for enhancing mathematical reasoning capabilities of large language models*.
- [162] Lucas Lehnert, Sainbayar Sukhbaatar, DiJia Su, Qinqing Zheng, Paul Mcvay, Michael Rabbat, and Yuandong Tian. 2024. *Beyond a\*: Better planning with transformers via search dynamics bootstrapping*.

- [163] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. 2024a. *Llava-onevision: Easy visual task transfer*. *ArXiv preprint*, abs/2408.03326.
- [164] Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. 2025a. *Imagine while reasoning in space: Multimodal visualization-of-thought*. *ArXiv preprint*, abs/2501.07542.
- [165] Dacheng Li, Shiyi Cao, Chengkun Cao, Xiuyu Li, Shangyin Tan, Kurt Keutzer, Jiarong Xing, Joseph E Gonzalez, and Ion Stoica. 2025b. *S\*: Test time scaling for code generation*. *ArXiv preprint*, abs/2502.14382.
- [166] Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Shishir G. Patil, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025c. *Llms can easily learn to reason from demonstrations structure, not content, is what matters!*
- [167] Jiazheng Li, Yuxiang Zhou, Junru Lu, Gladys Tyen, Lin Gui, Cesare Aloisi, and Yulan He. 2025d. *Two heads are better than one: Dual-model verbal reflection at inference-time*.
- [168] Jinming Li, Yichen Zhu, Zhibin Tang, Junjie Wen, Minjie Zhu, Xiaoyu Liu, Chengmeng Li, Ran Cheng, Yaxin Peng, and Feifei Feng. 2024b. *Improving vision-language-action models via chain-of-affordance*. *ArXiv preprint*, abs/2412.20451.
- [169] Loka Li, Zhenhao Chen, Guangyi Chen, Yixuan Zhang, Yusheng Su, Eric Xing, and Kun Zhang. 2024c. *Confidence matters: Revisiting intrinsic self-correction capabilities of large language models*.
- [170] Qingyao Li, Wei Xia, Kounianhua Du, Xinyi Dai, Ruiming Tang, Yasheng Wang, Yong Yu, and Weinan Zhang. 2024d. *Rethinkmcts: Refining erroneous thoughts in monte carlo tree search for code generation*.
- [171] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025e. *Search-o1: Agentic search-enhanced large reasoning models*.
- [172] Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025f. *Limr: Less is more for rl scaling*.
- [173] Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025g. *Torl: Scaling tool-integrated rl*.
- [174] Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2024e. *Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning*.
- [175] Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Rama-subramanian, and Radha Poovendran. 2025h. *Small models struggle to learn from strong reasoners*.
- [176] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d’Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. 2022. *Competition-level code generation with alphacode*. *Science*, 378(6624):1092–1097.
- [177] Ziniu Li. 2025. Can better cold-start strategies improve rl training for llms? <https://tangible-polo-203.notion.site/Can-Better-Cold-Start-Strategies-Improve-RL-Training-for-LLMs-17aa0742a51680828616c867ed53bc6b>. Notion Blog.
- [178] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2023. *Encouraging divergent thinking in large language models through multi-agent debate*.
- [179] Zhenyi Liao, Qingsong Xie, Yanhao Zhang, Zijian Kong, Haonan Lu, Zhenyu Yang, and Zhijie Deng. 2025. *Improved visual-spatial reasoning via rl-zero-like training*. *ArXiv preprint*, abs/2504.00883.
- [180] Shalev Lifshitz, Sheila A. McIlraith, and Yilun Du. 2025. *Multi-agent verification: Scaling test-time compute with multiple verifiers*.
- [181] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. *Let’s verify step by step*.
- [182] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. *Generating with confidence: Uncertainty quantification for black-box large language models*. *Trans. Mach. Learn. Res.*, 2024.

- [183] Fangfu Liu, Hanyang Wang, Yimo Cai, Kaiyan Zhang, Xiaohang Zhan, and Yueqi Duan. 2025a. **Video-t1: Test-time scaling for video generation.** *ArXiv preprint*, abs/2503.18942.
- [184] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.
- [185] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. **Visual instruction tuning.** In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- [186] Jiacheng Liu, Andrew Cohen, Ramakanth Pasunuru, Yejin Choi, Hannaneh Hajishirzi, and Asli Celikyilmaz. 2023b. **Don’t throw away your value model! generating more preferable text with value-guided monte-carlo tree search decoding.**
- [187] Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, Yuanxing Zhang, Zhuo Chen, Hangyu Guo, Shilong Li, Ziqiang Liu, Yong Shan, Yifan Song, Jiayi Tian, Wenhai Wu, Zhejian Zhou, Ruijie Zhu, Junlan Feng, Yang Gao, Shizhu He, Zhoujun Li, Tianyu Liu, Fanyu Meng, Wenbo Su, Yingshui Tan, Zili Wang, Jian Yang, Wei Ye, Bo Zheng, Wangchunshu Zhou, Wenhao Huang, Sujian Li, and Zhaoxiang Zhang. 2025b. **A comprehensive survey on long context language modeling.**
- [188] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. **Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.**
- [189] Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. 2025c. **Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling.**
- [190] Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2024b. Reducing hallucinations in vision-language models via latent space steering. *ArXiv preprint*, abs/2410.15778.
- [191] Siyao Liu, He Zhu, Jerry Liu, Shulin Xin, Aoyan Li, Rui Long, Li Chen, Jack Yang, Jinxiang Xia, ZY Peng, et al. 2024c. **Fullstack bench: Evaluating llms as full stack coder.** *ArXiv preprint*, abs/2412.00535.
- [192] Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2024d. **Can language models learn to skip steps?**
- [193] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023c. **G-eval: NLG evaluation using gpt-4 with better human alignment.** In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.
- [194] Yixin Liu, Avi Singh, C. Daniel Freeman, John D. Co-Reyes, and Peter J. Liu. 2023d. **Improving large language model fine-tuning for solving math problems.**
- [195] Yuecheng Liu, Dafeng Chi, Shiguang Wu, Zhanguang Zhang, Yaochen Hu, Lingfeng Zhang, Yingxue Zhang, Shuang Wu, Tongtong Cao, Guowei Huang, et al. 2025d. **Spatialcot: Advancing spatial reasoning through coordinate alignment and chain-of-thought for embodied task planning.** *ArXiv preprint*, abs/2501.10074.
- [196] Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. 2025e. **Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement.** *ArXiv preprint*, abs/2503.06520.
- [197] Zeyi Liu, Arpit Bahety, and Shuran Song. 2023e. Reflect: Summarizing robot experiences for failure explanation and correction. In *Conference on Robot Learning*, pages 3468–3484. PMLR.
- [198] Zhili Liu, Yunhao Gou, Kai Chen, Lanqing Hong, Jiahui Gao, Fei Mi, Yu Zhang, Zhenguo Li, Xin Jiang, Qun Liu, et al. 2024e. **Mixture of insightful experts (mote): The synergy of thought chains and expert mixtures in self-alignment.** *ArXiv preprint*, abs/2405.00557.
- [199] Zichen Liu, Changyu Chen, Wenjun Li, Tianyu Pang, Chao Du, and Min Lin. 2025f. There may not be aha moment in r1-zero-like training —a pilot study. <https://oatllm.notion.site/oat-zero>. Notion Blog.

- [200] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025g. Understanding r1-zero-like training: A critical perspective. <https://github.com/sail-sg/understand-r1-zero>.
- [201] Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025h. Inference-time scaling for generalist reward modeling.
- [202] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. 2025i. **Visual-rft: Visual reinforcement fine-tuning**. *ArXiv preprint*, abs/2503.01785.
- [203] Do Xuan Long, Duong Ngoc Yen, Anh Tuan Luu, Kenji Kawaguchi, Min-Yen Kan, and Nancy F. Chen. 2024. **Multi-expert prompting improves reliability, safety and usefulness of large language models**. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 20370–20401. Association for Computational Linguistics.
- [204] Jieyi Long. 2023. Large language model guided tree-of-thought.
- [205] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024a. **The AI Scientist: Towards fully automated open-ended scientific discovery**. *ArXiv preprint*, abs/2408.06292.
- [206] Dakuan Lu, Xiaoyu Tan, Rui Xu, Tianchu Yao, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi. 2025. **Scp-116k: A high-quality problem-solution dataset and a generalized pipeline for automated extraction in the higher education science domain**.
- [207] Jianqiao Lu, Zhiyang Dou, Hongru Wang, Zeyu Cao, Jianbo Dai, Yingjia Wan, and Zhijiang Guo. 2024b. **Autopsv: Automated process-supervised verifier**.
- [208] Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025a. **O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning**.
- [209] Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. 2024. **Improve mathematical reasoning in language models by automated process supervision**.
- [210] Michael Luo, Sijun Tan, Roy Huang, Xiaoxiang Shi, Rachel Xin, Colin Cai, Ameen Patel, Alpay Ariyak, Qingyang Wu, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025b. Deepcoder: A fully open-source 14b coder at o3-mini level. <https://pretty-radio-b75.notion.site/DeepCoder-A-Fully-Open-Source-14B-Coder-at-O3-mini-Level-1cf81902c14680b3bee5eb349a512a51>. Notion Blog.
- [211] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025c. 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.
- [212] Ruilin Luo, Zhuofan Zheng, Yifan Wang, Yiyo Yu, Xinzhe Ni, Zicheng Lin, Jin Zeng, and Yujiu Yang. 2025d. **Ursa: Understanding and verifying chain-of-thought reasoning in multimodal mathematics**. *ArXiv preprint*, abs/2501.04686.
- [213] Ruotian Ma, Peisong Wang, Cheng Liu, Xingyan Liu, Jiaqi Chen, Bang Zhang, Xin Zhou, Nan Du, and Jia Li. 2025a. **S<sup>2</sup>r: Teaching llms to self-verify and self-correct via reinforcement learning**.
- [214] Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025b. **Cot-valve: Length-compressible chain-of-thought tuning**.
- [215] Yan Ma, Steffi Chern, Xuyang Shen, Yiran Zhong, and Pengfei Liu. 2025c. **Rethinking rl scaling for vision language models: A transparent, from-scratch framework and comprehensive evaluation scheme**. *ArXiv preprint*, abs/2504.02587.
- [216] Yingwei Ma, Yongbin Li, Yihong Dong, Xue Jiang, Rongyu Cao, Jue Chen, Fei Huang, and Binhu Li. 2025d. **Thinking longer, not larger: Enhancing software engineering agents via scaling test-time compute**.
- [217] Yingwei Ma, Yue Liu, Yue Yu, Yuanliang Zhang, Yu Jiang, Changjian Wang, and Shanshan Li. 2023. **At which training stage does code data help llms reasoning?** *ArXiv preprint*, abs/2309.16298.

- [218] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. *Self-refine: Iterative refinement with self-feedback*. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- [219] Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. *SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models*. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.
- [220] Fanqing Meng, Lingxiao Du, and Xiangyan Liu. 2025a. R1-multimodal-journey: A journey to real multimodal r1. <https://github.com/FanqingM/R1-Multimodal-Journey>.
- [221] Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi, Wenhui Wang, Junjun He, Kaipeng Zhang, et al. 2025b. *Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning*. *ArXiv preprint*, abs/2503.07365.
- [222] William Merrill and Ashish Sabharwal. 2023. The expressive power of transformers with chain of thought.
- [223] Meta. 2023. *Llama 2: Open foundation and fine-tuned chat models*.
- [224] Meta. 2024. *The llama 3 herd of models*.
- [225] Janet Metcalfe and Arthur P Shimamura. 1994. *Metacognition: Knowing about knowing*. MIT press.
- [226] Zawalski Michał, Chen William, Pertsch Karl, Mees Oier, Finn Chelsea, and Levine Sergey. 2024. *Robotic control via embodied chain-of-thought reasoning*. *ArXiv preprint*, abs/2407.08693.
- [227] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. *FActScore: Fine-grained atomic evaluation of factual precision in long form text generation*. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- [228] Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. 2024. *Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems*.
- [229] Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. *Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models*.
- [230] ModelScope. 2024. *MMMU-Reasoning-Distill-Validation*. <https://modelscope.cn/datasets/modelscope/MMMU-Reasoning-Distill-Validation>. ModelScope.
- [231] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. *s1: Simple test-time scaling*.
- [232] Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. 2025. *Self-training elicits concise reasoning in large language models*.
- [233] Sania Nayab, Giulio Rossolini, Marco Simoni, Andrea Saracino, Giorgio Buttazzo, Nicolamaria Manes, and Fabrizio Giacomelli. 2024. *Concise thoughts: Impact of output length on llm reasoning and cost*.
- [234] Allen Newell, John C Shaw, and Herbert A Simon. 1959. Report on a general problem solving program. In *IFIP congress*, page 64. Pittsburgh, PA.
- [235] Allen Newell, Herbert Alexander Simon, et al. 1972. *Human problem solving*. Prentice-hall Englewood Cliffs, NJ.
- [236] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. *Large language diffusion models*.

- [237] Tobias Nipkow, Markus Wenzel, and Lawrence C Paulson. 2002. *Isabelle/HOL: a proof assistant for higher-order logic*. Springer.
- [238] Franz Nowak, Anej Svetec, Alexandra Butoi, and Ryan Cotterell. 2024. On the representational capacity of neural language models with chain-of-thought reasoning.
- [239] Rafael Núñez, Michael Allen, Richard Gao, Carson Miller Rigoli, Josephine Relaford-Doyle, and Arturs Semenuks. 2019. What happened to cognitive science? *Nature human behaviour*, 3(8):782–791.
- [240] Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models.
- [241] Theo X. Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. 2023. Is self-repair a silver bullet for code generation?
- [242] OpenAI. 2023. Gpt-4 technical report.
- [243] OpenAI. 2024. Openai o1 system card. Accessed: 2024-11-07.
- [244] OpenAI. 2025a. Computer-using agent. openai.com.
- [245] OpenAI. 2025b. Introducing deep research. openai.com.
- [246] OpenAI. 2025a. Introducing openai o3 and o4-mini. Accessed: 2025-04-17.
- [247] OpenAI. 2025b. Thinking with images. Accessed: 2025-04-17.
- [248] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- [249] Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. 2024a. Training software engineering agents and verifiers with swe-gym. *ArXiv preprint*, abs/2412.21139.
- [250] Jiayi Pan, Yichi Zhang, Nicholas Tomlin, Yifei Zhou, Sergey Levine, and Alane Suhr. 2024b. Autonomous evaluation and refinement of digital agents.
- [251] Jianhui Pang, Fanghua Ye, Derek Fai Wong, Xin He, Wanshun Chen, and Longyue Wang. 2024a. Anchor-based large language models.
- [252] Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024b. Iterative reasoning preference optimization.
- [253] Peter S. Park, Simon Goldstein, Aidan O’Gara, Michael Chen, and Dan Hendrycks. 2024. AI deception: A survey of examples, risks, and potential solutions. *Patterns*, 5(6):100988.
- [254] Dinesh Parthasarathy, Georgios D. Kontes, Axel Plinge, and Christopher Mutschler. 2023. C-MCTS: safe planning with monte carlo tree search. *ArXiv preprint*, abs/2305.16209.
- [255] Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2024. REFINER: Reasoning feedback on intermediate representations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1100–1126, St. Julian’s, Malta. Association for Computational Linguistics.
- [256] Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. 2025. Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. *ArXiv preprint*, abs/2503.07536.
- [257] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, and Josephina Hu et al. 2025. Humanity’s last exam.

- [258] Stanislas Polu and Ilya Sutskever. 2020. Generative language modeling for automated theorem proving.
- [259] Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. 2024. Agent q: Advanced reasoning and learning for autonomous ai agents.
- [260] Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyra Zhang, Fan Yang, and Mao Yang. 2024. Mutual reasoning makes smaller llms stronger problem-solvers.
- [261] Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Zhengzhong Liu, Yuanzhi Li, and Pengfei Liu. 2024. O1 replication journey: A strategic progress report –part 1. *ArXiv preprint*, abs/2410.18982.
- [262] Yiwei Qin, Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2023. T5Score: Discriminative finetuning of generative evaluation metrics. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15185–15202, Singapore. Association for Computational Linguistics.
- [263] Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, Xiaojun Xiao, Kai Cai, Chuang Li, Yaowei Zheng, Chaolin Jin, Chen Li, Xiao Zhou, Minchao Wang, Haoli Chen, Zhaojian Li, Haihua Yang, Haifeng Liu, Feng Lin, Tao Peng, Xin Liu, and Guang Shi. 2025. Ui-tars: Pioneering automated gui interaction with native agents.
- [264] Jiahao Qiu, Yifu Lu, Yifan Zeng, Jiacheng Guo, Jiayi Geng, Huazheng Wang, Kaixuan Huang, Yue Wu, and Mengdi Wang. 2024. Treebon: Enhancing inference-time alignment with speculative tree-search and best-of-n sampling.
- [265] Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, and Xiang Ren. 2023. Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement.
- [266] Yuxiao Qu, Matthew Y. R. Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. 2025. Optimizing test-time compute via meta reinforcement fine-tuning.
- [267] Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve.
- [268] Team Qwen. 2024. Qvq: To see the world with wisdom.
- [269] Gollam Rabby, Farhana Keya, Parvez Zamil, and Sören Auer. 2024. Mc-nest – enhancing mathematical reasoning in large language models with a monte carlo nash equilibrium self-refine tree.
- [270] Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, Bradley Knox, Chelsea Finn, and Scott Niekum. 2024. Scaling laws for reward model overoptimization in direct alignment algorithms.
- [271] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- [272] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3505–3506. ACM.
- [273] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark.
- [274] Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. 2023. Robots that ask for help: Uncertainty alignment for large language model planners. In *7th Annual Conference on Robot Learning*.

- [275] Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024. **RAGChecker: A fine-grained framework for diagnosing retrieval-augmented generation**. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- [276] Yangjun Ruan, Neil Band, Chris J. Maddison, and Tatsunori Hashimoto. 2025. **Reasoning to learn from latent thoughts**.
- [277] Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2024a. **Branch-solve-merge improves large language model evaluation and generation**. 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 8352–8370, Mexico City, Mexico. Association for Computational Linguistics.
- [278] Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, and Tianlu Wang. 2025. **Learning to plan & reason for evaluation with thinking-llm-as-a-judge**.
- [279] Swarnadeep Saha, Archiki Prasad, Justin Chih-Yao Chen, Peter Hase, Elias Stengel-Eskin, and Mohit Bansal. 2024b. **System-1.x: Learning to balance fast and slow planning with language models**.
- [280] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. **A systematic survey of prompt engineering in large language models: Techniques and applications**.
- [281] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. **High-dimensional continuous control using generalized advantage estimation**. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- [282] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. **Proximal policy optimization algorithms**.
- [283] Bytedance Seed. 2025. **Seed-thinking-v1.5: Advancing superb reasoning models with reinforcement learning**. <https://github.com/ByteDance-Seed/Seed-Thinking-v1.5>.
- [284] Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. 2024. **Rl on incorrect synthetic data scales the efficiency of llm math reasoning by eight-fold**.
- [285] Amrith Setlur, Nived Rajaraman, Sergey Levine, and Aviral Kumar. 2025. **Scaling test-time compute without verification or rl is suboptimal**.
- [286] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. **Deepseekmath: Pushing the limits of mathematical reasoning in open language models**.
- [287] Gerald Shen, Zhilin Wang, Olivier Delalleau, Jiaqi Zeng, Yi Dong, Daniel Egert, Shengyang Sun, Jimmy Zhang, Sahil Jain, Ali Taghibakhshi, Markel Sanz Ausin, Ashwath Aithal, and Oleksii Kuchaiev. 2024. **Nemo-aligner: Scalable toolkit for efficient model alignment**.
- [288] Haozhan Shen, Zilun Zhang, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. 2025a. **Vlm-r1: A stable and generalizable r1-style large vision-language model**. <https://github.com/om-ai-lab/VLM-R1>.
- [289] Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, and Shigu Lian. 2025b. **Dast: Difficulty-adaptive slow-thinking for large reasoning models**.
- [290] Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025c. **Codi: Compressing chain-of-thought into continuous space via self-distillation**.
- [291] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. **Hybridflow: A flexible and efficient rlhf framework**. *ArXiv preprint*, abs/2409.19256.
- [292] Freda Shi, Daniel Fried, Marjan Ghazvininejad, Luke Zettlemoyer, and Sida I. Wang. 2022. **Natural language to code translation with execution**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3533–3546, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

- [293] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. *Reflexion: language agents with verbal reinforcement learning*. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- [294] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.
- [295] Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshitij Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. 2023. *Beyond human data: Scaling self-training for problem-solving with language models*.
- [296] Nishad Singhi, Hritik Bansal, Arian Hosseini, Aditya Grover, Kai-Wei Chang, Marcus Rohrbach, and Anna Rohrbach. 2025. *When to solve, when to verify: Compute-optimal problem solving and generative verification for llm reasoning*.
- [297] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. *Scaling llm test-time compute optimally can be more effective than scaling model parameters*.
- [298] Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. *R1-searcher: Incentivizing the search capability in llms via reinforcement learning*. *ArXiv preprint*, abs/2503.05592.
- [299] Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. *To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning*.
- [300] Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. 2025. *Paperbench: Evaluating ai's ability to replicate ai research*.
- [301] Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. 2023. *Gpt-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems*.
- [302] Benedikt Stroebel, Sayash Kapoor, and Arvind Narayanan. 2024. *Inference scaling flaws: The limits of llm resampling with imperfect verifiers*.
- [303] DiJia Su, Sainbayar Sukhbaatar, Michael Rabbat, Yuandong Tian, and Qingqing Zheng. 2024. *Dualformer: Controllable fast and slow thinking by learning with randomized reasoning traces*.
- [304] Vighnesh Subramaniam, Antonio Torralba, and Shuang Li. 2024. *Debategpt: Fine-tuning large language models with multi-agent debate supervision*.
- [305] Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. 2023. *Cognitive architectures for language agents*.
- [306] Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024a. *Fast best-of-n decoding via speculative rejection*.
- [307] Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. 2024b. *Easy-to-hard generalization: Scalable alignment beyond human supervision*.
- [308] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.
- [309] Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. 2024. *Preference fine-tuning of llms should leverage suboptimal, on-policy data*.
- [310] Zhengyang Tang, Ziniu Li, Zhenyang Xiao, Tian Ding, Ruoyu Sun, Benyou Wang, Dayiheng Liu, Fei Huang, Tianyu Liu, Bowen Yu, and Junyang Lin. 2025. *Realcritic: Towards effectiveness-driven evaluation of language model critiques*.

- [311] AlphaCode2 Team. 2024a. Alphacode 2 technical report.
- [312] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. 2025. Gemini robotics: Bringing ai into the physical world. *ArXiv preprint*, abs/2503.20020.
- [313] Open O1 Team. 2024b. Open o1.
- [314] OpenThoughts Team. 2025a. Open Thoughts. <https://open-thoughts.ai>.
- [315] Qwen Team. 2024c. Qwq: Reflect deeply on the boundaries of the unknown.
- [316] Qwen Team. 2025b. Qwq-32b: Embracing the power of reinforcement learning.
- [317] Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ah-san, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, et al. 2025. Llamav-o1: Rethinking step-by-step visual reasoning in llms. *ArXiv preprint*, abs/2501.06186.
- [318] Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Haitao Mi, and Dong Yu. 2024. Toward self-improvement of llms via imagination, searching, and criticizing.
- [319] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*.
- [320] Gladys Tyen, Hassan Mansoor, Victor Cărbune, Peter Chen, and Tony Mak. 2023. Llms cannot find reasoning errors, but can correct them given the error location.
- [321] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback.
- [322] Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. 2023. Can large language models really improve by self-critiquing their own plans?
- [323] Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation.
- [324] Prakhar Verma, Sukruta Prakash Midigeshi, Gaurav Sinha, Arno Solin, Nagarajan Natarajan, and Amit Sharma. 2024. Plan\*rag: Efficient test-time planning for retrieval augmented generation.
- [325] Barbara Von Eckardt. 1995. *What is cognitive science?* MIT press.
- [326] Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. 2024. Dynamic self-consistency: Leveraging reasoning paths for efficient llm sampling.
- [327] Ante Wang, Linfeng Song, Ye Tian, Baolin Peng, Dian Yu, Haitao Mi, Jinsong Su, and Dong Yu. 2024a. Litesearch: Efficacious tree search for llm.
- [328] Ante Wang, Linfeng Song, Ye Tian, Dian Yu, Haitao Mi, Xiangyu Duan, Zhaopeng Tu, Jinsong Su, and Dong Yu. 2025a. Don't get lost in the trees: Streamlining llm reasoning by overcoming tree search exploration pitfalls.
- [329] Chaojie Wang, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. 2024b. Q\*: Improving multi-step reasoning for llms with deliberative planning.
- [330] Evan Wang, Federico Cassano, Catherine Wu, Yunfeng Bai, Will Song, Vaskar Nath, Ziwen Han, Sean Hendryx, Summer Yue, and Hugh Zhang. 2024c. Planning in natural language improves llm search for code generation.
- [331] Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024d. Interpretable preferences via multi-objective reward modeling and mixture-of-experts.
- [332] Haoyu Wang, Zeyu Qin, Li Shen, Xueqian Wang, Minhao Cheng, and Dacheng Tao. 2025b. Leveraging reasoning with guidelines to elicit and utilize knowledge for enhancing safety alignment. *ArXiv preprint*, abs/2502.04040.
- [333] Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. 2025c. Chain-of-retrieval augmented generation.

- [334] Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2023a. **Math-shepherd: Verify and reinforce llms step-by-step without human annotations.**
- [335] Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah D. Goodman. 2023b. **Hypothesis search: Inductive reasoning with language models.**
- [336] Tianlong Wang, Junzhe Chen, Xuetong Han, and Jing Bai. 2024e. **Cpl: Critical plan step learning boosts llm generalization in reasoning tasks.**
- [337] Xiaodong Wang and Peixi Peng. 2025. Open-r1-video. <https://github.com/Wang-Xiaodong1899/Open-R1-Video>.
- [338] Xinglin Wang, Shaoxiong Feng, Yiwei Li, Peiwen Yuan, Yueqi Zhang, Boyuan Pan, Heda Wang, Yao Hu, and Kan Li. 2024f. **Make every penny count: Difficulty-adaptive self-consistency for cost-efficient reasoning.**
- [339] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. **Self-consistency improves chain of thought reasoning in language models.** In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- [340] Yi Wang, Mushui Liu, Wanggui He, Longxiang Zhang, Ziwei Huang, Guanghao Zhang, Fangxun Shu, Zhong Tao, Dong She, Zhelun Yu, Haoyuan Li, Weilong Dai, Mingli Song, Jie Song, and Hao Jiang. 2025d. **Mint: Multi-modal chain of thought in unified generative models for enhanced image generation.** *ArXiv preprint*, abs/2503.01298.
- [341] Yiming Wang, Pei Zhang, Siyuan Huang, Baosong Yang, Zhuosheng Zhang, Fei Huang, and Rui Wang. 2025e. **Sampling-efficient test-time scaling: Self-estimating the best-of-n sampling in early decoding.**
- [342] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024g. **Mmlu-pro: A more robust and challenging multi-task language understanding benchmark.**
- [343] Yutong Wang, Pengliang Ji, Chaoqun Yang, Kaixin Li, Ming Hu, Jiaoyang Li, and Guillaume Sartoretti. 2025f. **Mcts-judge: Test-time scaling in llm-as-a-judge for code correctness evaluation.**
- [344] Zengzhi Wang, Xuefeng Li, Rui Xia, and Pengfei Liu. 2023d. **Mathpile: A billion-token-scale pretraining corpus for math.**
- [345] Zhenglin Wang, Jialong Wu, Yilong Lai, Congzhi Zhang, and Deyu Zhou. 2024h. **Seed: Accelerating reasoning tree construction via scheduled speculative decoding.**
- [346] Zihan Wang, Yunxuan Li, Yuexin Wu, Liangchen Luo, Le Hou, Hongkun Yu, and Jingbo Shang. 2024i. **Multi-step problem solving through a verifier: An empirical analysis on model-induced process supervision.**
- [347] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. **Chain-of-thought prompting elicits reasoning in large language models.** In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- [348] Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneau, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I. Wang. 2025. **Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution.**
- [349] Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. 2024. **From decoding to meta-generation: Inference-time algorithms for large language models.**
- [350] Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2023. **Generating sequences by learning to self-correct.** In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- [351] Han Wu, Yuxuan Yao, Shuqi Liu, Zehua Liu, Xiaojin Fu, Xiongwei Han, Xing Li, Hui-Ling Zhen, Tao Zhong, and Mingxuan Yuan. 2025. **Unlocking efficient long-to-short llm reasoning with model merging.**

- [352] Tianhao Wu, Janice Lan, Weizhe Yuan, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. 2024a. *Thinking llms: General instruction following with thought generation*.
- [353] Ting Wu, Xuefeng Li, and Pengfei Liu. 2024b. Progress or regress? self-improvement reversal in post-training.
- [354] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024c. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models.
- [355] Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2024d. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. 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 1819–1862, Mexico City, Mexico. Association for Computational Linguistics.
- [356] Zhiheng Xi, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang Hong, Shihuan Do, Wenyu Zhan, Xiao Wang, Rui Zheng, Tao Ji, Xiaowei Shi, Yitao Zhai, Rongxiang Weng, Jingang Wang, Xunliang Cai, Tao Gui, Zuxuan Wu, Qi Zhang, Xipeng Qiu, Xuanjing Huang, and Yu-Gang Jiang. 2024. Enhancing llm reasoning via critique models with test-time and training-time supervision.
- [357] Heming Xia, Yongqi Li, Chak Tou Leong, Wenjie Wang, and Wenjie Li. 2025. Tokenskip: Controllable chain-of-thought compression in llms.
- [358] Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2024. Evaluating mathematical reasoning beyond accuracy.
- [359] Violet Xiang, Charlie Snell, Kanishk Gandhi, Alon Albalak, Anikait Singh, Chase Blagden, Duy Phung, Rafael Rafailov, Nathan Lile, Dakota Mahan, Louis Castricato, Jan-Philipp Franken, Nick Haber, and Chelsea Finn. 2025. Towards system 2 reasoning in llms: Learning how to think with meta chain-of-thought.
- [360] Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning.
- [361] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024a. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments.
- [362] Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024b. Monte carlo tree search boosts reasoning via iterative preference learning.
- [363] Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Qizhe Xie. 2023. Self-evaluation guided beam search for reasoning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- [364] Huajian Xin, Z. Z. Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qushu Du, Wenjun Gao, Qihao Zhu, Dejian Yang, Zhibin Gou, Z. F. Wu, Fuli Luo, and Chong Ruan. 2024. Deepseek-prover-v1.5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search.
- [365] Ran Xin, Chenguang Xi, Jie Yang, Feng Chen, Hang Wu, Xia Xiao, Yifan Sun, Shen Zheng, and Kai Shen. 2025. Bfs-prover: Scalable best-first tree search for llm-based automatic theorem proving.
- [366] Wei Xiong, Hanning Zhang, Chenlu Ye, Lichang Chen, Nan Jiang, and Tong Zhang. 2025. Self-rewarding correction for mathematical reasoning.
- [367] Wang Xiyao, Yang Zhengyuan, Li Linjie, Lu Hongjin, Xu Yuancheng, Lin Chung-Ching Lin, Lin Kevin, Huang Furong, and Wang Lijuan. 2024. Scaling inference-time search with vision value model for improved visual comprehension. *ArXiv preprint*, abs/2412.03704.
- [368] Bin Xu, Yiguan Lin, Yinghao Li, and Yang Gao. 2024a. Sra-mcts: Self-driven reasoning augmentation with monte carlo tree search for code generation.

- [369] Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. 2024b. [Llava-cot: Let vision language models reason step-by-step](#). *ArXiv preprint*, abs/2411.10440.
- [370] Haotian Xu, Xing Wu, Weinong Wang, Zhongzhi Li, Da Zheng, Boyuan Chen, Yi Hu, Shijia Kang, Jiaming Ji, Yingying Zhang, Zhijiang Guo, Yaodong Yang, Muhan Zhang, and Debeng Zhang. 2025a. [Redstar: Does scaling long-cot data unlock better slow-reasoning systems?](#)
- [371] Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025b. [Chain of draft: Thinking faster by writing less](#).
- [372] Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Yang Wang. 2024c. [Pride and prejudice: Llm amplifies self-bias in self-refinement](#).
- [373] Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. 2025c. [Softcot: Soft chain-of-thought for efficient reasoning with llms](#).
- [374] Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. 2025d. [Kodcode: A diverse, challenging, and verifiable synthetic dataset for coding](#). *ArXiv preprint*, abs/2503.02951.
- [375] Yuchen Yan, Yongliang Shen, Yang Liu, Jin Jiang, Mengdi Zhang, Jian Shao, and Yueting Zhuang. 2025. [Inftythink: Breaking the length limits of long-context reasoning in large language models](#).
- [376] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jian-hong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024a. [Qwen2.5-math technical report: Toward mathematical expert model via self-improvement](#).
- [377] Kaiyu Yang, Gabriel Poesia, Jingxuan He, Wenda Li, Kristin Lauter, Swarat Chaudhuri, and Dawn Song. 2024b. [Formal mathematical reasoning: A new frontier in ai](#).
- [378] Wen Yang, Minpeng Liao, and Kai Fan. 2024c. [Markov chain of thought for efficient mathematical reasoning](#).
- [379] Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. 2025a. [Towards thinking-optimal scaling of test-time compute for llm reasoning](#).
- [380] Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, Bo Zhang, and Wei Chen. 2025b. [R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization](#).
- [381] Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, et al. 2024a. [Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search](#). *ArXiv preprint*, abs/2412.18319.
- [382] Jinwei Yao, Kaiqi Chen, Kexun Zhang, Jiaxuan You, Binhang Yuan, Zeke Wang, and Tao Lin. 2024b. [Deft: Decoding with flash tree-attention for efficient tree-structured llm inference](#).
- [383] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- [384] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023b. [React: Synergizing reasoning and acting in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- [385] Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. 2024. [Physics of language models: Part 2.2, how to learn from mistakes on grade-school math problems](#).
- [386] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. [Limo: Less is more for reasoning](#).
- [387] Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. [Demystifying long chain-of-thought reasoning in llms](#).
- [388] Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024a. [Distilling system 2 into system 1](#).

- [389] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. 2025a. **Dapo: An open-source llm reinforcement learning system at scale.**
- [390] Tian Yu, Shaolei Zhang, and Yang Feng. 2024b. **Auto-rag: Autonomous retrieval-augmented generation for large language models.**
- [391] Zhaojian Yu, Yinghao Wu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. 2025b. **Z1: Efficient test-time scaling with code.**
- [392] Zishun Yu, Tengyu Xu, Di Jin, Karthik Abinav Sankararaman, Yun He, Wenxuan Zhou, Zhouhao Zeng, Eryk Helenowski, Chen Zhu, Sinong Wang, Hao Ma, and Han Fang. 2025c. **Think smarter not harder: Adaptive reasoning with inference aware optimization.**
- [393] Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. 2024a. **Free process rewards without process labels.**
- [394] Weizhe Yuan, Pengfei Liu, and Matthias Gallé. 2024b. **Llmcrit: Teaching large language models to use criteria.** *ArXiv preprint*, abs/2403.01069.
- [395] Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. **Bartscore: Evaluating generated text as text generation.** In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 27263–27277.
- [396] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024c. **Self-rewarding language models.**
- [397] Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. 2025. **What's behind ppo's collapse in long-cot? value optimization holds the secret.**
- [398] Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. **Scaling relationship on learning mathematical reasoning with large language models.** *ArXiv preprint*, abs/2308.01825.
- [399] Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. 2025. **Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks.**
- [400] Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2024. **Inference scaling for long-context retrieval augmented generation.**
- [401] Milan Zeleny. 1987. Management support systems: Towards integrated knowledge management. *Human systems management*, 7(1):59–70.
- [402] Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. 2024a. **Quiet-star: Language models can teach themselves to think before speaking.**
- [403] Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. 2024b. **Self-taught optimizer (stop): Recursively self-improving code generation.** In *First Conference on Language Modeling*.
- [404] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. **Star: Bootstrapping reasoning with reasoning.** In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- [405] Weihao Zeng, Yuzhen Huang, Wei Liu, Keqing He, Qian Liu, Zejun Ma, and Junxian He. 2025a. **7b model and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient.** <https://hkust-nlp.notion.site/simplerl-reason>. Notion Blog.
- [406] Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Yunhua Zhou, and Xipeng Qiu. 2025b. **Revisiting the test-time scaling of o1-like models: Do they truly possess test-time scaling capabilities?**

- [407] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. [Rest-mcts\\*: Llm self-training via process reward guided tree search](#).
- [408] Di Zhang, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. 2024b. [Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b](#).
- [409] Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, Wanli Ouyang, and Dongzhan Zhou. 2024c. [Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning](#).
- [410] Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue. 2024d. [A careful examination of large language model performance on grade school arithmetic](#).
- [411] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024e. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [412] Jintian Zhang, Yuqi Zhu, Mengshu Sun, Yujie Luo, Shuofei Qiao, Lun Du, Da Zheng, Huajun Chen, and Ningyu Zhang. 2025a. [Lightthinker: Thinking step-by-step compression](#).
- [413] Qiyuan Zhang, Yafei Wang, Yuxin Jiang, Liangyou Li, Chuhan Wu, Yasheng Wang, Xin Jiang, Lifeng Shang, Ruiming Tang, Fuyuan Lyu, and Chen Ma. 2025b. [Crowd comparative reasoning: Unlocking comprehensive evaluations for llm-as-a-judge](#).
- [414] Shimao Zhang, Xiao Liu, Xin Zhang, Junxiao Liu, Zheheng Luo, Shujian Huang, and Yeyun Gong. 2025c. [Process-based self-rewarding language models](#).
- [415] Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B. Tenenbaum, and Chuang Gan. 2023. [Planning with large language models for code generation](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- [416] Wenqi Zhang, Mengna Wang, Gangao Liu, Xu Huixin, Yiwei Jiang, Yongliang Shen, Guiyang Hou, Zhe Zheng, Hang Zhang, Xin Li, et al. 2025d. [Embodied-reasoner: Synergizing visual search, reasoning, and action for embodied interactive tasks](#). *ArXiv preprint*, abs/2503.21696.
- [417] Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. 2024f. [Chain of preference optimization: Improving chain-of-thought reasoning in llms](#).
- [418] Yichi Zhang, Siyuan Zhang, Yao Huang, Zeyu Xia, Zhengwei Fang, Xiao Yang, Ranjie Duan, Dong Yan, Yinpeng Dong, and Jun Zhu. 2025e. [Stair: Improving safety alignment with introspective reasoning](#). *ArXiv preprint*, abs/2502.02384.
- [419] Yuxiang Zhang, Shangxi Wu, Yuqi Yang, Jiangming Shu, Jinlin Xiao, Chao Kong, and Jitao Sang. 2024g. [O1-coder: An o1 replication for coding](#).
- [420] Zijian Zhang, Kaiyuan Zheng, Zhaorun Chen, Joel Jang, Yi Li, Chaoqi Wang, Mingyu Ding, Dieter Fox, and Huaxiu Yao. 2024h. [Grape: Generalizing robot policy via preference alignment](#). *ArXiv preprint*, abs/2411.19309.
- [421] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. 2025. [Cot-vla: Visual chain-of-thought reasoning for vision-language-action models](#). *ArXiv preprint*, abs/2503.22020.
- [422] Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023a. [Verify-and-edit: A knowledge-enhanced chain-of-thought framework](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.
- [423] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023b. [A survey of large language models](#).
- [424] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. 2023c. [Pytorch fsdp: Experiences on scaling fully sharded data parallel](#).

- 
- [425] Yu Zhao, Hufeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024. [Marco-01: Towards open reasoning models for open-ended solutions](#).
  - [426] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023a. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
  - [427] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2023b. [Slang: Efficient execution of structured language model programs](#).
  - [428] Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. [Deepresearcher: Scaling deep research via reinforcement learning in real-world environments](#).
  - [429] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2023. [Language agent tree search unifies reasoning acting and planning in language models](#).
  - [430] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. 2024a. [Transfusion: Predict the next token and diffuse images with one multi-modal model](#). *ArXiv preprint*, abs/2408.11039.
  - [431] Fan Zhou, Zengzhi Wang, Qian Liu, Junlong Li, and Pengfei Liu. 2024b. [Programming every example: Lifting pre-training data quality like experts at scale](#).
  - [432] Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. 2025a. [R1-zero's "aha moment" in visual reasoning on a 2b non-sft model](#).
  - [433] Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. 2025b. [Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks](#).
  - [434] Qihao Zhu, Daya Guo, Zhihong Shao, Dejian Yang, Peiyi Wang, Runxin Xu, Y Wu, Yukun Li, Huazuo Gao, Shirong Ma, et al. 2024. [Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence](#). *ArXiv preprint*, abs/2406.11931.
  - [435] Yuchen Zhuang, Xiang Chen, Tong Yu, Saayan Mitra, Victor Bursztyn, Ryan A. Rossi, Somdeb Sarkhel, and Chao Zhang. 2023. [Toolchain\\*: Efficient action space navigation in large language models with a\\* search](#).
  - [436] Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbulin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. 2024. [Agent-as-a-judge: Evaluate agents with agents](#).