# RL “顿悟”秘籍：强化学习如何解锁并迁移大语言模型中的新算法？

## 摘要

**原文翻译** 一个悬而未决的问题是，大语言模型（LLMs）能否获得或泛化出真正新颖的推理策略，而不是仅仅将在预训练或后训练过程中编码于其参数中的技能进行打磨。为了尝试回答这一争议，我们引入了DELTA——算法编码中可学习性与可迁移性的分布式评估基准。这是一个由合成编码问题族构成的受控基准，旨在探究两个基本方面：**可学习性**——通过强化学习（RL），LLMs能否解决那些预训练模型在足够多次尝试后仍表现失败（pass@$K = 0$）的问题族？以及**可迁移性**——如果可学习性得以实现，这种技能能否系统性地迁移到分布外（OOD）测试集上？与以往的公开编码数据集不同，DELTA通过模板化的问题生成器来隔离推理技能，并引入了全新的OOD问题族，这些问题族要求的是新颖的策略，而非调用工具或记忆模式。我们的实验揭示了一个惊人的“顿悟”（grokking）阶段性转变：在经历了长时间近乎零奖励的阶段后，经过RL训练的模型会突然跃升至近乎完美的准确率。为了在先前无法解决的问题族上实现可学习性，我们探索了关键的训练要素，如使用密集奖励进行阶段性预热、经验回放、课程学习以及“验证在环”。在可学习性之外，我们利用DELTA来评估其在探索性、组合性和变革性三个轴向上的可迁移性或泛化能力，以及跨问题族的迁移。结果显示，在问题族内部和技能重组方面有显著提升，但在变革性案例中仍存在持续的弱点。因此，DELTA为探索RL驱动推理的极限，以及理解模型如何超越现有先验知识以获得新算法技能，提供了一个纯净的测试平台。代码已在 https://github.com/sunblaze-ucb/rl-grok-recipe 上提供。

**深度解读** 想象一下，我们想知道人工智能（AI）究竟是在“学新东西”，还是仅仅在“熟能生巧”。这篇论文的摘要开篇就抛出了这个核心问题。研究者们不想停留在哲学辩论上，他们想用实验来找到答案。为此，他们打造了一个名叫DELTA的“AI健身房”。这个健身房和我们平常见到的不一样，它不是一个大杂烩式的训练场，而是一个由许多精心设计的、独立的“训练室”组成的。每个训练室（也就是一个“问题族”）都只考验AI某一项特定的逻辑推理能力。

这个“健身房”有两个核心考核标准。第一是“可学习性”：对于那些AI之前无论尝试多少次都做不对的难题（通过率为零），我们能不能通过一种特殊的训练方法——强化学习（RL）——教会它？这就好比一个学生面对一道他完全没有头绪的奥数题，我们想看看有没有一种教学方法能让他彻底掌握解题思路。第二是“可迁移性”：如果AI真的学会了新本事，它能举一反三吗？比如，学会了解决A类难题，面对一个看起来有点像A但又不完全一样的B类难题，它还能不能应对自如？

最令人兴奋的发现是，AI在学习过程中出现了一个类似人类“顿悟”的现象。在训练初期，AI就像一个苦思冥想的学生，很长一段时间里都毫无进展，得分一直是零。但突然在某个时刻，它仿佛茅塞顿开，解题成功率瞬间飙升到接近100%。研究者们把这个过程比作找到了一个“秘籍”，这个秘籍的关键在于一种特殊的训练策略：先用一些简单的评分标准（比如解题步骤对了就给分）来“预热”和引导AI，让它不至于完全迷失方向；然后再切换到严格的最终评分标准（必须得到完美答案才给分）。

最后，研究者们测试了AI“顿悟”后学到的新技能。他们发现，对于同类型但更难的问题（探索性），或者把几个学会的技能组合起来使用（组合性），AI都表现得不错。但这就像一个武林高手，学会了“降龙十八掌”和“凌波微步”，他可以把这两招结合得很好。然而，当面对一个需要完全颠覆

原有武学理念、创造出全新招式才能解决的难题时（变革性），AI就束手无策了。这篇论文不仅证明了AI有能力学习全新的知识，还清晰地指出了它当前能力的边界在哪里，为未来的研究指明了方向。

---

## 1. 引言

**原文翻译** 关于语言模型强化学习的一个核心问题是，它仅仅是在打磨潜在的技能，还是能够催生出真正新颖的推理能力。一些人认为，强化学习只是在优化模型参数中已嵌入的现有启发式方法（Yue et al., 2025; Wu et al., 2025），而另一些人则将其视为一种解锁涌现式问题解决能力的方式（Liu et al., 2025b,a）。我们通过两个标准使这场辩论变得可检验：**可学习性**，即探究强化学习是否能灌输一种模型先前无法执行的程序；以及**泛化能力**，即探究该程序是否能迁移到多样的分布外（OOD）情况，而非仅仅是记忆化的模式。要解决这些问题，需要一个具有严格控制的训练-测试划分的数据集，以便系统性地探究这两种属性。

**深度解读** 这一段为整篇论文的研究拉开了序幕，它精准地定位了当前人工智能领域一个非常根本的争论：我们用强化学习（一种类似训练宠物，通过奖励和惩罚来学习的方法）去训练一个大语言模型，到底是在让它“百尺竿头，更进一步”，还是在教它“开天辟地，无中生有”？前者认为，模型本身就像一个知识渊博但有时会犯迷糊的学者，强化学习只是帮他梳理思路，让他已有的知识用得更溜。后者则认为，强化学习有潜力成为一把钥匙，能打开模型内心深处一扇从未开启的门，释放出它自己都不知道的、全新的解决问题的能力。

为了不让这个讨论停留在口水战上，研究者们提出了两个非常具体、可以量化的“试金石”。第一个叫“可学习性”，说白了就是：如果模型原来对某个问题是“一窍不通”（无论试多少次都失败），我们能不能通过强化学习，像一位导师一样，教会它完整的解题步骤，让它从0分变成100分？这考验的是“从无到有”的能力。第二个叫“泛化能力”，意思是：好，就算它学会了，那它是死记硬背了这个题目的解法，还是真正理解了背后的原理？检验方法就是给它换个“马甲”——题目背景、数字都变了，但核心逻辑不变——看看它还能不能做出来。这考验的是“举一反三”的能力。要做到这两点，就不能随随便便从网上扒数据来测试，而必须像在实验室里做实验一样，精心设计一套“考题”，确保训练和测试的题目之间界限分明，这样才能准确地衡量模型到底学到了什么。

---

**原文翻译 为什么受控的问题族很重要？** 在数学/编码领域的非受控开放基准（例如，Numina-Math (Li et al., 2024), DeepMath (He et al., 2025), OpenCodeReasoning (NVIDIA, 2025)）混合了不同的主题和难度，模糊了能力打磨与真正习得之间的界限。受控的合成问题族则消除了这些混淆因素：我们可以精确地改变分布和难度，将模型的进步归因于特定的技能，检测阶段性转变，并系统性地测试向OOD变体的迁移。

**深度解读** 这里，研究者们解释了他们为什么非要“小题大做”，自己去创造一个全新的测试数据集，而不是用现成的。他们把现在流行的那些AI竞赛榜单，比如数学或编程大赛的题库，比作一个“大杂烩考场”。在这样的考场里，题目五花八门，难度也参差不齐。一个模型在这样的考试中得了高分，我们很难说清楚它到底是靠“刷题”积累的经验，还是真的掌握了新的数学或编程原理。这就好比一个学生期末考了高分，我们分不清他是因为之前做过类似的题，还是因为他真正理解了背后的知识点。这种模糊性，对于想搞清楚AI学习本质的科学家来说，是无法接受的。

所以，他们选择了另一条路：自己动手，打造一个“精密的实验室”。在这个实验室里，每一组“实验器材”（也就是一个问题族）都是高度受控的。研究者可以像调节显微镜的焦距一样，精确地控制

题目的难度、考查的知识点类型。这样做的好处是显而易见的：如果模型在一个只考“排序算法”的问题族上取得了进步，我们就能非常有信心地说，它学会了“排序算法”，而不是别的。我们还能观察到，当难度提升到某个临界点时，模型的表现是否会发生突变（就像水在100度时会沸腾一样）。更重要的是，我们可以设计一些“前所未见”的题目变种，来严格测试模型的泛化能力。这种方法，虽然麻烦，但它把研究从一个模糊的“艺术”问题，变成了一个可以精确测量的“科学”问题，这正是这篇论文方法论的核心所在。

---

**原文翻译 为什么选择编程问题？** GRPO/PPO这类强化学习流程通常依赖于“通过/失败”的奖励：一个完美的解决方案获得+1分，其他任何情况都获得0分。在难题族上，这种稀疏的奖励可能会导致学习停滞。在数学领域，对中间步骤进行评分成本高昂且难以规模化。然而，编程天然地通过测试用例提供了细粒度的反馈，这些测试用例起到了密集奖励的作用。一个实用的方法是，开始时使用基于测试用例的奖励进行训练，以鼓励部分进展，然后过渡到二元结果奖励，以锁定精确的解决方案。这种分阶段的方案对于帮助LLMs获得真正新颖的程序化策略至关重要。尽管编码提供了一个独特的可扩展环境，但其核心思想——在强制要求完全正确之前使用中间信号——可能也适用于其他重推理领域，如数学或形式逻辑。

**深度解读** 这一段解释了为什么研究者们选择“编程”作为他们实验的“小白鼠”。在强化学习中，模型就像一个在黑暗中摸索的孩子，它需要知道自己做得好不好。最简单的反馈方式就是“非黑即白”：程序完全跑通了，奖励一个糖果（+1分）；否则，什么都没有（0分）。但问题是，对于一个很难的编程题，模型可能一开始连一行正确的代码都写不出来，这样它就永远也得不到那个+1的奖励，也就永远不知道该往哪个方向努力，学习就卡住了。这就好比教一个孩子投篮，如果只有投进篮筐才算得分，他可能会因为一直投不进而失去信心，最终放弃练习。

数学问题也有类似的困境，虽然我们可以去检查解题的中间步骤是否正确，但这太费劲了，就像需要一个老师随时盯着学生的草稿纸，很难大规模应用。而编程的妙处就在于，它天然地提供了一种“自动评分系统”——测试用例。一个程序需要通过10个测试用例才算完美，那么模型每多通过一个，我们就可以给它一点“小奖励”。这就把原来“0或1”的奖励，变成了“0.1, 0.2,..., 1.0”这样更平滑、更密集的奖励。

研究者们提出的“秘籍”就是利用这一点。他们设计了一个两步走的训练策略：第一阶段，采用“通过测试用例的比例”来打分。模型只要取得一点点进步（比如通过了10个测试用例中的1个），就能得到正向反馈，这会鼓励它继续探索。这就像告诉投篮的孩子：“不错，这次虽然没进，但打到篮板了，离成功更近了一步！”。当模型的能力被“预热”到一定程度，能够比较稳定地写出大部分正确的代码后，再进入第二阶段，切换回严格的“必须全部通过”的评分标准，来打磨细节，确保最终方案的完美。这个“先引导，再精确”的策略，是他们能够解开那些“不可能任务”的关键。这个思想非常有启发性，它告诉我们，在教AI学习复杂技能时，如何设计奖励机制，可能和算法本身一样重要。

---

**原文翻译** 为了满足这一需求，我们引入了DELTA，一个受控但多样化的编程问题基准。DELTA由来自不同领域的合成问题族组成，每个问题族都由模板化的问题生成器生成，使我们能够在纯净和隔离的环境中研究诸如难度扩展、知识迁移和可学习性等现象。

**深度解读** 这里正式推出了本研究的核心工具——DELTA基准测试集。你可以把它想象成一本为AI量身定做的、极其特殊的“奥林匹克竞赛习题集”。说它“特殊”，有以下几个原因：首先，它“受控

但多样化”。这意味着习题集里的题目不是东拼西凑来的，而是覆盖了几个完全不同的领域（比如逻辑解谜、物理模拟等），但在每个领域内部，题目的生成都遵循着严格的“模板”。这就好比一本数学练习册，既有代数章节，也有几何章节，但在代数章节里，所有的题目都是围绕着“解一元二次方程”这个核心知识点，只是数字和背景故事在变。

其次，它能让我们在“纯净和隔离”的环境中做研究。由于每个问题族都只针对一个特定的技能，这就创造了一个理想的“实验室环境”。当AI在某个问题族上表现提升时，我们可以非常确定地知道是哪项能力变强了。这避免了在混乱的真实世界数据中进行猜测。利用这个精密的工具，研究者们就可以像物理学家研究粒子一样，系统地研究AI智能的几个基本属性：比如，题目难度稍微增加一点，AI的表现会下降多少（难度扩展）？在一个领域学到的知识，能不能用到另一个领域去（知识迁移）？以及最核心的，AI到底能不能学会它原来完全不会的东西（可学习性）？DELTA不仅仅是一个数据集，它更是一个科学仪器，一个用来精确剖析和度量AI推理能力的“显微镜”。

---

**原文翻译 RL可学习性研究。** 我们揭示了在RL训练过程中一个未被充分探索的“顿悟”（grokking）现象。尽管最近的研究认为RL无法超越其参考模型的极限（Yue et al., 2025; Wu et al., 2025），但我们的证据表明并非如此。在基础模型表现为pass@$K = 0$[3]的难题上，使用二元奖励的标准RL会因缺乏正向信号而崩溃。相比之下，一种分阶段的方案——先用细粒度的代理奖励进行预热，再切换到严格的“通过/失败”奖励——首先引导探索进入一个能够触及完整解决方案的区域，然后将其打磨成经过验证的完整程序。这产生了一个漫长的探索平台期，随后突然“顿悟”，达到近乎完美的准确率（图1，右上角）。

[3] 此处pass@K指的是在K值很大（例如128）时。因此，pass@$K = 0$表示模型即使在多次采样尝试后也未能解决任务。

**深度解读** 这一段揭示了论文中最激动人心的发现，并直接挑战了当时AI领域的一些主流看法。当时，很多专家认为，强化学习（RL）最多只能让一个模型变得更好，但无法让它变得“不同”。也就是说，如果一个模型的基础能力有上限，那么无论怎么用RL训练，它也突破不了这个“天花板”。

而这篇论文的研究者们说：“不，我们有不同的发现。” 他们在一个特别设计的难题上进行了实验，这个难题对于未经训练的“基础模型”来说，是绝对无解的（即使用128次机会去尝试，成功率也是0）。他们发现，如果用传统的方法（做对了就奖，做错了不奖不罚）去训练，模型确实会“躺平”，因为一直得不到奖励，学习就停滞了。

但他们发明的“秘籍”——分阶段训练法——却创造了奇迹。这个方法分两步：第一步是“预热”，使用一种更宽容的评分标准，比如“程序通过了部分测试就算分”。这就像给在黑暗中摸索的模型点亮了一盏小灯，虽然光线微弱，但足以让它辨别大致正确的方向。这一阶段，模型的表现提升缓慢，看起来像是在一个很低的水平上“徘徊”，这就是所谓的“漫长的探索平台期”。第二步，当模型被引导到“正确答案”附近后，立刻切换回最严格的评分标准——“必须完全正确才给分”。就在这个切换之后，神奇的事情发生了：模型的成功率突然从几乎为零飙升到接近百分之百！这个现象，他们称之为“顿悟”（grokking）。这强有力地证明了，通过正确的引导，强化学习不仅能“打磨”模型，更能“解锁”它前所未有的新能力，彻底打破了原有的“天花板”。

---

**原文翻译 RL泛化能力研究。** DELTA扩展了OMEGA在Boden创造力类型学（Boden, 1998）指导下的三个轴向上的受控测试：（1）**探索性**：在同一问题族内扩展已知技能（例如，从六边形到八边形）；（2）**组合**

**性**：结合先前独立的技能（例如，一个弹跳球同时面对旋转的障碍物和移动的箱子）；（3）**变革性**：发现非常规的解决方案（例如，找到能保证周期性运动的特殊初始状态）。我们的结果显示，经过RL训练的模型能够泛化到更难和组合的变体，但随着复杂性的增加，性能有所下降，而变革性的案例仍然是最具挑战性的。

**深度解读** 在证明了AI能够“学会”新知识之后，研究者们接着要回答一个更深刻的问题：AI是真的“理解”了，还是只是“学会了应付考试”？这就是泛化能力研究。他们借鉴了一个关于人类创造力的理论，把AI的“举一反三”能力分成了三个等级，这就像是对AI创造力的“段位”评定。

第一级是“探索性”泛化，这是最基础的。好比AI学会了如何模拟一个球在六边形盒子里反弹，现在我们把盒子换成八边形，看看它还能不能搞定。这考验的是它在自己熟悉的领域内，处理稍微复杂一点情况的能力。

第二级是“组合性”泛化，难度更高。比如，AI分别学会了两种技能：1）处理旋转的盒子；2）处理移动的盒子。现在我们给它一个新任务：一个既在旋转又在移动的盒子。这考验的是它能否像搭积木一样，把两个独立的技能无缝地组合起来，解决一个复合型问题。

第三级是“变革性”泛化，这是最高段位，近乎于真正的“创造”。它要求AI发现一种全新的、甚至可以说是“取巧”的解法。比如，在模拟小球反弹时，大多数情况都是杂乱无章的。但在某些极其特殊的初始条件下，小球的运动轨迹会呈现完美的周期性。AI能不能不通过蛮力模拟，而是像一个物理学家一样，洞察到这个规律，并直接给出周期性的解？这考验的是它能否跳出常规思维，发现问题背后更深层次的规律。

实验结果很有趣：AI在前两个等级上表现出色，证明它确实学到了可以灵活运用的技能。但在最高级的“变革性”挑战面前，它几乎完全失败了。这表明，目前的AI虽然已经是一个出色的“工程师”，能够熟练应用和组合已有的知识，但离成为一个能够提出颠覆性理论的“科学家”，还有很长的路要走。

---

**原文翻译 主要贡献。** 1) **一个受控的数据集 (DELTA):** 我们设计了一套合成编程问题族，用以隔离推理技能，从而能够对**可学习性**（RL能否解锁基础模型中没有的程序）和**泛化能力**（这些程序能否系统性地迁移到OOD案例）进行纯净的测试。与以往的编码或数学数据集不同，DELTA引入了全新的OOD问题（Manufactoria）和丰富的渐进式奖励，避免了基于工具的捷径和数据混淆。2) **是打磨还是发现，取决于设置:** 我们提供了明确的证据，表明RL并不仅限于打磨参考模型中已有的能力。在基础模型失败的难题族上（pass@K=0），采用从密集到二元奖励的分阶段训练能够产生一个“顿悟”式的阶段性转变——从失败到精通的突然飞跃，这表明RL确实可以发现基础模型无法执行的策略。同时，在较简单的场景或较弱的设置下，RL主要是在打磨现有技能。最终出现哪种结果，关键取决于奖励设计、数据混合、任务难度和训练秘籍。3) **三轴泛化分析:** 我们评估了这些学到的策略如何沿着探索性、组合性和变革性三个轴向进行迁移。结果显示，在探索性和重组案例中泛化能力很强，但在变革性转变中持续失败，这既凸显了RL驱动推理的前景与局限，也指出了我们必须努力解决的泛化挑战。

**深度解读** 这部分总结了整篇论文的三大核心贡献，可以看作是研究团队的“成果汇报”。

第一个贡献是他们打造的“科学仪器”——**DELTA数据集**。这不仅仅是一堆题目，更是一种全新的研究方法。它的创新之处在于：首先，它像一个手术刀一样，能精确地把AI的各种“推理技能”分离

开来单独测试。其次，它包含了一个名为“Manufactoria”的“外星游戏”，这个游戏的规则和语法在地球上（也就是AI的训练数据里）从未出现过，这就杜绝了AI“作弊”（比如靠记忆或调用现有工具）的可能性，确保我们测试的是它真正的学习能力。最后，它内建了一套巧妙的“渐进式奖励”机制，这是实现“顿悟”的关键。

第二个贡献是他们对一个核心争论给出的明确答案：**强化学习（RL）到底是“磨刀石”还是“魔法棒”？答案是：两者都是，关键看你怎么用。** 这篇论文用实验证明，如果你给AI的任务比较简单，或者你的训练方法比较粗糙，那RL确实只能起到“打磨”的作用，让AI的现有技能更熟练。但是，如果你面对的是一个AI完全束手无策的难题，并使用他们发明的“秘籍”（比如从密集奖励到二元奖励的分阶段训练），RL就能化身“魔法棒”，让AI实现从0到1的突破，学会全新的解题策略。这个发现非常重要，它告诉我们，AI的潜力有多大，不仅取决于模型本身，更取决于我们如何去“教”它。

第三个贡献是他们绘制的一幅**AI泛化能力的“能力地图”**。通过探索性、组合性、变革性这三个维度的测试，他们清晰地标示出了当前AI能力的边界。地图显示，AI在“熟悉的领域内深耕”（探索性）和“将已有知识融会贯通”（组合性）方面已经相当强大，但在“跳出思维定势、实现范式创新”（变革性）方面，还是一片空白。这张地图不仅展示了RL技术已经取得的巨大成功，也为未来的研究者们指明了最需要攻克的“无人区”。

---

## 2. DELTA: 受控的编程问题族

**原文翻译** 我们通过DELTA——一个受控的合成编程问题族套件——来将可学习性和泛化能力操作化。

**深度解读** 这一句话是承上启下的引子。前面长篇大论地解释了为什么要研究“可学习性”和“泛化能力”，以及为什么需要一个特殊的工具。现在，研究者们正式亮出他们的核心工具——DELTA。这里的“操作化”是一个科学术语，意思是将一个抽象的概念（比如“学习能力”）转化为一个可以具体测量和执行的步骤或指标。也就是说，DELTA这个编程问题集合，就是他们用来把“AI能不能学会新东西”和“AI能不能举一反三”这两个哲学问题，变成一个个可以跑实验、出数据、画图表的科学问题的具体方法。从这里开始，文章将深入介绍这个“科学仪器”的内部构造。

---

**原文翻译 从OMEGA到DELTA。** OMEGA (Sun et al., 2025) 提供了40个可合成的数学问题族，用以研究符合Boden (Boden, 1998) 精神的探索性、组合性和变革性泛化。DELTA通过转向编程领域对此进行了补充，在编程领域，模板化的生成器可以产生具有可调难度和纯净分布控制的可自动验证的任务。与OMEGA相比，DELTA还提供了独特的优势和改进：a) **新颖的OOD问题族。** OMEGA中的数学任务仍属于熟悉的领域（如代数、几何），这些很可能出现在预训练语料库中。相比之下，DELTA包含一个名为Manufactoria的手工制作的分布外（OOD）问题域，它使用完全新颖的程序语法和问题解决策略。b) **更难用工具走捷径。** 许多合成的数学项目可以通过执行Python代码来解决（例如，计算矩阵的秩）。在DELTA中，目标是程序本身：模型必须合成一个正确的解决方案，而不是将计算委托给外部工具。c) **丰富的奖励信号。** 编程通过每个测试用例的通过率，实现了廉价的、分级的反馈，这支持了分阶段训练（先密集奖励，后二元完全通过奖励）。

**深度解读** 这里介绍了DELTA这个工具的“前世今生”。原来，这个研究团队之前已经开发过一个类似的工具叫OMEGA，但OMEGA是专注于数学问题的。DELTA可以看作是OMEGA的“升级版”和“跨界版”。

从OMEGA到DELTA的进化，主要体现在三个方面，这三个方面都旨在让对AI的考验变得更加“纯粹”和“严格”：

a) **更彻底的“陌生环境”考验**：之前的OMEGA虽然题目是新编的，但终究离不开代数、几何这些AI在网上早已“司空见惯”的数学领域。AI可能不是真的会推理，而是记住了类似的解题模式。而DELTA引入的“Manufactoria”问题，就像是把AI扔到一个外星球，那里有它从未见过的语言（程序语法）和物理规律（解题策略）。在这种环境下，AI无法依赖任何过去的经验，只能靠纯粹的现场学习和推理来解决问题，这才是对“可学习性”最严苛的考验。

b) **杜绝“抄作业”的可能**：在OMEGA的数学问题里，聪明的AI有时可以“偷懒”，比如遇到一个复杂的计算，它不去自己推导，而是偷偷写一小段Python代码，让计算器帮忙算出答案，然后把答案报出来。但在DELTA的编程任务里，任务本身就是要“写出那段代码”。这就好比考试，之前是问你“答案是多少”，现在是让你“写出完整的解题过程”，AI无法再把核心的计算任务外包给其他工具，必须亲力亲为。

c) **更精细的“计分板”**：正如之前提到的，编程任务天然地带有一个“多阶段计分”系统——通过的测试用例越多，得分越高。这为研究者们实施他们那个关键的“预热+冲刺”两阶段训练法提供了极大的便利。这种细粒度的反馈，就像一个循循善诱的教练，能更有效地引导AI学习。

总而言之，从OMEGA到DELTA，研究者们在不断地“堵上漏洞”，旨在创造一个越来越理想化的实验环境，以确保他们观察到的，是AI真正的推理能力，而非记忆力或取巧的能力。

---

**原文翻译** 在DELTA中，我们设计了来自五个主要领域的问题，如图1所示。接下来我们将详细介绍这些问题族。

**图1：DELTA概览及受控RL研究。**

- **左侧：合成编程问题族**
  - **Manufactoria**：具有自定义语法和类似解谜规则。
    - 谜题规则：输入是彩色纸带。拉取器（Puller）移除前端的一个颜色并根据颜色分流。涂色器（Painter）在末端添加颜色。
    - 玩具示例：问题是“接受没有红色的纸带”。解决方案代码展示了如何用 `START`，`NEXT`，`PULLER`，`END` 等指令构建一个状态机来检查纸带。
    - 问题族： `START`，`EXACT`，`REGEX`，`HAS` 等。
  - **BounceSim**：涉及物理模拟。
    - 玩具示例：任务是预测物体在时间t的位置。示例配置给出了盒子（六边形）和物体（三角形）的形状、尺寸、位置、速度等参数。
    - 问题族： `ROTAT_OBJ` (物体旋转)，`GRAVITY` (有重力)，`MOV_BOX` (箱子移动)等。
  - **Competition Coding (竞赛编程)**
    - 问题族： `SORT_DC` (排序分治)，`SEGMENT_TREE_DC` (线段树分治)，`CDQ_DC`，`MO_ALG`，`MEET_IN_MID` (中间相遇)等。
  - **SQL**

    - **LEAN** (一个定理证明器)
- **右侧：受控RL实验**
    - **RL可学习性研究**
        - 图表标题：**Grokking现象**
        - 图表展示了一条学习曲线，分为三个阶段：**探索阶段**（Exploration Phase，奖励很低）、**顿悟！**（Grokking!，奖励急剧上升）、**收敛阶段**（Convergence Phase，奖励维持在高位）。
        - 文字描述：可学习性研究显示了“顿悟”现象，其中RL从长时间的探索突然转变为快速收敛，揭示了超越参考模型的策略。
    - **RL泛化能力研究**
        - 文字描述：泛化能力研究扩展了OMEGA的四个轴向测试——探索性、组合性、变革性和领域级——测试对更难或重组任务的适应能力。
        - **探索性泛化 (Explorative Gen.)**：训练集和测试集问题类型相同，但测试集更难。例如，训练集是处理特殊六边形，测试集是处理旋转的八边形。
        - **组合性泛化 (Compositional Gen.)**：训练集包含独立技能，测试集要求组合这些技能。
        - **变革性泛化 (Transformative Gen.)**：训练集是常规情况，测试集是需要新颖思路的特殊情况。
        - **领域泛化 (Domain Generalization)**：训练集和测试集来自不同领域。

**深度解读** 这张图是整篇论文的“总览图”或“导航图”，它用一种非常直观的方式，展示了研究的两大支柱：左边的“实验材料”（DELTA问题集）和右边的“实验过程与发现”。

**左侧部分：AI的“铁人三项”赛场** 这里展示了DELTA包含的几大类编程挑战，每一类都像是一个独特的赛场，考验AI的不同能力：

- **Manufactoria**：这是一个纯粹的“逻辑迷宫”。AI需要学习一种全新的、类似汇编的语言，来编写程序处理虚拟的“彩色纸带”。这个赛场考验的是AI最底层的、与过往经验完全无关的抽象逻辑推理能力。
- **BounceSim**：这是一个“虚拟物理实验室”。AI需要编写程序，精确模拟小球在各种奇形怪状的、甚至会动会转的容器里如何反弹。这个赛场考验的是AI对几何、物理规律的理解和数值计算的精确性。
- **竞赛编程、SQL、LEAN**：这些是更贴近“现实世界”的赛场。竞赛编程考验的是复杂算法设计能力；SQL考验的是数据库查询和逻辑操作能力；LEAN则考验的是在形式化系统中进行严格数学证明的能力。

**右侧部分：实验的“剧情梗概”** 这里预告了论文后续将要讲述的两个核心故事：

- **可学习性研究与“顿悟”现象**：上面的小图生动地画出了一条“逆袭曲线”。一开始，AI的学习曲线几乎是平的，趴在底部（探索阶段），表示它毫无头绪。然后，突然之间，曲线像火箭一样

垂直蹿升（顿悟！），最后稳定在一个很高的水平上（收敛阶段）。这个戏剧性的转变，是论文关于“AI能学会新知识”的核心证据。

- **泛化能力研究的三个维度**：下面的四个小框则定义了如何评判AI是不是真的“学懂了”。它把“举一反三”分成了不同层次：
  - **探索性**：在同一条路上走得更远（比如从处理简单图形到复杂图形）。
  - **组合性**：把两条走过的路，融合成一条新路（比如把“旋转”和“移动”两种情况结合）。
  - **变革性**：放弃老路，发现一条全新的捷径（比如发现周期性运动的规律）。
  - **领域泛化**：在一个领域学到的本事，能不能用到另一个完全不同的领域。

通过这张图，我们就能对整个研究的框架有一个清晰的认识：研究者们先是精心搭建了几个高难度的“赛场”，然后通过一种特殊的训练方法，观察到了AI“顿悟”式的学习过程，并最终通过一套多维度的测试，精确地评估了AI学到的新技能的“含金量”。

---

### 2.1 Manufactoria (用于可学习性研究的分布外问题)

**原文翻译** Manufactoria是一款经典的Flash游戏（2010年），玩家在其中建造自动化工厂，根据机器人的彩色纸带模式对其进行分类。其底层逻辑类似于使用两种特殊的节点类型（拉取器、涂色器）来构建有限状态自动机或标签系统。虽然原始游戏是在二维空间中实现的，但我们将其重新形式化为一种自定义的程序化语法，如图1所示。详细信息在附录A.1中提供。

**深度解读** 这一段介绍了DELTA中最具特色、也是最核心的一个问题族——Manufactoria。它的来源很有趣，是一款十几年前的古董级网页小游戏。选择它的原因，恰恰是因为它的“古老”和“小众”。研究者们看中的是它背后独特的逻辑核心：玩家需要像设计电路一样，用有限的几种元件（拉取器、涂色器）搭建一个流水线，来处理和识别一串由颜色组成的“密码”（彩色纸带）。这本质上是在用一种可视化的方式，来构建一个计算机科学中的基础模型——“有限状态自动机”。

研究者们做了一件非常巧妙的转化工作：他们没有让AI去“玩”这个游戏，而是把游戏的规则和操作，提炼成了一套全新的、基于文本的编程语言。这就好比他们把围棋的规则，从棋盘和棋子，变成了一套可以用文字描述的指令集。这样做的好处是，AI必须从零开始学习这门“外星语言”的语法和逻辑，而不能依赖任何它在互联网上学到的关于现代编程语言（如Python、Java）的知识。这为测试AI的纯粹学习能力，创造了一个近乎完美的“无菌环境”。

---

**原文翻译 合理的OOD特性。** 这个任务之所以是分布外（OOD）的，有几个原因：a) 原始游戏解决方案仅以图片形式存储在老旧网站上。我们转换后的程序语法是全新的，任何LLM在预训练期间都不可能接触到；b) 我们没有复用已有的游戏挑战。相反，我们设计了受其机制启发但由作者合成的新问题族，这些对LLMs来说是完全未见的；c) 其解谜策略在性质上不同于传统的编程或图灵机任务。仅有两种功能有限的节点类型可用，解决问题需要独特的、未被标准编码策略所捕获的推理模式。

**深度解读** 这里详细解释了为什么Manufactoria是一个“货真价实”的分布外（Out-of-Distribution, OOD）任务。OOD是机器学习领域的一个术语，指的是测试数据与训练数据来自完全不同的分布，通俗讲就是“考题范围和复习资料完全不一样”。要证明一个任务是OOD的，需要非常有力的证据，研究者们从三个层面进行了论证：

a) **语法的“前所未见”**：AI的知识主要来源于它在预训练阶段“阅读”过的海量互联网文本。研究者们指出，这款老游戏的攻略在网上大多是图片形式，而他们发明的这套文本编程语法是100%原创的。这意味着，在AI庞大的知识库里，关于这门“语言”的信息是零。AI面对它，就像一个只学过英语的人突然看到一本写满甲骨文的书，完全无法依赖过去的知识。

b) **问题的“前所未见”**：即使AI侥幸在哪里见过这个游戏的玩法，研究者们也杜绝了它“背题”的可能。他们没有直接用游戏里的老关卡，而是自己当“出题人”，基于游戏的规则，设计了全新的、一系列难度递增的谜题。这确保了AI面对的每一个问题都是一次全新的挑战。

c) **思维方式的“前所未见”**：这一点最为关键。解决Manufactoria谜题所需要的思维方式，和我们平时写代码的逻辑有很大不同。常规编程我们有丰富的工具（变量、循环、函数等），而在这里，你手上只有“拉取器”和“涂色器”这两种极其简单的“积木”。你必须用这些有限的工具，通过巧妙的组合和状态转换，来搭建出复杂的逻辑。这种“戴着镣铐跳舞”的感觉，强迫AI去探索一种它在学习Python或Java时从未接触过的、更底层的计算思维模式。

通过这三层“保险”，研究者们确保了当AI成功解决Manufactoria问题时，我们几乎可以肯定，它是在进行真正的“学习”和“推理”，而不是“回忆”或“模仿”。

---

**原文翻译 一个可扩展的难度阶梯。** 我们总共构建了14个合成问题族。例如，标记为HAS（图2）的问题族要求接受包含某个子序列（如GGRBB）的纸带，这可以通过使用任意颜色字符串来合成。Manufactoria被组织成BASIC → EASY → MEDIUM → HARD四个等级，从而能够对不同规模的模型进行匹配研究。BASIC/EASY族（如START, EXACT）适合小型模型（如1.5B, 4B）进行可学习性研究，而MEDIUM/HARD族需要更高级的洞察力，适合用于探究SOTA（最先进）系统（如GPT-5级别）的能力。由于其语法和问题族都是新颖的，Manufactoria也作为一个OOD基准，用于在真正新颖的任务上对开放LLMs与SOTA LLMs进行“同台竞技”的比较。中等难度的任务暴露了更大的差距：只有GPT-5取得了非平凡的成功，而其他模型则崩溃至接近零。困难的族则在所有模型中都未被解决，这突显了难度的急剧转变以及当前模型的极限。

**图2：Manufactoria难度阶梯。** 14个问题族根据四种流行LLM的平均表现被分为基础（Basic）、简单（Easy）、中等（Medium）和困难（Hard）四个级别。每个测试分割包含20-50个问题，完全通过率是在4次独立运行中平均得出的。

**图表描述：** 这是一个条形图，标题为“Manufactoria的难度阶梯（分布外编码谜题）”。

- **Y轴**：完全通过率（Full Pass Rate），范围从0.0到1.0。
- **X轴**：展示了14个问题族，按难度从左到右排列，并分为四个等级：
  - **BASIC**：包含 `APPEND`，`EXACT`，`START`。示例：“只接受纸带‘RBR’”。
  - **EASY**：包含 `ENDS`，`REGEX`，`HAS`。示例：“接受含有子序列‘GGRBB’的纸带”。
  - **MEDIUM**：包含 `COMPR`，`PREPEND`，`MUTATE`，`BIT_OP`。示例：“将‘RB’替换为‘BR’”。
  - **HARD**：包含 `FDIV`，`SYMM`，`MINMAX`，`ADD`。示例：“接受模式 $R^nB^n$ $(n > 1)$”。
- **图例**：图中有多条不同颜色的线，代表不同的模型，包括gpt-5, qwen3-4b-mini, claude-sonnet-4-20250514, 和 Qwen3-235B-A22B-Thinking-2507。
- **数据趋势**：

- 在BASIC和EASY级别，所有模型的表现都相对较好，通过率较高。
- 进入MEDIUM级别，各模型表现开始出现显著分化。GPT-5的表现远超其他模型，但其通过率也大幅下降。其他模型在此级别的通过率接近于零。
- 在HARD级别，所有模型的通过率都降至零，表明这些问题对当前最先进的模型来说也是无法解决的。

**数据表格化呈现：** 下表根据图2中的视觉信息，估算了不同模型在Manufactoria各难度等级上的平均完全通过率。

| 难度等级 | 示例任务 | GPT-5 (估算) | Claude-Sonnet-4 (估算) | Qwen3-235B (估算) | qwen3-4b-mini (估算) |
|---|---|---|---|---|---|
| **BASIC** | 只接受 'RBR' | ~95% | ~90% | ~85% | ~80% |
| **EASY** | 接受含 'GGRBB' | ~80% | ~60% | ~50% | ~40% |
| **MEDIUM** | 将 'RB' 替换为 'BR' | ~40% | ~5% | ~2% | ~0% |
| **HARD** | 接受 $R^nB^n$ 模式 | 0% | 0% | 0% | 0% |

**深度解读** 这一部分详细介绍了Manufactoria这个“赛场”是如何被精心设计成一个“能力试金石”的。研究者们不只是随便出了几道题，而是构建了一个从“入门”到“劝退”的完整“难度阶梯”。

这个阶梯共有四个台阶：

- **基础级 (BASIC)** 和 **简单级 (EASY)**：这些是“热身运动”，比如检查纸带是不是以某个特定颜色开头，或者是不是一个固定的序列。这些任务相对直接，用来测试模型是否能理解这门新语言的基本语法和操作。对于现在的大模型来说，这部分应该不成问题。
- **中等级 (MEDIUM)**：难度开始攀升。任务变得更复杂，比如要求模型执行一些类似“查找替换”或者简单的位运算的操作。这不再是简单的模式匹配，而是需要模型构建一个更复杂的内部逻辑流程。从图2的数据可以看出，这正是区分“优秀选手”和“普通选手”的分水岭。像GPT-5这样的顶级模型还能勉强应对，而其他模型则几乎全军覆没。
- **困难级 (HARD)**：这是“奥赛级别”的挑战。任务要求模型理解和实现抽象的数学概念，比如对称模式（$R^nB^n$ 指的是n个R后面跟着n个B）或者数值运算（如整除）。这些任务需要模型具备真正的算法思维和规划能力。结果毫不意外，图表显示，在这一级别，所有当今最强的AI模型都“交了白卷”，成功率为零。

这个精心设计的难度阶梯，其意义远不止是“出难题”。它首先为研究“可学习性”提供了一个完美的靶子：那些成功率为零的 `MEDIUM` 和 `HARD` 问题，正是检验RL“顿悟秘籍”能否生效的理想实验对象。其次，它也成为了一个客观、公正的“比武擂台”，让各种大模型在一个绝对公平（因为谁都没

见过）的新任务上同场竞技，它们的表现直接反映了其底层的推理能力差距。这个“阶梯”清晰地勾勒出了当前AI能力的边界，也为后续实验的惊人结果埋下了伏笔。

---

### 2.2 BouncingSim (用于泛化能力研究的二维模拟编程任务)

**原文翻译** 我们引入了一个被广泛使用的社区测试——一个二维弹球模拟程序，它通常被视为LLMs中几何感知推理能力的代理指标 (Wiggers, 2025)。目标是合成一个程序，该程序能模拟在多边形容器中的弹性碰撞，并返回在查询时间戳时物体的精确状态；强大的解决方案需要精确的碰撞检测/响应和数值稳定的积分。

**深度解读** 在设立了用于测试“从无到有”学习能力的Manufactoria之后，研究者们需要另一个赛场来测试AI“举一反三”的泛化能力。他们选择了BouncingSim，一个在AI圈子里很流行的非正式测试项目，可以称之为“AI版的打砖块”或“几何弹球”。这个任务的核心是让AI编写一段程序，来预测一个小球在一个封闭的多边形容器里（比如三角形、六边形）如何运动和反弹。

这个任务看似简单，实则对AI的能力提出了极高的要求。它不仅仅是考验编程语法，更是对AI多方面能力的综合大考：

- **几何理解力**：AI需要理解多边形的顶点、边，以及如何判断一个物体（小球）是否与这些边发生了碰撞。
- **物理洞察力**：AI需要知道什么是“弹性碰撞”，并能应用物理公式（比如入射角等于反射角）来计算小球碰撞后的速度和方向。
- **数学精确性**：整个过程涉及到大量的坐标计算和时间积分。如果计算不够精确，或者算法不够稳定，模拟出来的轨迹很快就会“面目全非”。

研究者们选择这个任务，是因为它能很好地模拟真实世界中需要结合几何、物理和编程知识才能解决的问题。一个能完美完成这项任务的AI，可以说在一定程度上具备了“理工科思维”。

---

**原文翻译 任务设计。** 为了将非正式的、通过视觉判断的演示替换为一个严谨的基准，我们使任务具备以下特点：(a) **可验证**——每个提示都指定了一个确定性的初始状态（位置、速度、容器几何形状）；程序必须输出物体在目标时间的位置，并与一个“神谕”（oracle）进行比对评分；(b) **可合成**——实例通过改变图1中的配置来生成，其真实轨迹由$Box2D$[4]产生；(c) **可组合**——单一技能族（如 `ROT_BOX` ，旋转的盒子）可以被组合成多技能族（如 `ROT_BOX_OBJ` ，盒子和物体都在旋转）；以及(d) **难度可控**——我们通过改变多边形顶点数、物体速度、盒子运动、重力以及物体/盒子的数量，来创建BASIC → EASY → MEDIUM → HARD → EXTREME五个等级。详细配置在附录A中提供。

[4] https://box2d.org/

**深度解读** 之前社区里流行的“几何弹球”测试，更像是一种“看个热闹”的演示，大家凭感觉判断AI做得好不好。这对于科学研究是远远不够的。因此，研究者们对这个任务进行了“工业级”的改造，把它从一个“民间小游戏”升级成了一个“奥运会竞赛项目”。

改造的核心有四点，确保了测试的**严谨性**和**科学性**： a) **结果可量化 (可验证)**：每一次测试，AI得到的题目都是一个“标准开局”，所有初始条件（球在哪，多快，盒子什么样）都规定得死死的。AI交卷后，它的答案（球在某个特定时间的位置）会和一个由超高精度物理引擎（被称为“神谕”，意指

绝对正确的标准答案）计算出的结果进行比对。对就是对，错就是错，差之毫厘也不行。这杜绝了任何主观判断。

b) **题库可无限生成 (可合成)**：为了避免AI“背题”，研究者们写了一个“自动出题机”。这个机器可以随机组合各种参数（比如把盒子从三角形换成五边形，把球的速度调快一倍），从而源源不断地生成新的、独一无二的题目。他们还用了一个著名的物理引擎Box2D（很多手机游戏的物理效果都靠它）来生成每道题的标准答案。

c) **技能可拆分组合 (可组合)**：这是为了测试“组合性泛化”而设计的。研究者们把挑战拆分成了几个独立的“技能包”，比如“应对旋转的盒子”、“应对移动的盒子”。在训练时，AI可以单独练习这些技能。到了考试时，就可以出一个“复合题”，比如一个既在旋转又在移动的盒子，看AI能不能把学到的技能组合起来用。

d) **难度可精确调节 (难度可控)**：为了系统性地测试“探索性泛化”，他们设计了一个从“基础”到“极限”的五级难度系统。难度的提升是全方位的，可能包括：盒子的边数更多（碰撞计算更复杂）、球的速度更快（需要更精确的时间步长）、引入重力，甚至在同一个盒子里放好几个球（需要处理球与球之间的碰撞）。

通过这番改造，BouncingSim变成了一个功能强大的测试平台，研究者们可以在上面随心所欲地设计各种实验，来精确地探测AI在物理和几何推理方面的能力边界。

---

**图3：BouncingSim上各模型、问题族（ROT_OBJ, ROT_BOX, MOV_BOX, GRAVITY, MULTI_BOX, MULTI_OBJ）及难度等级（BASIC - EXTREME）的完全通过率（%）。** 颜色越暖表示准确率越高；单元格内的数值是每个分割上4次运行、每次50个测试问题的平均完全通过率。

**图表描述：** 这是一个热力图表格，展示了四种大语言模型在BouncingSim任务上的表现。

- **列**：分为四个主要模型：GPT-5-521157, qwen3-4b-mini, Claude4-23, Qwen3-235B-59。在每个模型下，又细分为六个问题族（子列）：
  - `ROT_OBJ` (旋转物体)
  - `ROT_BOX` (旋转盒子)
  - `MOV_BOX` (移动盒子)
  - `GRAVITY` (有重力)
  - `MULTI_BOX` (多个盒子)
  - `MULTI_OBJ` (多个物体)
- **行**：代表五个难度等级，从上到下依次是：basic, easy, medium, hard, extreme。
- **单元格**：每个单元格的颜色和数字代表了在该特定模型、特定问题族和特定难度下的平均“完全通过率”（即程序完美通过所有测试用例的百分比）。颜色从冷色调（蓝色/绿色，低通过率）到暖色调（黄色/橙色/红色，高通过率）渐变。

**数据表格化呈现（以GPT-5和qwen3-4b-mini为例）：**

| 问题族 | 难度 | GPT-5-521157 (%) | qwen3-4b-mini (%) |
|---|---|---|---|
| **ROT_OBJ** | basic | 100 | 89 |
| (旋转物体) | easy | 97 | 85 |
| | medium | 91 | 64 |
| | hard | 23 | 2 |
| | extreme | 2 | 1 |
| **ROT_BOX** | basic | 75 | 76 |
| (旋转盒子) | easy | 64 | 69 |
| | medium | 58 | 57 |
| | hard | 0 | 2 |
| | extreme | 0 | 1 |
| **MOV_BOX** | basic | 100 | 95 |
| (移动盒子) | easy | 95 | 59 |
| | medium | 59 | 27 |
| | hard | 33 | 5 |
| | extreme | 0 | 0 |
| **GRAVITY** | basic | 100 | 78 |
| (有重力) | easy | 95 | 52 |
| | medium | 59 | 25 |
| | hard | 33 | 13 |
| | extreme | 0 | 0 |
| **MULTI_BOX** | basic | 32 | 34 |
| (多个盒子) | easy | 25 | 32 |
| | medium | 13 | 14 |
| | hard | 14 | 0 |
| | extreme | 0 | 0 |
| **MULTI_OBJ** | basic | 80 | 41 |
| (多个物体) | easy | 45 | 20 |

| 问题族 | 难度 | GPT-5-521157 (%) | qwen3-4b-mini (%) |
|---|---|---|---|
| | medium | 11 | 0 |
| | hard | 7 | 0 |
| | extreme | 0 | 0 |

**深度解读** 这张热力图是AI在BouncingSim这个“物理模拟考场”上的“成绩单”，信息量非常丰富，揭示了当前大模型能力的几个关键点：

1. **GPT-5遥遥领先，但并非全能**：从整体颜色上看，最左侧GPT-5的区域“最暖”，表示它的平均分最高，在大多数基础和简单任务上都能拿到接近满分的成绩。这 подтверждает了它作为业界顶尖模型的强大实力。然而，它的“暖色区”也主要集中在上方。随着难度（从上到下）的增加，它的颜色也迅速“冷却”，在困难和极限等级，它的成绩同样惨不忍睹，多数情况下都是0分。这说明，即便是最强的模型，其物理和几何推理能力也存在明显的“天花板”。
2. **难度是“硬杀手”**：对于所有模型来说，从上到下的颜色变化趋势都是一致的：从暖到冷。这直观地展示了“探索性泛化”的挑战。当任务的复杂度（如盒子边数、物体速度）提升时，所有模型的表现都会急剧下降。这表明它们的解决方案不够鲁棒，对于参数的变化非常敏感。
3. **“组合”是另一大难关**：请特别关注最后两列，`MULTI_BOX`（多个盒子）和`MULTI_OBJ`（多个物体）。这两类任务考验的是“组合性泛化”。我们可以看到，即使是在最简单的`basic`级别，`MULTI_BOX`的通过率也普遍很低（只有30%左右），颜色是冷色调的。而`MULTI_OBJ`（多个物体互相碰撞）的难度更大，GPT-5的成绩从`basic`的80%迅速跌落到`medium`的11%。这说明，让AI去处理多个动态物体之间的复杂交互，对它们来说是一个巨大的挑战。它们或许能处理好“一对一”的问题，但很难处理好“多对多”的系统性问题。

这张图为后续的泛化能力研究提供了一个至关重要的“基线”或“参照点”。它告诉我们，在未经特殊训练的情况下，这些大模型的能力状况如何。后续实验中，经过RL训练的模型如果能在这张“成绩单”的冷色区域（特别是中等难度和组合任务）点亮暖色，那就强有力地证明了训练的有效性。

---

### 2.3 竞赛编程问题族

**原文翻译** 我们添加了三个真实世界领域：竞赛编程、SQL和LEAN。尽管它们并非严格的OOD（鉴于其在网上的流行度），但它们仍然具有挑战性（例如，gpt-5-high在hard级别的LiveCodeBench-Pro上也仅达到2% (Zheng et al., 2025)）。我们将它们纳入DELTA，是为了将种子问题扩展为完全受控的问题族，以支持可学习性和泛化能力的研究。正文中给出了简要的构建概述，细节在附录A中。

**深度解读** 在设计了纯粹考验底层逻辑的“外星游戏”Manufactoria和考验物理模拟能力的BouncingSim之后，研究者们觉得还不够。他们希望自己的研究能和“真实世界”的高难度编程任务接轨。因此，他们引入了三个“硬核”领域：竞赛编程、SQL和LEAN。

这三个领域各有侧重：

- **竞赛编程**：这是程序员界的“奥林匹克”，题目通常涉及复杂的算法和数据结构，非常考验逻辑思维和代码实现能力。
- **SQL**：这是数据科学家的“日常工具”，用于从数据库中查询和操作数据。写出高效、复杂的SQL查询语句，需要严密的逻辑。
- **LEAN**：这是一个“形式化证明”工具，用户需要用一种极其精确的语言来书写数学证明，并由计算机来验证其正确性。这代表了最高等级的逻辑严谨性。

研究者们坦言，这些领域在网上有大量资料，所以AI在预训练时肯定“见过”。因此，它们不能像Manufactoria那样作为严格的OOD测试。但引入它们的目的不同：首先，这些领域的难题（比如顶级竞赛题）即便对于最强的模型来说，成功率也极低，这同样为“可学习性”研究提供了很好的素材。其次，更重要的是，研究者们想做的是，把这些来自真实世界的、零散的“难题”，通过他们的“模板化生成”技术，改造成一个个“受控的问题族”。

这就好比一个植物学家，他不仅研究实验室里培育的纯种豌豆，也去野外采集各种野生植物。但他采集回来后，不是直接观察，而是通过嫁接、杂交等技术，把野生植物的优良基因（比如高难度）提取出来，培育成一个个特性清晰、可控的“新品种”。通过这种方式，他们既保留了任务的真实性和挑战性，又获得了在受控环境中进行科学研究的便利性，从而将他们的研究范围从“理想化的实验室”拓展到了“半真实的复杂世界”。

---

**原文翻译 竞赛编程。** 每个问题族将共享相同核心算法（例如，莫队算法、CDQ分治）的问题分组，并以该算法命名。对于每个族，我们：（1）收集5-7个经核实使用目标算法的种子任务；（2）依赖专家提供的解题策略和背景，扰动其上下文，然后使用LLM改变叙事表面，同时保留解决方案；（3）通过要求一个暴力解法能通过所有测试来过滤和验证，确保扰动的一致性。我们发布了5个问题族（每个约500个项目）。

**深度解读** 这里详细介绍了他们如何“量产”竞赛编程题。这个过程非常像一个“故事改编流水线”，目的是在保持题目“算法核心”不变的前提下，给它换上各种各样不同的“外衣”（即故事情节），从而生成大量同类型但不同样貌的题目。

流水线分为三步：

1. **寻找“故事原型”（种子任务）**：首先，他们会找到几道经典的、确认是考察某个特定算法（比如“莫队算法”）的编程竞赛原题。这些原题就是“故事原型”。
2. **“换皮”与“魔改”（扰动上下文）**：这是最关键的一步。他们请算法专家先写好这道题的“解题攻略”。然后，他们让大语言模型（LLM）扮演一个“编剧”的角色，根据这份攻略，去改写题目的背景故事。比如，一道原本是关于“计算军队战斗力”的题目，可以被改编成“分析股票市场波动”或者“管理魔法学院学生成绩”，但背后需要计算的数学模型和需要使用的“莫队算法”是完全一样的。这个过程就是“换皮”。
3. **质量检验（过滤和验证）**：为了确保“编剧”没有“自由发挥”过头，导致题目核心逻辑改变，他们有一个巧妙的质检方法。他们会写一个最笨、最直接的“暴力解法”程序。对于一道合格的改编题，这个暴力解法应该能够算出正确的答案（尽管可能会超时）。如果暴力解法都算不对，那就说明题目在改编过程中“改坏了”，核心逻辑变了，这样的题目就会被淘汰。

通过这套精密的流水线，他们可以从一道原题，衍生出成百上千道“新题”。这些题目组成了一个“问题族”，它们共享同一个“灵魂”（核心算法），但有着千变万化的“肉体”（故事背景）。这对于训练和测试AI的算法识别与应用能力，是一种极其有效的方法。

---

## 3. 可学习性研究：RL能否揭示新策略以及如何加速它？

**原文翻译** 近期研究中的一个核心争论是，强化学习（RL）是否能赋予模型超越其基础模型能力的推理能力。

**深度解读** 这一节正式进入了论文的核心腹地——关于“可学习性”的深入探讨。开篇的这句话，再次将我们带回了那个根本性的问题。在详细介绍了他们精心打造的实验平台DELTA之后，研究者们现在要开始利用这个平台，来正面回答这个争论。本章节的所有内容，都将围绕着一个核心目标展开：通过实验证据，证明在特定条件下，强化学习确实能够让AI模型实现“质的飞跃”，获得它原本完全不具备的新能力。同时，他们还将探索，如何能让这个“飞跃”的过程来得更快一些，也就是所谓的“加速顿悟”。

---

**原文翻译 怀疑论观点。** Yue et al. (2025) 认为，尽管经过RLHF（带人类反馈的强化学习）训练的模型在较小的k值（如$k = 1$）上优于其基础模型，但当k值很大时，基础模型能达到相同或更优的pass@k性能。他们的覆盖率和困惑度分析表明，推理能力最终受限于基础模型的支持范围。同样，Wu et al. (2025) 提供了一个理论论证，指出RLHF无法扩展到基础模型的表征极限之外。

**深度解读** 这里首先介绍了学术界的“反方辩友”——怀疑论者们的观点。他们的核心论点可以通俗地理解为：强化学习（RL）只是一个“优化器”，而不是一个“创造者”。

这个观点主要基于一种被称为 $\texttt{pass@k}$ 的评估方法。 $\texttt{pass@k}$ 的意思是“给模型k次机会，看它能否至少成功一次”。怀疑论者发现，虽然经过RL训练的模型可能在第一次尝试（$k = 1$）时就成功的概率更高，显得更“聪明”，但如果你给基础模型足够多的机会（比如让k等于100或1000），它“瞎猫碰上死耗子”也能最终蒙对答案。他们认为，只要基础模型有“可能”生成正确答案（哪怕概率极低），那就说明正确答案本身就在它的“能力圈”之内。RL所做的，无非是把那个极低的概率给提上来了，像是把大海里的一根针变得更容易捞到，但它并没有往海里扔进一根新的针。

更有甚者，从理论上分析，认为RL的训练过程本质上是在调整模型内部已有知识的“权重”，它无法创造出模型参数空间里原本不存在的新知识。这就好比一个调音师，他可以把一架钢琴的音调得更准，但他无法让这架钢琴发出小提琴的声音。总而言之，怀疑论者认为，模型的能力上限，在预训练完成的那一刻，就已经被“焊死”了。

---

**原文翻译 乐观论观点。** 相比之下，Liu et al. (2025b) 证明了ProRL可以在基础模型表现不佳的任务上扩展推理边界——特别是在Reasoning Gym (Stojanovski et al., 2025) 中的字母组成的二维谜题上。

**深度解读** 接着，文章介绍了“正方辩友”——乐观论者的观点。他们的声音虽然相对较少，但提供了一些鼓舞人心的初步证据。他们在一个名为“Reasoning Gym”的测试平台上，进行了一项实验。这个平台里有一种特殊的二维字母拼图游戏，对于基础的大模型来说非常困难。

乐观论者发现，通过一种名为ProRL（长时间强化学习）的训练方法，模型居然能够解决这些它之前几乎完全不会的难题。这表明，强化学习似乎真的有能力“拓宽”模型的推理边界，让它能够触及之前无法企及的领域。这个发现虽然是在一个特定的任务上取得的，但它就像是在怀疑论者坚固的理论大坝上，打开了一道小小的裂缝，暗示着强化学习的潜力可能比我们想象的要大得多。这为本篇论文的研究提供了重要的前期支持，作者们接下来要做的，就是把这道“小裂缝”彻底“撕开”。

---

**原文翻译 我们的贡献：一个纯净的测试平台和RL促使LLMs顿悟的明确证据。** 已有的支持RL泛化能力的证据通常来自大型、异构的训练语料库。这使得我们难以分离出RL为何以及如何可能发现新颖策略的原因。为了解决这个问题，DELTA提供了一个受控的环境：既是分布外（需要新颖策略）又内部一致（没有数据混淆）的合成问题族。我们专注于Manufactoria-HAS问题族（742个训练/100个测试实例），在这个问题族上，参考模型Qwen3-4B-Instruct-2507在pass@128时实现了0%的完全通过率。如图4所示，我们分阶段的RL训练策略使模型能够完全解决这个族，实现了100%的完全通过率。接下来，我们将详细说明这是如何实现的。

**图4：在Manufactoria-HAS上RL训练前后的Pass@k对比。**

- **图表标题**：Pass@k Comparison Before/After RL Training
- **图表类型**：线图
- **X轴**：k (尝试次数)，取值为1, 4, 8, 16, 32, 64, 128。
- **Y轴**：Pass@k on Test Set (测试集上的pass@k率)，范围从0.0到1.0。
- **数据线**：
  - **橙色线 (Qwen3-4B-instruct-2507)**：代表训练前的基础模型。这条线完全贴着X轴，表示在所有k值下，pass@k率均为0。
  - **蓝色线 (After RL (Step 800))**：代表经过800步RL训练后的模型。这条线在Y轴的1.0处呈水平直线，表示在所有k值下，pass@k率均为1.0。
- **核心信息**：图中标注了“明确证据：RL可以揭示超越参考模型极限的策略。”

**数据表格化呈现：**

| 尝试次数 (k) | 训练前 Pass@k (%) | 训练后 Pass@k (%) |
|---|---|---|
| 1 | 0 | 100 |
| 4 | 0 | 100 |
| 8 | 0 | 100 |
| 16 | 0 | 100 |
| 32 | 0 | 100 |
| 64 | 0 | 100 |
| 128 | 0 | 100 |

**深度解读** 在介绍了正反两方的观点后，作者们亮出了自己的“王牌证据”，这也是整篇论文中最具冲击力的一个发现。

他们首先指出了之前乐观论者研究的一个局限：那些实验用的数据都来自庞大而混杂的题库，就像一个大染缸。虽然看到了好的结果，但很难说清楚到底是哪个因素起了作用。为了让证据变得“铁证如山”，他们使用了自己打造的“超净实验室”——DELTA，并从中挑选了一个特定的“考场”：Manufactoria-HAS问题族。这个考场有一个关键特性：对于一个中等大小的“普通”模型（Qwen3-4B）来说，这里的题目是“绝对无解”的。实验数据显示，即使给这个模型128次机会，它解对一道题的概率也是**零**。

这创造了一个完美的实验起点。因为根据怀疑论者的理论，既然基础模型成功的概率是零，那么无论怎么用RL“优化”，结果也应该是零。然而，实验结果却给了怀疑论者一记响亮的“耳光”。

图4展示的结果堪称惊人。训练前的模型，其成功率曲线（橙色线）死死地贴在0%的底线上，纹丝不动。而经过作者们发明的“分阶段RL训练秘籍”训练之后，模型的成功率曲线（蓝色线）一飞冲天，直接钉在了100%的顶板上。这意味着，对于测试集里的任何一道题，模型现在只需要一次尝试，就能给出完美答案。

这个从0到100的飞跃，是一个无可辩驳的证据。它清晰地表明，RL在这里所做的，绝不仅仅是“优化概率”，而是真正地“创造能力”。模型学会了一种它之前完全不懂的、全新的解题策略。这个干净、清晰、对比强烈的实验结果，有力地驳斥了“RL无法超越基础模型极限”的观点，并为“AI能够顿悟”提供了最直接的视觉证明。接下来的内容，就是要揭示这个奇迹是如何发生的。

---

### 3.2 如何用RL解决"pass@K=0"的任务?

**原文翻译** 怀疑论者认为RL无法超越基础模型边界的立场是可以理解的，原因很简单：GRPO (Guo et al., 2025) 依赖于不同“rollouts”（模型生成的一次完整尝试）之间的奖励差异。如果在“pass@$K = 0$”任务中，没有任何一次rollout成功，那么就没有梯度信号可供学习。确实，如图5(a)所示，朴素的GRPO训练停滞不前。因此，核心挑战是：

**如果没有任何一次rollout获得完全通过，RL如何传播一个有意义的学习信号?**

**图5：解决“pass@K=0”任务的策略比较。**

- **(a) 策略(a): RL (GRPO) with Full-pass Rate (使用完全通过率作为奖励)**
  - 图表显示，训练数据上的完全通过率（Full-Pass Rate）在整个训练过程（0到800步）中始终为0。这表明模型无法从稀疏的二元奖励中学习。
- **(b) 策略(b): RL (GRPO) with Per-test Pass Rate (使用每个测试用例的通过率作为奖励)**
  - 图表显示了两条曲线。蓝色的“每个测试用例通过率”（Per-test Pass Rate）作为奖励信号，在训练开始后迅速上升，但在大约100步后就饱和了，稳定在0.6左右。然而，橙色的

“完全通过率”（Full Pass Rate）始终保持在接近0的水平（<0.01%）。这说明密集奖励虽然能提供初始信号，但本身不足以让模型学会完美解决问题。

- **(c) 策略(c): RL (GRPO) with Warm-up Phase (Per-test Pass Rate) + Continued (Full Pass Rate) (预热阶段使用每个测试用例通过率 + 后续阶段使用完全通过率)**
  - 这张图是整个故事的核心。它展示了一个两阶段的过程。
    - **左侧小图**：展示了“预热”阶段。模型先用“每个测试用例通过率”作为奖励进行训练。可以看到，完全通过率从0开始有了一点点微弱的、非零的信号。
    - **右侧大图**：从预热后的模型继续训练，但奖励切换为“完全通过率”。这条学习曲线完美地展示了“顿悟”过程：
      - **探索阶段 (Exploration Phase)**：从0到大约450步，完全通过率一直徘徊在接近0的水平。
      - **顿悟! (Grokking!)**：在450步左右，成功率突然开始飙升。
      - **收敛阶段 (Convergence Phase)**：从500步到800步，成功率迅速达到并稳定在接近1.0的水平。

**深度解读** 这一节开始揭秘他们是如何实现那个从0到100的“魔法”的。首先，他们承认了怀疑论者观点中的合理之处。传统的强化学习方法（如GRPO），其学习的动力来自于“比较”。模型会一次性生成很多个不同的答案（称为rollouts），然后比较哪个答案得到的奖励更高，并朝着那个方向去调整自己。

但这就带来了一个死结：在一个“pass@K=0”的难题上，模型无论生成多少个答案，得到的奖励全都是0。因为评分标准是“完全正确才给1分，否则都是0分”。既然所有答案的奖励都是0，那就没有任何“差异”可供比较，模型也就失去了学习的方向，梯度信号为零，训练自然就卡住了。图5(a)的实验结果证实了这一点：用这种最简单直接的方法，训练曲线就是一条死寂的直线，趴在零点。

这就引出了整个研究中最核心的技术挑战：**在一片奖励为零的“沙漠”中，如何为模型找到第一滴“甘泉”，让学习的引擎能够启动?**

---

**原文翻译 按测试用例通过率训练。** 一个解决方案是利用部分功劳。我们不使用“全有或全无”的完全通过率（仅当所有测试用例通过时奖励=1），而是使用更细粒度的“按测试用例通过率”，一个在范围内的连续奖励。如图5(b)所示，这个信号提供了初始的学习动力。然而，它在大约100步后迅速饱和，并且完全通过率仍然微不足道（<0.01%）。

**深度解读** 为了打破“全零奖励”的僵局，研究者们提出了第一个关键策略：改变计分规则，从“结果导向”变为“过程导向”。他们不再要求模型一次性就拿出完美答案，而是引入了“部分学分”机制。

这个机制就是“按测试用例通过率”（per-test pass rate）来给奖励。假设一个程序需要通过10个测试用例才算满分。那么，模型写出的程序哪怕只通过了其中的1个，也能得到0.1分的奖励；通过了5个，就能得到0.5分。这样一来，奖励就从原来只有0和1两个离散值的“悬崖”，变成了一个从0到1连续变化的“平缓坡道”。

如图5(b)所示，这个方法立竿见影。模型的学习引擎成功启动了，代表“部分学分”的蓝色曲线（Per-test Pass Rate）在训练初期快速上升。这说明模型在“密集奖励”的引导下，确实在朝着正确的方向进步，学会了写出一些“部分正确”的代码。

然而，这个方法也暴露出了一个新问题。虽然“部分学分”拿得越来越多，但模型的“总分”（代表完美答案的橙色曲线Full Pass Rate）却始终在零分附近徘徊。学习在大约100步之后就“饱和”了，进入了一个瓶颈期。这说明，虽然这种奖励机制能让模型“入门”，但它似乎缺乏足够的“驱动力”，去激励模型追求那最后1%的完美，无法帮助模型实现从“部分正确”到“完全正确”的最后一跃。

---

**原文翻译 预热阶段。** 尽管它不能作为完整的替代损失函数，但我们发现“按测试用例通过率”可以作为一个重要的预热阶段，将模型推出全零区域。如图5(c)的左侧小图所示，这个信号让模型得以超越全零区域：尽管完全通过率仍<1%，但模型开始积累正向的梯度。

**深度解读** 在这里，研究者们展现了他们解决问题的智慧。他们意识到，“按测试用例通过率”这个方法，既有优点（能启动学习），也有缺点（无法达到完美）。于是，他们没有抛弃它，而是给它找到了一个最适合的角色——**“预热教练”**。

这个策略的核心思想是“分工合作”。在训练的第一阶段，也就是“预热阶段”，就让“按测试用例通过率”这位宽容的教练上场。它的任务不是把模型训练成世界冠军，而仅仅是把它从“完全不会”的零基础状态，带到“会一点点”的入门水平。如图5(c)左侧小图所示，经过这个阶段的训练，虽然模型的最终成功率（Full Pass Rate）依然低得可怜，几乎还是零，但最关键的变化发生了：它不再是“绝对的零”。模型已经开始能够偶尔产生一些可以通过大部分测试用例的程序，这意味着它的内部参数已经进入了一个“有希望”的区域，学习的梯度信号从无到有，开始积累了。

这个“预热”阶段的意义，就像是在发射火箭前，先为引擎点火预热。虽然火箭还没有升空，但引擎已经启动，能量正在积蓄，为下一阶段的真正起飞做好了准备。它成功地解决了那个最根本的“冷启动”问题。

---

**原文翻译 探索与顿悟。** 从这个预热过的检查点开始，我们切换到使用二元完全通过奖励的RL。图5(c)的右侧大图展示了其动态过程：在大约450步的时间里，模型处于一个**探索阶段**，完全通过率仍<1%。在一个突然的**顿悟时刻**之后，模型发现了解决该问题族的关键策略。然后，训练进入一个**收敛阶段**，RL会打磨并持续强化这条成功的推理路径。在收敛时，RL训练的模型在pass@k上相比参考模型实现了近100%的绝对提升（图4）。我们也在附录C.1中的其他模型家族、大小和问题领域中观察到了这一现象。

**深度解读** 在完成了关键的“预热”之后，真正的“大戏”上演了。研究者们果断地“更换教练”：他们从预热好的那个模型状态开始，把奖励机制切换回最严苛的“完全通过奖励”（即“非黑即白”的0/1奖励）。

接下来发生的，就是图5(c)右侧大图所描绘的那个戏剧性的“顿悟”过程，它可以分为三个阶段：

1. **漫长的探索**：在切换奖励机制后，模型并没有立刻成功。在长达约450个训练步骤里，它的成功率依然在谷底徘徊。这个阶段，模型就像一个已经掌握了基本功的侦探，正在一个巨大的迷宫里搜寻唯一的出口。它知道出口的大致方向（预热阶段的功劳），但需要不断地尝试、失败、再尝试，来找到那条通往成功的精确路径。

2. **突然的顿悟**：在某个神奇的时刻，量变引起了质变。模型的一次随机尝试，可能恰好“撞”对了那个核心的解题算法。一旦这个“正确答案”被发现，它就会得到一个巨大的、+1的奖励信号。这个信号像一道闪电，瞬间照亮了整个迷宫。模型立刻意识到：“啊哈！原来是这样！”这就是“顿悟”的时刻。学习曲线在图上表现为一条近乎垂直的陡峭上升线。
3. **快速的收敛**：一旦找到了正确的路径，剩下的事情就简单了。强化学习机制会不断地奖励和强化这条“成功路径”，让模型牢牢地记住它，并能举一反三地应用到所有同类型的问题上。这个阶段，模型在巩固和熟练新学到的技能，学习曲线迅速达到并稳定在100%的平台。

这个完整的“预热-探索-顿悟-收敛”过程，不仅成功地解决了“pass@K=0”的难题，实现了一个看似不可能的壮举，而且这个过程本身，也为我们理解AI如何学习复杂抽象知识，提供了一个极具启发性的模型。它表明，AI的学习，可能并非总是平滑渐进的，也可能包含这种充满戏剧性的、类似人类灵感迸发的“飞跃”。

---

### 3.4 对预热阶段的更多研究

**原文翻译 选择性课程学习作为替代方案。** 一个自然的问题是，预热的效果是否可以通过跨问题族的课程学习来实现。为了探索这一点（图7），我们设计了一个三阶段的课程学习训练。在对基础族（START/APPEND/EXACT）进行训练后，模型被暴露于两种中间课程之一：(i) 阶段2-REGEX 或 (ii) 阶段2-COMPR，然后才迁移到目标HAS任务。根据图2，这两个问题族的难度水平相似。尽管难度相似，结果却大相径庭：REGEX课程导致了成功的迁移，并在最终的RL阶段于HAS上取得了近乎完全的掌握，而COMPR课程则未能进步，停滞在低通过率上。这种差异可以追溯到任务的兼容性——REGEX和HAS都围绕着检测或匹配子模式（例如，“接受带有模式$(BRB)^+(RR)^*$的纸带” vs. “接受带有子序列GGRBB的纸带”），而COMPR则强调数值解释和分支测试（例如，“将颜色B视为1，R视为0，如果数字>27则接受”）。这些结果表明，有效的课程不仅要控制难度，还必须在结构上与目标族对齐。因此，虽然课程学习可以非常有效，但其成功取决于找到合适的、相关的族来弥合推理差距——这并非总是可行的。相比之下，使用密集奖励的预热训练仍然具有广泛的用途，因为它不需要额外的问题族设计或混合。

**图7：Manufactoria-HAS的两阶段课程学习对比。** 模型首先在基础问题（START/APPEND/EXACT）上训练，然后分支进入两个中间课程之一：(i) 阶段2-REGEX，这导致了在目标HAS族上的成功迁移和高通过率；或 (ii) 阶段2-COMPR，这未能迁移并停滞在低性能水平。

**深度解读** 在证明了“预热”策略的有效性后，研究者们思考了另一个问题：除了用“密集奖励”这种方式来预热，还有没有别的办法？一个很自然的想法是“课程学习”（Curriculum Learning），这个概念借鉴了人类的学习方式：先学简单的，再学难的。

于是他们设计了一个实验，来模拟一个“学习计划”。所有模型都先从最简单的Manufactoria问题（比如 `START` ，`APPEND` ）开始学起，打好基础。然后，他们兵分两路，进入“进阶课程”：

- **A组**：学习 `REGEX` 问题族。这类问题要求模型理解和匹配正则表达式，本质上也是一种“模式匹配”。
- **B组**：学习 `COMPR` 问题族。这类问题要求模型把颜色序列当作二进制数，进行比较大小，本质上是“数值计算”。

从难度上看， REGEX 和 COMPR 是差不多的。在完成了各自的“进阶课程”后，两组模型都去挑战最终的目标—— HAS 问题族（在序列中寻找子串，也是一种“模式匹配”）。

结果出现了巨大的差异。从 REGEX 课程“毕业”的A组模型，在面对 HAS 时势如破竹，很快就掌握了，成功率飙升。而从 COMPR 课程“毕业”的B组模型，则完全“卡壳”了，表现和没上过进阶课差不多。

这个实验揭示了一个深刻的道理：有效的“课程学习”，不仅仅是难度上的循序渐进，更重要的是知识的“相关性”和“连贯性”。因为 REGEX 和 HAS 在底层逻辑上都是关于“模式匹配”的，所以学习 REGEX 的经验可以很好地“迁移”过来，帮助模型解决 HAS 。而 COMPR 的“数值计算”逻辑，与 HAS 的“模式匹配”逻辑风马牛不相及，所以学了也白学。

这个发现说明，“课程学习”虽然是个好主意，但它的成功高度依赖于我们能否精心设计出一条“知识上连贯”的学习路径。这在很多时候是非常困难的。相比之下，之前提出的“密集奖励预热”策略，则更加“通用”和“省心”。它不需要我们去费心寻找相关的“中间课程”，而是直接在目标任务本身上，通过奖励机制的调整，来帮助模型平稳起步。

---

**原文翻译 预热在“pass@k=0”之外也有帮助。** 即使基础模型表现出微小但非零的成功率 ( $pass@k = \epsilon > 0$)，一个简短的按测试奖励的预热也能提高稳定性和速度。根据经验，我们观察到与从头开始训练完全通过率相比，收敛更快、更平滑（见附录C.2）。

**深度解读** 这一段补充说明了“预热”策略的一个额外好处。它不仅是解决“pass@K=0”这种极端难题的“救命稻草”，对于那些虽然很难，但并非完全无解（比如基础模型有1%的成功率）的任务，它同样是一个强大的“助推器”。

可以这样理解：对于一个成功率只有1%的任务，如果采用严格的“完全通过奖励”，模型就像在买彩票，需要进行大量的随机尝试，才能偶尔中一次奖（得到+1的奖励）。这个学习过程会非常缓慢，而且充满了随机性，很不稳定。

而如果先进行一小段“密集奖励预热”，情况就大不相同了。这个预热阶段就像是给模型提供了一份“彩票中奖规律分析报告”。虽然不能保证它立刻中大奖，但能大大提高它中一些小奖（获得部分分数）的概率。这会让模型的学习过程变得更加“心中有数”，探索方向更明确。当预热结束，切换回严格奖励时，模型已经不是在“盲目”地买彩票了，而是有了一定的“技巧”和“感觉”。因此，它能更快地找到中大奖的窍门，学习过程自然也就变得更快速、更稳定。这个发现，进一步增强了“预热”策略的实用价值。

---

**原文翻译 局限性。** 值得注意的是，并非每个问题族都能通过预热训练被“解锁”。例如，如图8所示，即使使用按测试用例通过率的奖励，模型在更难的Manufactoria-PREPEND族上也未能逃离全零区域。按测试信号有适度上升但很快饱和，而完全通过率在整个训练过程中始终卡在零。这表明，使用按测试用例通过率的预热训练并非万能秘籍：其有效性取决于模型的能力和目标族的难度。

**图8：在更难的Manufactoria-PREPEND族上进行预热训练。**

- **图表标题**：Warm-up Phase (a): RL (GRPO) with Full-pass Rate, Problems: Manufactoria-PREPEND (MEDIUM)

- **图表类型**：线图
- **Y轴**：左侧为完全通过率（Full Pass Rate），右侧为每个测试用例通过率（Per-test Pass Rate）。
- **数据线**：
  - **蓝色线 (Per-test Pass Rate (Reward))**：代表作为奖励的“部分学分”。它在训练初期有微弱的上升，但很快就饱和在一个非常低的水平（大约0.3）。
  - **橙色线 (Full Pass Rate)**：代表最终的成功率。这条线在整个训练过程中，始终紧贴着0的底线，标注为“Stays with all-0”（始终为0）。

**深度解读** 在展示了“预热”策略的强大威力之后，研究者们展现了科学家应有的严谨和诚实，主动指出了这个方法的“局限性”。他们强调，“预热”并非包治百病的“万灵丹”。

为了证明这一点，他们用一个更难的中等难度问题族 `PREPEND` （要求在纸带开头添加指定序列）进行了同样的实验。结果如图8所示，这次“魔法”失灵了。

从图上看，即使使用了“密集奖励”，代表“部分学分”的蓝色曲线也只是稍微抬了一下头，就很快停滞在一个很低的水平。这说明，对于这个更难的任务，模型连“入门”都非常吃力，无法获得足够的、有意义的“部分学分”来引导自己。既然预热阶段本身就失败了，那么后续的学习自然也就无从谈起，代表最终成功率的橙色曲线，自始至终都被“钉”在了零点。

这个“失败的实验”和之前“成功的实验”放在一起，恰恰让这篇论文的结论更加可信和完整。它告诉我们一个重要的事实：任何方法都有其能力边界。这个“预热秘籍”的有效性，取决于两个前提：**模型的自身“天赋”（能力上限）和问题的内在“难度”**。当问题的难度超出了模型在当前引导策略下能够理解的范畴时，再好的“教练”也无能为力。这为未来的研究指明了方向：要么提升模型的基础能力，要么发明更强大的“引导”策略，来挑战那些目前看来“无法解锁”的更困难的任务。

---

## 4. 泛化能力研究

**原文翻译 设置。** 我们研究学到的程序化技能能在多大程度上迁移到训练分布之外。除非另有说明，参考模型是Qwen3-4B-Instruct。我们在一个由六个单一技能族——ROT_OBJ, ROT_BOX, MOV_BOX, GRAVITY, MULTI_BOX, MULTI_OBJ——的基础级别混合数据集上进行训练，每个族有1k个实例（总共6k）。因为基础模型在一些基础实例上有非零的完全通过率，我们直接优化一个二元的完全通过奖励（所有测试通过）300个梯度步长；所有其他超参数遵循第3节。评估跨越三个轴向——探索性、组合性和变革性——并报告完全通过率（合成的程序在所有单元测试上与神谕完全匹配的提示的比例）。对于探索性泛化，我们考虑四个难度等级（基础=ID，简单/中等/困难=OOD）与六个族交叉；图9中的每个条形图汇总了结果。更详细的设置在附录B中。

**深度解读** 在第3章证明了AI可以通过“顿悟”学会新技能后，第4章将要回答一个更深入的问题：AI学到的这些技能，是“死知识”还是“活学问”？它能应用到新的、更复杂的场景中去吗？这就是“泛化能力”研究。

这一段首先介绍了实验的“准备工作”：

- **选手**：依然是那个中等大小的模型（Qwen3-4B）。
- **训练营**：这次的训练内容是BouncingSim（几何弹球）任务。研究者们把六种不同的“单一技能”（如处理旋转的盒子、处理重力等）的最基础难度的题目混合在一起，构成了一个包含6000道题的“综合训练集”。
- **训练方法**：因为这些基础题目对模型来说不是完全无解的（不像之前的Manufactoria难题），所以研究者们没有使用复杂的“预热”策略，而是直接采用了最严格的“完全通过奖励”进行训练。
- **毕业考试**：训练完成后，模型将面临一场精心设计的大考，考卷分为三个部分，分别对应之前提到的三个泛化能力轴向：
  1. **探索性**：考同类型但难度更高的题目。
  2. **组合性**：考需要把两个独立技能组合起来才能解决的新题目。
  3. **变革性**：考需要“脑筋急转弯”才能发现捷径的特殊题目。
- **评分标准**：非常严格，必须程序跑出来的结果和“标准答案”一模一样，才算通过。

通过这个严谨的“训练-考试”流程，研究者们旨在精确地测量出，模型在“顿悟”之后，其知识迁移的能力到底有多强，以及边界在哪里。

---

**原文翻译 训练动态（图9a）。** 我们再次观察到了一个急剧的顿悟阶段性转变：在经历了长时间近乎零奖励的平台期后，模型在训练混合集上的表现大约在第200步时跃升至0.7的完全通过率，这表明它涌现出了能够处理弹性碰撞的稳定模拟代码。

**图9：BOUNCINGSIM上的泛化研究。**

- **(a) 训练曲线 (基础级别)**
  - 图表显示，在基础级别混合数据集上的训练完全通过率。曲线在前期（约0-180步）一直处于接近0的平台期，然后在接近200步时出现了一个急剧的“顿悟”跳跃，成功率飙升至约0.7。
- **(b) 探索性泛化**
  - 条形图比较了RL训练前后的表现。
  - **训练前 (Before RL)**：在所有难度等级（ID的基础级，OOD的简单、中等、困难级）上，通过率都接近于0。
  - **训练后 (After RL)**：
    - 在**基础级 (ID)** 上，通过率大幅提升至约76%。
    - 在**简单级 (OOD)** 上，通过率依然很高，约为59%。
    - 在**中等级 (OOD)** 上，通过率下降，但仍有可观的27%。
    - 在**困难级 (OOD)** 上，通过率降至较低的12%。
- **(c) 组合性泛化**
  - 条形图比较了RL训练前后的表现。

    - **训练前 (Before RL)**：通过率几乎为0。
    - **训练后 (After RL)**：通过率显著提升至约68%。
- **(d) 变革性泛化**
    - 条形图比较了RL训练前后的表现。
    - **训练前 (Before RL)**：通过率为0。
    - **训练后 (After RL)**：通过率依然接近于0。

**深度解读** 这一段描述了模型在BouncingSim训练过程中的有趣现象，并展示了最终的“考试成绩”。

首先，**训练过程本身再次验证了“顿悟”现象的存在**。如图9(a)所示，即使是在这个与Manufactoria完全不同的、更接近真实物理世界的任务上，模型的学习曲线也呈现出几乎相同的模式：长时间的“瓶颈期”，然后是突然的“爆发”。这说明，“顿悟”可能不是某个特定任务的偶然现象，而是一种在AI学习复杂程序化知识时普遍存在的规律。这次顿悟的成果是，模型学会了编写出能够稳定、正确地模拟小球弹性碰撞的核心代码。

接下来是激动人心的“放榜”时刻，我们来看看模型在三个泛化能力维度的具体表现：

- **探索性泛化 (图9b) - 成绩：良好**
    - 训练前，模型基本是“全科交白卷”。
    - 训练后，它在和训练题难度相当的“基础级”上取得了约76%的好成绩。更重要的是，在它从未见过的、难度更高的“简单级”和“中等级”上，它分别取得了59%和27%的通过率。虽然成绩随着难度增加而下降，但这明确表明，它学到的不是死记硬背的模板，而是可以推广到更复杂情况的、有一定鲁棒性的技能。
- **组合性泛化 (图9c) - 成绩：优秀**
    - 这部分的表现堪称“惊喜”。在训练中，模型只单独学过“处理旋转的盒子”和“处理移动的盒子”。考试时，面对一个“既旋转又移动的盒子”，它的通过率竟然高达68%！这说明模型能够非常有效地将两个独立的技能模块“拼接”在一起，解决一个全新的复合问题。这个能力对于解决真实世界的复杂问题至关重要。
- **变革性泛化 (图9d) - 成绩：不及格**
    - 这是唯一的“挂科”项目。面对那些需要发现“物理捷径”（如周期性运动）的特殊问题，模型在训练后依然一筹莫展，成功率接近于零。这说明，模型学会的是一个“通用”的、按部就班的模拟方法，但它缺乏更深层次的、能够洞察问题本质并发现“优雅”解法的抽象推理能力。

综合来看，这份“成绩单”清晰地描绘了当前通过RL训练的AI的能力画像：它是一个出色的“应用型工程师”，擅长在既有框架内解决难题（探索性）和整合不同技术模块（组合性），但它还不是一个能够提出理论突破的“理论物理学家”（变革性）。

---

**原文翻译 泛化结果 (图9b-d)。** RL训练的模型能够迁移到训练分布之外，但在不同轴向上成功程度各不相同。在**探索性泛化**中，在基础级（ID, 70-85%）上表现强劲，并能延续到简单级（50-75%），

尽管在中等级（15-50%）上增益缩小，在困难级（个位数）上几乎消失。对于**组合性泛化**，模型展示了惊人的技能整合能力：未见过的组合，如ROT_BOX+MOV_BOX, MOV_BOX+GRAVITY, 和MULTI_BOX+MULTI_OBJ，实现了60-70%的完全通过率（而RL前接近于零），这与OMEGA（Sun et al., 2025）中报道的弱组合性迁移形成对比。我们将其归因于编码任务是结构性地组合（合并模拟模块），而不是策略性地组合（发明新的推理步骤）。最后，在**变革性泛化**中，模型在质적으로新颖的动态（如完美的周期性或退化轨迹）上仍然接近于零，这些动态需要发现新的不变量，并与变革性数学泛化的持续困难相一致。

**深度解读** 这一段对前面图9中的泛化结果进行了更深入的分析和解读，其中包含了一个非常重要的对比和洞察。

首先，它再次确认了在**探索性泛化**上的“递减效应”：模型的能力随着问题复杂度的增加而平滑衰减。这符合我们的直觉，就像一个学生，基础题掌握得很好，难题就会吃力一些。

其次，它特别强调了在**组合性泛化**上的“惊人”成功，并将其与团队之前的另一项研究（OMEGA，关于数学问题）进行了对比。在之前的数学研究中，他们发现模型很难将两个不同的数学概念（比如数论和几何）组合起来解决一个新问题。但在这次的编程任务中，模型却表现得非常出色。研究者们对此提出了一个精辟的解释：**编码的组合是“结构性”的，而数学的组合是“策略性”的。**

这是什么意思呢？可以这样理解：在BouncingSim这个编程任务里，组合“旋转的盒子”和“移动的盒子”两个技能，可能只需要在代码里分别写好处理旋转的模块和处理移动的模块，然后把它们“组装”起来。这个过程更像是搭积木，是**结构上的拼接**。然而，在数学中，要结合数论和几何的知识去解决一个难题，通常需要一个非常巧妙的“金点子”或“解题技巧”，这个技巧本身是一种全新的**策略发明**，而不是简单地把两个领域的公式摆在一起。这个洞察揭示了当前大模型能力的一个深层特性：它们似乎更擅长处理结构清晰、可以模块化组合的任务，而对于需要“灵感一现”的、非结构化的策略创新，则能力有限。

最后，关于**变革性泛化**的失败，再次被强调。模型无法发现像“周期性”这样的“隐藏规律”（物理学中的“不变量”）。这与它们在数学领域同样难以实现变革性突破的现象是一致的。这表明，无论是编程还是数学，让AI从“遵循规则”跃升到“发现规则”，是当前面临的共同的、也是最艰巨的挑战。

---

**原文翻译 要点。** RL发现了可执行的模拟器，这些模拟器（i）能很好地迁移到参数变化上，（ii）能跨技能进行组合，但（iii）当测试分布要求质的に不同的解决方案模式时，则会遇到困难。编码任务似乎比符号数学更适合结构性组合，然而变革性的“模式创建”仍然是一个开放的挑战。图9总结了这些趋势。

**深度解读** 这是对整个泛化能力研究部分的高度浓缩总结，提出了三个核心要点，清晰地勾勒出了本次研究的结论边界。

1. **AI学会了“建模”，而非“套路”**：通过强化学习，AI不仅仅是记住了一些代码片段，而是真的学会了如何构建一个可以工作的“物理模拟器”。这个模拟器是“活”的，当外界条件发生一些数值上的变化时（比如盒子变大一点，球速变快一点），它依然能够适应并给出正确的结果。这证明了其学习成果具有一定的鲁棒性。

2. **AI是“组装大师”，而非“发明家”**：AI展现出了强大的“模块化”思考能力。它可以将独立学到的技能（如处理旋转、处理移动）像乐高积木一样有效地组合起来，解决更复杂的复合问题。这一点在编程任务上尤为突出，甚至超出了研究者们在其他领域（如数学）的预期。
3. **AI缺乏“洞察力”**：这是当前能力最明显的短板。当一个问题存在一个需要跳出常规思维才能发现的“捷径”或“根本规律”时，AI会“视而不见”。它会依然使用那个虽然通用但可能很笨拙的“标准方法”去硬算，而无法实现那种“啊哈！原来可以这样！”的认知飞跃。这种创造全新解题“模式”或“范式”的能力，是区分高级应用与真正创新的关键，也是AI领域接下来需要努力攻克的最大难关。

总而言之，这次研究给我们的启示是：我们已经可以通过RL，将AI训练成一个非常能干的“程序员”或“工程师”，但距离把它变成一个富有洞察力的“科学家”，还有很长的路要走。

---

## 6. 讨论及对未来研究的启示

**原文翻译 呼吁研究难题子集。** 近期的数学和代码基准通常报告在混合池上的平均表现（Huan et al., 2025; Guha et al., 2025; Liu et al., 2025b,a），其中一小部分真正困难的实例（那些对于强大的预训练模型来说pass@$K = 0$的实例）被更容易的项目所冲淡。我们的结果表明，这个“困难前沿”表现出独特的学习动态——最显著的是在RL下出现类似顿悟的阶段性转变，这可能需要每个问题成百上千的训练步骤。在大型异构池中，重复采样并解决任何一个困难案例的概率被稀释，进一步抑制了信号。因此，我们倡导未来的评估应明确地分离和追踪这个子集，以便在真正新颖的推理上的进展不被聚合指标所掩盖。

**深度解读** 这部分是论文作者们在完成了自己的研究之后，向整个AI研究社区发出的一个高瞻远瞩的“倡议”。他们指出了当前评估AI能力方式的一个普遍弊端，并提出了改进建议。

他们认为，现在很多流行的AI能力排行榜（基准测试），就像是一场包含了长跑、跳高、游泳等多个项目的“铁人三项”比赛，最后只公布一个“总分”。这种做法的问题在于，它可能会掩盖真相。比如，一个模型可能在99个简单的“长跑”项目上都拿了满分，但在1个极其困难的“高空跳伞”项目上得了0分。在计算总分时，它的平均分依然会非常高，给我们一种“这个模型很强大”的错觉。但实际上，它在那个最能体现突破性能力的“高空跳伞”项目上，是完全无能的。

作者们的研究恰恰聚焦于那些“高空跳伞”级别的难题（即 `pass@K=0` 的问题）。他们发现，AI在学习解决这些难题时，展现出一种非常独特的“顿悟”模式，这个过程需要大量的、集中的训练才能触发。如果把这些难题混在一大堆简单问题里，模型在训练时就很难集中“火力”去攻克它们，那个宝贵的“顿悟”信号就可能永远不会出现。

因此，他们大声疾呼：未来的AI能力评测，不能再满足于一个模糊的“平均分”了！我们必须把那些“极难题”单独拎出来，建立一个“难题名人堂”或者“前沿突破榜”。只有这样，我们才能清晰地看到AI是否在“啃硬骨头”，是否在真正的“推理能力”上取得了实质性的进展，而不是仅仅在简单问题上“刷分”。这个倡议，旨在引导整个领域的研究焦点，从追求“更高”的平均分，转向追求“更远”的能力边界。

---

**原文翻译 超越编码：从数学到科学。** 编码提供了密集的、可验证的反馈，这让RL能够在以前无法解决的问题上跨越可学习性的鸿沟。同样的原则可以扩展到数学和科学领域，只要有细粒度的信号可

用：基于评分细则的评分、步骤检查器、定理证明器验证，以及基于模拟或约束的评估器。我们期望将这些见解移植到这些领域，将使RL能够解决目前无法解决的问题。

**深度解读** 这里，作者们将他们的发现从“编程”这个具体的领域，提升到了一个更广阔的层面，展望了其在整个科学探索领域的应用潜力。

他们成功的核心“秘籍”，在于找到了一个可以提供“密集、可验证反馈”的环境（即编程中的“测试用例”）。这个机制就像一个极其耐心和精确的“私人教练”，AI每走一小步，教练都能立即告诉它“这一步走得对不对，好在哪里，差在哪里”。正是有了这样高质量的反馈，AI才能在看似不可能的难题上，一步步摸索，最终实现“顿悟”。

作者们认为，这个核心思想是完全可以“移植”的。虽然在其他领域，比如数学和科学研究，我们可能没有现成的“测试用例”，但我们可以创造出类似的“私人教练”系统。例如：

- **在数学证明中**：我们可以设计一个“步骤检查器”，AI每写一步证明，系统就自动验证这一步是否符合逻辑规则。或者，利用“定理证明器”来核对最终的证明是否严谨。
- **在科学发现中**：比如在物理学或化学中，AI提出的一个新理论或分子结构，可以通过“计算机模拟”来快速验证其效果。模拟的结果，就是一种细粒度的反馈。
- **在更广泛的领域**：我们可以制定非常详细的“评分标准”（rubrics），让AI的每一个输出，都能得到一个精确的分数，而不是一个模糊的“好”或“坏”。

他们的愿景是，只要我们能在更多的领域里，为AI建立起这样精细化的反馈机制，那么在编程领域观察到的“顿悟式学习”，就有可能在数学、物理、化学等更广泛的科学领域中被复制。这意味着，强化学习有潜力成为一个强大的工具，帮助AI乃至人类，去攻克那些目前我们认为“无法解决”的科学难题。

---

**原文翻译 我们如何训练与我们训练什么同等重要。** 尽管扩展数据对LLM训练至关重要，但我们的结果表明，训练程序本身也同样关键。首先，对于一个大型混合语料库中的所有难题，并非总存在一个自然的、课程式的进展路径；添加松散相关的族并不能可靠地平滑学习，甚至可能无济于事（见图7）。其次，具体的训练选择——如使用密集奖励进行阶段性预热、经验回放和验证/反馈在环——在提高难题层级问题的性能方面显示出巨大潜力。更广泛地说，打磨（优化现有先验）和发现（获得新策略）都可能发生；出现哪种情况取决于设置。正确的RL基础设施、奖励设计、数据混合和任务难度水平可以共同起到决定性作用，这些因素让我们能从RL中榨取更多性能，并得出在不同配置下似乎无法实现的结论。

**深度解读** 这是论文结论部分的点睛之笔，提出了一个极具分量的观点：对于训练AI来说，**“方法论”和“原材料”同样重要。**

在过去很长一段时间里，大模型领域的主流思想是“大力出奇迹”——只要有足够多的数据（原材料）和足够大的模型，AI的能力自然会提升。但这项研究用实验证明，这种想法是片面的。他们发现：

1. **“聪明的教学”胜过“盲目的喂料”**：他们通过课程学习的实验（图7）证明，不是随便给模型安排一些由易到难的“课程”就有用的。如果课程内容之间缺乏内在的逻辑关联，那么这种训练不仅没有帮助，反而可能是浪费时间。这说明，高质量的训练，需要精心设计学习路径。

2. **“训练秘籍”能化腐朽为神奇**：他们展示的一系列训练技巧，特别是“密集奖励预热”，就像一个“点金术”，能够让模型在一个它原本完全无法解决的问题上，实现从0到1的突破。这表明，训练过程中的具体策略选择，对于能否解锁模型的深层潜力，起着决定性的作用。

最终，他们得出了一个非常辩证和深刻的结论：**AI的学习，既可能是在“打磨”现有能力，也可能是在“发现”全新策略。** 最终会发生哪一种情况，不是由模型单方面决定的，而是由一个复杂的系统共同决定的，这个系统包括：

- **奖励设计**：我们如何定义“好”与“坏”？
- **数据组合**：我们给它什么样的“练习题”？
- **任务难度**：我们设定的挑战有多大？
- **训练设施**：我们使用的工具和流程是怎样的？

这就像培养一个孩子，他最终能成为一个普通工匠还是一个开创性的大师，不仅取决于他的天赋，更取决于我们为他提供的教育环境、激励方式和挑战机会。这篇论文最重要的贡献之一，就是为我们提供了一套经过验证的、能够引导AI走向“发现”和“顿悟”的“高级教育方法”。它告诉我们，AI的未来，不仅在于建造更大的“大脑”，更在于探索如何更智慧地去“点亮”它。