KIMI K1.5：使用LLMS扩展强化学习

KIMI K1.5 技术报告

Kimi团队

摘要

语言模型的预训练通过下一个标记预测已被证明在计算扩展方面有效，但受限于可用训练数据的数量。扩展强化学习（RL）为人工智能的持续改进解锁了一个新的维度，承诺大型语言模型（LLMs）可以通过学习探索和奖励来扩展其训练数据。然而，之前的已发布工作并未产生具有竞争力的结果。鉴于此，我们报告了Kimi k1.5的训练实践，这是我们最新的多模态LLM，采用RL进行训练，包括其RL训练技术、多模态数据配方和基础设施优化。长上下文扩展和改进的策略优化方法是我们方法的关键成分，它建立了一个简单有效的RL框架，而不依赖于更复杂的技术，如蒙特卡洛树搜索、价值函数和过程奖励模型。值得注意的是，我们的系统在多个基准和模态上实现了最先进的推理性能——例如，AIME上为77.5，MATH 500上为96.2，Codeforces上为94百分位，MathVista上为74.9——与OpenAI的o1相匹配。此外，我们提出了有效的long2short方法，利用长-CoT技术来改进短-CoT模型，产生最先进的短-CoT推理结果——例如，AIME上为60.8，MATH500上为94.6，LiveCodeBench上为47.3——大幅超越现有的短-CoT模型，如GPT-4o和Claude Sonnet 3.5（高达+550%）。Kimi k1.5在kimi.ai上的服务将很快上线。

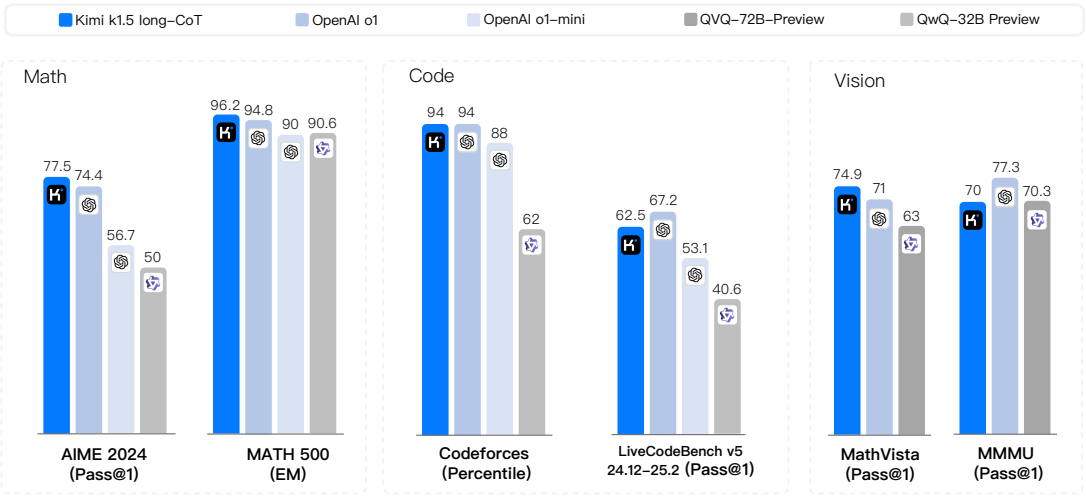


图 1：Kimi k1.5 长链思维结果。

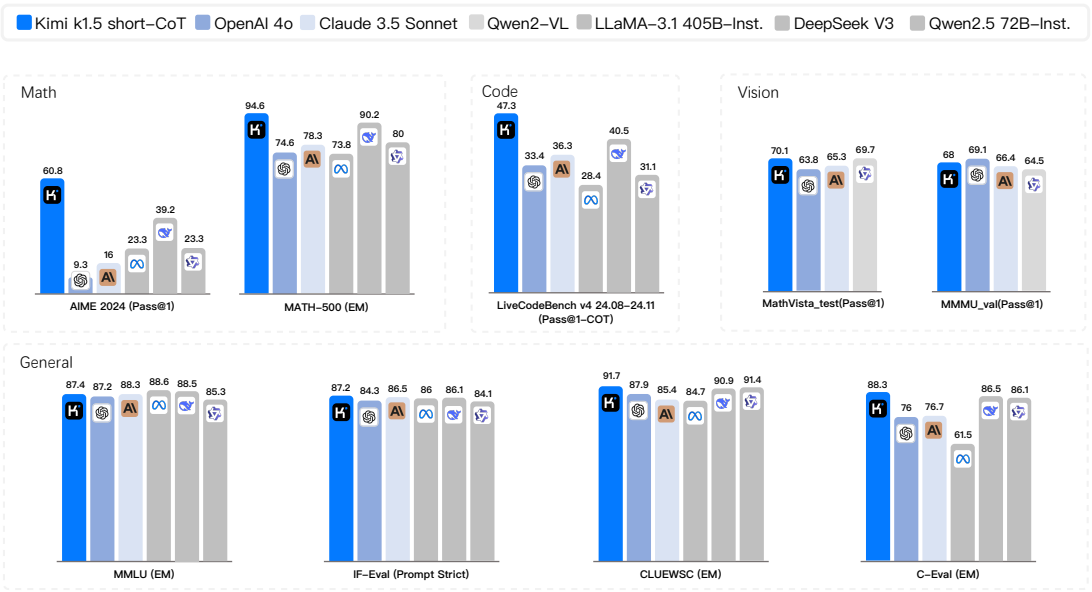


图 2：Kimi k1.5 短期 CoT 结果。

1 引言

语言模型预训练与下一个标记预测的研究是在规模法则的背景下进行的，其中按比例缩放模型参数和数据大小会导致智能的持续提升。（Kaplan et al. 2020; Hoffmann et al. 2022）然而，这种方法受到可用高质量训练数据量的限制（Villalobos et al. 2024; Muennighoff et al. 2023）。在本报告中，我们介绍了Kimi k1.5的训练配方，这是我们最新的多模态LLM，采用强化学习（RL）进行训练。目标是探索持续扩展的一个可能新方向。通过使用RL与LLM，模型学习通过奖励进行探索，因此不受现有静态数据集的限制。

关于k1.5的设计和训练，有几个关键要素。

- 长上下文缩放。我们将RL的上下文窗口扩展到128k，并观察到随着上下文长度的增加，性能持续改善。我们方法背后的一个关键思想是使用部分回合来提高训练效率——即通过重用大量先前轨迹来采样新的轨迹，避免从头重新生成新轨迹的成本。我们的观察表明，上下文长度是RL与LLM持续扩展的一个关键维度。
- 改进的策略优化。我们推导了带有长链的强化学习的公式，并采用了一种在线镜面下降的变体进行稳健的策略优化。通过我们有效的采样策略、长度惩罚和数据配方的优化，进一步改进了该算法。
- 简化框架。长上下文扩展结合改进的策略优化方法，建立了一个简化的强化学习框架，以便与大型语言模型（LLMs）进行学习。由于我们能够扩展上下文长度，学习到的链条（CoTs）展现了规划、反思和修正的特性。增加上下文长度会增加搜索步骤的数量。因此，我们展示了在不依赖于更复杂的技术，如蒙特卡洛树搜索、价值函数和过程奖励模型的情况下，可以实现强大的性能。
- 多模态。我们的模型在文本和视觉数据上进行联合训练，具备对这两种模态进行联合推理的能力。

此外，我们提出了有效的长到短的方法，这些方法利用长-CoT 技术来改进短-CoT 模型。具体而言，我们的方法包括使用长-CoT 激活应用长度惩罚和模型合并。

我们的长-CoT版本在多个基准和模态上实现了最先进的推理性能——例如，在AIME上为77.5，在MATH 500上为96.2，在Codeforces上为94百分位，在MathVista上为74.9——与OpenAI的o1相匹配。我们的模型还实现了最先进的短-CoT推理结果——例如，在AIME上为60.8，在MATH500上为94.6，在LiveCodeBench上为47.3——大幅超越现有的短-CoT模型，如GPT-4o和Claude Sonnet 3.5（高达+550%）。结果如图1和图2所示。

2 方法：使用大型语言模型的强化学习

Kimi k1.5 的开发包括几个阶段：预训练、普通监督微调（SFT）、长链思维（long-CoT）监督微调和强化学习（RL）。本报告重点关注 RL，首先概述 RL 提示集的策划（第 2.1 节）和长链思维监督微调（第 2.2 节），然后在第 2.3 节深入讨论 RL 训练策略。有关预训练和普通监督微调的更多细节可以在第 2.5 节找到。

2.1 RL 提示集策划

通过我们的初步实验，我们发现RL提示集的质量和多样性在确保强化学习的有效性方面起着关键作用。一个构建良好的提示集不仅引导模型进行稳健的推理，还减轻了奖励黑客攻击和对表面模式过拟合的风险。具体而言，三个关键属性定义了高质量的RL提示集：

- 多样化覆盖：提示应涵盖广泛的学科，如STEM、编码和一般推理，以增强模型的适应性，并确保在不同领域的广泛适用性。
- 平衡难度：提示集应包括易、中、难问题的良好分布范围，以促进逐步学习并防止对特定复杂度水平的过拟合。
- 准确的可评估性：提示应允许验证者进行客观和可靠的评估，确保模型性能是基于正确的推理而非表面模式或随机猜测来衡量的。

为了在提示集实现多样化覆盖，我们采用自动过滤器选择需要丰富推理且易于评估的问题。我们的数据集包括来自各个领域的问题，如STEM领域、竞赛和一般推理任务，涵盖文本和图像-文本问答数据。此外，我们开发了一个标签系统，以按领域和学科对提示进行分类，确保在不同学科领域之间的平衡代表性（M. Li et al. 2023; W. Liu et al. 2023）。

我们采用基于模型的方法，利用模型自身的能力自适应地评估每个提示的难度。具体来说，对于每个提示，SFT模型使用相对较高的采样温度生成十次答案。然后计算通过率，并将其作为提示难度的代理——通过率越低，难度越高。这种方法使得难度评估与模型的内在能力相一致，从而在RL训练中非常有效。通过利用这种方法，我们可以预先过滤掉大多数琐碎的案例，并在RL训练期间轻松探索不同的采样策略。

为了避免潜在的奖励黑客行为（Everitt et al. 2021; Pan et al. 2022），我们需要确保每个提示的推理过程和最终答案都能被准确验证。实证观察表明，一些复杂的推理问题可能有相对简单且容易猜测的答案，从而导致假阳性验证——模型通过不正确的推理过程得出正确答案。为了解决这个问题，我们排除了容易出现此类错误的问题，例如选择题、是非题和基于证明的问题。此外，对于一般的问答任务，我们提出了一种简单而有效的方法来识别和删除容易被黑客攻击的提示。具体而言，我们提示模型在没有任何链式推理步骤的情况下猜测潜在答案。如果模型在 N 次尝试内预测出正确答案，则该提示被认为过于容易被黑客攻击并被删除。我们发现设置 $N =$ 为 8 可以删除大多数容易被黑客攻击的提示。开发更先进的验证模型仍然是未来研究的一个开放方向。

2.2 长-CoT 监督微调

通过精炼的RL提示集，我们采用提示工程构建一个小而高质量的长链思维（long-CoT）热身数据集，包含对文本和图像输入的准确验证推理路径。这种方法类似于拒绝采样（RS），但专注于通过提示工程生成长链思维推理路径。生成的热身数据集旨在封装对人类推理至关重要的关键认知过程，例如规划，其中模型在执行之前系统地列出步骤；评估，涉及对中间步骤的批判性评估；反思，使模型能够重新考虑和完善其方法；以及探索，鼓励考虑替代解决方案。通过对这个热身数据集进行轻量级的SFT，我们有效地使模型内化这些推理策略。因此，经过微调的长链思维模型在生成更详细和逻辑连贯的响应方面表现出更强的能力，从而提升了其在各种推理任务中的表现。

2.3 强化学习

2.3.1 问题设置

给定一个训练数据集 $\mathcal{D} = \{(x_i, y_i^*)\}_{i=1}^n$，包含问题 x_i 和相应的真实答案 y_i^*，我们的目标是训练一个策略模型 π_θ，以准确解决测试问题。在复杂推理的背景下，问题 x 到解决方案 y 的映射并非简单。为了解决这个挑战，*chain of thought* (CoT) 方法提出使用一系列中间步骤 $z = (z_1, z_2, \ldots, z_m)$ 来连接 x 和 y，其中每个 z_i 是一个连贯的标记序列，作为解决问题的重要中间步骤（J. Wei 等，2022）。在解决问题 x 时，思路 $z_t \sim \pi_\theta(\cdot|x, z_1, \ldots, z_{t-1})$ 是自回归采样的，随后是最终答案 $y \sim \pi_\theta(\cdot|x, z_1, \ldots, z_m)$。我们用 $y, z \sim \pi_\theta$ 来表示这个采样过程。请注意，思路和最终答案都是作为语言序列进行采样的。

为了进一步增强模型的推理能力，采用 *planning* 算法来探索各种思维过程，在推理时生成改进的 CoT（Yao et al. 2024; Y. Wu et al. 2024; Snell et al. 2024）。这些方法的核心见解是明确构建一个由价值估计引导的思维搜索树。这使得模型能够探索思维过程的多种延续，或在遇到死胡同时回溯以调查新的方向。更详细地说，设 $\mathcal{T}$ 为一棵搜索树，其中每个节点代表一个部分解决方案 $s = (x, z_{1:|s|})$。这里 s 包含问题 x 和一系列思维 $z_{1:|s|} = (z_1, \ldots, z_{|s|})$，这些思维引导到该节点，$|s|$ 表示序列中的思维数量。规划算法使用一个评论模型 v 来提供反馈 $v(x, z_{1:|s|})$，这有助于评估当前解决问题的进展并识别现有部分解决方案中的任何错误。我们注意到，反馈可以通过判别分数或语言序列提供（L. Zhang et al. 2024）。在所有 $s \in \mathcal{T}$ 的反馈指导下，规划算法选择最有前景的节点进行扩展，从而扩展搜索树。上述过程迭代重复，直到得出完整的解决方案。

我们也可以从 *algorithmic perspective* 的角度来接近规划算法。考虑在 t 次迭代中可用的过去搜索历史 ($s_1, v(s_1), \ldots, s_{t-1}, v(s_{t-1})$)，规划算法 $\mathcal{A}$ 迭代地确定下一个搜索方向 $\mathcal{A}(s_t|s_1, v(s_1), \ldots, s_{t-1}, v(s_{t-1}))$ 并为当前搜索进展 $\mathcal{A}(v(s_t)|s_1, v(s_1), \ldots, s_t)$ 提供反馈。由于思想和反馈都可以视为中间推理步骤，并且这些组件都可以表示为语言标记的序列，我们使用 z 来替代 s，并使用 v 来简化符号。因此，我们将规划算法视为直接作用于推理步骤序列 $\mathcal{A}(\cdot|z_1, z_2, \ldots)$ 的映射。在这个框架中，规划算法使用的搜索树中存储的所有信息都被扁平化为提供给算法的完整上下文。这为生成高质量的 CoT 提供了一个有趣的视角：与其明确构建搜索树并实现规划算法，我们可以潜在地训练一个模型来近似这个过程。在这里，思想的数量（即语言标记）作为传统上分配给规划算法的计算预算的类比。最近在长上下文窗口方面的进展促进了训练和测试阶段的无缝可扩展性。如果可行，这种方法使模型能够通过自回归预测直接在推理空间上进行隐式搜索。因此，模型不仅学习解决一组训练问题，还发展出有效处理单个问题的能力，从而提高对未见测试问题的泛化能力。

因此，我们考虑训练模型通过强化学习（RL）生成 CoT（OpenAI 2024）。设 r 为一个奖励模型，它根据真实情况 y^* 为给定问题 x 的提议答案 y 的正确性提供依据，通过分配一个值 $r(x, y, y^*) \in \{0, 1\}$。对于可验证的问题，奖励直接由预定义的标准或规则决定。例如，在编码问题中，我们评估答案是否通过测试用例。对于具有自由形式真实情况的问题，我们训练一个奖励模型 $r(x, y, y^*)$，预测答案是否与真实情况匹配。给定一个问题 x，模型 π_θ 通过采样过程 $z \sim \pi_\theta(\cdot|x)$、$y \sim \pi_\theta(\cdot|x, z)$ 生成 CoT 和最终答案。生成的 CoT 的质量通过它是否能导致正确的最终答案来评估。总之，我们考虑以下目标来优化策略。

$$\max_\theta \mathbb{E}_{(x,y^*)\sim\mathcal{D},(y,z)\sim\pi_\theta} \left[r(x, y, y^*) \right] . \tag{1}$$

通过扩大强化学习训练，我们旨在训练一个模型，利用简单的基于提示的链式推理（CoT）和增强规划的链式推理（CoT）两者的优势。该模型在推理过程中仍然是自回归地采样语言序列，从而避免了在部署过程中高级规划算法所需的复杂并行化。然而，与简单的基于提示的方法的一个关键区别在于，该模型不应仅仅遵循一系列推理步骤。相反，它还应通过利用整个探索思维的集合作为上下文信息，学习关键的规划技能，包括错误识别、回溯和解决方案的细化。

2.3.2 策略优化

我们将在线策略镜像下降的变体作为我们的训练算法（Abbasi-Yadkori et al. 2019; Mei et al. 2019; Tomar et al. 2020）。该算法以迭代方式执行。在第 i 次迭代中，我们使用当前模型 π_{θ_i} 作为参考模型，并优化以下相对熵正则化的策略优化问题，

$$\max_{\theta} \mathbb{E}_{(x,y^*)\sim\mathcal{D}} \left[\mathbb{E}_{(y,z)\sim\pi_\theta} \left[r(x,y,y^*) \right] - \tau \mathrm{KL}(\pi_\theta(x) || \pi_{\theta_i}(x)) \right] , \tag{2}$$

其中 $\tau > 0$ 是一个控制正则化程度的参数。这个目标有一个闭式解。

$$\pi^*(y,z|x) = \pi_{\theta_i}(y,z|x) \exp(r(x,y,y^*)/\tau)/Z .$$

这里 $Z = \sum_{y',z'} \pi_{\theta_i}(y',z'|x) \exp(r(x,y',y^*)/\tau)$ 是归一化因子。对两边取对数，我们得到 *any* (y,z) 满足以下约束，这使我们能够在优化过程中利用离线数据。

$$r(x,y,y^*) - \tau \log Z = \tau \log \frac{\pi^*(y,z|x)}{\pi_{\theta_i}(y,z|x)} .$$

这激励了以下替代损失 {v*}

$$L(\theta) = \mathbb{E}_{(x,y^*)\sim\mathcal{D}} \left[\mathbb{E}_{(y,z)\sim\pi_{\theta_i}} \left[\left(r(x,y,y^*) - \tau \log Z - \tau \log \frac{\pi_\theta(y,z|x)}{\pi_{\theta_i}(y,z|x)} \right)^2 \right] \right] .$$

为了近似 $\tau \log Z$，我们使用样本 $(y_1,z_1),\ldots,(y_k,z_k) \sim \pi_{\theta_i}$：$\tau \log Z \approx \tau \log \frac{1}{k} \sum_{j=1}^{k} \exp(r(x,y_j,y^*)/\tau)$。我们还发现，使用采样奖励的经验均值 $r = \mathrm{mean}(r(x,y_1,y^*),\ldots,r(x,y_k,y^*))$ 可以产生有效的实际结果。这是合理的，因为 $\tau \log Z$ 在 $\tau \to \infty$ 时接近 π_{θ_i} 下的期望奖励。最后，我们通过对替代损失的梯度进行计算来结束我们的学习算法。对于每个问题 x，k 的响应是使用参考策略 π_{θ_i} 进行采样的，梯度由以下公式给出：

$$\frac{1}{k} \sum_{j=1}^{k} \left(\nabla_\theta \log \pi_\theta(y_j,z_j|x)(r(x,y_j,y^*) - \bar{r}) - \frac{\tau}{2} \nabla_\theta \left(\log \frac{\pi_\theta(y_j,z_j|x)}{\pi_{\theta_i}(y_j,z_j|x)} \right)^2 \right) , \tag{3}$$

对于熟悉策略梯度方法的人来说，这个梯度类似于（2）中使用采样奖励的均值作为基线的策略梯度（Kool et al. 2019; Ahmadian et al. 2024）。主要的区别在于响应是从 π_{θ_i} 中采样的，而不是基于策略的，并且应用了 l_2 正则化。因此，我们可以将其视为通常的基于策略的正则化策略梯度算法向离策略情况的自然扩展（Nachum et al. 2017）。我们从 $\mathcal{D}$ 中采样一批问题，并将参数更新为 θ_{i+1}，随后作为下一次迭代的参考策略。由于每次迭代由于参考策略的变化而考虑不同的优化问题，我们还在每次迭代开始时重置优化器。

我们在训练系统中排除了价值网络，这在之前的研究中也得到了利用（Ahmadian et al. 2024）。虽然这一设计选择显著提高了训练效率，但我们也假设在经典强化学习中，传统的价值函数用于信用分配可能不适合我们的背景。考虑一个场景，其中模型生成了部分链条推理（CoT）$(z_1,z_2,\ldots,z_t)$，并且有两个潜在的下一个推理步骤：z_{t+1}和z'_{t+1}。假设z_{t+1}直接导致正确答案，而z'_{t+1}包含一些错误。如果可以访问一个神谕价值函数，它将表明z_{t+1}相较于z'_{t+1}保留了更高的价值。根据标准的信用分配原则，选择z'_{t+1}将受到惩罚，因为它相对于当前策略具有负优势。然而，探索z'_{t+1}对于训练模型生成长链条推理是极其有价值的。通过使用从长链条推理中得出的最终答案的理由作为奖励信号，模型可以学习从采取z'_{t+1}中进行试错的模式，只要它成功恢复并达到正确答案。这个例子的关键启示是，我们应该鼓励模型探索多样的推理路径，以增强其解决复杂问题的能力。这种探索性的方法产生了丰富的经验，支持关键规划技能的发展。我们的主要目标不仅限于在训练问题上获得高准确率，而是专注于为模型提供有效的问题解决策略，最终提高其在测试问题上的表现。

2.3.3 长度惩罚

我们观察到一个过度思考的现象，即模型的响应长度在强化学习训练期间显著增加。尽管这导致了更好的性能，但过于冗长的推理过程在训练和推理中是昂贵的，且人类通常不喜欢过度思考。为了解决这个问题，我们引入了长度奖励，以抑制标记长度的快速增长，从而提高模型的标记效率。给定 k 采样的响应

$(y_1, z1), \ldots, (y_k, z_k)$ 的问题 x 的真实答案 y^*，设 len(i) 为 (y_i, z_i) 的长度，min_len = $\min_i$ len(i) 和 max_len = $\max_i$ len(i)。如果 max_len = min_len，我们对所有响应设置长度奖励为零，因为它们具有相同的长度。否则，长度奖励由以下公式给出：

$$\text{len\_reward(i)} = \begin{cases} \lambda & \text{If } r(x, y_i, y^*) = 1 \\ \min(0, \lambda) & \text{If } r(x, y_i, y^*) = 0 \end{cases}, \quad \text{where } \lambda = 0.5 - \frac{\text{len}(i) - \text{min\_len}}{\text{max\_len} - \text{min\_len}}.$$

本质上，我们鼓励较短的回答，并对正确答案中的较长回答进行惩罚，同时明确惩罚错误答案的长回答。这个基于长度的奖励随后与原始奖励通过一个权重参数相加。

在我们的初步实验中，长度惩罚可能会在初始阶段减慢训练。为了解决这个问题，我们建议在训练过程中逐渐增加长度惩罚。具体来说，我们在没有长度惩罚的情况下采用标准策略优化，然后在训练的其余部分施加一个恒定的长度惩罚。

2.3.4 采样策略

尽管RL算法本身具有相对良好的采样特性（更困难的问题提供更大的梯度），但它们的训练效率有限。因此，一些定义良好的先验采样方法可能会带来更大的性能提升。我们利用多种信号进一步改善采样策略。首先，我们收集的RL训练数据自然带有不同的难度标签。例如，数学竞赛问题比小学数学问题更困难。其次，由于RL训练过程对同一问题进行多次采样，我们还可以跟踪每个单独问题的成功率作为难度的指标。我们提出了两种采样方法，以利用这些先验信息来提高训练效率。

课程采样 我们首先在较简单的任务上进行训练，然后逐渐过渡到更具挑战性的任务。由于初始的强化学习模型性能有限，在非常困难的问题上花费有限的计算预算往往会产生很少的正确样本，从而导致训练效率降低。同时，我们收集的数据自然包含等级和难度标签，使得基于难度的采样成为提高训练效率的一种直观有效的方法。

优先采样 除了课程采样，我们还使用优先采样策略，专注于模型表现不佳的问题。我们跟踪每个问题 i 的成功率 s_i，并按比例采样问题为 $1 - s_i$，这样成功率较低的问题会获得更高的采样概率。这将模型的努力引导到其最薄弱的领域，从而实现更快的学习和更好的整体表现。

2.3.5 训练方案的更多细节

编码的测试用例生成 由于许多网络编码问题没有可用的测试用例，我们设计了一种方法来自动生成测试用例，以作为奖励来训练我们的模型与RL。我们的重点主要是那些不需要特殊评判的题目。我们还假设这些问题的真实解是可用的，以便我们可以利用这些解来生成更高质量的测试用例。

我们利用广泛认可的测试用例生成库 CYaRon[1] 来增强我们的方法。我们使用基础 Kimi k1.5 根据问题陈述生成测试用例。CYaRon 的使用说明和问题描述作为生成器的输入。对于每个问题，我们首先使用生成器生成 50 个测试用例，并随机抽取 10 个真实提交作为每个测试用例的样本。我们将测试用例与提交进行对比。如果至少 10 个提交中有 7 个产生匹配结果，则该测试用例被视为有效。在这一轮筛选后，我们获得一组选定的测试用例。如果至少 10 个提交中有 9 个通过整个选定测试用例集，则该问题及其相关的选定测试用例将被添加到我们的训练集中。

在统计学方面，从1,000个在线竞赛问题的样本中，大约614个不需要特殊评审。我们开发了463个测试用例生成器，产生了至少40个有效测试用例，从而将323个问题纳入我们的训练集。

数学奖励建模 评估数学解答的一大挑战是，不同的书写形式可以表示相同的基础答案。例如，$a^2 - 4$ 和 $(a + 2)(a - 2)$ 可能都是同一问题的有效解答。我们采用了两种方法来提高奖励模型的评分准确性：

1. 经典RM：我们借鉴了InstructGPT（Ouyang等，2022）的方法，实施了基于价值头的奖励模型，并收集了大约80万的数据点用于微调。该模型最终

[1] https://github.com/luogu-dev/cyaron

将“问题”、“参考答案”和“响应”作为输入，并输出一个单一的标量，指示响应是否正确。

2. 思维链 RM：最近的研究（Ankner 等，2024；McAleese 等，2024）表明，增强了思维链（CoT）推理的奖励模型可以显著超越经典方法，特别是在对细微正确性标准要求较高的任务上——例如数学。因此，我们收集了大约 80 万个 CoT 标记示例的同等大型数据集，以微调 Kimi 模型。在与经典 RM 相同的输入基础上，思维链方法明确生成逐步推理过程，然后以 JSON 格式提供最终的正确性判断，从而实现更强大和可解释的奖励信号。

在我们的手动抽查中，经典RM的准确率约为84.4，而链式思维RM的准确率达到了98.5。在强化学习训练过程中，我们采用了链式思维RM，以确保更正确的反馈。

视觉数据 为了提高模型在现实世界中的图像推理能力，并实现视觉输入与大型语言模型（LLMs）之间更有效的对齐，我们的视觉强化学习（Vision RL）数据主要来源于三个不同的类别：现实世界数据、合成视觉推理数据和文本渲染数据。

1. 真实世界的数据涵盖了不同年级水平的一系列科学问题，这些问题需要图形理解和推理、需要视觉感知和推断的位置猜测任务，以及涉及理解复杂图表的数据分析等多种数据类型。这些数据集提高了模型在真实场景中进行视觉推理的能力。 2. 合成视觉推理数据是人工生成的，包括程序生成的图像和场景，旨在提高特定的视觉推理技能，例如理解空间关系、几何模式和物体交互。这些合成数据集为测试模型的视觉推理能力提供了一个受控环境，并提供了源源不断的训练示例。

3. 文本渲染数据是通过将文本内容转换为视觉格式来创建的，使模型在处理不同模态的基于文本的查询时能够保持一致性。通过将文本文件、代码片段和结构化数据转换为图像，我们确保模型提供一致的响应，无论输入是纯文本还是渲染为图像的文本（如截图或照片）。这也有助于增强模型在处理文本密集型图像时的能力。

每种类型的数据在构建一个全面的视觉语言模型中都是必不可少的，该模型能够有效地管理广泛的现实世界应用，同时确保在各种输入模态下的一致性能。

2.4 Long2short：短链条模型的上下文压缩

尽管长-CoT模型表现强劲，但与标准短-CoT LLM相比，它在测试时消耗更多的token。然而，可以将长-CoT模型的思维先验转移到短-CoT模型，从而在有限的测试token预算下提高性能。我们提出了几种解决这个long2short问题的方法，包括模型合并（Yang et al. 2024）、最短拒绝采样、DPO（Rafailov et al. 2024）和long2short RL。以下是对这些方法的详细描述：

模型合并　模型合并被发现对维持泛化能力是有用的。我们还发现它在合并长上下文模型和短上下文模型时提高了令牌效率的有效性。这种方法将长上下文模型与较短的模型结合，以获得一个新的模型，而无需训练。具体而言，我们通过简单地平均它们的权重来合并这两个模型。

最短拒绝采样 我们观察到我们的模型对同一个问题生成的响应长度变化很大。基于此，我们设计了最短拒绝采样方法。该方法对同一个问题进行 n 次采样（在我们的实验中，$n = 8$），并选择最短的正确响应进行监督微调。

DPO与最短拒绝采样相似，我们利用长链模型生成多个响应样本。选择最短的正确解作为正样本，而较长的响应则被视为负样本，包括错误的较长响应和正确的较长响应（比选择的正样本长1.5倍）。这些正负对形成了用于DPO训练的成对偏好数据。

长到短的强化学习 在标准的强化学习训练阶段之后，我们选择一个在性能和令牌效率之间提供最佳平衡的模型作为基础模型，并进行单独的长到短的强化学习训练阶段。在第二阶段，我们应用第2.3.3节中引入的长度惩罚，并显著减少最大展开长度，以进一步惩罚可能正确但超出期望长度的响应。

2.5 其他训练细节

2.5.1 预训练

Kimi k1.5 基础模型在一个多样化、高质量的多模态语料库上进行训练。语言数据涵盖五个领域：英语、中文、代码、数学推理和知识。多模态数据，包括字幕生成、图像文本交错、OCR、知识和问答数据集，使我们的模型能够获得视觉语言能力。严格的质量控制确保了整体预训练数据集的相关性、多样性和均衡性。我们的预训练分为三个阶段：（1）视觉语言预训练，在此阶段建立强大的语言基础，随后逐步进行多模态整合；（2）冷却阶段，利用策划和合成数据巩固能力，特别是针对推理和基于知识的任务；（3）长上下文激活，将序列处理扩展到 131,072 个标记。有关我们预训练工作的更多细节，请参见附录 B。

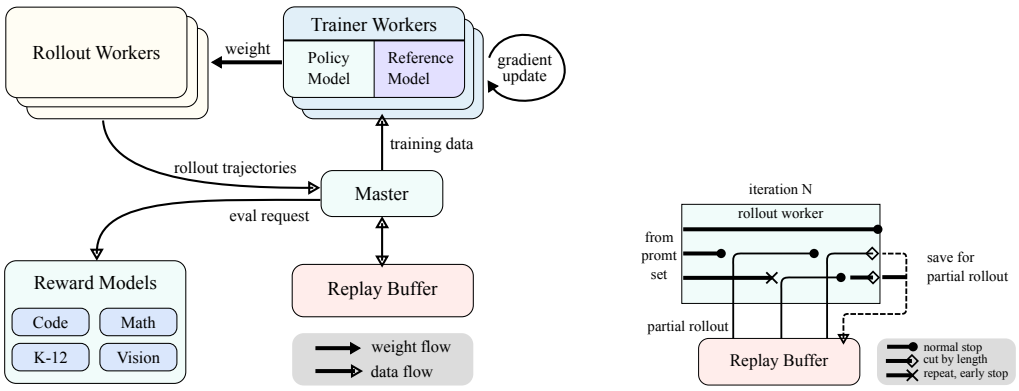
2.5.2 香草监督微调

我们创建了覆盖多个领域的基础 SFT 语料库。对于非推理任务，包括问答、写作和文本处理，我们最初通过人工标注构建一个种子数据集。这个种子数据集用于训练一个种子模型。随后，我们收集多样化的提示，并利用种子模型为每个提示生成多个响应。标注者随后对这些响应进行排名，并对排名最高的响应进行细化，以生成最终版本。对于数学和编码问题等推理任务，在这些任务中，基于规则和奖励建模的验证比人工判断更准确和高效，我们利用拒绝采样来扩展 SFT 数据集。

我们的香草SFT数据集包含大约100万个文本示例。具体来说，50万个示例用于一般问答，20万个用于编码，20万个用于数学和科学，5000个用于创意写作，以及2万个用于长文本任务，如摘要、文档问答、翻译和写作。此外，我们构建了100万个文本-视觉示例，涵盖了各种类别，包括图表解读、OCR、基于图像的对话、视觉编码、视觉推理以及带有视觉辅助的数学/科学问题。

我们首先在32k个标记的序列长度上训练模型1个周期，然后在128k个标记的序列长度上再训练1个周期。在第一阶段（32k），学习率从2×10^{-5}下降到2×10^{-6}，然后在第二阶段（128k）重新升温到1×10^{-5}，最后下降到1×10^{-6}。为了提高训练效率，我们将多个训练示例打包到每个单一训练序列中。

2.6 强化学习基础设施



(a) System overview (b) Partial Rollout

图 3：大规模强化学习训练系统用于 LLM

2.6.1 大规模强化学习训练系统用于LLM

在人工智能领域，强化学习（RL）已成为大型语言模型（LLMs）的关键训练方法（Ouyang et al. 2022）（Jaech et al. 2024），其灵感来源于在围棋、星际争霸 II 和 Dota 2 等复杂游戏中取得的成功，这些成功是通过 AlphaGo（Silver et al. 2017）、AlphaStar（Vinyals et al. 2019）和 OpenAI Dota Five（Berner et al. 2019）等系统实现的。沿袭这一传统，Kimi k1.5 系统采用了一种迭代同步的 RL 框架，精心设计以通过持续学习和适应来增强模型的推理能力。该系统的一项关键创新是引入了部分回放技术，旨在优化复杂推理轨迹的处理。

如图3a所示，RL训练系统通过迭代同步的方法进行操作，每次迭代包括一个展开阶段和一个训练阶段。在展开阶段，由中央主控协调的展开工作者通过与模型互动生成展开轨迹，产生对各种输入的响应序列。这些轨迹随后存储在重放缓冲区中，通过打破时间相关性来确保训练数据集的多样性和无偏性。在随后的训练阶段，训练工作者访问这些经验以更新模型的权重。这个循环过程使模型能够不断从其行动中学习，随着时间的推移调整其策略以提高性能。

中央主控器作为中央指挥者，管理着数据和通信在部署工作者、训练工作者、使用奖励模型进行评估以及重放缓冲区之间的流动。它确保系统和谐运行，平衡负载并促进高效的数据处理。

训练工人访问这些滚动轨迹，无论是在单次迭代中完成还是跨多个迭代进行，以计算更新梯度，从而优化模型的参数并提升其性能。这个过程由奖励模型监督，该模型评估模型输出的质量，并提供必要的反馈以指导训练过程。奖励模型的评估在确定模型策略的有效性和引导模型朝向最佳性能方面尤为关键。

此外，该系统包含一个代码执行服务，专门用于处理与代码相关的问题，并且是奖励模型的重要组成部分。该服务在实际编码场景中评估模型的输出，确保模型的学习与现实编程挑战紧密相关。通过将模型的解决方案与实际代码执行进行验证，这一反馈循环对于完善模型的策略和提高其在代码相关任务中的表现至关重要。

2.6.2 长期 CoT RL 的部分回滚

我们工作的一个主要思想是扩展长上下文强化学习训练。部分回放是一个关键技术，有效解决了通过管理长短轨迹的回放来处理长-CoT 特征的挑战。该技术建立了一个固定的输出令牌预算，限制每个回放轨迹的长度。如果在回放阶段轨迹超过令牌限制，未完成的部分将被保存到重放缓冲区，并在下一次迭代中继续。这确保了没有单个冗长的轨迹垄断系统的资源。此外，由于回放工作者异步操作，当一些工作者处理长轨迹时，其他工作者可以独立处理新的较短回放任务。异步操作通过确保所有回放工作者都积极参与训练过程，从而最大化计算效率，优化系统的整体性能。

如图3b所示，部分展开系统通过将长响应分解为跨迭代的段（从迭代 n-m 到迭代 n）来工作。重放缓冲区充当中央存储机制，维护这些响应段，其中只有当前迭代（迭代 n）需要进行策略计算。之前的段（迭代 n-m 到 n-1）可以从缓冲区高效重用，消除了重复展开的需要。这种分段方法显著减少了计算开销：系统不是一次性展开整个响应，而是逐步处理和存储段，从而允许生成更长的响应，同时保持快速的迭代时间。在训练过程中，可以排除某些段的损失计算，以进一步优化学习过程，使整个系统既高效又可扩展。

部分回滚的实现还提供了重复检测。系统识别生成内容中的重复序列并提前终止它们，从而减少不必要的计算，同时保持输出质量。检测到的重复可以被赋予额外的惩罚，有效地抑制提示集中的冗余内容生成。

2.6.3 训练和推理的混合部署

RL训练过程包括以下几个阶段:

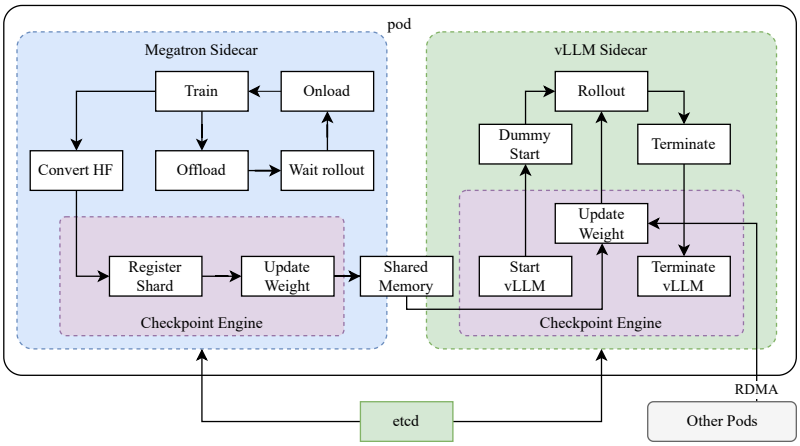


图 4：混合部署框架

- 训练阶段：一开始，Megatron（Shoeybi 等，2020）和 vLLM（Kwon 等，2023）在各自的容器中执行，由一个称为 checkpoint-engine 的中间过程封装（第 2.6.3 节）。Megatron 开始训练程序。训练完成后，Megatron 卸载 GPU 内存，并准备将当前权重转移到 vLLM。
- 推理阶段：在Megatron的卸载之后，vLLM以虚拟模型权重开始，并通过Mooncake（Qin等，2024）更新为从Megatron转移来的最新权重。完成发布后，检查点引擎停止所有vLLM进程。
- 后续训练阶段：一旦分配给 vLLM 的内存被释放，Megatron 将加载内存并启动另一轮训练。

我们发现现有的工作难以同时支持以下所有特性。

- 复杂的并行策略：Megatron 可能与 vLLM 采用不同的并行策略。在 Megatron 中，训练权重分布在多个节点上，这可能会与 vLLM 共享时面临挑战。
- 最小化闲置的GPU资源：对于在线策略强化学习，最近的研究如SGLang（L. Zheng等，2024）和vLLM可能在训练过程中保留一些GPU，这反过来可能导致训练GPU闲置。将训练和推理共享相同的设备会更高效。
- 动态扩展能力：在某些情况下，通过增加推理节点的数量，同时保持训练过程不变，可以实现显著的加速。我们的系统在需要时能够高效利用闲置的GPU节点。

如图4所示，我们在Megatron和vLLM之上实现了这个混合部署框架（第2.6.3节），从训练到推理阶段的时间少于一分钟，反之约为十秒。

混合部署策略 我们提出了一种混合部署策略，用于训练和推理任务，该策略利用Kubernetes Sidecar容器共享所有可用的GPU，将这两种工作负载放置在一个pod中。该策略的主要优点包括：

- 它促进了高效的资源共享和管理，防止了训练节点在等待推理节点时闲置，尤其是在两者部署在不同节点上的情况下。
- 利用不同的部署图像，训练和推理可以各自独立迭代，以获得更好的性能。
- 该架构不限于 vLLM，其他框架也可以方便地集成。

检查点引擎 检查点引擎负责管理 vLLM 进程的生命周期，暴露 HTTP API 以触发对 vLLM 的各种操作。为了整体的一致性和可靠性，我们利用由 etcd 服务管理的全局元数据系统来广播操作和状态。

通过vLLM卸载完全释放GPU内存可能会面临挑战，主要是由于CUDA图、NCCL缓冲区和NVIDIA驱动程序。为了尽量减少对vLLM的修改，我们在需要时终止并重新启动它，以提高GPU利用率和容错能力。

在Megatron中的工作者将拥有的检查点转换为共享内存中的Hugging Face格式。此转换还考虑了管道并行性和专家并行性，以便在这些检查点中仅保留张量并行性。共享内存中的检查点随后被划分为碎片，并在全球元数据系统中注册。我们使用Mooncake通过RDMA在对等节点之间传输检查点。需要对vLLM进行一些修改，以加载权重文件并执行张量并行性转换。

2.6.4 代码沙箱

我们开发了沙箱作为一个安全的环境，用于执行用户提交的代码，优化了代码执行和代码基准评估。通过动态切换容器镜像，沙箱通过 MultiPL-E (Cassano, Gouwar, D. Nguyen, S. Nguyen 等，2023)、DMOJ Judge Server [2]、Lean、Jupyter Notebook 和其他镜像支持不同的用例。

对于编码任务中的强化学习，沙箱通过提供一致且可重复的评估机制来确保训练数据判断的可靠性。其反馈系统支持多阶段评估，例如代码执行反馈和仓库级编辑，同时保持统一的上下文，以确保跨编程语言的公平和公正的基准比较。

我们在Kubernetes上部署服务以实现可扩展性和弹性，通过HTTP端点对外集成。Kubernetes的特性，如自动重启和滚动更新，确保了可用性和容错性。

为了优化性能并支持强化学习环境，我们在代码执行服务中引入了几种技术，以提高效率、速度和可靠性。这些技术包括：

- 使用Crun：我们使用crun作为容器运行时，而不是Docker，这显著减少了容器启动时间。
- Cgroup 重用：我们为容器使用预先创建 cgroups，这在高并发场景中至关重要，因为为每个容器创建和销毁 cgroups 可能成为瓶颈。
- 磁盘使用优化：使用一个上层挂载为tmpfs的覆盖文件系统来控制磁盘写入，提供一个固定大小的高速存储空间。这种方法对短暂工作负载是有益的。

Method	Time (s)
Docker	0.12
Sandbox	0.04

(a) 容器启动时间

Method	Containers/sec
Docker	27
Sandbox	120

(b) 每秒在16核机器上启动的最大容器数量

这些优化提高了代码执行中的强化学习效率，为评估强化学习生成的代码提供了一致且可靠的环境，这对于迭代训练和模型改进至关重要。

3 个实验

3.1 评估

由于k1.5是一个多模态模型，我们在不同模态的各种基准上进行了全面评估。详细的评估设置可以在附录C中找到。我们的基准主要包括以下三个类别：

- 文本基准：MMLU（Hendrycks等，2020），IF-Eval（J. Zhou等，2023），CLUEWSC（L. Xu等，2020），C-EVAL（Y. Huang等，2023）
- 推理基准：HumanEval-Mul，LiveCodeBench（Jain et al. 2024），Codeforces，AIME 2024，MATH-500（Lightman et al. 2023）
- 视觉基准：MMMU（Yue, Ni 等，2024），MATH-Vision（K. Wang 等，2024），MathVista（Lu 等，2023）

[2]https://github.com/DMOJ/judge-server

3.2 主要结果

K1.5 长-CoT 模型 Kimi k1.5 长-CoT 模型的性能如表 2 所示。通过长-CoT 监督微调（在第 2.2 节中描述）和视觉-文本联合强化学习（在第 2.3 节中讨论），模型的长期推理能力显著增强。测试时间计算的扩展进一步增强了其性能，使模型能够在多种模态中实现最先进的结果。我们的评估显示，模型在推理、理解和综合信息的能力上有显著改善，代表了多模态 AI 能力的进步。

K1.5 短链思维模型 Kimi k1.5 短链思维模型的性能如表 3 所示。该模型整合了多种技术，包括传统的监督微调（在第 2.5.2 节中讨论）、强化学习（在第 2.3 节中探讨）和长到短的蒸馏（在第 2.4 节中概述）。结果表明，k1.5 短链思维模型在多个任务中提供了与领先的开源和专有模型相媲美或更优的性能。这些任务包括文本、视觉和推理挑战，在自然语言理解、数学、编码和逻辑推理方面表现出显著优势。

	Benchmark (Metric)	Language-only Model		Vision-Language Model		
		QwQ-32B Preview	OpenAI o1-mini	QVQ-72B Preview	OpenAI o1	Kimi k1.5
Reasoning	MATH-500 (EM)	90.6	90.0	-	94.8	**96.2**
	AIME 2024 (Pass@1)	50.0	56.7	-	74.4	**77.5**
	Codeforces (Percentile)	62	88	-	**94**	**94**
	LiveCodeBench (Pass@1)	40.6	53.1	-	**67.2**	62.5
Vision	MathVista-Test (Pass@1)	-	-	71.4	71.0	**74.9**
	MMMU-Val (Pass@1)	-	-	70.3	**77.3**	70.0
	MathVision-Full (Pass@1)	-	-	35.9	-	**38.6**

表2：Kimi k1.5 long-CoT及旗舰开源和专有模型的性能。

	Benchmark (Metric)	Language-only Model			Vision-Language Model			
		Qwen2.5 72B-Inst.	LLaMA-3.1 405B-Inst.	DeepSeek V3	Qwen2-VL	Claude-3.5-Sonnet-1022	GPT-4o 0513	Kimi k1.5
Text	MMLU (EM)	85.3	**88.6**	88.5	-	88.3	87.2	87.4
	IF-Eval (Prompt Strict)	84.1	86.0	86.1	-	86.5	84.3	**87.2**
	CLUEWSC (EM)	91.4	84.7	90.9	-	85.4	87.9	**91.7**
	C-Eval (EM)	86.1	61.5	86.5	-	76.7	76.0	**88.3**
Reasoning	MATH-500 (EM)	80.0	73.8	90.2	-	78.3	74.6	**94.6**
	AIME 2024 (Pass@1)	23.3	23.3	39.2	-	16.0	9.3	**60.8**
	HumanEval-Mul (Pass@1)	77.3	77.2	**82.6**	-	81.7	80.5	81.5
	LiveCodeBench (Pass@1)	31.1	28.4	40.5	-	36.3	33.4	**47.3**
Vision	MathVista-Test (Pass@1)	-	-	-	69.7	65.3	63.8	**70.1**
	MMMU-Val (Pass@1)	-	-	-	64.5	66.4	**69.1**	68.0
	MathVision-Full (Pass@1)	-	-	-	26.6	**35.6**	30.4	31.0

表3：Kimi k1.5 短链条（short-CoT）和旗舰开源及专有模型的性能。VLM模型性能数据来自OpenCompass基准平台（https://opencompass.org.cn/）。

3.3 长上下文缩放

我们采用一个中等规模的模型来研究与大型语言模型（LLMs）相关的强化学习（RL）的扩展特性。图5展示了在数学提示集上训练的小型模型变体在训练迭代过程中训练准确率和响应长度的演变。随着训练的进行，我们观察到响应长度和性能准确率同时增加。值得注意的是，更具挑战性的基准测试显示出响应长度的陡峭增加，这表明模型学习生成更复杂问题的更详细解决方案。图6表明模型之间存在强相关性。

输出上下文长度及其解决问题的能力。我们最终的 k1.5 运行扩展到 128k 上下文长度，并在困难推理基准上观察到持续的改进。

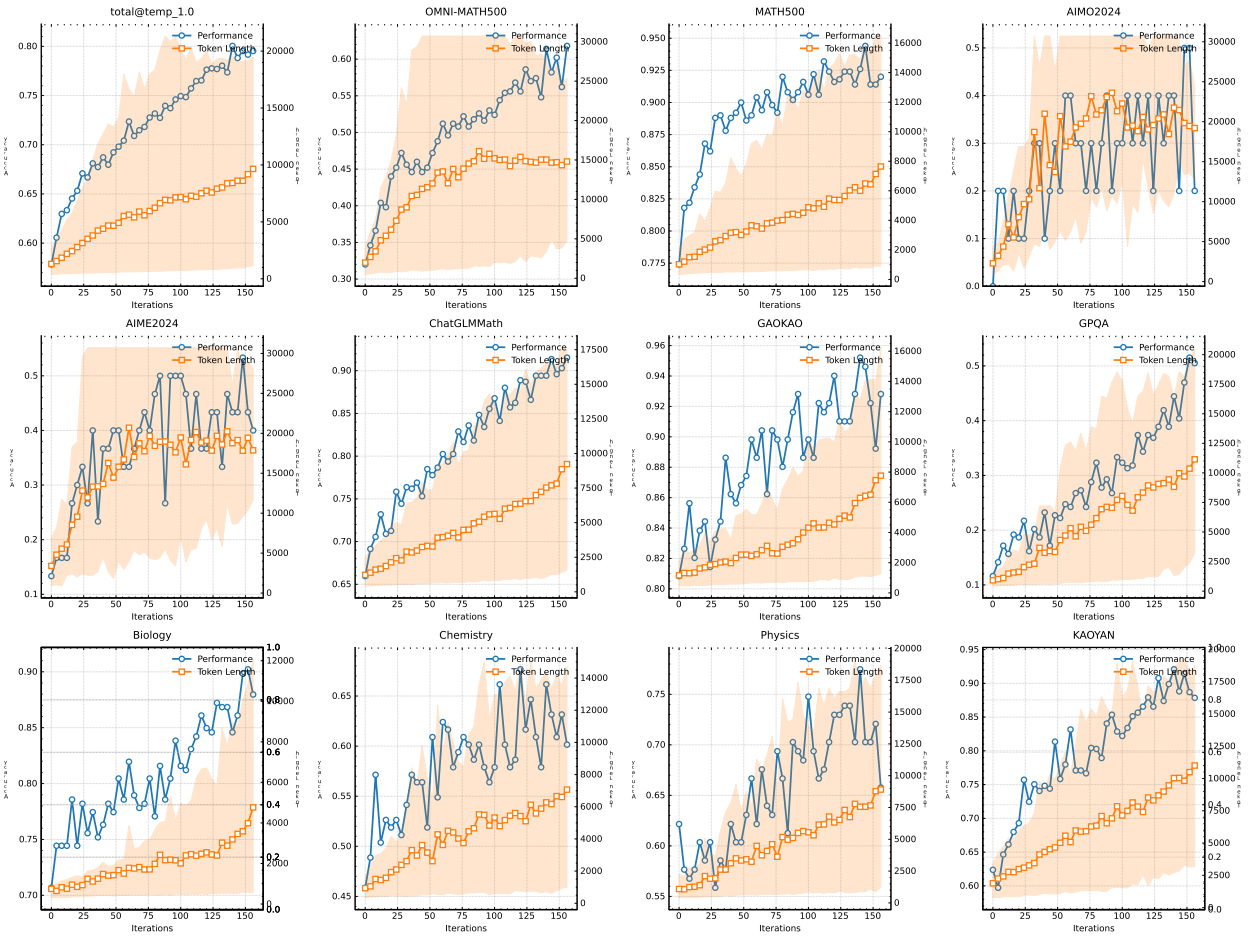


图5：随着训练迭代的增加，训练准确率和长度的变化。请注意，上述分数来自一个内部的长链模型，其模型大小远小于k1.5长链模型。阴影区域表示响应长度的95%分位数。

3.4 长期到短期

我们将提出的 long2short RL 算法与第 2.4 节中介绍的 DPO、最短拒绝采样和模型合并方法进行了比较，重点关注 long2short 问题的令牌效率（X. Chen et al. 2024），特别是获得的 long-cot 模型如何使短模型受益。在图 7 中，k1.5-long 代表我们为 long2short 训练选择的 long-cot 模型。k1.5-short w/ rl 指的是使用 long2short RL 训练获得的短模型。k1.5-short w/ dpo 表示通过 DPO 训练提高令牌效率的短模型。k1.5-short w/ merge 代表模型合并后的模型，而 k1.5-short w/ merge + rs 表示通过对合并模型应用最短拒绝采样获得的短模型。k1.5-shortest 代表我们在 long2short 训练期间获得的最短模型。如图 7 所示，提出的 long2short RL 算法在令牌效率上优于 DPO 和模型合并等其他方法。值得注意的是，k1.5 系列中的所有模型（标记为橙色）在令牌效率上优于其他模型（标记为蓝色）。例如，k1.5-short w/ rl 在 AIME2024 上的 Pass@1 得分为 60.8（平均 8 次运行），而平均仅使用 3,272 个令牌。同样，k1.5-shortest 在 MATH500 上的 Pass@1 得分为 88.2，同时消耗的令牌数量与其他短模型大致相同。

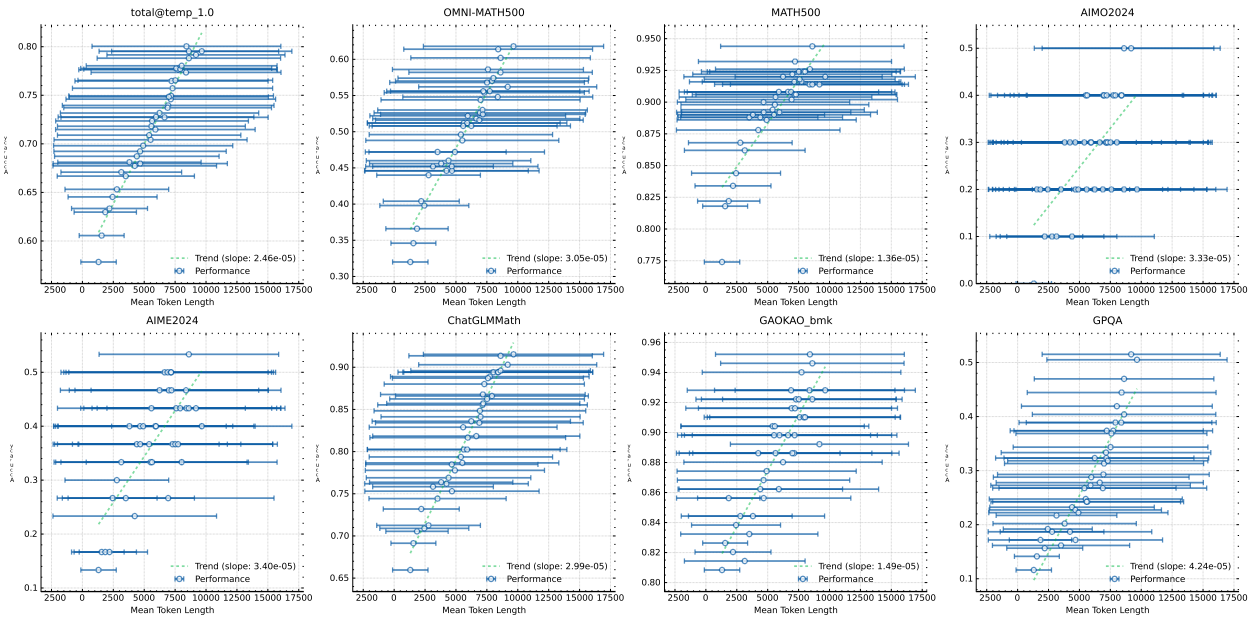


图6：模型性能随着响应长度的增加而提高

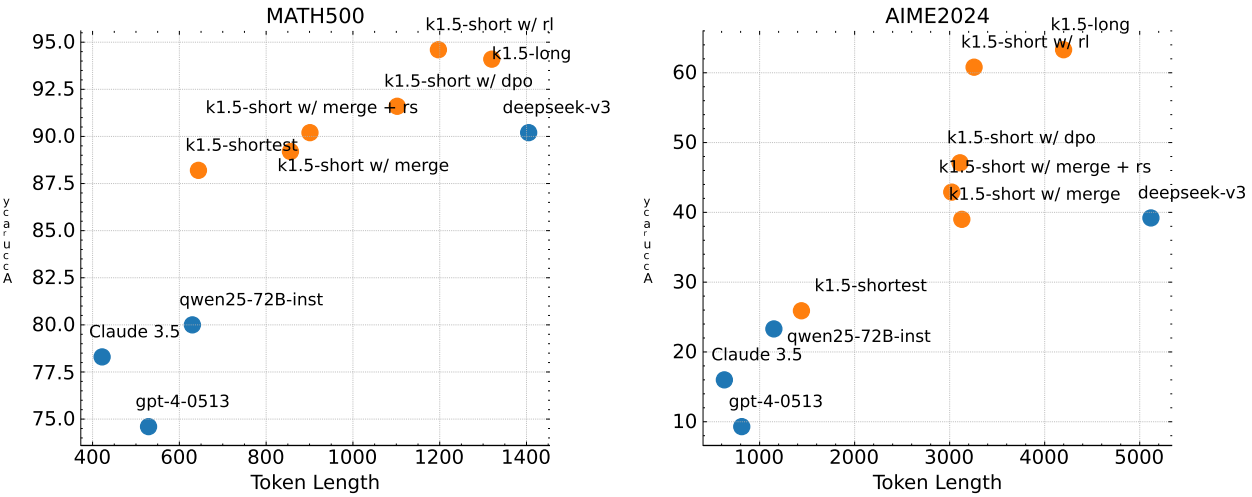


图7：Long2Short Pe 性能。所有k1.5系列展示了更好的令牌效率。 cy与其他模型相比。

3.5 消融研究

模型大小和上下文长度的扩展 我们的主要贡献是应用强化学习（RL）来增强模型生成扩展链式推理（CoT）的能力，从而提高其推理能力。一个自然的问题出现了：这与简单地增加模型大小相比如何？为了证明我们方法的有效性，我们使用相同的数据集训练了两个不同大小的模型，并记录了在RL训练期间所有检查点的评估结果和平均推理长度。这些结果如图8所示。值得注意的是，尽管较大模型最初的表现优于较小模型，但较小模型可以通过利用经过RL优化的更长CoT来实现可比的性能。然而，较大模型通常显示出比较小模型更好的标记效率。这也表明，如果目标是获得最佳性能，扩展较大模型的上下文长度具有更高的上限，并且更具标记效率。然而，如果测试时计算有预算，训练较小模型并使用更长的上下文长度可能是可行的解决方案。

使用负梯度的效果 我们研究了在我们的设置中使用 ReST (Gulcehre et al. 2023) 作为策略优化算法的有效性。ReST 与其他基于 RL 的方法之间的主要区别包括

我们的研究表明，ReST通过拟合当前模型中采样的最佳响应，迭代地优化模型，而不对错误响应施加负梯度以进行惩罚。如图10所示，我们的方法在样本复杂性上优于ReST，这表明引入负梯度显著提高了模型在生成长CoT时的效率。我们的方法不仅提升了推理的质量，还优化了训练过程，以更少的训练样本实现了稳健的性能。这个发现表明，在我们的设置中，策略优化算法的选择至关重要，因为ReST与其他基于RL的方法在其他领域的性能差距并不明显（Gulcehre等，2023）。因此，我们的结果强调了选择合适的优化策略以最大化生成长CoT的有效性的重要性。

采样策略 我们进一步展示了我们在第2.3.4节中介绍的课程采样策略的有效性。我们的训练数据集 $\mathcal{D}$ 包含了不同难度级别的多样化问题。通过我们的课程采样方法，我们最初使用 $\mathcal{D}$ 进行热身阶段，然后专注于难题来训练模型。该方法与一种基线方法进行比较，后者采用均匀采样策略，没有任何课程调整。如图9所示，我们的结果清楚地表明，所提出的课程采样方法显著提高了性能。这一改善可以归因于该方法逐步挑战模型的能力，使其能够在处理复杂问题时发展出更强的理解力和能力。在初步的总体介绍后，专注于更难的问题进行训练，模型能够更好地增强其推理和解决问题的能力。

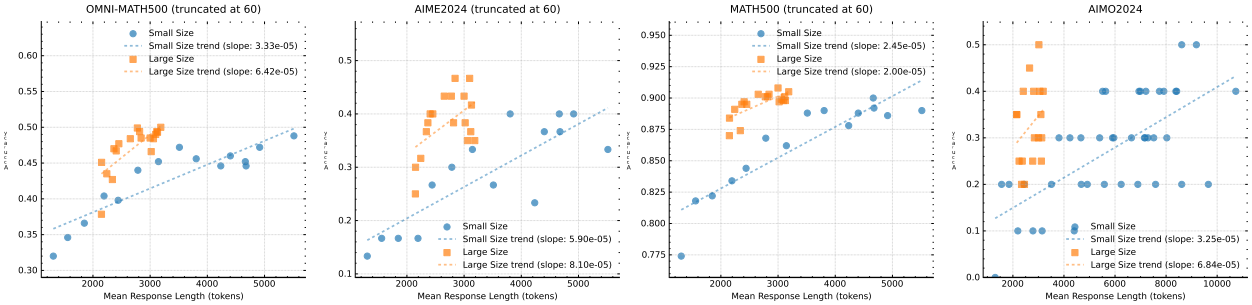


图 8：不同模型大小的模型性能与响应长度的关系

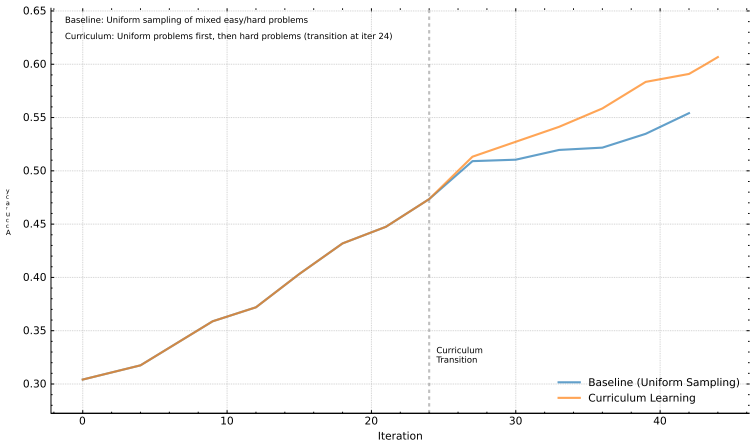


图 9：课程学习方法对模型性能的分析。

4 结论

我们介绍了k1.5的训练方案和系统设计，这是我们最新的通过强化学习训练的多模态LLM。我们从实践中提取的一个关键见解是，上下文长度的扩展对LLM的持续改进至关重要。我们采用了优化的学习算法和基础设施优化，例如部分回滚，以实现高效的长上下文强化学习训练。如何进一步提高长上下文强化学习训练的效率和可扩展性仍然是一个重要的问题。

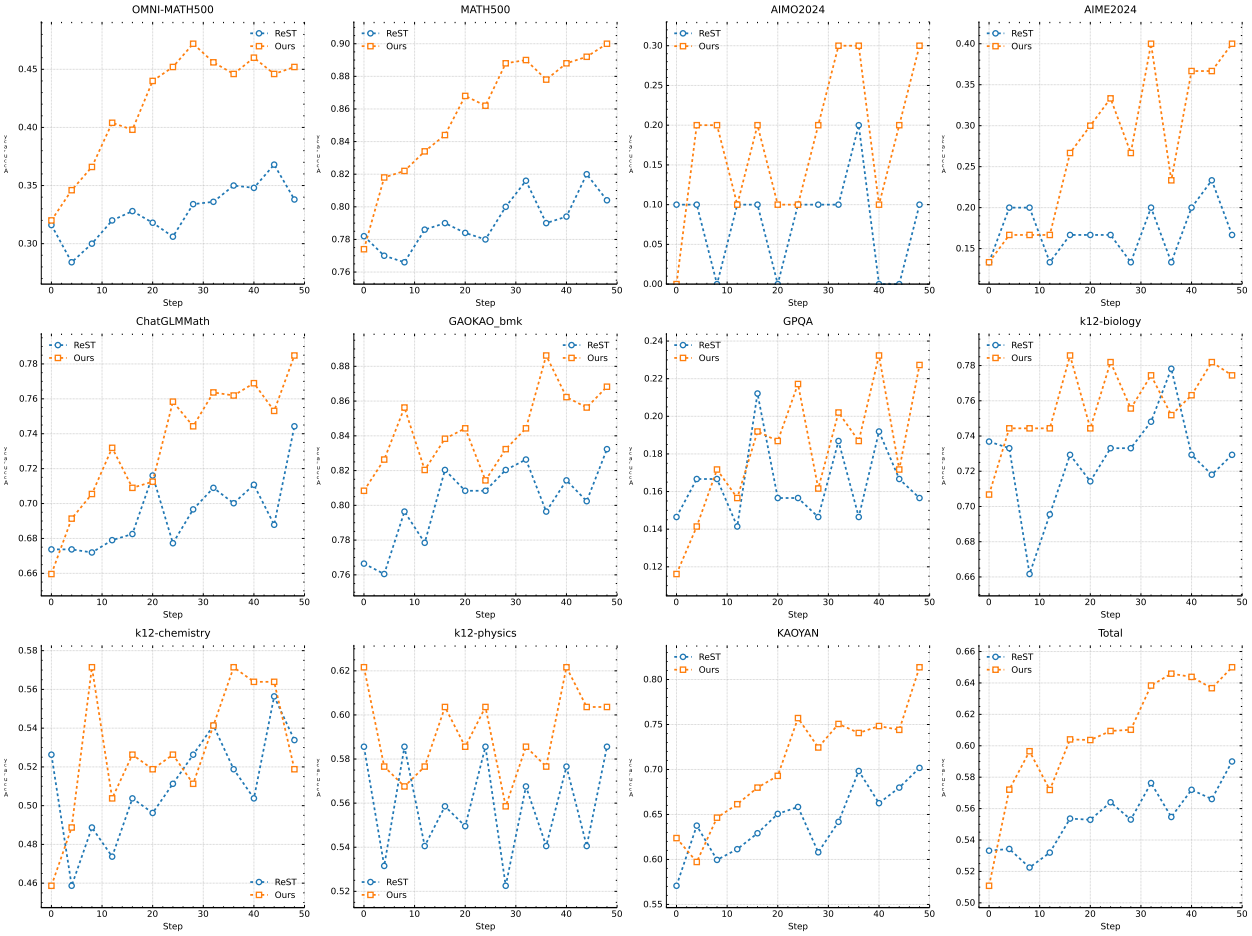


图10：使用ReST进行策略优化的比较。

我们所做的另一个贡献是结合多种技术，以实现更好的策略优化。具体来说，我们使用大型语言模型（LLMs）来构建长链思维（long-CoT）强化学习（RL），并推导出一种在线镜面下降（online mirror descent）的变体，以实现稳健优化。我们还实验了采样策略、长度惩罚以及优化数据配方，以实现强大的强化学习性能。

我们展示了通过长上下文扩展和改进策略优化可以实现强大的性能，即使不使用更复杂的技术，如蒙特卡洛树搜索、价值函数和过程奖励模型。在未来，研究改善信用分配和减少过度思考而不损害模型的探索能力也将是非常有趣的。

我们还观察到了长转短方法的潜力。这些方法在很大程度上提高了短CoT模型的性能。此外，可以以迭代的方式将长转短方法与长-CoT RL结合，以进一步提高令牌效率，并在给定的上下文长度预算中提取最佳性能。

参考文献

Abbasi-Yadkori, Yasin 等. “Politex: 使用专家预测的策略迭代的遗憾界限”。在: *International Conference on Machine Learning*。PMLR. 2019, 第 3692–3702 页。Ahmadian, Arash 等. “回归基础: 重新审视强化风格优化以从人类反馈中学习在 llms 中”。在: *arXiv preprint arXiv:2402.14740* (2024)。Ankner, Zachary 等. *Critique-out-Loud Reward Models*。2024. arXiv: 2408.11791 [cs.LG]。网址: https://arxiv.org/abs/2408.11791。Berner, Christopher 等. “使用大规模深度强化学习的 Dota 2”。在: *arXiv preprint arXiv:1912.06680* (2019)。

Cassano, Federico, John Gouwar, Daniel Nguyen, Sy Duy Nguyen 等. “MultiPL-E: 一种可扩展和可扩展的神经代码生成基准方法”。在: *ArXiv* (2022)。网址: https://arxiv.org/abs/2208.08227。Cassano, Federico, John Gouwar, Daniel Nguyen, Sydney Nguyen 等. “MultiPL-E: 一种可扩展和多语言的神经代码生成基准方法”。在: *IEEE Transactions on Software Engineering* 49.7 (2023), pp. 3675–3691。DOI: 10.1109/TSE.2023.3267446。Chen, Jianlv 等. “Bge m3-embedding: 通过自我知识蒸馏实现多语言、多功能、多粒度的文本嵌入”。在: *arXiv preprint arXiv:2402.03216* (2024)。Chen, Xingyu 等. “不要想太多 2+ 3=? 关于 o1 类 LLM 的过度思考”。在: *arXiv preprint arXiv:2412.21187* (2024)。Everitt, Tom 等. *Reward Tampering Problems and Solutions in Reinforcement Learning: A Causal Influence Diagram Perspective*。2021。arXiv: 1908.04734 [cs.AI]。网址: https://arxiv.org/abs/1908.04734。Gadre, Samir Yitzhak 等. “Datacomp: 寻找下一代多模态数据集”。在: *Advances in Neural Information Processing Systems* 36 (2024)。Grattafiori, Aaron 等. *The Llama 3 Herd of Models*。2024。arXiv: 2407.21783 [cs.AI]。网址: https://arxiv.org/abs/2407.21783。Gulcehre, Caglar 等. “用于语言建模的强化自我训练 (rest)”。在: *arXiv preprint arXiv:2308.08998* (2023)。Hendrycks, Dan 等. “测量大规模多任务语言理解”。在: *ArXiv* abs/2009.03300 (2020)。网址: https://arxiv.org/abs/2009.03300。Hoffmann, Jordan 等. *Training Compute-Optimal Large Language Models*。2022。arXiv: 2203.15556 [cs.CL]。网址: https://arxiv.org/abs/2203.15556。Huang, Yuzhen 等. “C-Eval: 一种多层次多学科的中文评估套件，用于基础模型”。在: *ArXiv* abs/2305.08322 (2023)。网址: https://arxiv.org/abs/2305.08322。Jaech, Aaron 等. “Openai o1 系统卡”。在: *arXiv preprint arXiv:2412.16720* (2024)。Jain, Naman 等. “LiveCodeBench: 对大型语言模型进行整体和无污染的代码评估”。在: *ArXiv* abs/2403.07974 (2024)。网址: https://arxiv.org/abs/2403.07974。Joulin, Armand 等. “高效文本分类的技巧包”。在: *arXiv preprint arXiv:1607.01759* (2016)。Kaplan, Jared 等. *Scaling Laws for Neural Language Models*。2020。arXiv: 2001.08361 [cs.LG]。网址: https://arxiv.org/abs/2001.08361。Kool, Wouter, Herke van Hoof 和 Max Welling. “购买 4 个强化样本，免费获得基线！”在: (2019)。Kwon, Woosuk 等. “使用 PagedAttention 的大型语言模型服务的高效内存管理”。在: *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*。2023。Laurençon, Hugo 等. “Obelics: 一个开放的网络规模过滤数据集，包含交错的图像-文本文档”。在: *Advances in Neural Information Processing Systems* 36 (2024)。Li, Jeffrey 等. “Datacomp-lm: 寻找下一代语言模型训练集”。在: *arXiv preprint arXiv:2406.11794* (2024)。Li, Ming 等. “从数量到质量: 通过自我引导的数据选择提升 LLM 性能以进行指令调优”。在: *arXiv preprint arXiv:2308.12032* (2023)。Li, Raymond 等. *StarCoder: may the source be with you!* 2023。arXiv: 2305.06161 [cs.CL]。网址: https://arxiv.org/abs/2305.06161。Lightman, Hunter 等. “让我们一步一步验证”。在: *arXiv preprint arXiv:2305.20050* (2023)。Liu, Wei 等. “什么样的数据适合对齐？关于指令调优中自动数据选择的综合研究”。在: *arXiv preprint arXiv:2312.15685* (2023)。Lozhkov, Anton 等. *StarCoder 2 and The Stack v2: The Next Generation*。2024。arXiv: 2402.19173 [cs.SE]。网址: https://arxiv.org/abs/2402.19173。Lu, Pan 等. “Mathvista: 在视觉上下文中评估基础模型的数学推理”。在: *arXiv preprint arXiv:2310.02255* (2023)。McAleese, Nat 等. *LLM Critics Help Catch LLM Bugs*。2024。arXiv: 2407.00215 [cs.SE]。网址: https://arxiv.org/abs/2407.00215。Mei, Jincheng 等. “关于策略优化中的原则性熵探索”。在: *Proceedings of the 28th International Joint Conference on Artificial Intelligence*。2019, pp. 3130–3136。Muennighoff, Niklas 等. *Scaling Data-Constrained Language Models*。2023。arXiv: 2305.16264 [cs.CL]。网址: https://arxiv.org/abs/2305.16264。Nachum, Ofir 等. “弥合基于价值和基于策略的强化学习之间的差距”。在: *Advances in neural information processing systems* 30 (2017)。OpenAI. “学习与 LLMs 推理”。在: (2024)。网址: https://openai.com/index/learning-to-reason-with-llms/。

欧阳龙等。“训练语言模型以遵循人类反馈的指令”。在：*Advances in neural information processing systems* 35 (2022)，第 27730–27744 页。潘亚历克斯，库什·巴蒂亚和雅各布·斯坦哈特。“奖励错误指定的影响：映射和缓解不对齐模型”。在：*International Conference on Learning Representations*。2022。网址：https://openreview.net/forum?id=JYtwGwIL7ye。帕斯特，基兰等。“Openwebmath：高质量数学网络文本的开放数据集”。在：*arXiv preprint arXiv:2310.06786* (2023)。佩内多，吉尔赫梅等。“fineweb 数据集：为最优文本数据大规模提取网络”。在：*arXiv preprint arXiv:2406.17557* (2024)。秦若宇等。*Mooncake: A KVCache-centric Disaggregated Architecture for LLM Serving*。2024。arXiv: 2407.00079 [cs.DC]。网址：https://arxiv.org/abs/2407.00079。拉法伊洛夫，拉法尔等。“直接偏好优化：你的语言模型实际上是一个奖励模型”。在：*Advances in Neural Information Processing Systems* 36 (2024)。舒曼，克里斯托夫等。“Laion-5b：用于训练下一代图像-文本模型的开放大规模数据集”。在：*Advances in Neural Information Processing Systems* 35 (2022)，第 25278–25294 页。肖伊比，穆罕默德等。*Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism*。2020。arXiv: 1909.08053 [cs.CL]。网址：https://arxiv.org/abs/1909.08053。西尔弗，大卫等。“在没有人类知识的情况下掌握围棋游戏”。在：*nature* 550.7676 (2017)，第 354–359 页。斯内尔，查理等。“在测试时优化 LLM 计算的规模可能比扩展模型参数更有效”。在：*arXiv preprint arXiv:2408.03314* (2024)。苏丹等。“Nemotron-CC：将 Common Crawl 转换为精炼的长时间预训练数据集”。在：*arXiv preprint arXiv:2412.02595* (2024)。苏建林等。“Roformer：带有旋转位置嵌入的增强型变换器”。在：*Neurocomputing* 568 (2024)，第 127063 页。团队，双子等。*Gemini: A Family of Highly Capable Multimodal Models*。2024。arXiv: 2312.11805 [cs.CL]。网址：https://arxiv.org/abs/2312.11805。托马尔，马南等。“镜像下降策略优化”。在：*arXiv preprint arXiv:2005.09814* (2020)。瓦斯瓦尼，阿希什等。“注意力是你所需要的一切”。在：*Advances in Neural Information Processing Systems*。由 I. Guyon 等编辑。第 30 卷。Curran Associates, Inc.，2017。网址：https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf。维拉洛博斯，巴勃罗等。*Will we run out of data? Limits of LLM scaling based on human-generated data*。2024。arXiv: 2211.04325 [cs.LG]。网址：https://arxiv.org/abs/2211.04325。维尼亚尔斯，奥里奥尔等。“使用多智能体强化学习在《星际争霸 II》中达到大师级水平”。在：*nature* 575.7782 (2019)，第 350–354 页。王可等。“使用数学视觉数据集测量多模态数学推理”。在：*arXiv preprint arXiv:2402.14804* (2024)。魏浩然等。“通用 OCR 理论：通过统一的端到端模型迈向 OCR-2.0”。在：*arXiv preprint arXiv:2409.01704* (2024)。魏杰森等。“链式思维提示引发大型语言模型中的推理”。在：*Advances in neural information processing systems* 35 (2022)，第 24824–24837 页。吴扬震等。“推理规模法则：对使用语言模型解决问题的计算最优推理的实证分析”。在：*arXiv preprint arXiv:2408.00724* (2024)。徐亮等。“CLUE：中文语言理解评估基准”。在：*International Conference on Computational Linguistics*。2020。网址：https://arxiv.org/abs/2004.05986。杨恩能等。“在 LLM、MLLM 及其他领域的模型合并：方法、理论、应用和机会”。在：*arXiv preprint arXiv:2408.07666* (2024)。姚顺宇等。“思维树：使用大型语言模型进行深思熟虑的问题解决”。在：*Advances in Neural Information Processing Systems* 36 (2024)。岳翔，袁胜尼等。“Mmmu：一个针对专家 AGI 的大规模多学科多模态理解和推理基准”。在：*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*。2024，第 9556–9567 页。岳翔，邢伟曲等。“Mammoth：通过混合指令调优构建数学通用模型”。在：*arXiv preprint arXiv:2309.05653* (2023)。张伦军等。“生成验证器：将奖励建模视为下一个令牌预测，2024”。在：*URL https://arxiv.org/abs/2408.15240* (2024)。郑连敏等。*SGLang: Efficient Execution of Structured Language Model Programs*。2024。arXiv: 2312.07104 [cs.AI]。网址：https://arxiv.org/abs/2312.07104。周杰弗里等。“大型语言模型的指令遵循评估”。在：*ArXiv* abs/2311.07911 (2023)。网址：https://arxiv.org/abs/2311.07911。

朱, 万荣等. “多模态c4：一个开放的、十亿规模的图像与文本交织的语料库”。在: *Advances in Neural Information Processing Systems* 36 (2024).

附录

A 贡献

研究与开发

安刚 杜博飞 高
长久 姜城 陈城
李晨俊 肖陈庄
杜德浩 张恩铭
袁恩哲 陆洪泛
宋国坤 赖海青
郭汉 朱浩 丁浩
胡浩 杨浩 张浩
天 姚浩天 赵浩
宇 陆宏城 高欢
袁华彬 郑景元
刘建林 苏建洲
王进 张俊杰 闫
立东 施龙辉 于
梦南 董尼奥 张
启伟 潘曲城 龚
书鹏 魏少伟 刘
涛 姜维民 熊维
然 何伟浩 高维
晓 黄文浩 吴文
扬 何先青 贾
徐欣然 朱新
宇 周新星 祖
阳 李阳阳 胡
阳阳 刘妍如
陈叶杰 王怡
道 秦逸博 刘
怡平 包宇伦
杜宇志 王宇
欣 吴Y. 查尔
斯 扎伊达 周
兆基 王兆伟
李正 张哲旭
王志奇 黄志
林 杨子尧 许
宗汉 杨

数据标注

唐春宁, 汪聪聪,
唐凤翔, 唐光达,
魏浩泽, 李浩珍,
余佳, 陈建航,
郭杰, 赵俊彦,
吴玲, 叶生灵,
马思涵, 曹思颖,
黄向辉, 魏颖,
杨阳阳, 刘震,
朱子豪, 黄

The listing of authors is in alphabetical order based on their first names.

B 预训练

强化学习（RL）的效率与基础模型的性能密切相关。前沿模型如Gemini（Team et al. 2024）和Llama（Grattafiori et al. 2024）突显了预训练数据质量在实现高性能中的重要性。然而，许多近期的开源模型在其数据处理流程和配方方面缺乏完全透明性，这给更广泛的社区理解带来了挑战。虽然我们目前不打算开源我们的专有模型，但我们致力于提供我们数据流程和方法论的全面披露。在本节中，我们主要关注多模态预训练数据配方，随后简要讨论模型架构和训练阶段。

B.1 语言数据

我们的预训练语料库旨在为训练大型语言模型（LLMs）提供全面且高质量的数据。它涵盖了五个领域：英语、中文、代码、数学与推理以及知识。我们对每个领域采用复杂的过滤和质量控制机制，以确保训练数据的最高质量。对于所有预训练数据，我们对每个数据源进行了严格的单独验证，以评估其对整体训练配方的具体贡献。这种系统的评估确保了我们多样化数据组成的质量和有效性。

我们开发了一个多维质量过滤框架，该框架结合了多种评分方法，以减少个体偏见并确保全面的质量评估。我们的框架包含：

1. 基于规则的过滤：我们实施特定领域的启发式方法来去除有问题的内容，包括重复内容、机器翻译文本和低质量的网络抓取。我们还过滤掉包含过多特殊字符、异常格式或垃圾邮件模式的文档。 2. 基于FastText的分类：我们训练了专门的FastText（Joulin et al. 2016; J. Li et al. 2024）模型，以根据语言特征和语义连贯性识别内容质量。这有助于识别具有自然语言流和适当语法结构的文档。 3. 基于嵌入的相似性分析：使用文档嵌入（Jianlv Chen et al. 2024），我们计算文档级相似性分数，以识别和去除近重复项，同时保留语义上有价值的变体。这种方法有助于保持我们训练语料库的多样性。 4. 基于LLM的质量评估：根据（Penedo et al. 2024），我们利用LLM对文档进行评分，基于连贯性、信息量和潜在教育价值。这种方法特别有效于识别简单方法可能遗漏的细微质量指标。

每个文档的最终质量分数是通过这些个体分数的组合计算得出的。基于广泛的实证分析，我们实施动态采样率，其中高质量文档在训练期间被上采样，而低质量文档则被下采样。

代码数据 代码数据主要由两类组成。对于从代码文件中提取的纯代码数据，我们遵循了BigCode (R. Li et al. 2023; Lozhkov et al. 2024)的方法论，并对数据集进行了全面的预处理。最初，我们消除了杂项语言，并应用了基于规则的清理程序以提高数据质量。随后，我们通过战略抽样技术解决了语言不平衡的问题。具体而言，JSON、YAML和YACC等标记语言被下采样，而包括Python、C、C++、Java和Go在内的32种主要编程语言则被上采样，以确保平衡的代表性。关于来自各种数据源的文本-代码交错数据，我们使用基于嵌入的方法来召回高质量数据。这种方法确保了数据的多样性并保持其高质量。

数学与推理数据 我们数据集中的数学和推理部分对于培养强大的分析和解决问题的能力至关重要。数学预训练数据主要来自于从公开可用的互联网来源收集的网页文本和PDF文档。(Paster et al. 2023) 最初，我们发现我们的通用领域文本提取、数据清理过程和OCR模型在数学领域表现出较高的假阴性率。因此，我们首先开发了专门针对数学内容的数据清理程序和OCR模型，旨在最大化数学数据的召回率。随后，我们实施了两阶段的数据清理过程：

1. 使用FastText模型进行初步清理，以去除大多数无关数据。

2. 利用微调的语言模型进一步清理剩余数据，从而生成高质量的数学数据。

知识数据 知识语料库经过精心策划，以确保在学术学科中的全面覆盖。我们的知识库主要由学术练习、教科书、研究论文和其他一般教育文献组成。这些材料中有很大一部分通过OCR处理进行了数字化，我们开发了专有模型，针对学术内容进行了优化，特别是处理数学公式和特殊符号方面。

我们使用内部语言模型为文档标注多维标签，包括：

1. 用于评估识别准确性的OCR质量指标 2. 衡量教学相关性的教育价值指标 3. 文档类型分类（例如，练习、理论材料）

基于这些多维注释，我们实现了一个复杂的过滤和采样管道。首先，文档通过OCR质量阈值进行过滤。我们的OCR质量评估框架特别关注检测和过滤常见的OCR伪影，特别是那些通常表明识别失败的重复文本模式。

超越基本的质量控制，我们通过评分系统仔细评估每个文档的教育价值。具有高教学相关性和知识深度的文档被优先考虑，同时保持理论深度和教学清晰度之间的平衡。这有助于确保我们的训练语料库包含高质量的教育内容，能够有效地促进模型的知识获取。

最后，为了优化我们训练语料库的整体组成，不同文档类型的采样策略通过广泛的实验经验确定。我们进行独立评估，以识别对模型知识获取能力贡献最大的文档子集。这些高价值子集在最终训练语料库中被上采样。然而，为了保持数据多样性并确保模型的泛化能力，我们仔细保留其他文档类型的平衡表示，按适当比例进行分配。这种数据驱动的方法帮助我们优化了专注于知识获取与广泛泛化能力之间的权衡。

B.2 多模态数据

我们的多模态预训练语料库旨在提供高质量的数据，使模型能够处理和理解来自多种模态的信息，包括文本、图像和视频。为此，我们还从五个类别中策划了高质量的数据——字幕、交错、OCR（光学字符识别）、知识和一般问答——以形成该语料库。

在构建我们的训练语料库时，我们开发了几个多模态数据处理管道，以确保数据质量，包括过滤、合成和去重。在视觉和语言的联合训练过程中，建立有效的多模态数据策略至关重要，因为它既保留了语言模型的能力，又促进了跨多种模态的知识对齐。

我们在本节中提供了对这些来源的详细描述，内容分为以下几个类别：

标题数据 我们的标题数据为模型提供了基本的模态对齐和广泛的世界知识。通过整合标题数据，多模态 LLM 获得了更广泛的世界知识和高效的学习能力。我们整合了各种开源的中文和英文标题数据集，如 (Schuhmann et al. 2022; S. Y. Gadre et al. 2024)，并从多个来源收集了大量内部标题数据。然而，在整个训练过程中，我们严格限制合成标题数据的比例，以减轻因现实世界知识不足而导致的幻觉风险。

对于一般的标题数据，我们遵循严格的质量控制流程，以避免重复并保持高图像-文本相关性。我们还在预训练期间改变图像分辨率，以确保视觉塔在处理高分辨率和低分辨率图像时仍然有效。

图像-文本交错数据 在预训练阶段，模型从交错数据中受益于多个方面，例如，通过交错数据可以增强多图像理解能力；交错数据总是为给定图像提供详细知识；通过交错数据还可以获得更长的多模态上下文学习能力。此外，我们还发现交错数据对维持模型的语言能力有积极贡献。因此，图像-文本交错数据是我们训练语料库的重要组成部分。我们的多模态

语料库考虑了开源的交错数据集，如（Zhu et al. 2024; Laurençon et al. 2024），并使用教科书、网页和教程等资源构建了大规模的内部数据。此外，我们还发现，合成交错数据有助于多模态LLM在保持文本知识方面的表现。为了确保每个图像的知识得到充分研究，对于所有交错数据，除了标准的过滤、去重和其他质量控制流程外，我们还集成了一种数据重新排序程序，以保持所有图像和文本的正确顺序。

OCR数据 光学字符识别（OCR）是一种广泛采用的技术，它将图像中的文本转换为可编辑格式。在k1.5中，强大的OCR能力被认为对更好地将模型与人类价值观对齐至关重要。因此，我们的OCR数据来源多样，从开源到内部数据集，涵盖了干净和增强的图像。

除了公开可用的数据，我们还开发了大量内部OCR数据集，涵盖多语言文本、密集文本布局、基于网络的内容和手写样本。此外，遵循OCR 2.0（H. Wei等，2024）中概述的原则，我们的模型还能够处理各种光学图像类型，包括图形、表格、几何图表、海洋图和自然场景文本。我们应用广泛的数据增强技术——如旋转、扭曲、颜色调整和噪声添加——以增强模型的鲁棒性。因此，我们的模型在OCR任务中达到了高水平的熟练度。

知识数据 多模态知识数据的概念类似于之前提到的文本预训练数据，不同之处在于我们专注于从多种来源汇集全面的人类知识库，以进一步增强模型的能力。例如，我们数据集中精心策划的几何数据对于发展视觉推理能力至关重要，确保模型能够解释人类创建的抽象图表。

我们的知识库遵循标准化的分类法，以平衡各个类别的内容，确保数据源的多样性。与仅包含文本的语料库类似，这些语料库从教科书、研究论文和其他学术材料中收集知识，多模态知识数据则使用布局解析器和OCR模型来处理来自这些来源的内容。同时，我们还包括来自互联网和其他外部资源的过滤数据。

由于我们知识库中很大一部分来源于基于互联网的材料，信息图表可能导致模型仅关注基于OCR的信息。在这种情况下，单靠基本的OCR管道可能会限制训练的有效性。为了解决这个问题，我们开发了一个额外的管道，更好地捕捉嵌入在图像中的纯文本信息。

一般问答数据 在训练过程中，我们观察到将大量高质量的问答数据集纳入预训练带来了显著的好处。具体而言，我们包括了针对基础、表格/图表问答、网络代理和一般问答等任务的严格学术数据集。此外，我们还编制了大量内部问答数据，以进一步增强模型的能力。为了保持难度和多样性的平衡，我们对一般问答数据集应用了评分模型和细致的手动分类，从而实现了整体性能的提升。

B.3 模型架构

Kimi k系列模型采用了一种变体的Transformer解码器（Vaswani等，2017），该解码器集成了多模态能力，并在架构和优化策略上进行了改进，如图11所示。这些进展共同支持稳定的大规模训练和高效的推理，特别针对大规模强化学习和Kimi用户的操作需求进行了定制。

广泛的扩展实验表明，大多数基础模型的性能来自于预训练数据质量和多样性的提升。关于模型架构扩展实验的具体细节超出了本报告的范围，将在未来的出版物中进行讨论。

B.4 训练阶段

Kimi k1.5 模型的训练分为三个阶段：视觉-语言预训练阶段、视觉-语言冷却阶段和长上下文激活阶段。Kimi k1.5 模型的每个训练阶段都专注于特定能力的增强。

视觉-语言预训练阶段 在这个阶段，模型首先仅在语言数据上进行训练，建立一个稳健的语言模型基础。然后，模型逐渐接触交错的视觉-语言数据，获得多模态能力。视觉塔最初在隔离状态下进行训练，而不更新语言模型参数，然后我们解冻语言模型层，最终增加视觉-文本数据的比例。

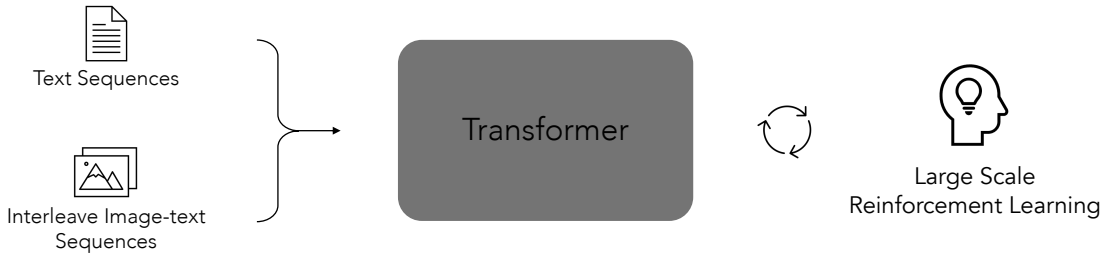


图11：Kimi k1.5 支持交错的图像和文本作为输入，利用大规模强化学习来增强模型的推理能力。

达到30%。最终的数据混合及其各自的权重是通过在较小模型上进行的消融研究确定的。

视觉语言冷却阶段 第二阶段作为冷却阶段，模型继续使用高质量的语言和视觉语言数据集进行训练，以确保卓越的性能。通过实证研究，我们观察到在冷却阶段引入合成数据显著提高了性能，特别是在数学推理、基于知识的任务和代码生成方面。冷却数据集的英语和中文部分来自于高保真度的预训练语料库子集。对于数学、知识和代码领域，我们采用混合方法：利用选定的预训练子集，同时用合成生成的内容进行增强。具体而言，我们利用现有的数学、知识和代码语料作为源材料，通过专有语言模型生成问答对，实施拒绝采样技术以保持质量标准（Yue, Qu, et al. 2023; D. Su et al. 2024）。这些合成的问答对在整合到冷却数据集之前经过全面验证。

长上下文激活阶段 最后，在第三阶段，k1.5使用上采样的长上下文冷却数据进行训练，使其能够处理扩展序列并支持需要更长上下文的任务。为了确保基础模型的优秀长文本能力，我们上采样了长上下文数据，并在长上下文训练期间使用了40%的全注意力数据和60%的部分注意力数据。全注意力数据部分来自高质量的自然数据，部分来自合成的长上下文问答和摘要数据。部分注意力数据来自冷却数据的均匀采样。RoPE频率（J. Su et al. 2024）设置为1,000,000。在此阶段，我们通过将最大序列长度从4,096逐渐增加到32,768，最终达到131,072，逐步扩展长度激活训练。

C 评估细节

C.1 文本基准测试

MMLU（Hendrycks 等，2020）涵盖了57个学科，包括STEM、人文学科、社会科学等。其难度从初级水平到高级专业水平不等，测试了世界知识和解决问题的能力。

IF-Eval (J. Zhou et al. 2023) 是一个评估大型语言模型遵循可验证指令能力的基准。共有500+个提示，指令包括“写一篇超过800字的文章”等。由于版本变更，表3中报告的IFEval数量来自一个中间模型。我们将根据最终模型更新分数。

CLUEWSC（L. Xu 等，2020）是 CLUE 基准中的一个共指消解任务，要求模型判断句子中的代词和名词短语是否共指，数据来自中文小说。

C-EVAL（Y. Huang 等，2023）是一个全面的中文评估套件，用于评估基础模型的高级知识和推理能力。它包含来自52个学科和四个难度级别的13,948道选择题。

C.2 推理基准

HumanEval-Mul 是 Multipl-E 的一个子集（Cassano, Gouwar, D. Nguyen, S. D. Nguyen 等，2022）。MultiPL-E 扩展了 HumanEval 基准和 MBPP 基准，支持 18 种语言，涵盖了一系列编程。

范式和流行度。我们选择了8种主流编程语言（Python、Java、Cpp、C#、JavaScript、TypeScript、PHP和Bash）的HumanEval翻译。

LiveCodeBench（Jain 等，2024）作为一个全面且无污染的基准，用于评估大型语言模型（LLMs）在编码任务中的表现。它具有实时更新以防止数据污染、在多个编码场景中的整体评估、高质量的问题和测试，以及平衡的问题难度。我们使用来自 2408-2411（发布 v4）的短链模型进行测试，以及使用来自 2412-2502（发布 v5）的长链模型进行测试。

AIME 2024 包含了 2024 年 AIME 的竞赛问题。AIME 是一项享有盛誉的仅限邀请的数学竞赛，面向顶尖高中生，评估高级数学技能，并要求扎实的基础和高水平的逻辑思维。

MATH-500（Lightman 等，2023）是一个综合性的数学基准，包含 500 道关于各种数学主题的问题，包括代数、微积分、概率等。测试计算能力和数学推理能力。更高的分数表明更强的数学问题解决能力。

Codeforces 是一个知名的在线评测平台，并且是评估长链条编码模型的热门测试平台。为了在 Div2 和 Div3 竞赛中获得更高的排名，我们对由 k1.5 长链条模型生成的代码片段进行多数投票，使用的测试用例也是由同一模型生成的。

C.3 图像基准测试

MMMU（Yue, Ni 等，2024）包含了一组精心策划的11.5K多模态问题，这些问题来源于大学考试、测验和教科书。这些问题涵盖六个主要学术领域：艺术与设计、商业、科学、健康与医学、人文学科与社会科学，以及技术与工程。

MATH-Vision (MATH-V) (K. Wang et al. 2024) 是一个精心策划的集合，包含 3,040 个高质量的数学问题，这些问题具有视觉背景，来源于真实的数学竞赛。它涵盖了 16 个不同的数学学科，并按 5 个难度级别进行分级。该数据集提供了一套全面而多样的挑战，非常适合评估 LMM 的数学推理能力。

MathVista（Lu et al. 2023）是一个基准，整合了来自各种数学和视觉任务的挑战，要求参与者展示细致的深度视觉理解以及组合推理，以成功完成任务。

本文由 AINLP 公众号通过大模型 API 进行翻译，更多资源请扫码关注！

长按扫码关注我们

AINLP

我爱自然语言处理

一个能聊天有趣有AI的NLP公众号