KIMI K1.5：

利用 LLM 扩展强化学习

KIMI K1.5 的技术报告

基米团队

摘要

利用下一个标记预测进行语言模型预训练已被证明对扩展计算有效，但却受限于可用的训练数据量。扩展强化学习（RL）为人工智能的持续改进打开了一条新的轴线，大型语言模型（LLM）有望通过学习探索奖励来扩展其训练数据。然而，之前发表的研究成果并不具有竞争力。有鉴于此，我们报告了使用 RL 训练的最新多模态 LLM Kimi k1.5 的训练实践，包括其 RL 训练技术、多模态数据配方和基础架构优化。长上下文扩展和改进的策略优化方法是我们方法的关键要素，它建立了一个简单有效的 RL 框架，而无需依赖蒙特卡罗树搜索、值函数和过程奖励模型等更复杂的技术。值得注意的是，我们的系统在多个基准和模式上都达到了最先进的推理性能，例如在 AIME 上达到 77.5，在 MATH 500 上达到 96.2，在 Codeforces 上达到 94%，在 MathVista 上达到 74.9，与 OpenAI 的 o1 相媲美。此外，我们还提出了有效的long2short方法，利用长CoT技术改进短CoT模型，从而获得最先进的短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--优于现有的短CoT模型，如GPT-4o和Claude Sonnet。
3.5 的幅度较大（高达 +550%）。kimi.ai 上的 Kimi k1.5 服务推出。

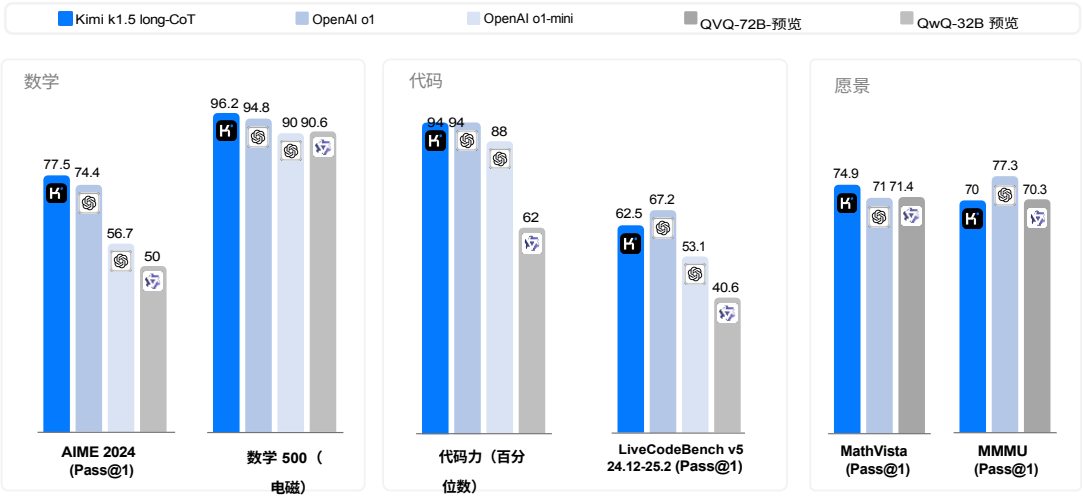


图 1：Kimi k1.5 long-CoT 结果。

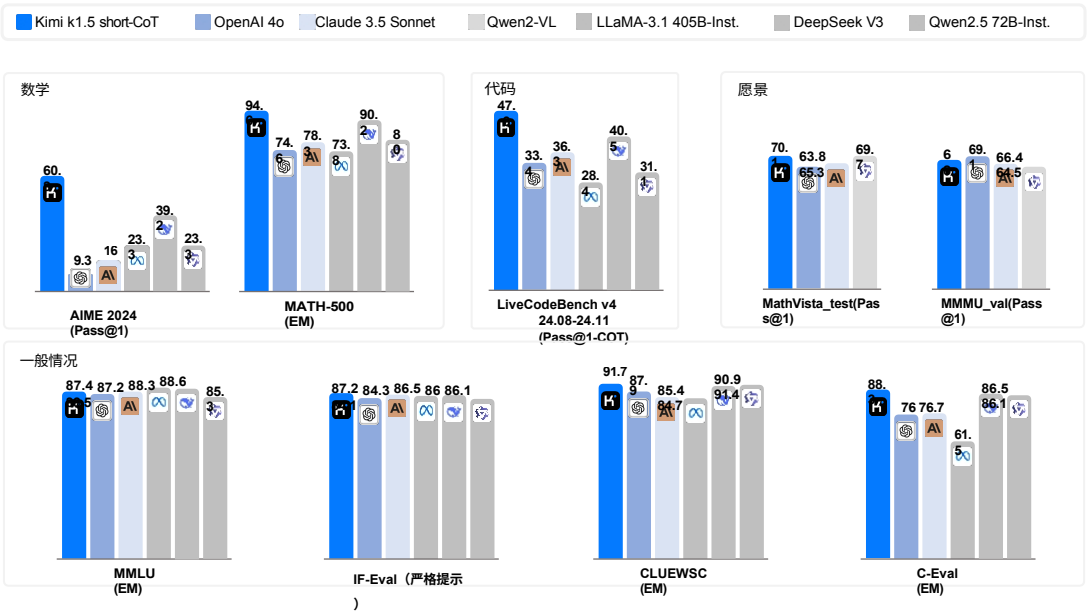


图 2：Kimi k1.5 短-CoT 结果。

1 导言

根据缩放定律，按比例缩放模型参数和数据大小会导致智能的持续提高。(Kaplan 等人，2020；Hoffmann 等人，2022）然而，这种方法受限于可用的高质量训练数据量（Villalobos 等人，2024；Muennighoff 等人，2023）。在本报告中，我们将介绍使用强化学习（RL）训练的最新多模态 LLM Kimi k1.5 的训练配方。我们的目标是探索一个可能的新轴心，以便继续扩展。利用 RL 与 LLM，模型学会了利用奖励进行探索，因此不局限于预先存在的静态数据集。

k1.5 的设计和培训有几个关键要素。

- **长上下文扩展**。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，性能也在不断提高。我们的方法背后的一个关键理念是利用部分滚动来提高训练效率，即通过重复使用以前的一大部分轨迹来采样新轨迹，从而避免了从头开始重新生成新轨迹的成本。我们的观察结果表明，上下文长度是 RL 与 LLM 持续扩展的一个关键维度。
- **改进策略优化**。我们推导出了一种具有长 CoT 的 RL 方案，并采用了在线镜像下降的变体来进行稳健的策略优化。我们的有效采样策略、长度惩罚和数据配方优化进一步改进了这一算法。
- **简化框架**。长上下文缩放与改进的策略优化方法相结合，为学习 LLM 建立了一个简单的 RL 框架。由于我们可以扩展上下文长度，因此学习到的 CoTs 具有规划、反射和修正的特性。增加上下文长度会增加搜索步骤的数量。因此，我们证明，无需依赖蒙特卡罗树搜索、值函数和过程奖励模型等更复杂的技术，也能实现强大的性能。
- **多模态**。我们的模型是在文本和视觉数据的基础上联合训练的，具有对两种模式进行联合推理的能力。

此外，我们还提出了有效的 long2short 方法，利用长 CoT 技术改进短 CoT 模型。具体来说，我们的方法包括利用

长-CoT 激活和模型合并进行长度惩罚。

我们的 Long-CoT 版本在多个基准和模式中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 上为第 94 百分位数，在 MathVista 上为 74.9，与 OpenAI 的 o1 相当。我们的模型还取得了最先进的短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--远远现有的短CoT模型，如GPT-4o和Claude Sonnet 3.5（最高达+550%）。结果如图 1 和图 2 。所示

2 方法：利用 LLM 进行强化学习

Kimi k1.5 的开发由几个阶段组成：预培训、vanilla 监督微调 (SFT)、long-CoT 监督微调和强化学习 (RL)。本报告的重点是强化学习，首先概述了强化学习的提示集整理（第 2.1 ）节和 long-CoT 监督微调（第 2.2 ），节然后在第 2.3 。节深入讨论了强化学习的训练策略有关预训练和 vanilla 监督微调的更多详情，请参阅第 2.5 。节

2.1 RL 提示集编辑

通过初步实验，我们发现 RL 提示集的质量和多样性对确保强化学习的有效性起着至关重要的作用。精心构建的提示集不仅能引导模型稳健的推理，还能降低奖励黑客和过度拟合肤浅模式的风险。具体来说，高质量的 RL 提示集有三个关键属性：

- **多元化覆盖**：提示应涵盖一系列学科，如科学、技术、工程和数学、编码和一般推理，以增强模型的适应性，确保其在不同领域的广泛适用性。
- **均衡难度**：提示集应包括容易、适中和困难的问题，以促进循序渐进的学习，防止过度适应特定的复杂程度。
- **准确的可评估性**：提示应允许核查人员进行客观可靠的评估，确保根据正确的推理而不是肤浅的模式或随意的猜测来衡量模型的性能。

为了实现提示集的多样化覆盖，我们采用了自动筛选器来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不同领域的问题，如科学、技术、工程和数学领域、竞赛和一般推理任务，其中既有纯文本数据，也有图像文本答题数据。此外，我们还开发了一个标签系统，按领域和学科对提示进行分类，确保不同学科领域的均衡代表性（M. Li 等人，2023 年；W. Liu 等人，2023 年）。

我们采用基于模型的方法，利用模型自身的能力来自适应地评估每个提示的难度。具体来说，对于每个提示，SFT 模型都会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表--通过率越低，难度越高。这种方法可以使难度评估与模型的内在能力保持一致，因此对 RL 训练非常有效。利用这种方法，我们可以预过滤大多数琐碎的案例，并在 RL 训练过程中轻松探索不同的抽样策略。

为了避免潜在的奖励黑客（Everitt 等人，2021 年；Pan 等人，2022 年），我们需要确保每个提示的推理过程和最终答案都能得到准确验证。经验观察显示，一些复杂的推理问题可能会有相对简单和容易猜到的答案，从而导致假阳性验证--即模型通过错误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，如多项选择题、真/假题和基于证明的问题。此外，对于一般的答题任务，我们提出了一种简单而有效的方法来识别和删除易被破解的提示。具体来说，我们会提示模型猜测可能的答案，而无需任何 CoT 推理步骤。如果模型在 N 次尝试内预测出正确答案，则认为该提示太容易被破解，并将其删除。我们发现，设置 $N= 8$ 可以删除大部分易被破解的提示。开发更先进的验证模型仍是未来研究的一个开放方向。

2.2 Long-CoT 监督微调

利用改进后的 RL 提示集，我们采用提示工程构建了一个小型但高质量的长 CoT 热身数据集，其中包含经过准确验证的文本和图像输入推理路径。这种方法类似于拒绝采样 (RS)，但重点是通过提示工程生成长 CoT 推理路径。

由此产生的热身数据集旨在囊括对类人推理至关重要的关键认知过程，例如**规划**，即模型在执行前系统地列出步骤；**评估**，包括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。通过在热身数据集上执行轻量级 SFT，我们有效地将这些推理策略内化到模型中。因此，经过微调的长CoT模型在生成更详细、逻辑更连贯的回答方面表现出了更强的能力，从而提高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

给定训练数据集 $D=\{(x_i, y_i^*)\}_{i=1}^n$ 的问题 x_i 和相应的地面真实答案 y_i^*，我们的目标是训练策略模型 π_θ，以准确解决测试问题。在复杂推理中，将问题 x 映射到解决方案 y 并非易事。为了应对这一挑战，*思维链*（CoT）方法建议使用一系列中间步骤 $z=(z_1, z_2, \ldots)$ 其中每个 z_i 都是一个连贯的标记序列，是解决问题的重要中间步骤（J. Wei 人，2022 年）。当解决 x 问题时，思想 $z_t \sim \pi_\theta(\cdot|x, z_1, \ldots, z_{t-1})$ 进行自动回归采样，然后得出最终答案 $y \sim \pi_\theta(\cdot|x, z_1, \ldots, z_m)$。我们用 $y, z \sim \pi_\theta$ 来表示这个抽样过程。请注意，想法和最终答案都是作为语言序列进行抽样的。

为了进一步增强模型的推理能力，*规划*算法被用来探索各种思维过程，从而在推理时生成改进的CoT（Yao等人，2024年；Y. Wu等人，2024年；Snell等人，2024年）。这些方法的核心洞察力在于明确构建一棵由价值估计引导的思维搜索树。这样，模型就能探索思维过程的不同延续，或在遇到死胡同时回溯到新的方向。更详细地说，假设 T 是一棵搜索树，其中每个节点代表一个部分解 $s=(x, z_{1:|s|})$。这里的 s 包括问题 x 和一系列想法 $z_{1:|s|}=(z_1, \ldots, z_{|s|})$，其中 $|s|$ 表示序列中的想法数。规划算法使用评论模型 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度，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.）在所有 $s \in T$ 的反馈指导下，规划算法会选择最有希望的节点进行扩展，从而不断扩大搜索树。上述过程反复进行，直到得出完整的解决方案。

我们还可以从*算法的角度来*研究规划算法。给定*第* t 次迭代时的搜索历史记录 $(s_1, v(s_1), \ldots, s_{t-1}, v(s_{t-1}))$，规划算法 A 通过迭代确定下一个搜索方向 $A(s_t|s_1, v(s_1), \ldots, s_{t-1}, v(s_{t-1}))$ 并为当前搜索进度提供反馈 $A(v(s_t)|s_1, v(s_1), \ldots, s_t)$。由于思考和反馈都可以看作是中间推理步骤，而且这些部分都可以用语言标记序列来表示，因此我们用 z 来代替 s 和 v，以简化符号。因此，我们将规划算法视为直接作用于推理步骤序列 $A(\cdot|z_1, z_2, \ldots)$*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：与明确构建搜索树和实施规划算法相比，我们可以训练一个模型来近似这一过程。在这里，思维（即语言标记）的数量可以类比传统上分配给规划算法的计算预算。长语境窗口的最新进展有助于在训练和测试阶段实现无缝扩展。如果可行，这种方法可使模型直接通过自动回归预测对推理空间进行隐式搜索。因此，模型不仅能学会解决一系列训练问题，还能发展出有效解决单个问题的能力，从而提高对未见测试问题的泛化能力。

因此，我们考虑用强化学习（RL）（OpenAI 2024）。来训练模型以生成 CoT让 r 成为奖励模型，通过赋值 $r(x, y, y^*) \in \{0, 1\}$ 来证明基于基本事实 y^* 的给定问题 x 的建议答案 y *的*正确性。对于可验证问题，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。对于具有自由形式基本事实的问题，我们会训练一个奖励模型 $r(x, y, y^*)$ 来预测答案是否与基本事实相符。给定问题 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 的质量取决于它是否能得出正确的最终答案。总之，我们认为优化策略的目标如下

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

通过扩大 RL 训练规模，我们的目标是训练出一种模型，利用基于简单提示的 CoT 和规划增强型 CoT 的优势。该模型在推理过程中仍会自动回归采样语言序列，从而避免了高级规划算法在部署过程中所需的复杂并行化。然而

，与简单的基于提示的方法一个关键区别是，模型不应仅仅遵循一系列推理步骤。相反，它还应学习关键的规划技能，包括错误识别、回溯和解决方案完善，方法是利用作为上下文信息的整套已探索思路。

2.3.2 政策优化

我们采用在线策略镜像算法的变体作为训练算法（Abbasi-Yadkori 等，2019 年；Mei 等，2019 年；Tomar 等，2020 年）。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策略优化问题、

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

其中，$\tau > 0$ 是控制正则化程度的参数。该目标有一个封闭式解

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

这里 $Z = \sum_{y',z'} \pi_{\theta_i}(y', z' \mid x) \exp(r(x, y', y^*)/\tau)$ 是归一化系数。对两边取对数得
对于*任意*(y, z) 都满足以下约束条件，这使我们能够在优化过程中利用非政策数据

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

因此，代用损失如下

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

为了近似 $\tau \log Z$，我们使用样本 (y_1, z_1)，……。，$(y_k, z_k) \sim \pi_{\theta_i}$：$\tau \log Z \approx \tau \log \frac{1}{k} \sum_{j=1}^{k} \exp(r(x, y_j, y^*)/\tau)$.
我们还发现，使用采样奖励的经验平均值 $\bar{r} = \mathrm{mean}(r(x, y_1, y^*), \ldots, r(x, y_k, y^*))$ 会产生有效的实际结果。这是合理的，因为当 $\tau \to \infty$ 时，$\tau \log Z$ 接近 π_{θ_i}下的预期奖励。最后，我们通过代偿损失的梯度来结束我们的学习算法。对于每个问题 x，使用参考策略 π_{θ_i}对 k 个响应进行采样，梯度为

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

对于熟悉政策梯度方法的人来说，这种梯度类似于以采样奖励平均值为基线的 (2) 政策梯度（Kool 等，2019 年；Ahmadian 等人，2024 年）。主要区别在于，响应是从 π_{θ_i}而非政策上采样的，并且应用了 l_2 规则化。因此，我们可以把它看作是通常的政策上正则化政策梯度算法在非政策情况下的自然扩展（Nachum 等，2017 年）。我们从 D 中抽取一批问题，并将参数更新为 θ_{i+1}，作为下一次迭代的参考策略。由于每次迭代都会因参考策略的变化而考虑不同的优化问题，因此我们也会在每次迭代开始时重置优化器。

在我们的训练系统中，我们排除了价值网络，这在以前的研究中也被利用过（Ahmadian 等，2024 年）。虽然这一设计选择大大提高了训练效率，但我们也假设，在经典 RL 中使用价值函数进行学分分配的传统方法可能并不适合我们的情况。考虑以下情况
模型生成了部分 CoT $(z_1, z_2, \ldots, z_t)$，并且有两个潜在的下一步推理步骤：z_{t+1} 和 z'_{t+1}。
假设 z_{t+1}能直接得出正确答案，而 z'_{t+1} 包含一些错误。如果甲骨文值函数是
如果z_{t+1}的值比z'_{t+1}的值更高，则表明z_{t+1}的值更高。根据标准信用
原则上，选择 z'_{t+1} 将受到惩罚，因为相对于现行政策，它具有负面优势。
然而，探索 z'_{t+1} 对于训练模型生成长 CoT 极具价值。利用
如果把长 CoT 得出的最终答案作为奖励信号，模型就能从试错模式中学习出
只要它能成功恢复并得出正确答案，就可以取 z'_{t+1}。本例的主要启示是
我们应该鼓励模型探索不同的推理路径，以增强其解决复杂问题的能力。这种探索性的方法会产生丰富的经验，有助于关键规划技能的发展。我们的主要目标并不局限于在训练问题上获得高准确率，而是侧重于为模型配备有

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 长度处罚

我们观察到一种过度思考现象，即在 RL 训练过程中，模型的响应长度显著增加。虽然这会带来更好的性能，但在训练和推理过程中，过长的推理过程代价高昂，而且人类往往不喜欢过度思考。为了解决这个问题，我们引入了长度奖励来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 个采样响应

$(y_1, z_{(1)})$，…。，(y_k, z_k) of problem x with true answer y^*，设 len(i) 为 $(y_i, z_{(i)})$ 的长度，min_len= $\min_{(i)}$len($i_{)}$, max_len= $\max_{(i)}$len$_{(i)}$。如果 max_len= min_len，我们会将所有回复的长度奖励设为零，因为它们的长度相同。否则，长度奖励的计算公式为

$$\text{len\_reward(i=} \begin{matrix} \lambda & \text{If } r(x, y_i, y)^* = 1 \\ \min(0, \lambda) & \text{If } r(x, y_i, y)^* = 0 \end{matrix}_{(.)}, \quad \text{where } \lambda = 0(.)(5) - \frac{\text{len}(i) -}{\text{max\_len} - \text{min\_len}}$$

从本质上讲，我们鼓励较短回答，惩罚正确回答中较长的回答，同时明确惩罚错误的长回答。这种基于长度的奖励会在原始奖励的基础上加上一个加权参数。

在我们的初步实验中，长度惩罚可能会减慢初始阶段的训练速度。为了缓解这一问题，我们建议在训练过程中逐渐预热长度惩罚。具体来说，我们先采用不含长度惩罚的标准策略优化，然后在剩余的训练中采用恒定的长度惩罚。

2.3.4 取样策略

虽然 RL 算法本身具有相对较好的采样特性（难度越大的问题梯度越大），但其训练效率有限。因此，一些定义明确的先验采样方法可能会带来更大的性能提升。我们利用多种信号来进一步改进采样策略。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，数学竞赛题比小学数学题更难。其次，由于 RL 训练过程会对同一问题进行采样，因此我们还可以跟踪每个问题的成功率，以此来衡量难度。我们提出了两种利用这些先验来提高训练效率的采样方法。

课程采样 我们从较容易的任务开始训练，然后逐步过渡到更具挑战性任务。由于初始 RL 模型的性能有限，将有限的计算预算花在非常难的问题上往往只能得到很少的正确样本，从而降低了训练效率。同时，我们收集的数据自然包括等级和难度标签，因此基于难度的抽样是提高训练效率的直观而有效的方法。

优先抽样 除了课程抽样外，我们还使用优先抽样策略来关注模型表现不佳的问题。我们跟踪每个问题 i 的成功率 s_i，并按照 $1 - s_i$ 的比例对问题进行抽样，这样成功率较低的问题就能获得较高的抽样概率。这将引导模型向其最薄弱的地方努力，从而加快学习速度，提高整体性能。

2.3.5 培训食谱的更多详情

编码测试用例生成 由于许多编码问题无法从网络上获得测试用例，因此我们设计了一种自动生成测试用例的方法，这些测试用例可作为使用 RL 训练模型的奖励。我们的重点主要放在不需要特殊法官的问题上。我们还假定这些问题都有地面实况解决方案，这样我们就可以利用这些解决方案生成更高质量的测试用例。

我们利用广受认可的测试用例生成库 CYaRon[1] 来增强我们的方法。我们使用基于问题陈述的 Kimi k1.5 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，并为每个测试用例随机抽取 10 份地面实况报告。我们根据提交的内容运行测试用例。如果 10 份提交材料中至少有 7 份产生了匹配结果，则该测试用例被视为有效。这一轮筛选，我们得到了一组选定的测试用例。如果在 10 个提交的测试用例中至少有 9 个通过了整个测试用例集，那么问题及其相关的测试用例就会被添加到我们的训练集中。

从统计数据来看，在 1000 个在线竞赛问题样本中，约有 614 个问题不需要特别评委。我们开发了 463 个测试用例生成器，这些生成器至少能生成 40 个有效测试用例，因此我们的训练集中包含了 323 个问题。

数学奖励模型 评估数学解决方案的一个挑战是，不同的书面形式可以代表相同的基本答案。，a^2-4 和 $(a+2)(a-2)$可能都是同一问题的有效解决方案。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：从 InstructGPT（欧阳等人，2022 年）方法中汲取灵感，我们实施了基于值头的奖励模型，并收集了约 800k 个数据点进行微调。该模型最终

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

将 "问题"、"参考答案 "和 "回答 "作为输入，并输出一个标量来表示回答是否正确。

2. 思维链 RM：最近的研究（Ankner 等，2024 年；McAleese 等，2024 年）表明，使用思维链（CoT）推理增强的奖励模型可以大大优于传统方法，尤其是在数学等涉及细微正确性标准的任务中。因此，我们收集了一个同样庞大的数据集，其中包含约 80 万个 CoT 标签示例，用于微调 Kimi 模型。基于与经典 RM 相同输入，思维链方法在以 JSON 格式提供最终正确性判断之前，会明确生成一个逐步推理的过程，从而使奖励信号更具鲁棒性和可解释性。

在人工抽查中，经典 RM 的准确率约为 **84.4**，而思维链 RM 的准确率则达到了 **98.5**。在 RL 培训过程中，我们采用了思维链 RM，以确保更正确的反馈。

视觉数据 为了提高模型的真实世界图像推理能力，并在视觉输入和大型语言模型（LLM）之间实现更有效的协调，我们的视觉强化学习（Vision RL）数据主要来自三个不同的类别：真实世界数据、合成视觉推理数据和文本渲染数据。

1. 真实世界的数据包括不同年级的一系列需要图形理解和推理的科学问题、需要视觉感知和推理的位置猜测任务，以及需要理解复杂图表的数据分析等类型的数据。这些数据集提高了模型在真实世界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，包括程序化创建的图像和场景，旨在提高特定的视觉推理技能，如理解空间关系、几何图案和物体互动。这些合成数据集为测试模型的视觉推理能力提供了受控环境，并提供了无穷无尽的训练示例。
3. 文本渲染数据是通过将文本内容转换为可视化格式而创建的，从而使模型在不同模式下处理基于文本的查询时保持一致性。通过将文本文档、代码片段和结构化数据转换为图像，无论输入的是纯文本还是渲染为图像（如屏幕截图或照片）的文本，我们都能确保模型提供一致的响应。这也有助于增强模型在处理文本较多的图像时的能力。

每种类型的数据对于建立一个全面的视觉语言模型都至关重要，该模型可有效管理现实世界中的各种应用，同时确保在各种输入模式下性能一致。

2.4 Long2short：短 CoT 模型的上下文压缩

虽然长CoT模型的性能很强，但与标准的短CoT LLM相比，它需要消耗更多的测试时间令牌。不过，可以将长-CoT 模型的思维先验转移到短-CoT 模型中，这样即使测试时间令牌预算有限，也能提高性能。我们针对这个长2短问题介绍了几种方法，包括模型合并（Yang 等人，2024 年）、最短拒绝采样、DPO（Rafailov 等人，2024 年）和长2短 RL。下文将详细介绍这些方法：

模型合并 研究发现，模型合并有助于保持泛化能力。我们还发现，在合并长点模型和短点模型时，它在提高标记效率方面也很有效。这种方法将长点模型与短点模型合并，无需训练即可得到一个新模型。具体来说，我们合并两个模型时，只需简单地求出它们的权重平均值。

最短剔除采样法 我们发现，我们的模型在处理同一问题时会产生长度差异很大的响应。基于这一点，我们设计了最短拒绝采样法。该方法对同一问题进行 n 次采样（在我们的实验中，$n=8$），并选择最短的正确回答进行监督微调。

DPO 与最短拒绝采样类似，我们利用长 CoT 模型生成多个响应样本。最短的正确解决方案被选为正样本，而较长的回复则被视为负样本，包括错误的较长回复和正确的较长回复（比所选正样本长 1.5 倍）。这些正负对构成了用于 DPO 训练的成对偏好数据。

长2短 RL 在标准 RL 训练阶段之后，我们会选择一个在性能和令牌效率之间达到最佳平衡的模型作为基础模型，并进行单独的长2短 RL 训练阶段。在第二阶段中，我们将采用第 2.3.3 节中介绍的长度惩罚，并大幅减少最大滚动长度，以进一步惩罚超出所需长度但可能正确的响应。

2.5 其他培训细节

2.5.1 预培训

Kimi k1.5 基础模型是在一个多样化、高质量的多模态语料库中训练出来的。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、图像-文本交错、OCR、知识和 QA 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个预训练数据集的相关性、多样性和平衡性。我们的预训练分三个阶段进行：(1) 视觉语言预训练，建立坚实的语言基础，然后逐步进行多模态整合；(2) 冷却，利用策划和合成数据巩固能力，特别是推理和基于知识的任务；(3) 长语境激活，将序列处理扩展到 131,072 个标记。有关预培训工作的更多详情，请参阅附录 B。

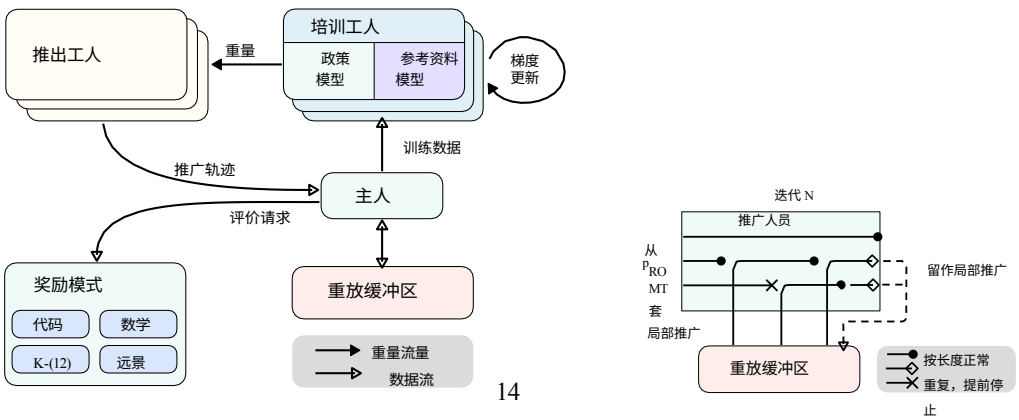
2.5.2 香草监督微调

我们创建的虚构 SFT 语料库涵盖多个领域。对于非推理任务，包括问题解答、写作和文本处理，我们首先通过人工标注构建一个种子数据集。该种子数据集用于训练种子模型。随后，我们收集各种提示，并利用种子模型为每个提示生成多个回复。然后，注释者对这些回答进行排序，并完善排序靠前的回答，生成最终版本。对于数学和编码问题等推理任务，基于规则和奖励建模的验证比人工判断更准确、更高效，因此我们利用拒绝抽样来扩展 SFT 数据集。

我们的普通 SFT 数据集包含约 100 万个文本示例。其中，50 万个示例用于一般问题解答，20 万个用于编码，20 万个用于数学和科学，5 千个用于创意写作，2 万个用于摘要、文档问答、翻译和写作等长语境任务。此外，我们还构建了 100 万个文本视觉示例，涵盖各种类别，包括图表解读、OCR、图像对话、视觉编码、视觉推理以及带有视觉辅助工具的数学/科学问题。

我们首先以 32k 个字节的序列长度对模型进行 1 次历时训练，然后再以 128k 个字节的序列长度进行另一次历时训练。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）回升到 $1\times 10^{-(5)}$，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训练序列中。

2.6 RL 基础设施



(a) 系统概览 (b) 部分推广

图 3：用于 LLM 的大规模强化学习训练系统

2.6.1 用于 LLM 的大规模强化学习训练系统

在人工智能领域，强化学习（RL）已成为大型语言模型（LLMs）（2022）（Jaech Ouyang et al. et al. 2024）的重要训练方法，其灵感来自于通过AlphaGo（Silver et al. 2017）、AlphaStar（Vinyals et al. 2019）和OpenAI Dota Five（Berner et al. 2019）。等系统掌握围棋、星际争霸II和Dota 2等复杂游戏的成功经验秉承这一传统，Kimi k1.5 系统采用了迭代同步 RL 框架，其精心设计旨在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滚动技术，旨在优化复杂推理轨迹的处理。

如图 3a 所示，RL 训练系统通过同步迭代的方式运行，每次迭代包括展开阶段和训练阶段。在推出阶段，推出工作者在中央主控的协调下，通过与模型交互生成推出轨迹，对各种输入做出响应序列。这些轨迹随后被存储在重放缓冲区中，通过破坏时间相关性，确保为训练提供多样化和无偏见的数据集。在随后的训练阶段，训练员会利用这些经验来更新模型的权重。这一循环过程使模型能够不断从其行动中学习，并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器，负责管理数据流以及推广工作者、培训工作者、奖励模型评估和重放缓冲器之间的通信。它确保系统和谐运行，平衡负载，促进高效的数据处理。

无论是在一次迭代中完成，还是在多次迭代中完成，训练员都会访问这些推出轨迹，以计算梯度更新，从而完善模型参数并提高其性能。这一过程由奖励模型监督，奖励模型对模型输出的质量进行评估，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达到最佳性能方面尤为关键。

此外，该系统还包含一个代码执行服务，专门用于处理与代码相关的问题，是奖励模型不可或缺的一部分。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世界的编程挑战紧密结合。通过根据实际代码执行情况验证模型的解决方案，这一反馈回路对于完善模型的策略和提高其在代码相关任务中的性能至关重要。

2.6.2 长 CoT RL 的部分推出

我们工作的主要思路之一是扩展长语境 RL 训练。部分滚动是一项关键技术，它通过管理长轨迹和短轨迹的滚动，有效地应对了处理长语境特征的挑战。该技术建立了固定的输出令牌预算，为每个滚动轨迹的长度设定了上限。如果轨迹在推出阶段超过了令牌限制，未完成的部分将保存到重放缓冲区，并在下一次迭代中继续。这就确保了没有任何一条冗长的轨迹会垄断系统资源。此外，由于滚动工作者是异步操作的，因此当一些工作者在处理较长的轨迹时，其他工作者可以独立处理新的、较短的滚动任务。异步操作确保了所有滚动工人都能积极地参与到训练过程中，从而最大限度地提高了计算效率，优化了系统的整体性能。

如图 3b ，所示部分滚动系统的工作原理是将长响应分解为跨迭代（从迭代 n-m 到迭代 n）的响应段。重放缓冲区作为中央存储机制，负责维护这些响应段，其中只有当前迭代（迭代 n）需要进行策略计算。之前的分段（迭代 n-m 至 n-1）可以从缓冲区中有效地重复使用，从而消除了重复滚动的需要。这种分段方法大大降低了计算开销：系统不是一次性推出整个响应，而是以增量方式处理和存储分段，这样既能生成更长的响应，又能保持快速的迭代时间。在训练过程中，可以将某些片段排除在损失计算之外，以进一步优化学习过程，从而使整个系统既高效又可扩展。

部分滚动的实现还提供了重复检测功能。该系统能识别生成内容中的重复序列，并提前终止它们，从而在保持输出质量的同时减少不必要的计算。被检测到的重复内容可被处以额外的惩罚，从而有效阻止提示集中多余内容生成。

2.6.3 训练与推理的混合部署

RL 训练过程包括以下几个阶段：

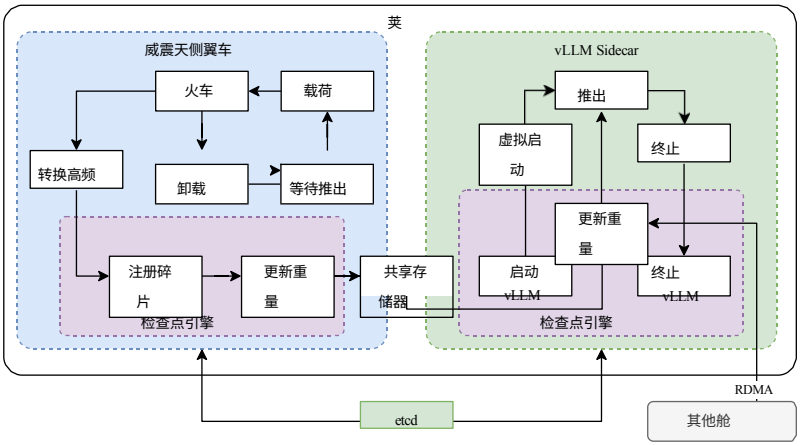


图 4：混合部署框架

1. **训练阶段：**一开始，Megatron（Shoeybi 等人，2020 年）和 vLLM（Kwon 等人，2023 年））分别在不同的容器中执行，并由一个称为检查点引擎（checkpoint-engine的临时进程封装（第 2.6.3 ）。节威震天开始训练程序。训练完成后，Megatron 会卸载 GPU 内存，并准备将当前权重传输到 vLLM。
2. **推理阶段：**在 Megatron 卸载之后，vLLM 开始使用虚拟模型权重，并用通过月饼（Qin 等，2024 年）。从 Megatron 转移过来的最新权重更新它们在推出完成后，检查点引擎会停止所有 vLLM 进程。
3. **后续训练阶段：**一旦分配给 vLLM 的内存被释放，威震天就会重新加载内存并启动新一轮训练。

我们发现，要同时支持以下所有特征，现有的工作具有挑战性。

4. 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，与 vLLM 共享可能具有挑战性。
5. 尽量减少闲置的 GPU 资源：对于 On-Policy RL，SGLang（L. Zheng 等人，2024 年）和 vLLM 等最新作品可能会在训练过程中保留一些 GPU，这反过来可能会导致训练 GPU 闲置。在训练和推理之间共享相同的设备会更有效。
6. 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。我们的系统能在需要时有效利用闲置的 GPU 节点。

如图 4 ，所示我们在 Megatron 和 vLLM 的基础上实现了这种混合部署框架（第 2.6.3 ）节，从训练到推理阶段的时间不到一分钟，反之则约为 10 秒。

混合部署策略 我们为训练和推理任务提出了一种混合部署策略，它利用 Kubernetes Sidecar 容器共享所有可用 GPU，将两种工作负载集中在一个 pod 中。这种策略的主要优势在于

7. 它有助于高效的资源共享和管理，当推理节点和列车节点分别部署在不同的节点上时，可以防止列车节点在等待推理节点时闲置。
8. 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。

9. 该架构并不局限于 vLLM，还可以方便地集成其他框架。

检查点引擎（Checkpoint Engine） 检查点引擎负责管理 vLLM 进程的生命周期，提供 HTTP API，以便在 vLLM 上触发各种操作。为了实现整体一致性和可靠性，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。

由于 CUDA 图形、NCCL 缓冲区和英伟达驱动程序的存在，通过 vLLM 卸载完全释放 GPU 内存可能具有挑战性。为了尽量减少对 vLLM 的修改，我们在需要时终止并重新启动 vLLM，以提高 GPU 利用率和容错性。

Megatron 中的 Worker 会将拥有的检查点转换为共享内存中的 Hugging Face 格式。这种转换还考虑了管道并行性和专家并行性，因此这些检查点中只保留了张量并行性。共享内存中的检查点随后会被划分为碎片，并在全局元数据系统中注册。我们采用 Mooncake 在对等节点之间通过 RDMA 传输检查点。加载权重文件和执行张量并行性转换需要对 vLLM 进行一些修改。

2.6.4 代码沙盒

我们开发的沙箱是执行用户提交代码的安全环境，并针对代码执行和代码基准评估进行了优化。通过动态切换容器镜像，沙箱可通过 MultiPL-E (Cassano, Gouwar, D. Nguyen, S. Nguyen, et al. 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 RL，沙箱通过提供一致且可重复的评估机制，确保了训练数据判断的可靠性。沙箱的反馈系统支持多阶段评估，如代码执行反馈和资源库级编辑，同时保持统一的上下文，以确保对不同编程语言进行公平公正的基准比较。

我们将该服务部署在 Kubernetes 上，以实现可扩展性和弹性，并通过 HTTP 端点进行外部集成。Kubernetes 的自动重启和滚动更新等功能可确保可用性和容错性。

为了优化性能并支持 RL 环境，我们在代码执行服务中采用了多项技术，以提高效率、速度和可靠性。这些技术包括

1. **使用 Crun：**我们使用 `crun` 代替 Docker 作为容器运行时，从而大大缩短了容器启动时间。
2. **cgroup 重用：**在高并发场景中，为每个容器创建和销毁 C 组可能会成为瓶颈，而这一点在这种场景中至关重要。
3. **优化磁盘使用：**使用上层挂载为 `tmpfs` 的覆盖文件系统来控制磁盘写入，提供固定大小的高速存储空间。这种方法有利于短暂性工作负载。

方法	时间（秒）
Docker	0.12
沙盒	0.04

(a) 容器启动时间

方法	集装箱/秒
Docker	27
沙盒	120

(b) 16 核机器上每秒启动的最大容器数

这些优化提高了 RL 代码执行的效率，为评估 RL 生成的代码提供了一致、可靠的环境，这对于迭代训练和模型改进至关重要。

3 实验

3.1 评估

由于 k1.5 是一个多模态模型，我们针对不同模态的各种基准进行了综合评估。详细的评估设置见附录 C。我们的

基准主要包括以下三类：

- **文本基准**：MMLU（Hendrycks 等，2020 年）、IF-Eval（J. Zhou 等人，2023 年）、CLUEWSC（L. Xu 等人，2020 年）、C-EVAL（Y. Huang 等人，2023 年）
- **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 等人，2024 年）、MATH-Vision（K. Wang 等人，2024 年）、MathVista（Lu 等人，2023 年）

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

3.2 主要成果

K1.5 long-CoT 模型 Kimi k1.5 long-CoT 模型的性能见表 2。通过 long-CoT 监督微调（见第 2.2 节）和视觉-文本联合强化学习（见第 2.3 ），节该模型的长期推理能力得到了显著增强。测试时间计算的扩展进一步增强了模型的性能，使模型能够在各种模式下取得最先进的结果。我们的评估结果表明，该模型在推理、理解和综合扩展上下文信息方面的能力有了显著提高，这代表了多模态人工智能能力的进步。

K1.5 短-CoT 模型 Kimi k1.5 短-CoT 模型的性能见表 3。该模型集成了多种技术，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）和从长到短的蒸馏（在第 2.4 ）。节中概述研究结果表明，k1.5 short-CoT 模型在多个任务中的性能与领先的开源和专有模型相比具有竞争力或更胜一筹。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具有显著优势。

	基准 (指标)	纯语言模式 QwQ-32B 预览版	OpenAI o1-mini	视觉语言模型 QVQ-72B 预览	OpenAI o1	Kimi k1.5
推理	MATH-500 (EM)	90.6	90.0	-	94.8	**96.2**
	AIME 2024 (Pass@1)	50.0	56.7	-	74.4	**77.5**
	代码力 (百分位数)	62	88	-	**94**	**94**
	LiveCodeBench (Pass@1)	40.6	53.1	-	**67.2**	62.5
愿景	MathVista 测试 (通过@1)	-	-	71.4	71.0	**74.9**
	MMMU-Val (Pass@1)	-	-	70.3	**77.3**	70.0
	数学视力-满分 (Pass@1)	-	-	35.9	-	**38.6**

表 2：Kimi k1.5 long-CoT、旗舰开源模型和专有模型的性能。

	基准 (指标)	纯语言模式 Qwen2.5 72B-Inst.	LLaMA-3.1	DeepSeek V3	视觉语言模型 Qwen2-VL	Claude-3.5-十四行诗-1022	GPT-4o 0513	Kimi k1.5
文本	MMLU (EM)	85.3	**88.6**	88.5	-	88.3	87.2	87.4
	IF-Eval (严格提示)	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**
(推理)	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**
愿景	MathVista 测试 (通过@1)	-	-	-	69.7	65.3	63.8	**70.1**
	MMMU-Val (Pass@1)	-	-	-	64.5	66.4	**69.1**	68.0
	数学视力-满分 (及格@1)	-	-	-	26.6	**35.6**	30.4	31.0

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

3.3 长语境缩放

我们使用了一个中型模型来研究带有 LLM 的 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变化情况。随着训练的进行，我们观察到响应长度和性能准确度同时增加。值得注意的是，在更具挑战性的基准测试中，响应长度的增长速度更快，这表明模型学会了为复杂问题生成更复杂的解决方案。图 6 显示，模型的响应长度和准确率之间存在很强的相关性。

输出上下文长度及其解决问题的能力。我们最后运行的 k1.5 可扩展到 128k 上下文长度，并观察到其在困难推理基准上的持续改进。

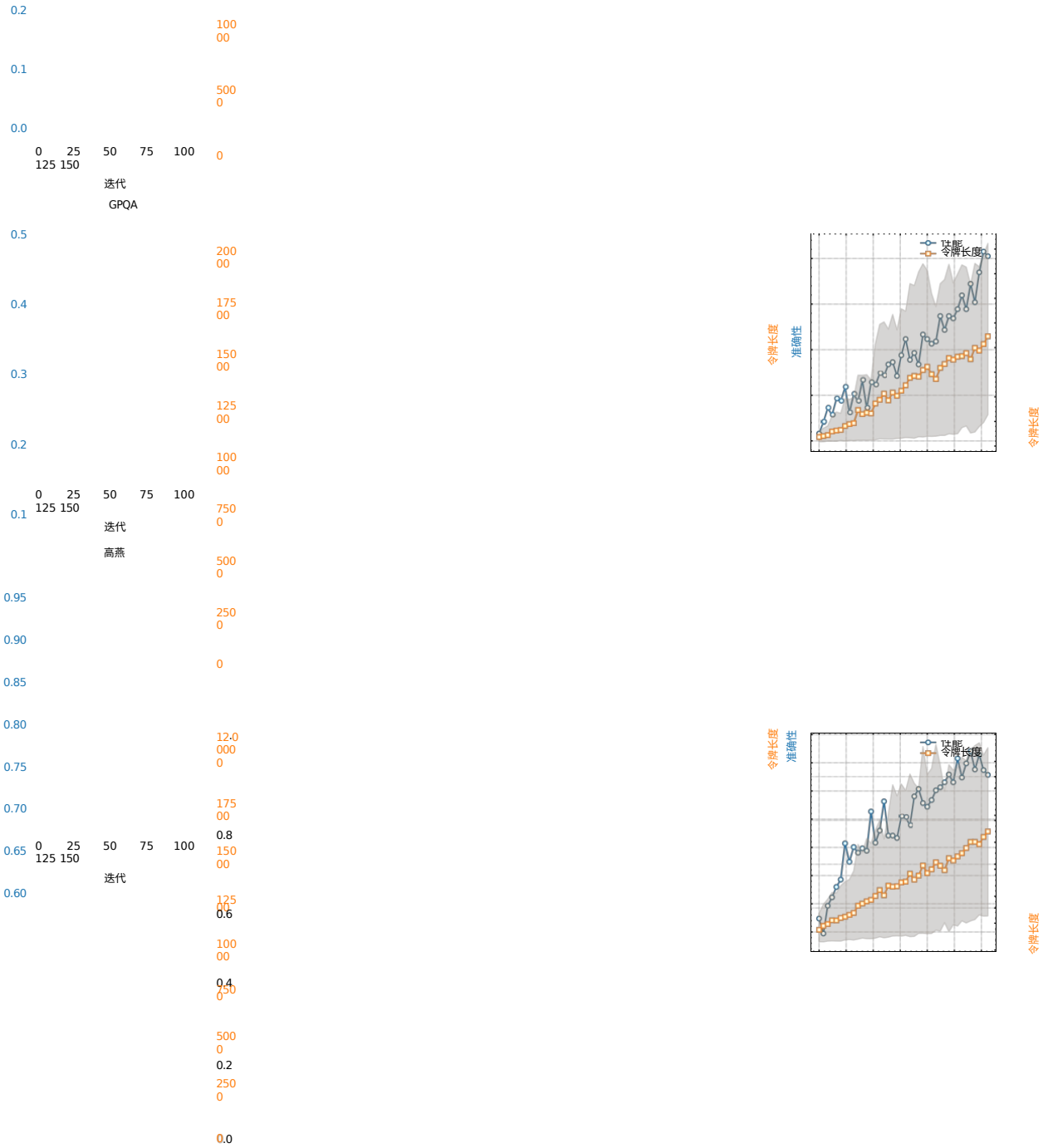


图 5：随着训练迭代次数的增加，训练精度和训练长度的变化。请注意，上面的分数来自内部 long-cot 模型，其模型规模远小于 k1.5 long-CoT 模型。阴影区域代表响应长度的 95% 百分位数。

3.4 Long2short

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是 long2short 问题的令牌效率（X. Chen 等，2024 年），特别是所获得的 long-cot 模型如何使 short 模型受益。在图 7

，中k1.5-long 代表我们选择用于 long2short 训练的 long-cot 模型；k1.5-short w/ rl 指的是通过 long2short RL 训练获得的短模型；k1.5-short w/ dpo 表示通过 DPO 训练提高了标记效率的短模型。k1.5-short w/ merge 表示模型合并后的模型，而 k1.5-short w/ merge+ rs 则表示通过对合并后的模型应用最短拒绝采样而得到的短模型。k1.5-shortest 表示我们在 long2short 训练中得到的最短模型。如图 7 ，所示与 DPO 和模型合并等其他方法相比，拟议的 long2short RL 算法具有最高的标记效率。值得注意的是，与其他模型（蓝色标记）相比，k1.5 系列的所有模型（橙色标记）都表现出更高的标记效率。，k1.5-short w/ rl 在 AIME2024 上获得了 60.8 的 Pass@1 分数（8 次运行的平均值），而平均只使用了 3272 个标记符。类似地，k1.5-shortest 在 MATH500 上获得了 88.2 的 Pass@1 分数，而消耗的令牌数量与其他短模型大致相同。

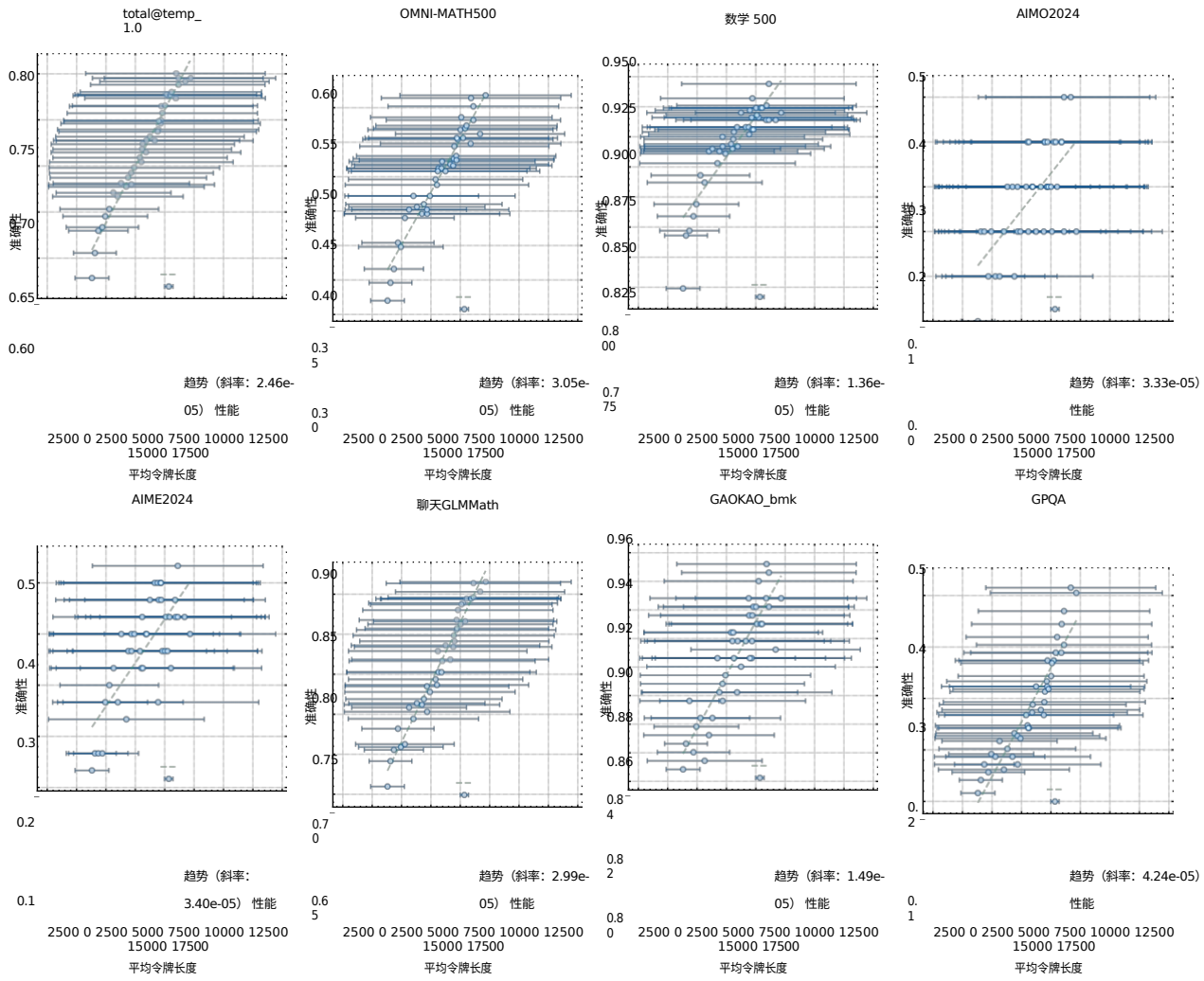


图 6：模型性能随响应长度增加而提高

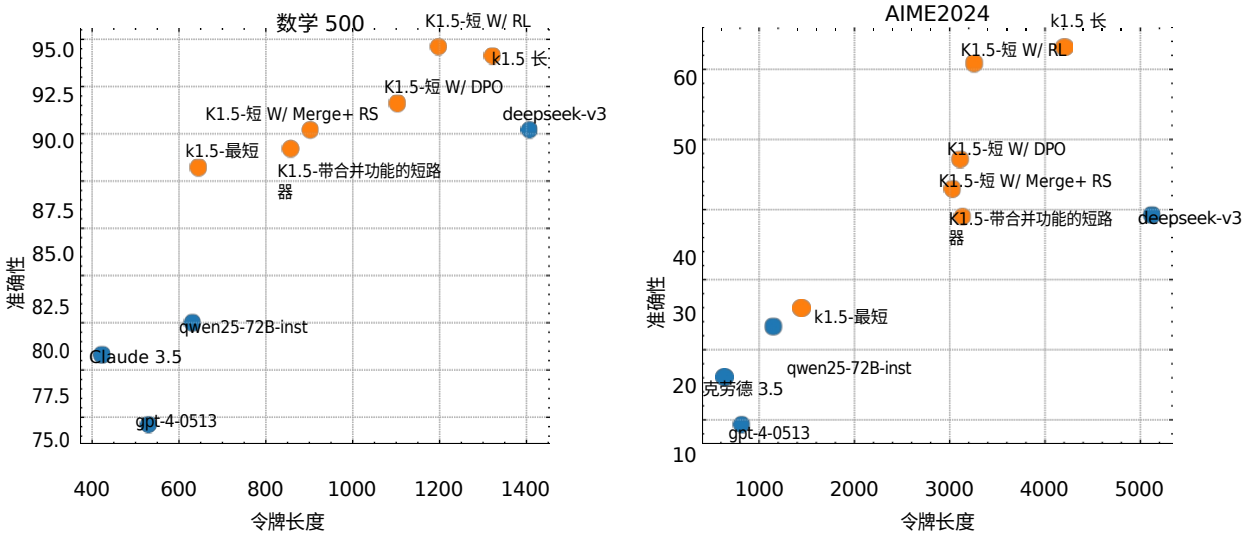


图 7：Long2Short 性能。与其他型号相比，所有 k1.5 系列都表现出更高的令牌效率。

3.5 消融研究

模型大小和上下文长度的缩放 我们的主要贡献在于应用 RL 增强了模型生成扩展 CoT 的能力，从而提高了推理能力。一个自然而然的问题随之而来：这与单纯增加模型大小相比效果如何？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 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，虽然较大的模型最初的表现优于较小模型，但较小的模型可以通过利用通过 RL 优化的较长的 CoT 达到相当的性能。不过，大模型的令牌效率通常比小模型高。这也表明，如果以尽可能好的性能为目标，扩展较大模型的上下文长度具有更高的上限和更高的标记效率。不过，如果测试时间计算有预算，那么用较大的上下文长度训练较小的模型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环境中使用 ReST（Gulcehre 等人，2023 年）作为策略优化算法的效果。ReST 与其他基于 RL 的方法的主要区别包括

这是因为 ReST 通过拟合从当前模型中采样到的最佳响应来迭代完善模型，而不应用负梯度来惩罚不正确的响应。如图 10 ，，所示与 ReST 相比我们的方法表现出更高的样本复杂度，表明负梯度的加入明显提高了模型生成长 CoT 的效率。我们的方法不仅提高了推理的质量，还优化了训练过程，以更少的训练样本实现了稳健的性能。这一发现表明，在我们的环境中，策略优化算法的选择至关重要，因为 ReST 与其他基于 RL 的方法之间的性能差距在其他领域并不明显（Gulcehre 等人，2023 年）。因此，我们的结果凸显了选择适当优化策略的重要性，以最大限度地提高生成长 CoT 的效率。

取样策略 我们将进一步证明 2.3.4 。节中介绍的课程取样策略的有效性我们的训练数据集 $\mathcal{D}$ 包含各种不同难度的问题。利用我们的课程抽样方法，我们首先使用 $\mathcal{D}$ 作为热身阶段，然后只关注来训练模型。这种方法与采用统一抽样策略、不做任何课程调整的基准方法进行了比较。如图 9 ，所示我们的结果清楚地表明，建议的课程抽样方法显著提高了性能。这种改进可归因于该方法能够逐步挑战模型，使其在处理复杂问题时形成更稳健的理解和能力。在最初的一般介绍之后，通过将训练重点放在更难的问题上，模型可以更好地加强其推理和解决问题的能力。

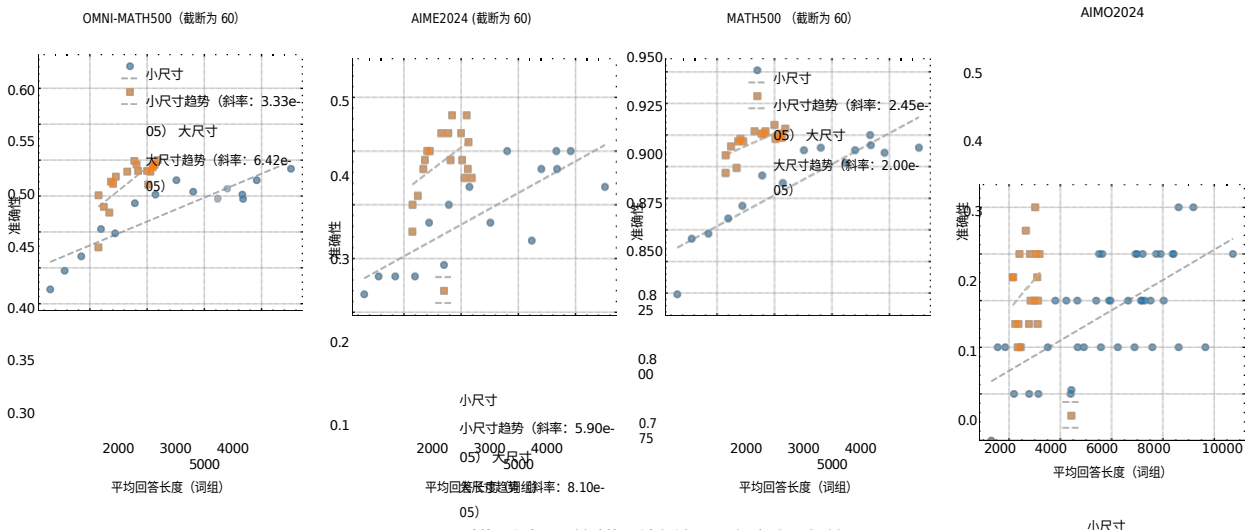
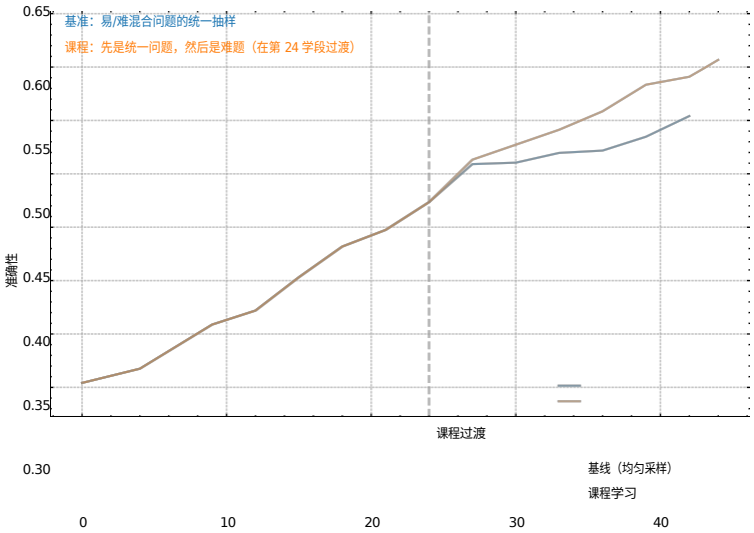


图 8：不同模型大小的模型性能与响应长度的关系



迭代

图 9：课程学习方法对模型性能的影响分析。

4 结论

我们介绍了 k1.5 的训练配方和系统设计，这是我们最新的使用 RL 训练的多模式 LLM。我们从实践中获得的一个重要启示是，上下文长度的缩放对 LLM 的持续改进至关重要。我们采用优化的学习算法和基础架构优化（如部分滚动）来实现高效的长上下文 RL 训练。进一步提高长语境 RL 训练的效率和可扩展性，仍然是未来的一个重要问题。

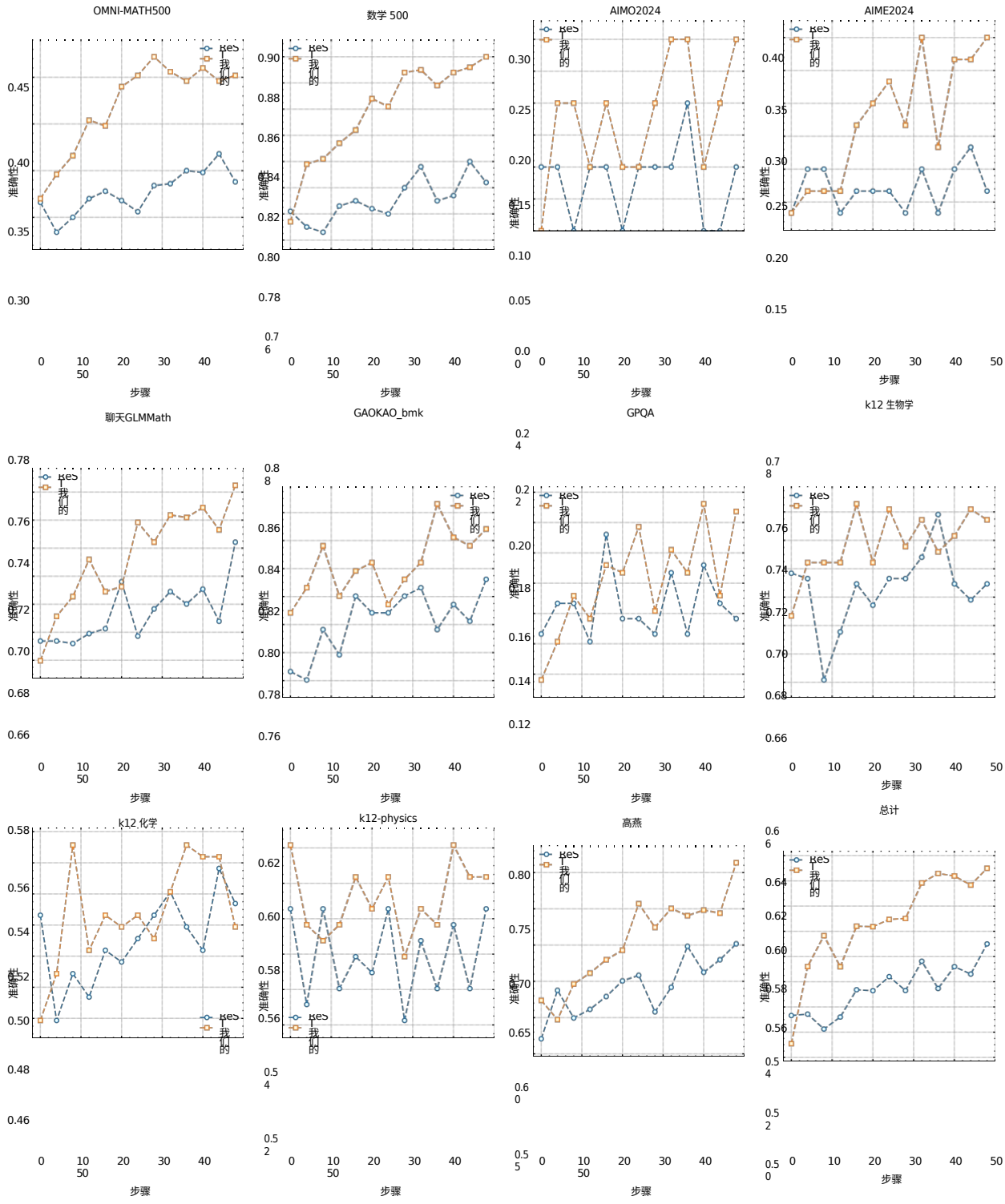


图 10：与使用 ReST 进行策略优化的比较。

我们的另一项贡献是将各种技术相结合，改进了策略优化。具体来说，我们利用 LLM 制定了 long-CoT RL，并推导出在线镜像下降的变体，以实现稳健优化。我们还尝试了采样策略、长度惩罚和优化数据配方，以实现强大的 RL 性能。

我们表明，即使不使用蒙特卡洛树搜索、价值函数和过程奖励模型等更复杂的技术，也可以通过长上下文缩放和改进策略优化来实现强大的性能。未来，在不损害模型探索能力的前提下，研究如何改进信用分配和减少过度思考也将非常有趣。

我们还发现了长2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，还可以通过迭代的方式将 long2short 方法与长 CoT RL 结合起来，进一步提高令牌效率，并在给定上下文长度预算的情况下获得最佳性能。

参考资料

Abbasi-Yadkori, Yasin 等人，"Politex：利用专家预测进行政策迭代的遗憾界限"。In：*国际 机器学习会议*。PMLR.2019, pp.

Ahmadian, Arash et al：在 llms 中重新审视从人类反馈中学习的强化风格优化"。In: *arXiv preprint arXiv:2402.14740* (2024).

Ankner, Zachary *et* al.2024. ArXiv: 2408.11791 [cs.LG].URL: https://arxiv. org/abs/2408.11791.

Berner, Christopher et al. "Dota 2 with large scale deep reinforcement learning".载于：*arXiv preprint arXiv:1912.06680* (2019).

Cassano、Federico、John Gouwar、Daniel Nguyen、Sy Duy Nguyen 等：《MultiPL-E：一种可扩展的神经代码生成基准方法》。In：*ArXiv* (2022).URL: https://arxiv.org/abs/2208.08227.

Cassano、Federico、John Gouwar、Daniel Nguyen、Sydney Nguyen ："MultiPL-E：神经代码生成基准测试的可扩展多语言方法"。In：*IEEE Transactions on Software Engineering* 49.7 (2023), pp.doi: 10.1109/tse.2023.3267446.

Chen, Jianlv 等，"Bge m3-embedding：多语言、多功能、多粒度的文本嵌入自我知识提炼"。In: *arXiv preprint arXiv:2402.03216* (2024).

Chen, Xingyu et al.论 o1 类 LLM 的过度思考"。In: *arXiv preprint arXiv:2412.21187* (2024).

Everitt, Tom 等.*强化学习中的奖励篡改问题及解决方案.因果影响图：因果影响图 透视*。2021. arXiv: 1908.04734 [cs.AI].URL: https://arxiv.org/abs/1908.04734.

Gadre, Samir Yitzhak et al：寻找下一代多模态数据集"。In： *神经信息处理系统进展* 36 (2024)。

Grattafiori, Aaron 等人. *Llama 3 模型群*.2024. ArXiv: 2407.21783 [cs.AI].URL: https://arxiv.org/abs/2407.21783.

Gulcehre, Caglar et al. "Reinforced self-training (rest) for language modeling".In: *arXiv preprint arXiv:2308.08998* (2023).

Hendrycks, Dan et al. "Measuring Massive Multitask Language Understanding".In：*ArXiv* abs/2009.03300 (2020). URL: https://arxiv.org/abs/2009.03300.

霍夫曼、乔丹等.*训练计算最优的大型语言模型*.2022. arXiv: 2203.15556 [cs.CL]. URL: https://arxiv.org/abs/2203.15556.

Huang, Yuzhen et al. "C-Eval: A Multi-Level Multi-Discipline Chinese Evaluation Suite for Foundation Models".In：*ArXiv* abs/2305.08322 (2023)。URL: https://arxiv.org/abs/2305.08322.

Jaech、Aaron ：《Openai o1 系统卡》。In: *arXiv preprint arXiv:2412.16720* (2024).

Jain, Naman 等，"LiveCodeBench：代码大型语言模型的整体和无污染评估"。In：*ArXiv* abs/2403.07974 (2024)。URL: https://arxiv.org/abs/2403.07974.

Joulin, Armand et al. "高效文本分类的锦囊妙计"。In: *arXiv preprint arXiv:1607.01759* (2016).Kaplan, Jared et al.

Scaling Laws for Neural Language Models.ArXiv: 2001.08361 [cs.LG].URL: https://arxiv.org/abs/2001.08361.

Kool, Wouter, Herke van Hoof, and Max Welling."买 4 个强化样本，送一个基线！"收录于：（2019）.Kwon, Woosuk et al. "Efficient Memory Management for Large Language Model Serving with PagedAttention".In：*ACM SIGOPS 第 29 届操作系统原理研讨会论文集*》。2023.

Laurençon, Hugo et al：交错图像-文本文档的开放式网络规模过滤数据集"。In：*Advances in Neural Information Processing Systems* 36 (2024).

李杰夫等 "Datacomp-lm：寻找下一代语言模型训练集"。In: *arXiv preprint arXiv:2406.11794* (2024).

李明等人 "从数量到质量：利用自导数据选择提升指令 调整的 llm 性能"。In: *arXiv preprint arXiv:2308.12032* (2023).

李雷蒙德等. *StarCoder：愿源泉与你同在！* 2023. ArXiv: 2305 .06161 [cs.CL].URL: https ：//arxiv.org/abs/2305.06161.

莱特曼、亨特：《让我们一步步验证》。In: *arXiv preprint arXiv:2305.20050* (2023).

Liu、Wei 等. "What makes good data for alignment? A comprehensive study of automatic data selection in instruction tuning".见：*arXiv preprint arXiv:2312.15685* (2023).

Lozhkov, Anton 等. *StarCoder 2 和 The Stack v2：下一代*。2024. ArXiv: 2402.19173 [cs.SE]。URL: https://arxiv.org/abs/2402.19173。

Lu, Pan et al：评估视觉语境中基础模型的数学推理"。In: *arXiv preprint arXiv:2310.02255* (2023).

McAleese, Nat et al. *LLM Critics Help Catch LLM Bugs*.2024. ArXiv: 2407.00215 [cs.SE].URL: https://arxiv.org/abs/2407.00215.

Mei、Jincheng 等人 "论政策优化中的原则性熵探索"。In：*Proceedings of the 28th International Joint Conference on Artificial Intelligence*.2019, pp.

Muennighoff, Niklas et *al*.2023. ArXiv: 2305.16264 [cs.CL].URL: https://arxiv.org/abs/2305.16264。

Nachum、Ofir 等人 "弥合基于价值和策略的强化学习之间的差距"。In：*神经 信息处理系统的进展* 30 (2017)。

OpenAI."用 LLM 学习推理"。In: (2024).URL: https://openai.com/index/learning-to-reason- with-llms/ 。

欧阳龙等人 "训练语言模型，使其遵循人类反馈指令"。In. Advances in neural Information Processing Systems 35 (2022), pp：*神经 信息处理系统进展* 35 (2022)，第 27730-27744 页。

Pan, Alexander, Kush Bhatia, and Jacob Steinhardt."奖励失范的影响：Mapping and Mitigating Misaligned Models".In：*学习表征国际会议*。2022.URL: https://openreview. net/forum?id=JYtwGwIL7ye。

Paster, Keiran et al. "Openwebmath：高质量数学网络文本的开放数据集"。In: *arXiv preprint arXiv:2310.06786* (2023).

Penedo, Guilherme et al：为最精细的大规模文本数据对网络进行分拣"。In: *arXiv preprint arXiv:2406.17557* (2024).

Qin, Ruoyu et al：*用于 LLM 服务的以 KVCache 为中心的分解架构*。2024. ArXiv: 2407.00079 [cs.DC].URL: https://arxiv.org/abs/2407.00079.

Rafailov, Rafael et al：你的语言模型其实是一个奖励模型"。In：*Advances in Neural Information Processing Systems* 36 (2024).

Schuhmann, Christoph 等. "Laion-5b：用于训练下一代图像-文本模型的开放式大规模数据集"。In：*神经信息处理系统进展* 35 (2022)，第 25278-25294 页。

Shoeybi、Mohammad 等人. *Megatron-LM：利用模型并行性训练多亿参数语言模型*.2020. ArXiv: 1909.08053 [cs.CL]。URL: https://arxiv.org/abs/1909.08053.

西尔弗、戴维等人：《在没有人类知识的情况下掌握围棋》。见：《*自然*》550.7676 (2017)，第 354-359 页。Snell,

Charlie et al. "Scaling llm test-time compute optimally can be more effective than scaling model parameters".In：*arXiv preprint arXiv:2408.03314* (2024).

Su, Dan 等. "Nemotron-CC：将普通抓取转化为精炼的长视距预训练数据集"。In：*arXiv preprint arXiv:2412.02595* (2024).

Su, Jianlin et al：具有旋转位置嵌入功能的增强型变压器"。In：*Neurocomputing* 568 (2024)、p.127063.

双子座团队等：*《双子座：高能力多模态模型家族》*（Gemin*: A Family of Highly Capable Multimodal Models*.2024. ArXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

Tomar、Manan：《镜像下降策略优化》。In: *arXiv preprint arXiv:2005.09814* (2020).

Vaswani, Ashish et al. "Attention is All You Need".In：*神经信息处理系统的进展*》。编者 I.Guyon et al.Curran Associates, Inc., 2017.URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Villalobos, Pablo et al.*基于人类生成数据的 LLM 扩展极限*。2024. ArXiv：2211.04325 [cs.LG].URL: https://arxiv.org/abs/2211.04325.

Vinyals、Oriol 等. "使用多代理强化学习的《星际争霸 II》大师级".见：*nature* 575.7782 (2019)，第 350-354 页。

Wang, Ke 等.《用数学视觉数据集衡量多模态数学推理》。见：*arXiv preprint arXiv:2402.14804* (2024).

Wei, Haoran et al：Towards OCR-2.0 via a Unified End-to-end Model".In: *arXiv preprint arXiv:2409.01704* (2024).

Wei, Jason 等人. "Chain-of-thought prompting elicits reasoning in large language models".In：*神经 信息处理系统进展* 35 (2022)，第 24824-24837 页。

Wu, Yangzhen et al：用语言模型解决问题的计算最优推理实证分析 "。In: *arXiv preprint arXiv:2408.00724* (2024).

徐亮等 CLUE：中文理解评估基准"。In： *计算语言学国际会议*。2020.URL: https://arxiv.org/abs/2004.05986.

Yang, Enneng et al. "Model merging in llms, mllms, and beyond：方法、理论、应用与机遇"。In：*arXiv preprint arXiv:2408.07666* (2024).

Yao, Shunyu et al：利用大型语言模型慎重解决问题"。In：*神经 信息处理系统进展* 36 (2024)。

Yue、Xiang、Yuansheng Ni 等："Mmmu：面向 agi 专家的大规模多学科多模态理解与推理基准"。In：*IEEE/CVF 计算机视觉与模式识别会议论文集*》。2024, pp.

Yue, Xiang, Xingwei Qu, et al：通过混合指令调整建立数学通才模型"。In：*arXiv preprint arXiv:2309.05653* (2023).

Zhang, Lunjun et al：奖励建模作为下一个标记预测，2024 年"。In：*URL https://arxiv. org/abs/2408.15240* (2024).

Zheng, Lianmin et . *SGLang: Efficient Execution of Structured Language Model Programs*.2024. arXiv: 2312.07104 [cs.AI].URL: https://arxiv.org/abs/2312.07104.

Zhou、Jeffrey 等人 "大型语言模型的指令跟踪评估"。In：*ArXiv* abs/2311.07911 (2023).

URL: https://arxiv.org/abs/2311.07911.

Zhu, Wanrong et al. "Multimodal c4: An open, billion-scale corpus of images interleaved with text".In：*神经信息处理系统进展* 36 (2024).

附录

A 会费

研发

杜鞍钢 高博飞
蒋昌久 陈诚 李
晨军 肖晨庄 杜
德浩 张恩铭 袁
恩哲 卢恩哲
Flood Sung
Guokun Lai
Haiqing Guo
Han Zhu Hao
Ding Hao Hu
杨浩 张浩天
姚浩天 赵浩
宇 卢浩
洪城 高欢 袁华
斌 郑静远 刘建
林 苏建洲 王进
张俊杰 闫立东
史龙辉 于梦楠
董尼奥 张启伟
潘求成 龚树鹏
魏少伟 刘涛 蒋
为民 熊蔚然 何
伟豪 高伟晓 黄
文浩 吴文洋 何
贤清 贾兴哲 吴
文洋
Xinran Xu
Xinyu Zhou
Xinxing Zu
Xuehai Pan
Yang Li Yang
Yang Hu
Yang Yang
Liu Yanru
Chen Yejie
Wang Yidao
Qin Yibo Liu
Yiping Bao
Yulun Du
Yuzhi Wang
Yuxin Wu 李
洋洋，胡洋洋
，刘艳茹。
Y.Charles
Zaida Zhou
Zhaoji Wang
Zhaowei Li
Zheng Zhang
Zhexu Wang
Zhiqi Huang
Zhilin Yang
Ziyao Xu
Zonghan Yang
杨宗翰

数据注释

Chuning Tang
Congcong Wang
Fengxiang Tang
Guangda Wei
Haoze Li
Haozhen Yu
Jia Chen
Jianhang Guo
Jie Zhao
Junyan Wu
Ling Ye
Shengling Ma
Sihan Cao
Siying Huang
Xianghui Wei
Yang Yang Liu
Ying Yang
Zhen Zhu
Zihao Huang

作者姓名按字母顺序排列。

B 预培训

强化学习（RL）的效率与基础模型的性能密切相关。Gemini（Team 等人，2024 年）和 Llama（Grattafiori 等人，2024 年）等前沿模型强调了预训练数据质量对实现高性能的重要性。然而，最近的许多开源模型在数据处理管道和配方方面缺乏完全的透明度，给更广泛的社区理解带来了挑战。虽然我们目前没有开源我们的专有模型，但我们致力于全面披露我们的数据管道和方法。在本节中，我们将主要关注多模态预训练数据配方，然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大型语言模型（LLM）的训练提供全面而高质量的数据。它包括五个领域：英语、中文、代码、数学与推理以及知识。我们对每个领域都采用了复杂的过滤和质量控制机制，以确保获得最高质量的训练数据。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，以评估其对整个训练配方的具体贡献。这种系统性评估确保了我们多样化数据组成的质量和有效性。

针对**中英文文本数据**，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。我们的框架包括

1. **基于规则的过滤**：我们采用特定领域的启发式方法来删除有问题的内容，包括重复内容、机器翻译文本和低质量的网络搜刮。我们还能过滤掉包含过多特殊字符、异常格式或垃圾模式的文档。
2. **基于 FastText 的分类**：我们训练了专门的 FastText（Joulin 等人，2016 年；J. Li 等人，2024 年）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**：利用文档嵌入（Jianlv Chen 等人，2024 年），我们计算文档级的相似性得分，以识别和删除近似重复的内容，同时保留语义上有价值的变化。这种方法有助于保持训练语料库的多样性。
4. **基于 LLM 的质量评估**：根据（Penedo 等人，2024 年），我们利用 LLM 对文档的连贯性、信息量和潜在教育价值进行评分。这种方法在识别细微的质量指标方面特别有效，而简单的方法可能会忽略这些指标。

每份文档的最终质量得分由这些单个得分组合计算得出。基于广泛的经验分析，我们采用了动态采样率，在训练过程中对高质量文档进行上采样，而对低质量文档进行下采样。

代码数据 代码数据主要包括两类。对于源于代码文件的纯代码数据，我们遵循 BigCode 的方法（R. Li 等人，2023；Lozhkov 等人，2024），对数据集进行了全面的预处理。首先，我们剔除了杂项语言，并应用基于规则的清理程序来提高数据质量。随后，我们通过策略性抽样技术解决了语言失衡问题。具体来说，JSON、YAML 和 YACC 等标记语言被降低采样率，而包括 Python、C、C++、Java 和 Go 在内的 32 种主要编程语言被提高采样率，以确保均衡的代表性。对于来自不同数据源的文本代码交错数据，我们使用基于嵌入的方法来调用高质量数据。这种方法既保证了数据的多样性，又保持了数据的高质量。

数学和推理数据 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取自从公开互联网资源中收集的网络文本和 PDF 文档。(Paster 等人，2023 年）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很高的假阴性率。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，旨在最大限度地提高数学数据的召回率。随后，我们实施了两阶段数据

清理流程：

1. 使用 FastText 模型进行初步清理，删除大部分无关数据。

2. 利用微调语言模型进一步清理剩余数据，从而获得高质量的数学数据。

知识数据 知识语料库经过精心策划，以确保全面覆盖各学科。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。这些资料中有很大一部分是通过 OCR 处理数字化的，为此我们开发了专有模型，针对学术内容进行了优化，特别是在处理数学公式和特殊符号方面。

我们采用内部语言模型为文档标注多维标签，包括

1. 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. 文件类型分类（如练习、理论材料等）

基于这些多维注释，我们实施了一个复杂的过滤和采样管道。，通过 OCR 质量阈值对文档进行过滤。我们的 OCR 质量评估框架特别注重检测和过滤常见的 OCR 伪特征，尤其是重复文本模式，这些模式通常表明识别失败。

除了基本的质量控制外，我们还通过评分系统仔细评估每份文档的教育价值。具有高度教学相关性和知识深度的文档会被优先考虑，同时在理论深度和教学清晰度之间保持平衡。这有助于确保我们的培训语料库包含高质量的教育内容，能够有效促进模型的知识获取。

最后，为了优化训练语料库的整体构成，我们通过大量实验，根据经验确定了不同文档类型的采样策略。我们进行单独评估，以确定对模型知识获取能力贡献最大的文档子集。这些高价值子集会在最终的训练语料库中被增加采样。不过，为了保持数据的多样性并确保模型的通用性，我们还是以适当的比例谨慎地保留了其他文档类型的均衡代表性。这种以数据为导向的方法有助于我们在重点知识获取和广泛泛化能力之间进行优化权衡。

B.2 多模式数据

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文本、图像和视频等多种模态的信息。，我们还从五个类别（字幕、交错、OCR（光学字符识别）、知识和一般问题解答）中收集了高质量数据，形成了该语料库。

在构建训练语料库时，我们开发了多个多模态数据处理管道，以确保数据质量，包括过滤、合成和重复数据删除。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，因为它既能保留语言模型的能力，又能促进不同模态知识的协调。

在本节中，我们将对这些来源进行详细描述，并将其分为以下几类：

字幕数据 我们的字幕数据为模型提供了基本的模态对齐和广泛的世界知识。通过整合字幕数据，多模态 LLM 可以以较高的学习效率获得更广泛的世界知识。我们整合了各种开源的中英文字幕数据集（Schuhmann 等人，2022 年；S. Y. Gadre 等，2024 年），还从多个来源收集了大量内部字幕数据。不过，在整个训练过程中，我们严格限制合成标题数据的比例，以降低因真实世界知识不足而产生幻觉的风险。

对于一般标题数据，我们遵循严格的质量控制流程，以避免重复并保持图像与文本的高度相关性。我们还在预训练过程中改变图像分辨率，以确保视觉塔在处理高分辨率和低分辨率图像时依然有效。

图像-文本交错数据 在预训练阶段，交错数据能使模型在很多方面受益，，交错数据能提高多图像理解能力；交错数据总是能提供给定图像的详细知识；交错还能获得更长的多模态语境学习能力。此外，我们还发现，交织数据对保持模型的语言能力也有积极作用。因此，图像-文本交错数据是我们训练语料的重要组成部分。我们的多模态

语料库考虑了开源交错数据集（Zhu 等人，2024 年；Laurençon 等，2024 年），还利用教科书、网页和教程等资源构建了大规模内部数据。此外，我们还发现合成交错数据有利于多模态 LLM 保持文本知识的性能。为了确保每个图像的知识都得到充分研究，对于所有交织数据，除了标准的过滤、去重和其他质量控制流程外，我们还集成了数据重排序程序，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，可将图像中的文本转换为可编辑的格式。在 k1.5 中，强大的 OCR 功能被认为是使模型更好地与人类价值观保持一致的关键。因此，我们的 OCR 数据源多种多样，既有开源数据，也有内部数据集，既有纯净图像，也有增强图像。

除了公开数据外，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，根据 OCR 2.0（H. Wei 人，2024 年）中概述的原则，我们的模型还能处理各种光学图像类型，包括数字、表格、几何图形、人鱼图和自然场景文本。我们应用了大量的数据增强技术，如旋转、变形、颜色调整和噪声添加等，以增强模型的鲁棒性。因此，我们的模型在 OCR 任务中达到了很高的熟练程度。

知识数据 多模态知识数据的概念与前面提到的文本预训练数据类似，只不过这里我们侧重于从不同来源收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开发视觉推理能力至关重要，可确保模型能够解释人类创建的抽象图表。

我们的知识语料库采用标准化分类法，以平衡不同类别的内容，确保数据来源的多样性。与从教科书、研究论文和其他学术材料中收集知识的纯文本语料库类似，多模态知识数据采用了布局解析器和 OCR 模型来处理这些来源的内容。同时，我们还包括来自互联网和其他外部资源的过滤数据。

由于我们的知识语料库中有很大一部分来自互联网资料，因此信息图表可能会导致模型只关注基于 OCR 的信息。在这种情况下，完全依赖基本的 OCR 管道可能会限制训练效果。为了解决这个问题，我们开发了一个额外的管道，可以更好地捕捉图像中嵌入的纯文本信息。

通用质量保证数据 在训练过程中，我们发现将大量高质量的质量保证数据集纳入预训练具有显著的优势。具体来说，我们纳入了严格的学术数据集，用于处理接地、表格/图表问题解答、网络代理和一般质量保证等任务。此外，我们还汇编了大量内部质量保证数据，以进一步增强模型的能力。为了保持难度和多样性的平衡，我们对一般问题解答数据集采用了评分模型和细致的人工分类，从而提高了整体性能。

B.3 模型架构

Kimi k 系列模型采用了 Transformer 解码器的变体（Vaswani 等人，2017 年），该解码器在改进架构和优化策略的同时还集成了多模态功能，如图 11 所示。这些进步共同支持稳定的大规模训练和高效的推理，专为大规模强化学习和 Kimi 用户的操作要求量身定制。

广泛的扩展实验表明，基础模型的大部分性能来自预训练数据质量和多样性的提高。有关模型架构扩展实验的具体细节超出了本报告的范围，将在今后的出版物中讨论。

B.4 培训阶段

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语境激活阶段。Kimi k1.5 模型训练的每个阶段都侧重于特定能力的提升。

视觉-语言预训练阶段 在这一阶段，首先只对语言数据进行模型训练，建立稳健的语言模型基础。然后逐步将模型引入交错的视觉-语言数据，从而获得多模态能力。视觉塔最初是在不更新语言模型参数的情况下进行孤立训练的，然后我们将语言模型层解冻，并最终增加视觉-文本数据的比例

图 11：Kimi k1.5 支持将交错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。

至 30%。最终的数据混合物及其各自的权重是通过在较小模型上进行的烧蚀研究确定的。

视觉语言冷却阶段 第二阶段为冷却阶段，在这一阶段，模型将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。通过经验调查，我们发现在冷却阶段加入合成数据能显著提高性能，尤其是在数学推理、基于知识的任务和代码生成方面。冷却数据集的英文和中文部分均来自预训练语料库的高保真子集。对于数学、知识和代码领域，我们采用了一种混合方法：利用选定的预训练子集，同时用合成生成的内容对其进行扩充。具体来说，我们利用现有的数学、知识和代码语料库作为源材料，通过专有的语言模型生成问答对，并采用剔除抽样技术来保持质量标准（Yue, Qu, et al. 2023; D. Su et al. 2024）。这些合成的 QA 对在整合到冷却数据集之前要经过全面的验证。

长语境激活阶段 最后，在第三阶段，k1.5 将使用上采样的长语境冷却数据进行训练，使其能够处理扩展序列并支持需要更长语境的任务。为了确保基础模型具有出色的长语境能力，我们对长语境数据进行了上采样，并在长语境训练中使用了 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，部分来自合成的长语境问答和摘要数据。部分注意力数据来自冷却数据的统一采样。RoPE 频率（J. Su 等，2024 年）设定为 1,000,000。在这一阶段，我们通过将最大序列长度从 4,096 增加到 32,768 并最终增加到 131,072 来逐步延长激活训练的长度。

C 评估细节

C.1 文本基准

MMLU (Hendrycks et al. 2020) 涵盖 57 个学科，包括科学、技术、工程和数学、人文科学、社会科学等。其难度从初级水平到高级专业水平等，既测试世界知识，也测试解决问题的能力。

IF-Eval (J. Zhou et al. 2023) 是评估大型语言模型遵循可验证指令能力的基准。其中有 500 多条提示指令，如 "写一篇 800 字以上的文章 "。由于版本变化，表 3 中报告的 IFEval 数量来自中间模型。我们将根据最终模型更新分数。

CLUEWSC（L. Xu 人，2020 年）是 CLUE 基准中的核心参照解析任务，要求模型判断句子中的代词和名词短语是否共指，数据来自中文小说书籍。

C-EVAL（Y. Huang 人，2023 年）是用于评估基础模型高级知识和推理能力的综合性中文评估套件。它包括 13,948 道选择题，涉及 52 个学科和 4 个难度级别。

C.2 推理基准

HumanEval-Mul 是 Multipl-E 的子集（Cassano、Gouwar、D. Nguyen、S. D. Nguyen 等，2022 年）。MultiPL-E 将 HumanEval 基准和 MBPP 基准扩展到 18 种语言，包括一系列编程语言

范式和受欢迎程度。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 等人，2024 年）是评估编码任务中大型语言模型（LLM）的一个全面、无污染的基准。它具有实时更新以防止数据污染、跨多种编码场景的整体评估、高质量的问题和测试以及均衡的问题难度等特点。我们用 2408-2411（版本 4）中的问题测试了短-CoT 模型，用 2412-2502（版本 5）中的问题测试了长-CoT 模型。

AIME 2024 包含 2024 年 AIME 的竞赛试题。AIME 是一项著名的数学竞赛，只邀请顶尖高中学生参加，评估高级数学技能，要求扎实的基础和高度的逻辑思维。

MATH-500 (Lightman et al. 2023) 是一项综合性数学基准测试，包含 500 个问题，涉及代数、微积分、概率等多个数学主题。测试计算能力和数学推理能力。分数越高，说明数学问题解决能力越强。

Codeforces 是一个著名的在线评判平台，也是评估 long-CoT 编码模型的常用测试平台。为了在 Div2 和 Div3 竞赛中获得更高的排名，我们对 k1.5 long-CoT 模型生成的代码片段进行了多数投票，并采用了同样由该模型生成的测试用例。

C.3 图像基准

MMMU（Yue、Ni 等人，2024 年）是一个经过精心策划的题库，包含 11.5K 个多模态问题，这些问题来自大学考试、测验和教科书。这些问题横跨六大学术领域：艺术与设计、商业、科学、健康与医学、人文与社会科学以及技术与工程。

MATH-Vision (MATH-V) (K. Wang 人，2024 年) 是一个经过精心策划的集合，包含 3040 个高质量的数学问题，这些问题的可视化背景均来自真实的数学竞赛。它涵盖 16 个不同的数学学科，难度分为 5 级。该数据集提供了全面而多样的挑战，因此非常适合用于评估 LMM 的数学推理能力。

MathVista（Lu 等人，2023 年）是一个基准，它整合了各种数学和视觉任务的挑战，要求参与者表现出精细、深刻的视觉理解力，并具备组合推理能力，以成功完成任务。