arXiv:2602.01120v1 [cs.LG] 1 Feb 2026

MarkovScale：迈向推理时的最优序贯扩展

Youkang Wang、Jian Wang、Rubing Chen、Tianyi Zeng、Xiao-Yong Wei，*IEEE*高级会员，以及Qing Li，*IEEE*会士

摘要—序贯扩展是一种重要的推理时扩展范式，但其性能提升通常较为有限，且尚未得到充分理解，这主要是因为启发式、缺乏理论依据的方法普遍存在，从而掩盖了清晰的最优性界。为此，我们提出一个有理论依据的框架，将序贯扩展建模为双状态马尔可夫过程。该方法揭示了序贯扩展的内在性质，并为若干关键方面给出了闭式解，例如准确率得以提升的具体条件，以及理论上的性能上界、中性界和下界。基于这一表述，我们开发了MarkovScale，这是一个实用系统，它应用这些最优性准则，在准确率与效率之间实现有理论依据的平衡。在3个骨干大语言模型、5项基准测试以及超过20种配置上开展的全面实验表明，MarkovScale始终优于最先进的并行扩展方法与序列扩展方法，标志着向大语言模型最优且资源高效的推理迈出了重要一步。源代码将在录用后于https://open-upon-acceptance公开。

索引词—推理时扩展、马尔可夫决策过程、大语言模型推理、概率框架

I. 引言

THE 长期以来，大语言模型（LLM）推理能力的提升一直遵循缩放定律范式。该范式认为，性能提升主要通过增加计算资源实现，并使模型能够在更大的数据集上进行训练、使用更多模型参数。然而，近期出现了一种范式转变，即推理时扩展 [1]–[4]，：在不修改基础模型的情况下，通过增加推理时的计算预算来提升性能。这种方法具有显著优势：它保留了原有的模型架构，减少了对庞大计算基础设施的依赖，并为传统的基于训练的扩展方法提供了一种灵活而高效的替代方案。对于难以获取模型参数或高性能计算资源的用户而言，这一点尤为重要。

当前关于推理时扩展方法的研究大致区分了两种途径：并行扩展 和序贯扩展。并行扩展技术（例如，Best-of-N [5],多数投票 [6], [7]）会汇总多个相互独立的模型输出，从中选出最优回答；而序贯方法（例如思维链 [8], [9], 多轮思考 [10]）通过将先前的推理步骤纳入后续迭代，逐步优化输出。尽管实证研究表明，与并行或混合方法相比，序贯扩展通常带来的性能提升较小，但其在令牌效率方面具有独特优势 [11]。此外，序贯范式在多智能体辩论等交互式推理场景中尤为重要，因为每个智能体的论点都必须与先前的交流连贯衔接。我们的研究致力于改进序贯扩展方法，使其性能达到或超过并行方法，同时保留其固有的效率优势。我们将这一双重目标——最大化性能，同时最小化资源消耗——视为实现**扩展最优性**的核心挑战。

Youkang Wang、Rubing Chen、Tianyi Zeng、Xiaoyong Wei 和 Qing Li 任职于香港理工大学计算学系。Jian Wang 和 Xiao-Yong Wei 任职于四川大学计算机科学学院（电子邮箱：{yk2022.wang、rubing.chen、tianyi.zeng}@connect.polyu.hk；jwanglvy@gmail.com；{cs007.wei、qing-prof.li}@polyu.edu.hk）。通信作者为 Xiao-Yong Wei（cs007.wei@polyu.edu.hk）。

当前方法难以实现最优性，原因在于它们依赖启发式方法，而非有原则的统计基础。因此，难以推导性能或 token 使用量的上界和下界的闭式表达式。为弥补这一不足，我们证明序贯扩展过程可以建模为双状态马尔可夫过程，其中状态间的扩展转移对应于大语言模型将答案修正为正确或错误答案。通过纳入大语言模型在初始迭代中输出正确答案的零样本概率，我们可以形式化地表示任意迭代时两个状态上的概率分布。这使我们能够建立最优性界限的闭式解。

我们的主要贡献如下：

- **有原则的公式**：我们提出一种新颖的离散时间双状态马尔可夫公式，为序贯扩展动态提供概率表征。该框架不仅增强了可解释性，还揭示了序贯扩展的基本属性。值得注意的是，可以预先解析地预测给定大语言模型的性能收敛，从而在大规模实验之前支持模型评估与选择。

- **性能特征分析**：我们的分析揭示，序贯扩展性能取决于模型固有的推理能力（由转移概率刻画）以及依赖于问题的零样本准确率。该公式能够在早期计算出兼顾准确率与效率的最优停止迭代次数，同时也为判断缩放何时提升或降低性能确立了明确标准。

- **实际实现**：我们开发了MarkovScale，一个

实用框架，通过在理论最优状态终止生成来应用所推导的最优性准则。在3个骨干大语言模型、5个代表性推理基准以及20余种配置上的全面评估表明，MarkovScale在序列缩放和并行缩放两种设置下均优于最先进的方法。

II. 相关工作

A. 大语言模型推理时扩展

推理时扩展是一种很有前景的范式，无需修改模型参数即可增强大语言模型在复杂推理任务上的表现 [8]。它通过在推理时增加计算量而非重新训练，提供了一种轻量且灵活的替代方案，以提升推理性能。推理时扩展方法大致可分为并行扩展 和序贯扩展 [12]。自洽性 [13] 和 Best-of-N（BoN）等并行方法会在一次前向传播中生成多个相互独立的输出 [14], , 再通过投票或选择汇总结果。序贯方法则逐步优化输出，将先前的推理作为上下文，以进行更深入的推理。尽管序贯扩展带来的性能提升通常小于并行或混合方法，但它在令牌效率方面具有显著优势 [11], , 这促使我们专注于推进这一方向。

B. 序贯扩展

序贯扩展通过多步迭代增强大语言模型的推理能力，每一步都基于先前的输出，从而实现反思、精炼和纠正 [15]。然而，朴素的迭代方法往往会遭遇收益递减，甚至导致性能下降 [15]。例如，多轮思考 [10]试图通过重新审视和精炼初始输出提升答案质量，但收益往往很快便会趋于平稳。这揭示了一个关键认识：仅仅增加步骤并不足够，有效的序贯扩展需要策略性引导，以确保取得有意义的进展。一类重要的序贯方法建立在大语言模型自我纠正和自我验证的内在能力之上。ReVISE [16] 使模型能够评估并修订自身的推理路径，而 SETS [17] 则将采样、验证和纠正整合到一个统一循环中。Self-Refine [15], [18] 采用递归反馈循环，让同一个模型分别充当生成器、批评器和修订者。尽管这些方法前景可观，但由于大语言模型可靠检测并纠正自身错误的能力有限，它们可能不够稳定。此外，随着人们日益关注预算意识，出现了诸如 s1 [19], 这样的方法，通过解码时干预控制推理成本；还有一些方法研究输出长度约束下的推理，或对过长的推理链施加惩罚 [20], [21]。“过度思考” [22] 和“思考不足” [23] 等概念进一步凸显了序贯扩展中有效且高效地分配资源的必要性。作为一种代表性方法，思维原子 [24]将推理分解为迭代分解-收缩过程，并将计算资源用于直接关联当前原子问题状态的推理。

与以往基于启发式或经验驱动的方法不同，我们提出的 MarkovScale 框架为序贯扩展提供了一种有原则且具有坚实理论基础的方法。通过将迭代过程建模为马尔可夫链，我们推导出明确的停止准则和理论界，以刻画其收敛至固有准确率上界的过程，从而应对先前工作中观察到的性能不稳定性和资源使用效率低下等挑战。

III. 马尔可夫序贯扩展

本节将序贯扩展建模为一个马尔可夫过程。这一形式化不仅揭示了序贯扩展的核心性质，还使我们能够推导出扩展有益时所满足的闭式条件。

A. 将序贯扩展视为马尔可夫过程

序贯扩展是一个迭代推理过程。给定问题 q，大语言模型在第 i^{th} 次迭代时生成答案 o_i，并以第 $(i-1)^{th}$ 次迭代时的前一个答案 o_{i-1} 作为 $o_i = f(q, o_{i-1})$。我们通过标记答案是否正确来表示每次迭代的状态，记为 $X_i \in \{C, W\}$，其中 C表示答案 o_i 正确， W 则表示不正确。这就形成了一个两状态马尔可夫链，因为

$$\mathbb{P}(X_i|X_{i-1}, X_{i-2}, \cdots, X_0) = \mathbb{P}(X_i|X_{i-1}). \tag{1}$$

a) 建模假设与范围。： 我们强调，两状态形式化是一种有意采用的粗粒度抽象。我们并不对推理质量或置信度轨迹进行建模，而是专注于迭代优化所引发的正确性动态。这种抽象捕捉了预期的正确性转移行为，而这正是推理时扩展直接优化的量。

该表述既不假设推理质量是二元的，也不假设内部推理状态具有马尔可夫性。相反，我们将马尔可夫性质施加于可观测的正确性结果，而经验结果表明，这会产生足够稳定的转移统计，从而实现准确的预测和界估计（见第 V-E 节）。

我们将转移概率定义如下：1) $a = P(X_i = W \mid X_{i-1} = C)$：从正确答案转变为错误答案的概率。2) $b = P(X_i = C \mid X_{i-1} = W)$：从错误答案转变为正确答案的概率。由此得到转移矩阵

$$\mathbf{P} = \begin{bmatrix} 1-a & a \\ b & 1-b \end{bmatrix}.$$

令 $p_0(q) = \mathbb{P}(X_0 = C)$ 表示大语言模型在初始迭代中输出正确答案的零样本概率。第 i^{th} 次迭代时的状态概率分布向量 $\boldsymbol{\pi}_i$ 为：

$$\begin{aligned} \boldsymbol{\pi}_i &= \big[\mathbb{P}(X_i = C), \mathbb{P}(X_i = W)\big] \\ &= \boldsymbol{\pi}_0 \underbrace{\mathbf{P}\mathbf{P}\cdots\mathbf{P}}_{i \text{ times}} = \boldsymbol{\pi}_0 \mathbf{P}^i = [p_0, 1-p_0]\mathbf{P}^i \end{aligned} \tag{2}$$

B. 正确状态收敛概率

令 $p_i(q) = \mathbb{P}(X_i = C)$ 表示第 i^{th} 次迭代时获得正确答案的概率。根据式（2），我们有$p_i = \boldsymbol{\pi}_0\mathbf{P}^i\mathbf{e}_1$ 其中，$\mathbf{e}_1 = [1,\ 0]^\top$ 是正确状态对应的基向量。我们进一步使用矩阵对角化对$\mathbf{P}$ 进行分解，得到$\mathbf{P} = \mathbf{S}\mathbf{D}\mathbf{S}^{-1}$，其中

$$\mathbf{D} = \begin{bmatrix} 1 & 0 \\ 0 & \lambda \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 1 & a \\ 1 & -b \end{bmatrix}, \quad \mathbf{S}^{-1} = \frac{1}{a+b}\begin{bmatrix} b & a \\ 1 & -1 \end{bmatrix}$$

以及 $\lambda = 1 - a - b$。由此可得

$$\mathbf{P}^i = \frac{1}{a+b}\begin{bmatrix} b + a\lambda^i & a(1-\lambda^i) \\ b(1-\lambda^i) & a + b\lambda^i \end{bmatrix}. \tag{3}$$

随后计算 p_i ，如下：

$$p_i = p_0 \cdot \mathbf{P}^i_{11} + (1-p_0)\cdot\mathbf{P}^i_{21} = \frac{b}{a+b} + \frac{\lambda^i}{a+b}\left(ap_0 - b(1-p_0)\right). \tag{4}$$

C. 有益扩展条件

先前研究通过实验证明，序贯扩展有时能够提升性能，而在其他情况下则会导致性能下降 [10], [11]。然而，这些发现仍停留在观察层面，缺乏理论解释，尤其是对其内在机制以及扩展在哪些确切条件下有益的解释。如式（4）所示，马尔可夫序贯扩展过程由骨干模型的能力和问题的难度决定。这两个主要因素共同体现在零样本概率 p_0 以及转移概率 a 和 b中。基于这些参数，这一表述使我们能够建立闭式判据，明确序贯扩展何时具有优势。

为量化序贯扩展的优势，我们引入一个收益函数

$$g_i(q) = p_i(q) + \sigma - p_0(q). \tag{5}$$

对于给定问题 q ，该函数通过比较迭代性能（即 $p_i(q)$）与基线（即 $p_0(q)$），衡量答案质量的提升，并考虑理论偏差项 σ。该偏差项 σ 对以下因素引入的总体建模误差给出上界：(i) 验证器噪声，(ii) 推理中的非马尔可夫依赖，(iii) 随机解码变异性，以及 (iv) 预测误差界。因此， σ 定义的是鲁棒性裕度，而非自由超参数。当缩放提升答案质量时， $g_i(q)$ 的值大于零；当性能下降时，其值小于零。

将式 (4) 中 $p_i(q)$ 的闭式表达式代入 $g(q)$，可得

$$\begin{aligned} g_i(q) &= \left[L + \lambda^i(p_0 - L)\right] - p_0 + \sigma \\ &= (L - p_0) + \lambda^i(p_0 - L) + \sigma = (L - p_0)(1 - \lambda^i) + \sigma \end{aligned} \tag{6}$$

其中$L = \frac{b}{a+b}$ 且 $\lambda = 1 - a - b$。由于 $\lambda \in (0,1)$，$\lambda^i \to 0$当 $i \to \infty$时，渐近收益

$$\lim_{i\to\infty} g_i(q) = (L - p_0)(1 - 0) + \sigma = L - p_0 + \sigma. \tag{7}$$

$\lim g(q)$的符号定义了以下三种扩展机制：1) **有益扩展** ($\lim g(q) > 0$)：当$p_0(q) < L + \sigma$ 时出现，且该缩放单调提升置信度；2) **有害扩展** ($\lim g(q) < 0$)：当 $p_0(q) > L + \sigma$ 时出现，且该缩放导致渐近退化；以及3) **中性扩展** ($\lim g(q) = 0$)：当 $p_0(q) = L + \sigma$ 时出现，且置信度保持不变。这些闭式条件使得可将缩放作为准则进行选择性应用，具体如下：

定理 III.1（有益的序贯扩展判据）。 对于给定问题 q，当且仅当满足以下条件时，序贯扩展可被证明有益：

$$p_0(q) < \frac{b}{a+b} + \sigma. \tag{8}$$

D. 模型选择与性能界量化

除了判断何时缩放有利之外，实践者通常还面临两项主要挑战：1）在实验之前，从可用选项中选出最优的骨干大语言模型；2）量化所选缩放策略与理论性能上限和下限的接近程度。我们的框架为这两项挑战都提供了有原则的解决方案。

a) 模型选择： 对于给定的候选模型，我们的闭式解（式 (4)）使实践者能够在进行任何实验之前，直接估算其期望收敛准确率 p_i （在朴素序贯缩放下），如下所示：

定理 III.2（期望收敛准确率）。 对于任何满足 p_0（q） $< L$ 且 $|\lambda| < 1$ 的问题，其中 $\lambda = 1-a-b$，正确概率将以指数速度收敛至：

$$\lim_{i\to\infty} p_i = L = \frac{b}{a+b} \tag{9}$$

证明。 根据式 (4)， $p_i = L + \lambda^i(p_0 - L)$，瞬态项$\lambda^i(p_0 - L)$随着 $i \to \infty$ 而消失，因为 $|\lambda| = |1-a-b| < 1$。剩余项 L 构成不动点。然而在实践中，最终的收敛准确率需要考虑偏差 σ，因此收敛至 $L + \sigma$。 □

该分析揭示了模型固有的序贯扩展能力，其中 L表示无界缩放下的预期准确率。它可作为模型选择的指标。

b) 性能界：该定理还可以扩展为估计给定模型的性能界。具体而言，对数据集应用朴素序贯缩放后，我们得到：

- **中性界**： $\bar{L} = \frac{\bar{b}}{\bar{a}+\bar{b}}$，其中 $\bar{a}$ 和 $\bar{b}$ 是观测到的转移概率。这代表了大多数扩展策略的基线收敛点，并可作为选择策略的基线（即，如果 L 不足，则应考虑其他策略）。
- **上界**： $L^+ = \frac{b^+}{a^+ + b^+}$，其中 a^+ 和 b^+ 是与有益扩展问题相对应的转移概率（如公式（7）所示）。这是理论上可达到的最高准确率，可用于评估某种策略距离最优性能有多近。

• **下界**：$L^- = \frac{b^-}{a^-+b^-}$，其中 a^- 和 b^- 是与有害扩展问题相对应的转移概率（如公式（7）所示）。这是最差情况下的准确率，代表扩展产生负面影响时的性能，从而限制模型的效能。

IV. 最优马尔可夫扩展

A. 采用门控策略的 *MarkovScale*

我们的马尔可夫公式使我们可以闭式推导出基于原理的扩展策略。借助定理 III.1，当初始准确率 $p_0(q)$ 达到或超过阈值 $\frac{b}{a+b} + \sigma$ 时，我们可以轻松实现一种完全跳过缩放的基本门控策略。这会自动过滤掉落入有害扩展区间或中性扩展区间的问题。我们将该策略记为**MarkovScale**^{t}。

B. 具备扩展最优性的 *MarkovScale*

虽然 MarkovScale^t 能够通过选择性缩放来减少token使用量，但对于有益扩展区间内的问题，它未必是最优的。为此，我们将**扩展最优性** 定义为一种运行状态，其中序贯扩展同时满足两个条件：1）所得准确率超过指定的置信水平 $\tau \in (0,1)$，以及 2）token使用量最小化。我们将此建模为一个优化问题，以确定最优迭代次数 i^* 为

$$i^* = \arg\min_{i\in[0,\infty)} p_i \geq \tau,\ \tau \in (0,1),\ i \in \mathbb{Z}^+. \tag{10}$$

将公式 (4) 代入公式 (10)，可得最优停止准则：

$$\frac{b}{a+b} + \frac{\lambda^i}{a+b}\left(ap_0 - b(1-p_0)\right) \geq \tau \tag{11}$$

$$\Longrightarrow\ \lambda^i\left((a+b)p_0 - b\right) \geq (a+b)\tau - b. \tag{12}$$

对于 $ap_0 \neq b(1-p_0)$ 且 $\lambda \neq 0$ 的非平凡情况，我们得到闭式解：

$$i^* = \left\lceil \log\left(\frac{|(a+b)\tau - b|}{|ap_0 - b(1-p_0)|}\right)\log^{-1}(\lambda)\right\rceil. \tag{13}$$

对于其他特殊情况，我们有（推导过程见附录 A）：1）$i^* = 1$（立即满足），当$\tau \leq \frac{b}{a+b}$时；2）$i^* = \frac{b}{a+b}$（一致收敛）ap_0 当 b —— 3 i^* ~~=1~~ 1 $(1-p_0)$ 时；以及 ）迭代中的精确收敛，当 $\lambda = 0$ 时。通过将缩放最优性与这些值进行比较，并在最优迭代处停止缩放，可以节省令牌消耗量。

1）***MarkovScale****：最大后验估计：为实现最优缩放，我们需要估计转移概率 a 和 b，以及答案正确的初始概率 p_0。在我们的方法中，转移概率被视为模型特有的属性，反映了模型修正答案的能力（即其自我反思能力）。这些概率可以根据历史推理数据（例如训练期间观察到的状态转移）计算经验均值，从而轻松估计。另一方面，初始概率 p_0 表示模型的零样本性能，并取决于特定问题 q。

算法 1 MarkovScale* 与最大后验估计和最优停止

输入： 问题 q，置信水平 τ，最大迭代次数 N，先验强度 γ，偏差 σ。
输出： 最终补全 o_i。

```
// 离线预测
计算 p̂0 ← M0(大语言模型(q))。
计算 â ← Ma(大语言模型(q)), b̂ ← Mb(大语言模型(q))。
计算 λ ← 1 − â − b̂.
// 门控（MarkovScale^t）
如果 p̂0 ≥ b/(â+b̂) + σ 则
    返回零样本输出。
结束如果
// 初始化最大后验先验
计算 α0 ← p̂0 · γ。
计算 β0 ← (1 − p̂0) · γ。
对于 i = 1 至 N 执行
    生成补全 oi。
    计算验证器得分 ϕ(oi) ∈ [0, 1]。
    αi ← αi−1 + ϕ(oi)。
    βi ← βi−1 + (1 − ϕ(oi))。
    通过式 (18) 计算最大后验估计 p_i^MAP。
    // 最优停止检查
    如果 λ^i ((â + b̂)p_i^MAP − b̂) ≥ (â + b̂)τ − b̂ 则
        跳出
    结束如果
结束循环
返回最终补全 oi。
```

我们采用最大后验（MAP）来估计 p_0，通过离线学习估计初始值，并通过在线优化动态调整。我们将这一策略称为 **MarkovScale***。整体工作流程总结见算法1。

先验的离线学习。我们在冻结模型之上采用多层感知机（记为 $\mathcal{M}_0$），学习从给定问题 q 到概率 $p_0(q)$ 的映射，该概率表示大语言模型将为 q 生成正确答案的可能性。我们的训练数据由问题-概率对 $(q, \bar{p})$ 组成，其中 $\bar{p}$ 是通过对每个问题 q的特定数量生成答案的结果进行聚合得到的。多层感知机层训练完成后，$p_0(q)$ 的离线估计可通过

$$\hat{p}_0(q) = \mathcal{M}_0(\mathrm{LLM}(q)). \tag{14}$$

类似地，我们使用额外的多层感知机层来学习 a和b，将问题 q 映射为转移概率。为了构造问题—转移概率对，我们首先在训练数据集上使用朴素的多轮序贯扩展进行推理。随后，我们通过统计推理步骤之间的状态转移来进行估计，具体而言，包括大语言模型从先前的正确答案转为错误答案的情况（以正确状态的数量为分母进行归一化，从而计算a），以及从先前的错误答案转为正确答案的情况（以错误状态的数量为分母进行归一化，从而计算 b）。构造好训练数据后，我们训练两个多层感知机作为$\hat{}$的估计器，用于预测$\hat{a}(q)$和 $b(q)$，其定义如下：

$$\hat{a}(q) = \mathcal{M}_a(\mathrm{LLM}(q)),\quad \hat{b}(q) = \mathcal{M}_b(\mathrm{LLM}(q)), \tag{15}$$

其中， $\mathcal{M}_a$ 和 $\mathcal{M}_b$ 是不同的多层感知机。尽管这种形式可行，但我们发现，针对每个问题预测$\hat{a}$和$\hat{b}$ 具有较高方差，且缺乏稳定性。为缓解这一问题，我们对每个问题的估计值进行聚合，即对每个数据集中的所有问题取平均，从而得到统一的数据集级转移概率 a 和 b ，并在推理时使用。使用数据集级转移概率时，MarkovScale 优化的是问题分布上的期望正确性，而非严格意义上的逐问题最优性。在温和的集中性假设下，数据集级转移概率能够在期望意义上近似逐问题的动态，从而平均而言实现近乎最优的停止决策。

后验概率在线精化。 为了在推理期间精化离线估计，我们需要为 p 定义概率密度函数（PDF）。由于大语言模型对每个问题的输出（正确或错误）都可以视为一次伯努利试验，因此我们将 p 建模为多个试验上的贝塔分布。由此得到 $p \sim Beta(\alpha_0, \beta_0)$，其参数初始化如下

$$\alpha_0 = \hat{p}_0 \cdot \gamma, \quad \beta_0 = (1 - \hat{p}_0) \cdot \gamma. \tag{16}$$

其中， $\gamma > 0$ 控制先验强度。对于第 i^{th} 次迭代，我们按如下方式更新后验概率：

$$\alpha_i = \alpha_{i-1} + \phi(o_i), \quad \beta_i = \beta_{i-1} + (1 - \phi(o_i)) \tag{17}$$

其中， $\phi(o_i)$ 是一个用于估计 i^{th} 答案质量的函数。该函数可以直接使用常见的验证器实现，例如过程奖励模型（PRMs）。随后，最大后验估计可表述如下：

$$p_i^{\text{MAP}} = \begin{cases} \frac{\alpha_i}{\alpha_i + \beta_i}, & \text{if } \alpha_i \leq 1 \text{ or } \beta_i \leq 1, \\ \frac{\alpha_i - 1}{\alpha_i + \beta_i - 2}, & \text{otherwise.} \end{cases} \tag{18}$$

*2)**MarkovScale***^{0}：免训练估计： 尽管第 IV-B1 节所述的最大后验估计方法效果良好，但它依赖于对初始概率预测器 $\hat{p}_0$ 及转移概率预测器$\hat{a}$ 和 b 进行离线训练。这会带来额外的训练开销和模型特定依赖。为减轻这些依赖，我们提出**MarkovScale^0**，，一种完全免训练的变体。具体而言，我们使用启发式先验初始化贝塔分布参数，以设定 p_0。对于转移概率 a 和 b，我们对每个数据集中的所有问题进行最少轮数的推理，并利用这些样本基于现有样本统计计算转移概率。这相当于对 $\hat{a}$ 和 b 进行近似的经验估计。随后，我们采用式（17）和式（18）中所述的相同在线精化策略，其中 $\phi(o_i)$ 使用 PRMs 计算。该方法利用序贯扩展的迭代特性，从头开始引导概率估计，无需进行任何模型训练。因此，MarkovScale^0 能够适应各种不同的模型。

V. 实验

A. 实验设置

基准测试。 我们采用了 5 个具有挑战性的推理基准测试：(1) **MATH-500** [25], 包含 500 道高中竞赛题；(2) **GSM8K** [5],包含 8.5K 道小学题目；(3) **AIME 2024** [26] 和 (4) **AIME 2025**[1], 各包含 30 道奥赛预备级数学题；(5) **AMC 2023**[2], 包含美国数学竞赛的中级题目。这些多样化的测试平台用于评估不同难度级别下的缩放性能。

骨干模型。 我们采用广泛使用的开源大语言模型作为骨干模型。具体而言，我们使用 DeepSeek-R1-Distill-Qwen-7B[27] 和 DeepSeek-R1-Distill-Llama-8B [27]作为采用不同架构的蒸馏推理模型。为评估高容量、大规模设置下的性能，我们还纳入了 QwQ-32B [28]。

基线方法。我们将 MarkovScale 与以下具有竞争力的推理时扩展方法进行评估：(1) *Budget-Forcing* [19], 一种简单的预算约束顺序缩放策略；(2)*MR-Thinking* [10], 逐轮迭代重新思考先前答案；(3)思维原子（*AoT*）[3] [24], 采用迭代分解-收缩过程进行缩放；(4) *Best-of-N*（*BoN*） [5], 并行生成多个输出，并依据外部验证器评分选出最佳答案；(5)自一致性（*SC*） [6],依靠对多个采样生成结果进行多数投票；(6) 自适应自一致性（*ASC*） [29]: 当已有生成结果形成明显多数时，根据轻量级停止准则停止采样；(7) 早停自一致性（*ESC*） [30], 通过早停策略增强自一致性；以及(8) 难度自适应自一致性（*DSC*） [31], 利用问题难度信息自适应分配预算。

实现细节。为实现MarkovScale*，我们针对每个MATH-500训练问题生成60个答案，以估计零样本正确率。它们作为多层感知机预测器的训练数据，该预测器以1e-5的学习率训练30个训练轮次。我们采用Qwen2.5-Math-PRM-7B作为验证器。在MarkovScale^0，中，我们使用经验先验初始化贝塔分布参数： $\alpha_0 = 9$ 和$\beta_0 = 1$。我们将 N 定义为生成过程中允许的最大序贯迭代次数或并行采样数，并在 {8, 16, 32, 64}中变化 N 。为确保公平比较，我们对所有方法使用固定的解码温度0.7和top-p 值0.95。所有实验均使用 4×H20 GPU进行。我们同时报告准确率 和词元数 ，以分别评估推理时扩展的有效性与效率。

B. 性能比较与准确率-*Token* 消耗权衡

表 I 汇报了不同骨干模型和基准测试配置下的定量结果，而图 1 展示了不同方法在准确率与令牌消耗量方面的缩放对比

[1]https://huggingface.co/datasets/math-ai/aime25. [2] https://huggingface.co/datasets/math-ai/amc23.[3]实验使用 https://github.com/qixucen/atom 上的官方实现进行。我们发现，当前代码库的 AoT 深度配置存在问题，导致表 I 中报告的性能未达最优。

表 I：不同推理时扩展方法的性能比较，结果以准确率（%）报告。每个基准测试列显示 $N \in \{8, 16, 32, 64\}$ 的结果。

基准测试	MATH-500				GSM8K				AIME 2024				AIME 2025				AMC 2023			
N	8	16	32	64	8	16	32	64	8	16	32	64	8	16	32	64	8	16	32	64
DeepSeek-R1-Distill-Llama-8B																				
基线（不扩展）	75.2				71.1				43.3				13.3				72.5			
预算强制	75.8	75.8	75.8	75.8	86.9	86.9	86.9	86.9	23.3	23.3	23.3	23.3	23.3	23.3	23.3	23.3	65.0	65.0	65.0	65.0
思维原子	33.0	33.0	33.0	33.0	50.9	50.9	50.9	50.9	16.7	16.7	16.7	16.7	6.7	6.7	6.7	6.7	35.0	35.0	35.0	35.0
N选优	82.0	83.0	83.8	84.6	74.4	75.7	76.3	76.9	43.3	50.0	53.3	66.7	26.7	30.0	30.0	33.3	75.0	75.0	80.0	82.5
自一致性	88.4	90.2	91.2	91.8	81.8	84.6	86.7	87.5	53.3	50.0	50.0	56.7	23.3	23.3	23.3	23.3	77.5	80.0	75.0	80.0
MR-Thinking	85.4	88.2	86.4	88.2	**90.5**	91.2	90.9	90.8	43.3	50.0	60.0	56.7	33.3	40.0	26.7	30.0	**80.0**	82.5	85.0	82.5
ESC	88.4	90.2	91.0	91.4	81.8	84.6	86.7	87.2	53.3	50.0	50.0	56.7	23.3	23.3	23.3	23.3	77.5	80.0	75.0	80.0
ASC	88.4	90.0	91.0	91.4	81.8	84.5	86.2	86.9	53.3	50.0	50.0	56.7	23.3	23.3	23.3	23.3	77.5	80.0	75.0	80.0
DSC	84.0	89.4	86.8	91.0	81.8	84.5	86.2	86.9	53.3	50.0	50.0	56.7	13.3	13.3	23.3	23.3	77.5	80.0	75.0	80.0
MarkovScale^t （我们的方法）	**90.2**	**92.0**	**92.4**	**93.2**	89.7	**92.0**	**93.2**	**93.9**	**60.0**	**73.3**	**73.3**	**76.7**	**40.0**	**43.3**	**43.3**	**43.3**	**80.0**	**87.5**	**95.0**	**92.5**
MarkovScale* （我们的方法）	**90.2**	**92.0**	**92.4**	**93.2**	89.7	**92.0**	**93.2**	**93.9**	**60.0**	**73.3**	**73.3**	**76.7**	**40.0**	**43.3**	**43.3**	**43.3**	**80.0**	**87.5**	**95.0**	**92.5**
MarkovScale^0 （我们的方法）	**90.2**	**92.0**	**92.4**	**93.2**	89.7	**92.0**	**93.2**	**93.9**	**60.0**	**73.3**	**73.3**	**76.7**	**40.0**	**43.3**	**43.3**	**43.3**	**80.0**	**87.5**	**95.0**	**92.5**
DeepSeek-R1-Distill-Qwen-7B																				
基线（不扩展）	89.6				87.0				53.3				36.7				82.5			
预算强制	85.8	85.8	85.8	85.8	90.7	90.7	90.7	90.7	36.7	36.7	36.7	36.7	30.0	30.0	30.0	30.0	80.0	80.0	80.0	80.0
思维原子	46.8	46.8	46.8	46.8	77.7	77.7	77.7	77.7	36.7	36.7	36.7	36.7	33.3	33.3	33.3	33.3	70.0	70.0	70.0	70.0
N选优	**94.8**	93.8	94.4	94.0	92.4	93.5	93.1	94.0	**70.0**	**76.7**	70.0	**80.0**	43.3	46.7	50.0	**53.3**	**95.0**	**95.0**	**95.0**	**95.0**
自一致性	93.4	93.4	93.2	93.4	90.1	90.8	90.9	91.2	60.0	66.7	63.3	60.0	40.0	36.7	40.0	40.0	85.0	87.5	90.0	92.5
MR-Thinking	91.2	91.6	91.4	92.0	88.4	88.3	88.4	88.2	56.7	53.3	63.3	70.0	40.0	43.3	33.3	40.0	87.5	87.5	90.0	90.0
ESC	93.4	93.4	93.4	93.4	90.1	90.8	90.9	91.1	60.0	66.7	63.3	60.0	40.0	36.7	40.0	40.0	85.0	87.5	90.0	92.5
ASC	93.4	93.4	93.2	93.4	90.1	90.8	90.9	91.1	60.0	66.7	63.3	60.0	40.0	36.7	40.0	40.0	85.0	87.5	90.0	92.5
DSC	93.2	93.2	93.0	93.2	90.1	90.7	90.9	91.1	60.0	66.7	63.3	60.0	40.0	36.7	40.0	40.0	82.5	87.5	87.5	90.0
MarkovScale^t （我们的方法）	94.6	**95.0**	**95.0**	**95.6**	**93.3**	**94.1**	**94.3**	**94.6**	**70.0**	70.0	**73.3**	73.3	**53.3**	**56.7**	50.0	46.7	92.5	**95.0**	**95.0**	**95.0**
MarkovScale* （我们的方法）	94.6	**95.0**	**95.0**	**95.6**	**93.3**	**94.1**	**94.3**	**94.6**	**70.0**	70.0	**73.3**	73.3	**53.3**	**56.7**	50.0	46.7	92.5	**95.0**	**95.0**	**95.0**
MarkovScale^0 （我们的方法）	94.6	**95.0**	**95.0**	**95.6**	**93.3**	**94.1**	**94.3**	**94.6**	**70.0**	70.0	**73.3**	73.3	**53.3**	**56.7**	**53.3**	46.7	92.5	**95.0**	**95.0**	**95.0**
QwQ-32B																				
基础（不扩展）	94.2				94.8				70.0				56.7				85.0			
预算强制	92.2	92.2	92.2	92.2	87.6	87.6	87.6	87.6	**73.3**	73.3	73.3	73.3	**70.0**	70.0	70.0	70.0	67.5	67.5	67.5	67.5
思维原子	56.0	56.0	56.0	56.0	89.0	89.0	89.0	89.0	43.3	43.3	43.3	43.3	20.0	20.0	20.0	20.0	80.0	80.0	80.0	80.0
N选优	94.6	94.6	95.2	94.8	**95.8**	95.5	95.5	95.8	**73.3**	73.3	73.3	76.7	66.7	63.3	63.3	66.7	**95.0**	**97.5**	**97.5**	**97.5**
自一致性	95.2	95.2	95.0	95.4	95.6	95.5	95.7	95.8	70.0	66.7	70.0	70.0	66.7	60.0	63.3	63.3	92.5	92.5	92.5	92.5
MR-Thinking	94.6	94.4	94.4	93.8	95.3	**95.6**	95.4	94.8	70.0	70.0	63.3	73.3	53.3	56.7	53.3	56.7	87.5	80.0	75.0	80.0
ESC	95.2	95.2	95.0	95.4	95.6	95.5	95.7	95.8	70.0	66.7	70.0	70.0	66.7	60.0	60.0	60.0	92.5	92.5	92.5	92.5
ASC	95.2	95.2	95.2	95.4	95.6	95.5	**95.8**	**95.9**	70.0	66.7	70.0	70.0	66.7	60.0	63.3	63.3	92.5	92.5	92.5	92.5
DSC	95.0	94.4	94.6	94.6	95.2	95.1	95.1	95.3	70.0	66.7	70.0	70.0	66.7	60.0	63.3	63.3	92.5	92.5	92.5	92.5
MarkovScale^t （本文）	**96.6**	96.6	96.4	**96.6**	95.5	**95.6**	95.6	95.6	**73.3**	**76.7**	**80.0**	**83.3**	**70.0**	**73.3**	**76.7**	**76.7**	87.5	87.5	87.5	87.5
MarkovScale* （我们的方法）	**96.6**	96.6	**96.6**	**96.6**	95.4	**95.6**	95.6	95.6	**73.3**	**76.7**	**80.0**	**83.3**	**70.0**	**73.3**	**76.7**	**76.7**	87.5	87.5	87.5	87.5
MarkovScale^0 （我们的方法）	**96.6**	**96.8**	**96.6**	**96.6**	95.5	**95.6**	95.6	95.6	**73.3**	**76.7**	**80.0**	**83.3**	**70.0**	**73.3**	**76.7**	**76.7**	87.5	87.5	87.5	90.0

。两者共同凸显了 MarkovScale 在优化性能与效率之间权衡方面的有效性，以下关键发现便是明证：

在不同骨干模型上均取得更高准确率。如表 I 所示，在所有测试的骨干模型上，MarkovScale 的表现始终显著优于基线。例如，MarkovScale 在 DeepSeek-R1-Distill-Llama-8B 上的准确率平均提高了 19.7%。即使在更强大的 QwQ-32B 模型上，准确率平均也提升了 7.7%。这些结果凸显了 MarkovScale 在推理时缩放方面的泛化能力，无论底层模型采用何种架构。此外，我们的结果表明，MarkovScale 在 R1-Distill-Qwen-2.5-7B、R1-Distill-Llama-3.1-8B 和 QwQ-32B 上均展现出稳健的性能。从 7B 到 32B 都能保持一致的表现，验证了所提出的方法具有良好的泛化

在不同模型规模之间有效地

MarkovScale 各变体的独特优势。表 I 显示，MarkovScale 的三个变体表现均良好。其中，最容易实现的是 MarkovScale^t。MarkovScale^0 和 MarkovScale* 的准确率相近，而免训练的 MarkovScale^0 通常会使用更多词元。这可能是因为 MarkovScale^0 要求大语言模型达到更高的内部置信度阈值，因此往往需要更多迭代。这体现了训练轻量级预测器所需投入与节省推理词元所带来的效率之间的实际权衡。

准确率与成本之间的帕累托最优权衡。MarkovScale 能够持续识别缩放过程中的最优停止点；超过该点后，增加迭代次数带来的准确率提升会逐渐减小。这一特性对 QwQ-32B 等更大的模型尤其有益，因为不必要的

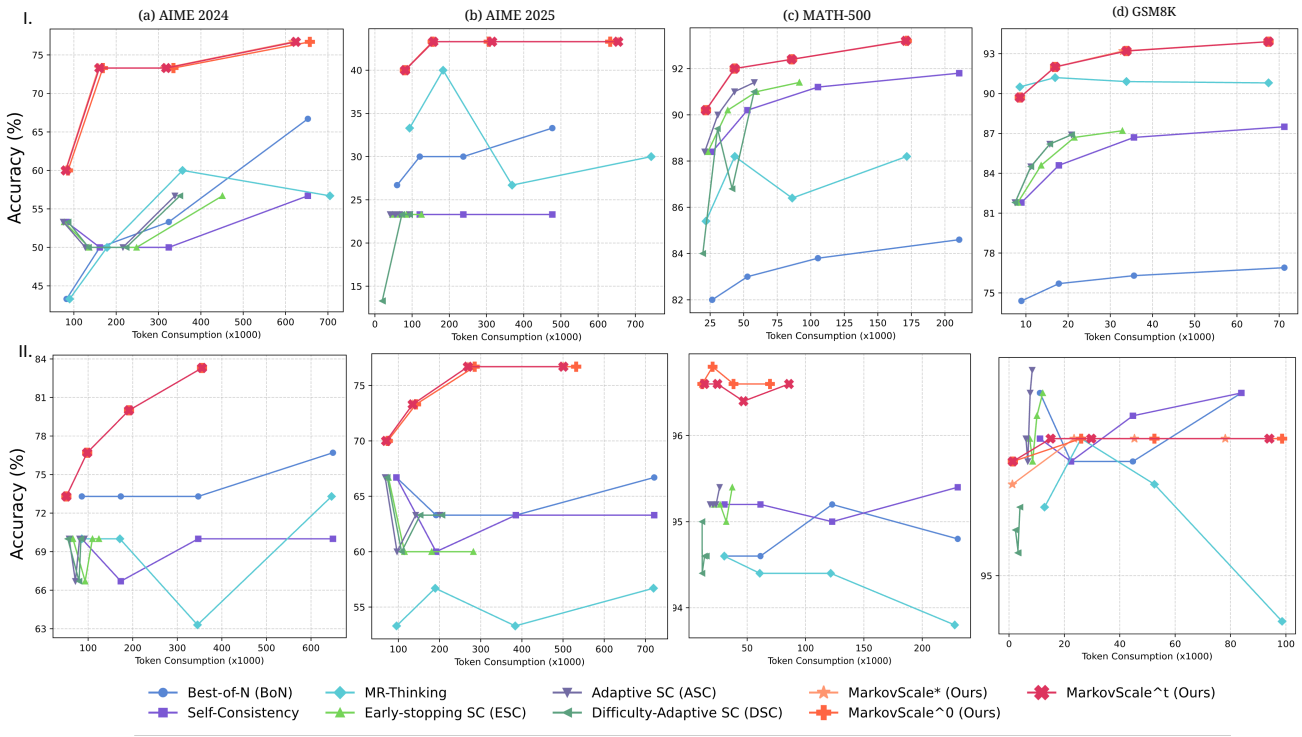


图1：以 (i) DeepSeek-R1-Distill-Llama-8B（上排）和 (ii) QwQ-32B（下排）为骨干模型时，不同方法的缩放性能（准确率与平均token消耗）。

缩放可能造成高昂成本。如图 1 所示，与基线相比，MarkovScale 能以显著更低的 token 预算达到更高的准确率。例如，在 MATH-500 上，其准确率达到 97%，同时相比 MR-Thinking 基线节省约 70% 的 token。在 AIME 2024 和 AIME 2025 上也分别观察到了类似趋势。这些结果表明，MarkovScale 在准确率-成本权衡方面实现了显著的帕累托最优。

预算变化下的稳健缩放。MarkovScale 对缩放预算表现出显著的稳健性。如表 I 所示，在改变缩放预算N且处于 $\{8, 16, 32, 64\}$范围内时，它在大多数配置下始终保持最先进的性能。最重要的是，根据具体配置的不同，它在取得这些提升的同时，还将 token使用量降低了 5% 至 70%。这凸显了 MarkovScale 高效适应推理约束的潜力。

C. 对扩展最优性影响因素的消融研究

表 II 展示了 MarkovScale 在 Deepseek-R1-Distill-Qwen-7B 上结合 $N = 8$ 、跨不同基准测试的消融研究。我们得出两项关键发现：**1）最优停止的有效性**：与式 (13) 所定义的无最优停止（w/o OS）策略变体相比，完整的 MarkovScale* 在保持相同准确率的同时显著降低了 token使用量。例如，在 MATH-500 上，完整的 MarkovScale* 使用 17,114 个 token，较 w/o OS 变体的 20,123 个 token 减少了 17.6%。在其他基准测试上也观察到类似趋势。这些结果证实，所提出的最优停止策略能够避免不必要的缩放步骤，从而在不牺牲性能的情况下带来显著的效率提升。**2）缩放门控（SG）的作用**：同时去除最优停止与缩放门控的 MarkovScale* 变体（w/o OS+SG）在准确率和 token使用量方面表现明显更差。例如，在 GSM8K 上，完整的 MarkovScale* 以 4,238 个 token 达到 93.9% 的准确率，而 w/o OS+SG 变体使用 4,792 个 token 仅获得 88.4% 的准确率，有效性与效率均明显下降。这些结果支持定理 III.1，验证了缩放门控是实现最优缩放的一项至关重要的原则性策略。

D. 置信水平 τ 的敏感性研究

我们通过在 $[0.90, 0.99]$范围内改变超参数 τ ，对 MarkovScale* 进行了敏感性研究。如表 III 所示，在 R1-Distill-Qwen-7B 模型上使用 $N = 64$ 和 $\gamma = 10$时，实验观察到的最优值分别为：AIME 2024 上的 $\tau = 0.93$ 、AIME 2025 上的 $\tau = 0.92$ 、AMC 2023 上的 $\tau = 0.93$ 、GSM8K 上的 $\tau = 0.97$ 以及 MATH-500 上的$\tau = 0.98$ 。结果表明， τ 的最优选择与基准难度相关：较难的任务（例如 AIME 2024 、 AIME 2025 、 AMC 2023 ）通常偏好约为 0.88–0.93的取值，而较简单的数据集（例如 GSM8K、MATH-500）则在约为 0.95–0.98的较高取值下表现最佳。

E. 马尔可夫公式的有效性及其拓展

为评估马尔可夫公式的有效性，我们将理论推导的界与观察到的经验结果进行比较，以回答以下问题：

表 II：MarkovScale 的消融研究结果（以 Deepseek-R1-Distill-Qwen7B 为骨干模型，$N = 8$），涵盖不同基准测试。“准确率”表示准确率（%），“词元数”表示词元数量。

基线方法	MATH-500		GSM8K		AIME 2024		AIME 2025		AMC 2023	
	Acc.	词元数 (↓)	Acc.	词元数 (↓)	Acc.	词元数 (↓)	Acc.	词元数 (↓)	Acc.	词元数 (↓)
MarkovScale* （完整）	95.0	**17114**	93.9	4238	63.3	41192	50.0	**51991**	92.5	**16220**
不使用缩放门控（SG）	95.0	17515	93.9	**2801**	63.3	**40151**	50.0	52938	92.5	16235
不使用最优停止（OS）	95.0	20123	93.9	5056	63.3	41906	50.0	54772	92.5	17962
不使用OS +SG	91.2	21396	88.4	4792	56.7	78432	40.0	86568	87.5	36780

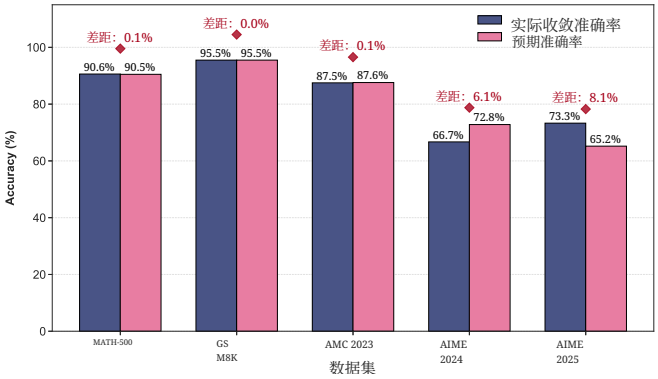


图 2：期望准确率与实际收敛结果的比较。

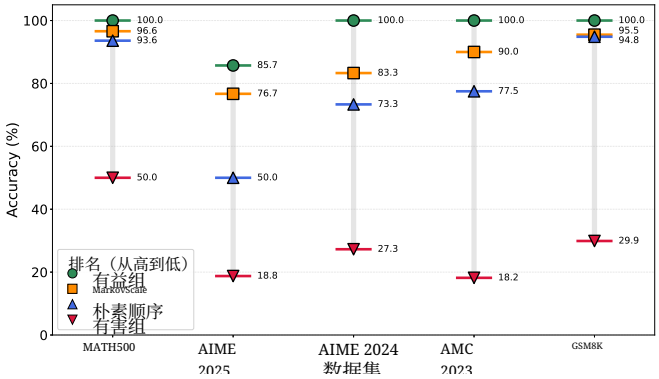


图 3：MarkovScale 相对于理论界的性能。

表 III：使用 Deepseek-R1-Distill-Qwen-7B 作为骨干模型，在不同基准测试上采用不同阈值 τ设置时 MarkovScale* 的实验结果。

阈值 τ	MATH-500		GSM8K		AIME 2024		AIME 2025		AMC 2023	
	Acc.	词元数(↓)	Acc.	词元数(↓)	Acc.	词元数(↓)	Acc.	词元数(↓)	Acc.	词元数(↓)
0.90	89.6	2,939	87.0	1,027	66.7	320,742	43.3	468,315	85.0	34,110
0.91	90.8	3,491	88.3	2,150	70.0	381,891	43.3	468,315	90.0	60,334
0.92	92.6	5,718	89.5	3,637	70.0	405,844	**46.7**	**541,722**	92.5	117,790
0.93	94.2	40,591	92.7	6,250	**73.3**	**481,951**	46.7	618,765	**95.0**	**118,222**
0.94	95.0	55,209	93.9	9,634	73.3	537,113	46.7	619,701	95.0	158,662
0.95	95.0	81,217	94.4	14,537	73.3	600,713	46.7	623,294	95.0	192,809
0.96	95.2	104,107	94.5	18,554	73.3	602,617	46.7	659,270	95.0	224,076
0.97	95.4	134,130	**94.6**	**23,573**	73.3	606,782	46.7	675,129	95.0	251,998
0.98	**95.6**	**159,118**	94.6	30,601	73.3	622,478	46.7	684,446	95.0	280,571
0.99	95.6	168,226	94.6	37,971	73.3	625,850	46.7	684,446	95.0	286,638

1）能否预测预期准确率？ 定理 III.2 给出了朴素序列缩放方法的理论准确率上限，该上限取决于骨干模型的能力，并由转移概率体现。图 2 比较了使用 Deepseek-R1-Distill-Qwen7B 在多个基准测试上得到的理论准确率与实际收敛结果。理论预测与经验准确率高度吻合，平均绝对误差（MAE）仅为 2.88%。值得注意的是，在五个数据集中，有三个数据集的预测误差可以忽略不计（≤0.1%），凸显了马尔可夫公式的稳健性。由于这些理论准确率可以提前计算，因此有助于实现预期性能，并能为是否值得提前开展大规模实验提供参考。

2）理论界能否用于评估性能？ 在第 III-D 节中，我们介绍了用于确定理论中性界、上界和准确率下界的方法。为评估 MarkovScale 的相对性能，我们将其实际准确率与这些界进行比较。如图 3 所示，MarkovScale 始终接近理论上界，明显优于中性界和下界。这不仅证明了 MarkovScale 的有效性，也为评估未来的序列扩展方法提供了一个实用框架，清晰揭示它们与最优性能的接近程度，以及它们是否超越平均（中性界）或最坏情况（下界）。

VI. 结论

本文提出 MarkovScale，这是一种用于大语言模型（LLM）推理时最优扩展的系统化框架。通过将序贯扩展过程建模为

离散时间马尔可夫链，我们的工作奠定了首个理论基础，并提出了具有可证明最优性性质和理论界的实用算法。MarkovScale 解决了现有方法的一项关键局限：这些方法依赖启发式词元扩展，在准确率与计算成本之间难以实现高效或可预测的权衡。通过揭示序贯扩展的底层结构，MarkovScale 还为分析自我修正动态开辟了新途径，例如大语言模型中的“过度思考”现象 [32]以及视频搜索中的语义推理 [33], [34]，从而为复杂推理任务提供有价值的见解。将这一表述扩展到多状态或非齐次马尔可夫链（例如置信度校准状态或验证器条件状态），是未来值得探索的方向。

附录

详细解的推导

这里我们详细推导式（12）的解：

$$((a+b)p_0 - b) \cdot \lambda^i \geq (a+b)\tau - b,$$

求满足不等式的最小整数解 i 。沿用之前的符号，其中 $\lambda = 1-a-b \in (0,1)$ 且 $L = \frac{b}{a+b}$。

a) 情形结构与解释。： 在枚举各种情形之前，我们注意到，该不等式完全由两个线性项的相对符号决定，即 $(a+b)p_0 - b$ 和 $(a+b)\tau - b$，其中 $\lambda \in (0,1)$。因此，分析可分为四种互斥情形：（i）任一侧为零的退化情形；（ii）符号一致且通过对数反演可得到有限解的情形；（iii）符号一致但不存在可行整数解的情形；（iv）符号相反，意味着不等式要么立即满足，要么无解。每种情形对应一种定性上不同的缩放行为。

情况 1：方程左侧等于零（即$(a+b)p_0 - b = 0$）。

这对应于马尔可夫链的平稳准确率，此时缩放在期望意义上既不会提高也不会降低正确性。

我们有

$$(a+b)p_0 - b = 0 \Rightarrow p_0 = \frac{b}{a+b} = L.$$

不等式变为：

$$0 \cdot \lambda^i \geq (a+b)\tau - b \Rightarrow 0 \geq (a+b)\tau - b \Rightarrow \frac{b}{a+b} \geq \tau \Rightarrow L \geq \tau.$$

因此，情况 1 的解为：

- 若 $p_0 = L \geq \tau$，则任意 i 都满足该不等式（因此，最小值为 $i = 1$）。
- 如果 $p_0 = L < \tau$，则没有 i 满足该不等式。

情况2：方程右侧等于零（即（$a+b$）$\tau - b = 0$）。

此情况表示置信阈值等于马尔可夫不动点，因此停止条件仅取决于初始准确率高于还是低于均衡值。

我们有：

$$(a+b)\tau - b = 0 \Rightarrow \tau = \frac{b}{a+b} = L.$$

不等式变为：

$$[(a+b)p_0 - b]\lambda^i \geq 0.$$

因此，情况2的解为

- When $(a+b)p_0 - b \geq 0$ (i.e., $p_0 > L$):

$$\lambda^i \geq 0.$$

这对所有 i 都成立，因为 $\lambda > 0$ （最小值 $i = 1$）。

- 当 $(a+b)p_0 - b < 0$ （即 $p_0 < L$）：

$$\lambda^i \leq 0.$$

永远不成立，因为 $\lambda^i > 0$ 对所有 i 都成立（无解）。

情况3：左侧和右侧各项的符号一致。现在可以取对数（即 $(a+b)\tau - b$ $(a+b)p_0 - b > 0$）。

在这种情况下，不等式两侧的符号相同，这意味着缩放可以单调地使准确率接近（或远离）置信阈值，从而可以通过对数求出最优停止时间。

情况3.1： $(a+b)p_0 - b > 0$ **且** $(a+b)\tau - b > 0$。
</style}不等式变为：

$$\lambda^i \geq \frac{(a+b)\tau - b}{(a+b)p_0 - b} = \frac{\tau - L}{p_0 - L}.$$

取对数（$\lambda \in (0,\ 1) \Rightarrow \lambda^i > 0$ 且 $\frac{\tau - L}{p_0 - L} > 0$），并将 log λ 移至右侧（log $\lambda < 0$）：

$$i \leq \frac{\log\left(\frac{\tau - L}{p_0 - L}\right)}{\log \lambda}.$$

有两种子情况：

- 当 $p_0 > \tau > L$ 时：

$$\frac{\tau - L}{p_0 - L} \in (0,1) \Rightarrow \log\left(\frac{\tau - L}{p_0 - L}\right) < 0,$$

since $\log \lambda < 0 \Rightarrow i \leq$ positive number.

因此，所有 $0 < i \leq$ 对数$\left(\frac{\tau - L}{p_0 - L}\right)$ 对数 λ 都满足该不等式（最小值为 $i = 1$）。

- 当 $\tau > p_0 > L$ 时：

$$\frac{\tau - L}{p_0 - L} > 1 \Rightarrow \log\left(\frac{\tau - L}{p_0 - L}\right) > 0,$$

因为 $\log \lambda < 0 \Rightarrow i \leq$ 是负数。

无解。

因此，当初始准确率已经超过稳态点 L 时，仅当 $p_0 > \tau$ 时缩放才有益；否则，无论进行多少缩放，都无法达到所需的置信度。

情形 3.2： $(a+b)p_0 - b < 0$ **且** $(a+b)\tau - b < 0$。 不等式方向反转：

$$\lambda^i \leq \frac{(a+b)\tau - b}{(a+b)p_0 - b} = \frac{\tau - L}{p_0 - L}.$$

有两种子情况：

- 当 $p_0 < \tau < L$：

$$\frac{\tau - L}{p_0 - L} \in (0, 1).$$

同样，取对数可得：

$$i \geq \frac{\log\left(\frac{\tau-L}{p_0-L}\right)}{\log \lambda} > 0.$$

最小值 i 为向上取整 $\left\lceil \frac{\log\left(\frac{\tau-L}{p_0-L}\right)}{\log \lambda} \right\rceil$。

- 当 $\tau < p_0 < L$：

$$\frac{\tau - L}{p_0 - L} > 1.$$

由于对所有 $i \geq 1$，$\lambda^i < 1$ 均成立，因此不存在解。

这是唯一存在非平凡有限最优迭代次数的区间，并由此得到式（13）中的闭式向上取整表达式。

情形 4：左侧与右侧的符号不同。对数不适用（即 $(a+b)\tau - b$ $(a+b)p_0 - b < 0$）。

当两侧符号相反时，不等式要么对所有迭代都显然成立，要么对所有迭代都不成立，因此无需进行对数分析。

情形 4.1： $(a + b)p_0 - b < 0$ **且** $(a + b)\tau - b > 0$。不等式变为：

$$\lambda^i \leq \frac{(a + b)\tau - b}{(a + b)p_0 - b} = \frac{\tau - L}{p_0 - L} < 0.$$

不存在解，因为对于所有 i，$\lambda^i > 0$。

这对应于有害扩展区间，在该区间中，相较于目标阈值，缩放会严格降低正确性。

情形 4.2： $(a + b)p_0 - b > 0$ **和** $(a + b)\tau - b < 0$。不等式变为：

$$\lambda^i \geq \frac{(a + b)\tau - b}{(a + b)p_0 - b} = \frac{\tau - L}{p_0 - L} < 0.$$

这对所有 $i \geq 1$ 均成立，因为 $\lambda^i > 0 > \frac{\tau-L}{p_0-L}$（最小值 $i = 1$）。

这种情形意味着立即满足停止条件，因此采用了 MarkovScale^t 中的常数时间停止规则。

b) 总结： 总而言之，该不等式会产生三种性质迥异的结果：**(i)** 立即满足（$i^* = 1$），**(ii)** 无可行解（应跳过缩放），或 **(iii)** 存在唯一的有限最优迭代次数，由式 **(13)** 给出。按情形进行的这一分析，完整刻画了 MarkovScale 在所有参数区间内的最优停止行为。

不同 MarkovScale 变体的比较

表 IV 对三个 MarkovScale 变体在准确率、token 使用量和实际部署考量方面进行了并列比较。我们的发现和见解如下：**1)** 尽管设计极简，采用最简单的首轮门控规则的 MarkovScale^t，表现却意外地接近更复杂的变体，因此在优先考虑实现简便性的情况下颇具吸引力。**2)** 完全免训练的 MarkovScale^0，始终能取得较高的准确率，有时甚至超过 MarkovScale*，但通常会产生更高的推理阶段 token 成本。**3)** 相比之下，MarkovScale* 利用已训练预测器减少推理阶段计算，因此在大多数设置下往往具有更高的 token 效率。这凸显了一种实际权衡：MarkovScale^0 无需承担任何训练开销，但推理时会消耗更多计算资源；而 MarkovScale* 则在预测器训练阶段预先投入计算，以提升推理阶段的使用效率。

超参数分析

MarkovScale主要使用两个超参数：1）停止阈值 τ，它规定推理过程停止缩放之前所需的准确率概率；2）先验强度因子 γ（仅用于MarkovScale*），它控制离线先验$\hat{}p_0$ 在贝塔分布参数的在线MAP更新中的权重。

表V报告了先验强度因子 γ 的不同设置以及置信水平 τ 在 $[0.80, 0.99]$ 范围内变化（间隔为0.01）时的缩放性能，使用Deepseek-R1-Distill-Qwen-7B作为骨干模型，并在AIME 2024 基准测试上进行评估。我们采用 $\gamma = 10$，这是一个具有代表性的中等范围设置。表中的经验结果支持这一选择：较低的先验强度（例如 $\gamma = 1$ 或 $\gamma = 4$）会导致性能明显更差，$\gamma = 1$ 的峰值准确率为63.33%，$\gamma = 4$为60.00%，而 $\gamma = 10$则为73.33%。敏感性分析还表明，将γ 从非常小的值向上调通常会带来稳定的准确率提升，而过弱的先验则会在几乎所有阈值上造成显著的准确率下降。综合来看，这些趋势验证了 $\gamma = 10$ 是一个稳健且校准良好的设置。

参考文献

[1] B. Brown、J. Juravsky、R. Ehrlich、R. Clark、Q. V. Le、C. R´e 和 A. Mirhoseini，《大型语言猴子：通过重复采样扩展推理计算》，*arXiv* 预印本 *arXiv:2407.21787*，2024年。[2] Y. Wu、Z. Sun、S. Li、S. Welleck 和 Y. Yang，《扩展推理计算：语言模型问题求解的计算最优推理》，发表于*NeurIPS'24* 数学推理与人工智能研讨会，2024年。[3] C. Snell、J. Lee、K. Xu 和 A. Kumar，《最优扩展大语言模型测试时计算，比扩展模型参数更有效》，*arXiv* 预印本 *arXiv:2408.03314*，2024年。[4] Q. Zhang、F. Lyu、Z. Sun、L. Wang、W. Zhang、W. Hua、H. Wu、Z. Guo、Y. Wang、N. Muennighoff等，《大语言模型测试时扩展综述：是什么、如何实现、应用于何处，以及效果如何？》，*arXiv* 预印本 *arXiv:2503.24235*，2025年。[5] K. Cobbe、V. Kosaraju、M. Bavarian、M. Chen、H. Jun、L. Kaiser、M. Plappert、J. Tworek、J. Hilton、R. Nakano等，《训练验证器以求解数学文字题》，*arXiv* 预印本 *arXiv:2110.14168*，2021年。[6] X. Wang、J. Wei、D. Schuurmans、Q. V. Le、E. H. Chi、S. Narang、A. Chowdhery 和 D. Zhou，《自一致性提升语言模型中的思维链推理》，发表于第十一届国际学习表征会议，2023年。[7] A. Nguyen、D. Mekala、C. Dong 和 J. Shang，《一致性预测何时可能是正确预测？》，*arXiv* 预印本 *arXiv:2407.05778*，2024年。

表 IV：MarkovScale变体在具有 $N \in \{8, 16, 32, 64\}$ 的代表性推理基准上的比较。“准确率”表示准确率（%），“词元数”表示推理词元总数。

基线方法	MATH-500		GSM8K		AIME 2024		AIME 2025		AMC 2023	
	Acc.	词元数 (↓)	Acc.	词元数 (↓)	Acc.	词元数 (↓)	Acc.	词元数 (↓)	Acc.	词元数 (↓)
DeepSeek-R1-Distill-Llama-8B										
MR-Thinking ($N=8$)	85.4	21868	**90.5**	**8552**	43.3	89923	33.3	93307	**80.0**	34500
MarkovScale^t ($N=8$)	**90.2**	21809	89.7	**8552**	**60.0**	80956	**40.0**	81296	**80.0**	**33837**
MarkovScale* ($N=8$)	**90.2**	**21800**	89.7	**8552**	**60.0**	**80325**	**40.0**	81296	**80.0**	33908
MarkovScale^0 ($N=8$)	**90.2**	21868	89.7	**8552**	**60.0**	86120	**40.0**	**80492**	**80.0**	34500
MR-Thinking（$N=16$）	88.2	43338	91.2	16916	50.0	178390	40.0	183332.0	82.5	67480.0
MarkovScale^t （$N=16$）	**92.0**	43203	**92.0**	16916	**73.3**	160233	**43.3**	157090	**87.5**	66049
MarkovScale* （$N=16$）	**92.0**	**43178**	**92.0**	**16863**	**73.3**	**158395**	**43.3**	156790	**87.5**	**65538**
MarkovScale^0 （$N=16$）	**92.0**	43338	**92.0**	16916	**73.3**	168548	**43.3**	**155195**	**87.5**	67480
MR-Thinking（$N=32$）	86.4	86127	90.9	33819	60.0	356745	26.7	368740	85.0	133628
MarkovScale^t （$N=32$）	**92.4**	85804	**93.2**	33819	**73.3**	318040	**43.3**	315051	**95.0**	130376
MarkovScale* ($N=32$)	**92.4**	**85747**	**93.2**	**32988**	**73.3**	**314751**	**43.3**	311657	**95.0**	**126306**
MarkovScale^0 ($N=32$)	**92.4**	86127	**93.2**	33819	**73.3**	335142	**43.3**	**306124**	**95.0**	133628
MR-Thinking ($N=64$)	88.2	171565	90.8	67478	56.7	704771	30.0	742767	82.5	266692
MarkovScale^t ($N=64$)	**93.2**	170930	**93.9**	67478	**76.7**	624233	**43.3**	653900	**92.5**	260360
MarkovScale* ($N=64$)	**93.2**	**170687**	**93.9**	**67370**	**76.7**	**617252**	**43.3**	653900	**92.5**	**244158**
MarkovScale^0 ($N=64$)	**93.2**	171565	**93.9**	67478	**76.7**	657353	**43.3**	**632523**	**92.5**	266692
Deepseek-R1-Distill-Qwen-7B										
MR-Thinking ($N=8$)	91.2	21396	88.4	4792.0	56.7	78432	40.0	86568	87.5	36780
MarkovScale^t ($N=8$)	**94.6**	20995	**93.3**	4785	**70.0**	35972	**53.3**	81231	**92.5**	20377
MarkovScale* ($N=8$)	**94.6**	**20817**	**93.3**	**4770**	**70.0**	**34021**	**53.3**	**80524**	**92.5**	**18617**
MarkovScale^0 ($N=8$)	**94.6**	21396	**93.3**	4789	**70.0**	41428	**53.3**	81281	**92.5**	36780
MR-Thinking ($N=16$)	91.6	42422	88.3	9615	53.3	157507	43.3	171853	87.5	72877
MarkovScale^t ($N=16$)	**95.0**	42262	**94.1**	9600	**70.0**	64369	**56.7**	159080	**95.0**	38055
MarkovScale* ($N=16$)	**95.0**	**42108**	**94.1**	**8573**	**70.0**	**60624**	**56.7**	**156835**	**95.0**	**33421**
MarkovScale^0 ($N=16$)	**95.0**	42422	**94.1**	9609	**70.0**	76211	**56.7**	157591	**95.0**	72877
MR-Thinking ($N=32$)	91.4	84234	88.4	19135	63.3	313463	33.3	341517	90.0	143178
MarkovScale^t ($N=32$)	**95.0**	82522	**94.3**	19104	**73.3**	254489	50.0	**235833**	**95.0**	71659
MarkovScale* ($N=32$)	**95.0**	**57993**	**94.3**	**14016**	**73.3**	245441	50.0	**235833**	**95.0**	**61585**
MarkovScale^0 ($N=32$)	**95.0**	84234	**94.3**	19122	**73.3**	**244472**	**53.3**	236835	**95.0**	143178
MR-Thinking ($N=64$)	92.0	168331	88.2	38305	70.0	625850	40.0	684446	90.0	286639
MarkovScale^t ($N=64$)	**95.6**	167678	**94.6**	38235	**73.3**	501492	**46.7**	541722	**95.0**	139123
MarkovScale* ($N=64$)	**95.6**	**159118**	**94.6**	**23574**	**73.3**	481952	**46.7**	541722	**95.0**	**118222**
MarkovScale^0 ($N=64$)	**95.6**	168331	**94.6**	38277	**73.3**	**480287**	**46.7**	**469318**	**95.0**	286639
QwQ-32B										
MR-Thinking ($N=8$)	94.6	30378	95.3	12832	70.0	85864	53.3	95536	**87.5**	47288
MarkovScale^t ($N=8$)	**96.6**	13415	**95.5**	1358	**73.3**	**51087**	**70.0**	**70323**	**87.5**	13445
MarkovScale* ($N=8$)	**96.6**	**11113**	95.4	**1221**	**73.3**	**51087**	**70.0**	**70323**	**87.5**	13445
MarkovScale^0 ($N=8$)	**96.6**	11477	**95.5**	1385	**73.3**	**51087**	**70.0**	74022	**87.5**	**8888**
MR-Thinking ($N=16$)	94.4	60801	**95.6**	26043	70.0	171256	56.7	189500	80.0	95434
MarkovScale^t ($N=16$)	96.6	24539	**95.6**	**15016**	**76.7**	**97863**	**73.3**	**134737**	**87.5**	22713
MarkovScale* ($N=16$)	96.6	**12122**	**95.6**	23488	**76.7**	**97863**	**73.3**	**134737**	**87.5**	22713
MarkovScale^0 ($N=16$)	**96.8**	20366	**95.6**	26043	**76.7**	**97863**	**73.3**	142663	**87.5**	**12834**
MR-Thinking ($N=32$)	94.4	121565	95.4	52509	63.3	345355	53.3	383810	75.0	191141
MarkovScale^t ($N=32$)	96.4	46806	**95.6**	**29705**	**80.0**	**191659**	**76.7**	**268905**	**87.5**	40965
MarkovScale* ($N=32$)	**96.6**	**12122**	**95.6**	45293	**80.0**	**191659**	**76.7**	**268905**	**87.5**	40965
MarkovScale^0 ($N=32$)	**96.6**	38298	**95.6**	52509	**80.0**	**191659**	**76.7**	285287	**87.5**	**20354**
MR-Thinking ($N=64$)	93.8	227838	94.8	98607	73.3	646200	56.7	719654	80.0	360068
MarkovScale^t ($N=64$)	**96.6**	85797	**95.6**	94012	**83.3**	**355525**	**76.7**	**500679**	87.5	**73133**
MarkovScale* ($N=64$)	**96.6**	**12122**	**95.6**	**78088**	**83.3**	**355525**	**76.7**	**500679**	87.5	**73133**
MarkovScale^0 ($N=64$)	**96.6**	69552	**95.6**	98607	**83.3**	**355525**	**76.7**	531857	**90.0**	87667

表 V：先验强度因子 γ 不同设置下的实验结果。此处，MarkovScale* 在 AIME 2024 基准测试上使用 Deepseek-R1-Distill-Qwen-7B。

阈值 τ	$\gamma = 1$		$\gamma = 4$		$\gamma = 7$		$\gamma = 10$		$\gamma = 20$	
	Acc.	**词元数** (↓)	**Acc.**	**词元数** (↓)	**Acc.**	**词元数** (↓)	**Acc.**	**词元数** (↓)	**Acc.**	**词元数** (↓)
0.80	60.00	47,360	53.33	9,884	53.33	9,884	53.33	9,884	53.33	9,884
0.81	60.00	47,360	53.33	9,884	53.33	9,884	53.33	9,884	53.33	9,884
0.82	60.00	47,360	53.33	9,884	53.33	9,884	53.33	9,884	53.33	9,884
0.83	63.33	68,998	53.33	9,884	53.33	9,884	53.33	9,884	53.33	9,884
0.84	63.33	69,498	53.33	9,884	53.33	9,884	53.33	9,884	53.33	9,884
0.85	63.33	70,511	53.33	9,884	53.33	9,884	53.33	9,884	53.33	9,884
0.86	63.33	97,876	53.33	9,884	53.33	9,884	53.33	9,884	60.00	91,438
0.87	63.33	98,403	53.33	9,884	53.33	9,884	53.33	9,884	66.67	256,804
0.88	63.33	98,403	53.33	9,884	53.33	9,884	53.33	37,725	73.33	423,459
0.89	63.33	98,547	53.33	9,884	53.33	9,884	66.67	224,497	73.33	483,868
0.90	63.33	99,068	53.33	9,884	53.33	37,725	66.67	320,742	73.33	563,873
0.91	63.33	131,530	53.33	9,884	66.67	200,576	70.00	381,891	73.33	583,646
0.92	63.33	131,964	53.33	9,884	66.67	257,174	70.00	405,844	73.33	595,669
0.93	63.33	162,589	53.33	9,884	70.00	353,703	73.33	481,951	73.33	604,932
0.94	63.33	162,589	53.33	9,884	70.00	353,703	73.33	537,113	73.33	607,838
0.95	63.33	162,589	53.33	9,884	70.00	381,891	73.33	600,713	73.33	615,435
0.96	63.33	225,103	53.33	69,710	73.33	463,866	73.33	602,617	73.33	622,944
0.97	63.33	225,103	60.00	117,653	73.33	501,492	73.33	606,782	73.33	625,850
0.98	66.67	250,354	63.33	185,195	73.33	547,403	73.33	622,478	73.33	625,850
0.99	66.67	250,354	66.67	269,499	73.33	618,059	73.33	625,850	73.33	625,850

[8] J. Wei、X. Wang、D. Schuurmans、M. Bosma、F. Xia、E. Chi、Q. V. Le、D. Zhou 等，“思维链提示激发大语言模型中的推理”，神经信息处理系统进展，第35卷，第24 824–24 837页，2022年。[9] Z. Li、H. Liu、D. Zhou和T. Ma，“思维链使Transformer能够解决固有串行问题”，*arXiv* 预印本 *arXiv:2402.12875*，第1卷，2024年。[10] X. Tian、S. Zhao、H. Wang、S. Chen、Y. Ji、Y. Peng、H. Zhao和X. Li，“再想一次：通过缩放多轮测试时思考增强大语言模型推理”，*arXiv*预印本 *arXiv:2503.19855*，2025年。[11] J. Wang、S. Zhu、J. Saad-Falcon、B. Athiwaratkun、Q. Wu、J. Wang、S. L. Song、C. Zhang、B. Dhingra和J. Zou，“深思、快想：探究无验证器推理时扩展方法的效率”，*arXiv*预印本 *arXiv:2504.14047*，2025年。[12] J. Wang、B. Zhu、C. T. Leong、Y. Li和W. Li，“缩放之上的缩放：探索大型推理模型中的测试时扩展帕累托”，*arXiv*预印本 *arXiv:2505.20522*，2025年。[13] X. Wang、J. Wei、D. Schuurmans、Q. Le、E. H. Chi、S. Narang、A. Chowdhery和D. Zhou，“自一致性改进大语言模型中的思维链推理”，神经信息处理系统进展，第35卷，第24 824–24 837页，2022年。[14] Y. Wang、J. Wang、R. Chen和X.-Y. Wei，“OptScale：面向推理时扩展的概率最优性”，载于*AAAI*人工智能会议论文集，2026年。[15] A. Madaan、N. Tandon、P. Sahoo、S. V. Kaushik和A. Sinha，“Self-Refine：基于自我反馈的迭代优化”，*arXiv*预印本 *arXiv:2303.17651*，2023年。[16] H. Lee、S. Oh、J. Kim、J. Shin和J. Tack，“ReVISE：通过内在自我验证学习在测试时精炼”，*arXiv*预印本 *arXiv:2502.14565*，2025年。[17] J. Chen、J. Ren、X. Chen、C. Yang、R. Sun、J. Yoon和S. ¨O. Arık，“SETS：利用自我验证与自我纠正改进测试时扩展”，*arXiv*预印本 *arXiv:2501.19306*，2025年。[18] X. Chen、M. Lin、N. Sch¨arli和D. Zhou，“教导大语言模型进行自我调试”，*arXiv*预印本 *arXiv:2304.05128*，2023年。[19] N. Muennighoff、Z. Yang、W. Shi、X. L. Li、L. Fei-Fei、H. Hajishirzi、L. Zettlemoyer、P. Liang、E. Cand`es和T. Hashimoto，“s1：简单测试时扩展”，*arXiv*预印本 *arXiv:2501.19393*，2025年。[20] Y. Sun、H. Wang、J. Li、J. Liu、X. Li、H. Wen、Y. Yuan、H. Zheng、Y. Liang、Y. Li 等，“严格输出长度约束下大语言模型推理能力的实证研究”，*arXiv*预印本 *arXiv:2504 14350*，2025年。[21]P. Aggarwal和S. Welleck，“L1：控制一段rea-的时长通过强化学习进行思考的推理模型，” *arXiv* 预印本 *arXiv:2503.04697*，2025年。[22] X. Chen、J. Xu、T. Liang、Z. He、J. Pang、D. Yu、L. Song、Q. Liu、M. Zhou、Z. Zhang 等，“别为 2+ 3= 想那么多？关于类 o1 大语言模型的过度思考”，*arXiv* 预印本 *arXiv:2412.21187*，2024年。[23] Y. Wang、Q. Liu、J. Xu、T. Liang、X. Chen、Z. He、L. Song、D. Yu、J. Li、Z. Zhang 等，“思绪四散：关于类 o1 大语言模型的思考不足”，*arXiv* 预印本 *arXiv:2501.18585*，2025年。[24] F. Teng、Z. Yu、Q. Shi、J. Zhang、C. Wu 和 Y. Luo，“用于马尔可夫大语言模型测试时扩展的思维原子”，*arXiv* 预印本 *arXiv:2502 12018*，2025。.[25] D. Hendrycks、C. Burns、S. Kadavath、A. Arora、S. Basart、E. Tang、D. Song 和 J. Steinhardt，“使用 MATH 数据集衡量数学问题求解能力”，载于第三十五届神经信息处理系统会议数据集与基准测试专场（第二轮），2021年。[26] 美国数学协会，“美国数学邀请赛 - AIME”，https: //huggingface.co/datasets/AI-MO/aimo-validation-aime，2024年2月。[27] D. Guo、D. Yang、H. Zhang、J. Song、R. Zhang、R. Xu、Q. Zhu、S. Ma、P. Wang、X. Bi 等，“DeepSeek-R1：通过强化学习激励大语言模型的推理能力”，*arXiv* 预印本 *arXiv:2501 12948*，2025。.[28] A. Yang、A. Li、B. Yang、B. Zhang、B. Hui、B. Zheng、B. Yu、C. Gao、C. Huang、C. Lv 等，“Qwen3 技术报告”，*arXiv* 预印本 *arXiv:2505.09388*，2025年。[29] P. Aggarwal、A. Madaan、Y. Yang 等，“让我们逐步采样：使用大语言模型实现高效推理与代码生成的自适应一致性”，*arXiv* 预印本 *arXiv:2305.11860*，2023年。[30] Y. Li、P. Yuan、S. Feng、B. Pan、X. Wang、B. Sun、H. Wang 和 K. Li，“摆脱高昂成本：面向多步推理的提前停止自一致性”，*arXiv* 预印本 *arXiv:2401.10480*，2024年。[31] X. Wang、S. Feng、Y. Li、P. Yuan、Y. Zhang、C. Tan、B. Pan、Y. Hu 和 K. Li，“让每一分钱都花得值得：面向高成本效益推理的难度自适应自洽性”，*arXiv* 预印本 *arXiv:2408.13457*，2024年。[32] J. Su、J. Healey、P. Nakov 和 C. Cardie，“思考不足与过度思考之间：关于大语言模型推理长度与正确性的实证研究”，*arXiv* 预印本 *arXiv:2505.00127*，2025年。[33] C.-W. Ngo、Y.-G. Jiang、X. Wei、F. Wang、W. Zhao、H.-K. Tan 和 X. Wu，“Vireo-374 实验：用于语义视频索引与搜索的视觉词袋和基于视觉的本体”，载于*IEEE* 计算机学会，2007年。[34] C.-W. Ngo、Y.-G. Jiang、X. Wei、W. Zhao、F. Wang、X. Wu 和 H.-K. Tan，“超越语义搜索：你观察到的未必是你所想的”，载于*IEEE* 计算机学会，2008年。