ByteDance | Seed

Seed-Thinking-v1.5: Advancing Superb Reasoning Models with Reinforcement Learning

ByteDance Seed

Full author list in Contributions

Abstract

We introduce Seed-Thinking-v1.5, capable of reasoning through thinking before responding, resulting in improved performance on a wide range of benchmarks. Seed-Thinking-v1.5 achieves 86.7 on AIME 2024, 55.0 on Codeforces and 77.3 on GPQA, demonstrating excellent reasoning abilities in STEM and coding. Beyond reasoning tasks, the method demonstrates notable generalization across diverse domains. For instance, it surpasses DeepSeek R1 by 8% in win rate on non-reasoning tasks, indicating its broader applicability. Compared to other state-of-the-art reasoning models, Seed-Thinking-v1.5 is a Mixture-of-Experts (MoE) model with a relatively small size, featuring 20B activated and 200B total parameters. As part of our effort to assess generalized reasoning, we develop two internal benchmarks, BeyondAIME and Codeforces, both of which will be publicly released to support future research.

Date: April 10, 2025

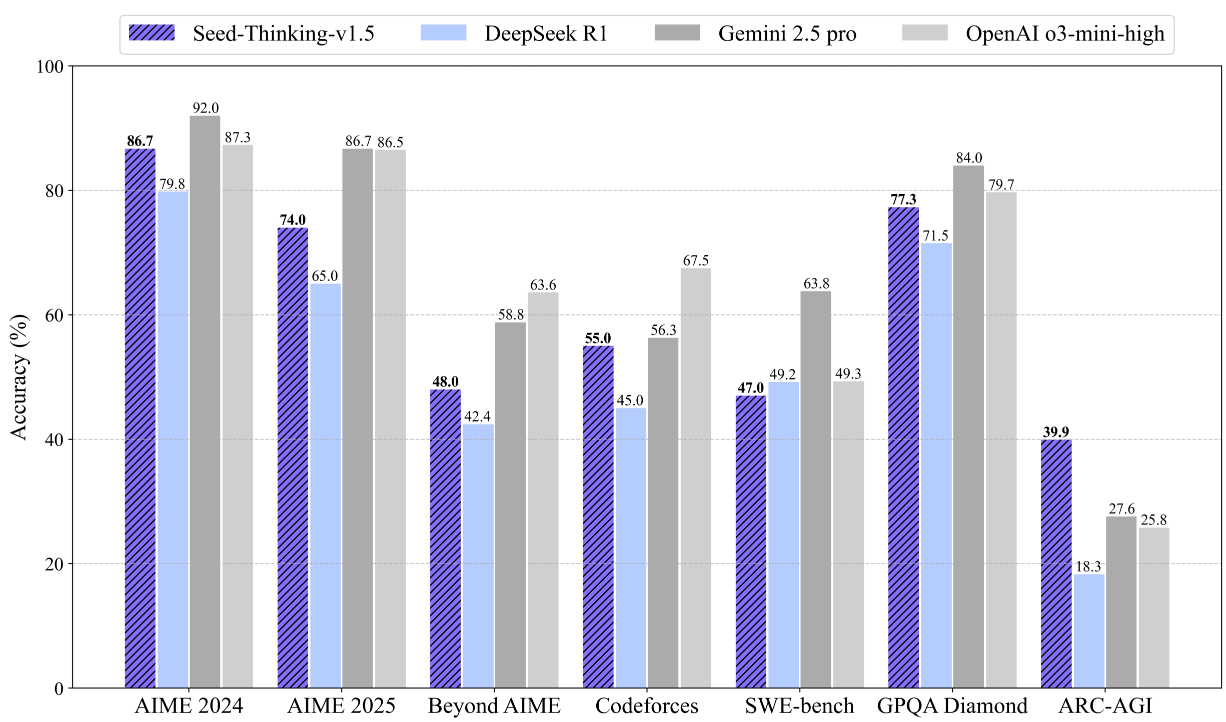

Figure 1 Benchmark performance on reasoning tasks

ByteDance | Seed

Seed-Thinking-v1.5: 通过强化学习推进卓越的推理模型

ByteDance 种子

贡献中的完整作者名单

摘要

我们介绍了Seed-Thinking-v1.5，它能够在回应之前通过思考进行推理，从而在各种基准测试中表现出色。Seed-Thinking-v1.5在AIME 2024上取得了86.7的成绩，在Codeforces上取得了55.0的成绩，在GPQA上取得了77.3的成绩，展示了在STEM和编程方面的卓越推理能力。除了推理任务，该方法在不同领域的多样化任务中也表现出显著的泛化能力。例如，在非推理任务的胜率上，它超过了DeepSeek R1 8%，表明其更广泛的应用潜力。与其他最先进的推理模型相比，Seed-Thinking-v1.5是一个专家混合（MoE）模型，具有相对较小的规模，拥有20B激活参数和200B总参数。作为我们评估泛化推理能力的一部分，我们开发了两个内部基准测试，BeyondAIME和Codeforces，这两个基准测试将公开发布以支持未来的研究。

日期: 2025年4月10日

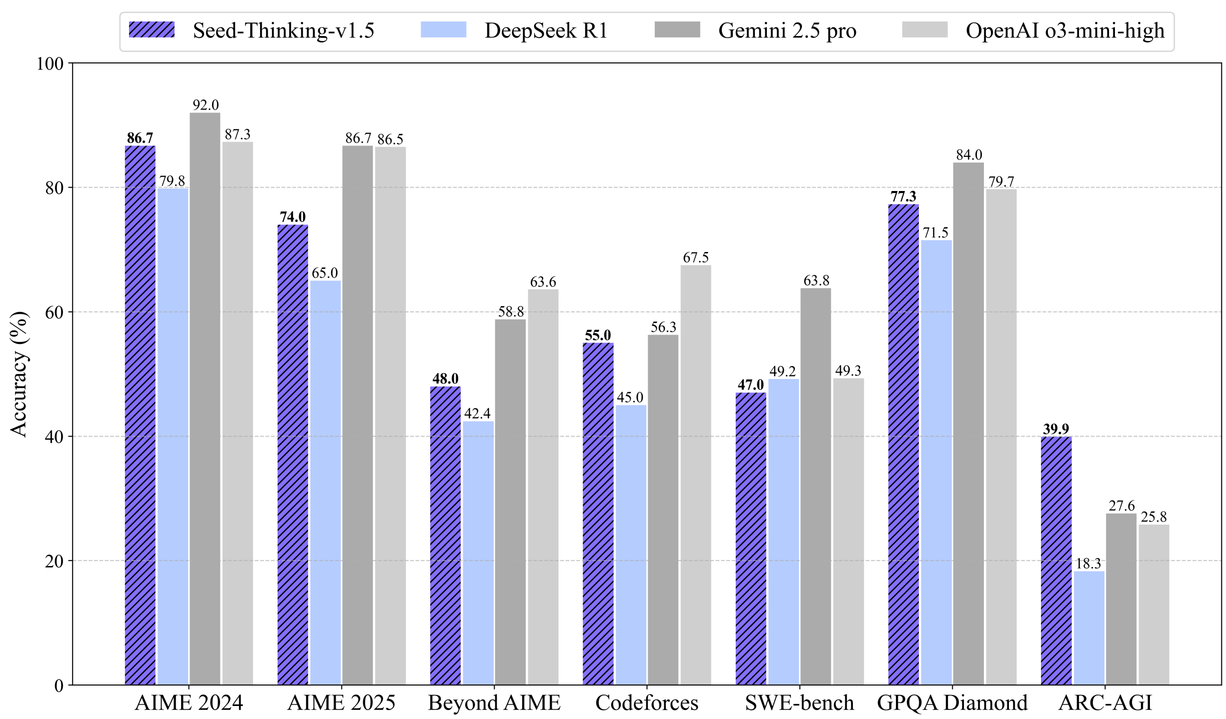

图1 {v*}推理任务的基准性能

1 Introduction

Driven by large-scale reinforcement learning on large language models, reasoning models have seen significant advancements. Notably, OpenAI's o1 series [1], DeepSeek's R1 [2], Google's Gemini 2.5 [3], and Anthropic's Claude 3.7 [4] have emerged as state-of-the-art models, each making substantial progress in logical reasoning, mathematical problem-solving, and code generation. These advancements underscore a shift toward more structured, efficient and scalable reasoning models, with ongoing research focusing on training efficiency, long chain-of-thought, and large-scale reinforcement learning.

In this work, we present a new reasoning model, called Seed-Thinking-v1.5. This model has achieved strong performance in both reasoning and non-reasoning tasks.

Mathematical Reasoning : For math competition, Seed-Thinking-v1.5 achieves 86.7 on AIME 2024, matching the performance of o3-mini-high and significantly outperforming o1 and DeepSeek R1, demonstrating competitive strength. Since AIME 2024 no longer provides sufficient discrimination, we construct a more challenging evaluation set named BeyondAIME. All problems in BeyondAIME are newly curated by human experts and designed to minimize the chance of being solved through memorization or guessing. While Seed-Thinking-v1.5 surpasses both o1 and R1, there remains a performance gap compared to o3 and Gemini pro 2.5. This also further demonstrates the discriminative power of the new evaluation set.

Competitive Programming : For the evaluation of competitive programming, we adopt Codeforces as our benchmark. Unlike some prior works that rely on Elo Scores, which contains estimation and are not directly comparable, we adopt a concrete evaluation protocol based on the most recent 12 Codeforces contests. Specifically, we report pass@1 and pass@8 metrics, where pass@k indicates whether the model solves the problem within k attempts, i.e., selecting the best result from k generated submissions. We choose to report pass@8 since it provides more stable results and aligns more closely with actual user submission patterns. Seed-Thinking-v1.5 outperforms DeepSeek R1 on both metrics, though a performance gap remains compared to o3. The evaluation set will be made publicly available in a future release.

Science : Seed-Thinking-v1.5 reaches a score of 77.3 on GPQA, close to o3-level performance. Importantly, this gain is largely attributed to improved generalization from mathematical training, rather than an increase in domain-specific science data.

Non-reasoning Tasks : For non-reasoning tasks, Seed-Thinking-v1.5 is evaluated using a test set designed to replicate real-world user needs. Through human evaluations conducted against DeepSeek R1 across diverse scenarios, Seed-Thinking-v1.5 demonstrates significant advancements: it attains an 8.0% overall rise in users' positive feedback, thereby highlighting its augmented ability to manage intricate user scenarios.

There are three key points in the development of high-quality reasoning models: training data, RL algorithm, and RL infrastructure. We have devoted considerable effort to these three areas, and we will discuss them in detail.

Data For SFT training, unlike conventional post-training data, reasoning models rely on chain-of-thought data, which explicitly outlines the step-by-step reasoning process. Our preliminary experiments showed that too much non-CoT SFT data can significantly reduce the model's ability to explore. For RL training, we incorporate four categories of data: STEM problems, code-related tasks, logic reasoning and non-reasoning data like creative writing and dialogue. Among these, the logic reasoning data contributes to performance improvements on the ARC-AGI benchmark significantly. The math data exhibits strong generalization capabilities and can lead to broad performance improvements across tasks.

RL Algorithm RL training of reasoning models is highly unstable and often crashes, especially for models without SFT. Sometimes, the score difference between two runs can be as high as 10 points. The stable training of RL systems is crucial for the success of reasoning models. To address these long-standing issues, we have pioneered VAPO[5] and DAPO[6]—two distinct frameworks tailored for actor-critic and policy-gradient RL paradigms, respectively. VAPO now stands as the state-of-the-art (SOTA) solution in actor-critic methods, while DAPO establishes a new SOTA result for policy-gradient approaches

1 引言

由大规模语言模型上的大规模强化学习驱动，推理模型已经取得了显著的进展。特别是，OpenAI 的 o1 系列 [1]、DeepSeek 的 R1 [2]、Google 的 Gemini 2.5 [3] 和 Anthropic 的 Claude 3.7 [4] 已经成为最先进的模型，每个模型在逻辑推理、数学问题解决和代码生成方面都取得了实质性的进展。这些进展强调了向更加结构化、高效和可扩展的推理模型的转变，当前的研究重点在于训练效率、长链思考和大规模强化学习。

在这项工作中，我们提出了一种新的推理模型，称为Seed-Thinking-v1.5。该模型在推理和非推理任务中都取得了优异的性能。

数学推理：在2024年AIME数学竞赛中，Seed-Thinking-v1.5取得了86.7的成绩，与o3-mini-high的表现持平，并且显著优于o1和DeepSeek R1，展示了其竞争力。由于2024年AIME已无法提供足够的区分度，我们构建了一个更具挑战性的评估集，命名为BeyondAIME。BeyondAIME中的所有问题均由人类专家全新策划，旨在尽量减少通过记忆或猜测解决的可能性。尽管Seed-Thinking-v1.5超越了o1和R1，但与o3和Gemini pro 2.5相比仍存在性能差距。这也进一步证明了新评估集的区分能力。

竞赛编程 : 为了评估竞赛编程，我们采用 Codeforces 作为我们的基准。与一些依赖 Elo 分数的先前工作不同，这些分数包含估计且不可直接比较，我们采用基于最近 12 场 Codeforces 比赛的具体评估协议。具体来说，我们报告 pass@1 和 pass@8 指标，其中 pass@k 表示模型是否在 k 次尝试内解决问题，即从 k 次生成的提交中选择最佳结果。我们选择报告 pass@8，因为它提供了更稳定的结果，并且更符合实际用户的提交模式。Seed-Thinking-v1.5 在这两个指标上都优于 DeepSeek R1，尽管与 o3 相比仍存在性能差距。评估集将在未来的版本中公开发布。

科学 : Seed-Thinking-v1.5 在 GPQA 上达到了 77.3 分，接近 o3 级表现。重要的是，这一进步主要归功于数学训练中改进的泛化能力，而不是特定科学领域数据的增加。

非推理任务 : 对于非推理任务，Seed-Thinking-v1.5 使用一个旨在复制现实世界用户需求的测试集进行评估。通过在多种场景下与 DeepSeek R1 进行的人类评估，Seed-Thinking-v1.5 展现了显著的进步：它在用户正面反馈方面总体提高了 8.0%，从而突显了其增强处理复杂用户场景的能力。

在开发高质量推理模型时，有三个关键点：训练数据、RL算法和RL基础设施。我们在这些领域投入了大量努力，并将详细讨论它们。

对于SFT训练的数据，与传统的后训练数据不同，推理模型依赖于链式思维数据，这些数据明确地概述了逐步推理过程。我们的初步实验表明，过多的非CoT SFT数据会显著降低模型的探索能力。对于RL训练，我们纳入了四类数据：STEM问题、代码相关任务、逻辑推理和非推理数据，如创意写作和对话。其中，逻辑推理数据对ARC-AGI基准测试的性能提升贡献显著。数学数据表现出强大的泛化能力，可以导致跨任务的广泛性能提升。

RL算法 在推理模型的RL训练中，训练高度不稳定，经常崩溃，特别是对于没有SFT的模型。有时，两次运行之间的分数差异可能高达10分。RL系统的稳定训练对于推理模型的成功至关重要。为了解决这些长期存在的问题，我们开创了VAPO[5]和DAPO[6]——分别为演员-评论家和策略梯度RL范式量身定制的两个不同框架。VAPO现在是演员-评论家方法中的最先进（SOTA）解决方案，而DAPO为策略梯度方法建立了新的SOTA结果。

without critic models. By targeting the core instability issues in RL training, both methods deliver robust and consistent training trajectories, effectively enabling reliable optimization of reasoning models.

RL Infrastructure The complexity of Large Language Models (LLM) based reinforcement learning systems demands robust infrastructure to ensure scalability, reproducibility, and computational efficiency. To handle heterogeneous workloads, we decouple streaming rollout architecture that asynchronously processes partial trajectory generations through prioritized sample pools, achieving $3\times$ faster iteration cycles than synchronous frameworks. The system also supports mixed-precision training with automatic fault recovery, critical for maintaining stability during large-scale RL runs.

2 Data

2.1 RL Training Data

Our RL training data consists of two main parts: verifiable problems with definitive answers and non-verifiable problems without definitive answers. The model's reasoning ability primarily comes from the first part and can be generalized to the second part.

2.1.1 Verifiable Problems

The Verifiable problems primarily comprise STEM questions paired with answers, coding problems equipped with unit tests, and logic reasonings that are amenable to automated verification.

STEM Data

Our dataset consists of several hundred thousand high-quality, competition-grade problems spanning mathematics, physics, and chemistry, with mathematics comprising the majority (over 80%). These problems are drawn from a mix of open-source datasets, public competitions (both domestic and international), and proprietary collections.

For data cleaning, we first eliminate questions with incomplete statements, inconsistent notation, or unclear requirements. For the remaining questions, we use our model (Doubao-Pro 1.5) to generate multiple responses. Problems for which the model achieved a woN score (worst of N) of 1 are deemed too simple and removed. Finally, some questions may have an inaccurate reference answer. We use SOTA reasoning models to generate multiple candidate responses for each question. If the model's answers were inconsistent with the reference answer, but the model's outputs showed high internal consistency, or involved only a very small number of reasoning tokens, we consider the reference answer to be incorrect. Human experts then conduct manual verification on these questions to ensure that the reference answers are correct. We also apply data augmentation to make the data more suitable for learning and evaluation. Specifically, we convert multiple-choice questions into fill-in-the-blank or short-answer formats to eliminate the possibility of guessing and to better assess reasoning ability. And we modify certain math problems to ensure that the answers are integers whenever possible.

After data cleaning and augmentation, we finally obtain a training set of 100k STEM problems. During training, we use model-based Seed-Verifier to evaluate response correctness, which is introduced in 3.1.

Code Data

For coding problems, we prioritize the source of high-quality and challenging algorithmic tasks, primarily drawn from esteemed competitive programming contests.

We filter data to ensure that each problem includes a comprehensive specification: a clear problem description, a set of unit tests, and a checker script. Unit tests validate the functional correctness of solutions, while the checker script enforces additional constraints such as output formatting and edge cases. We also perform difficulty filtering, ensuring that problems possess an appropriate level of complexity and applicability to real-world algorithmic reasoning.

For evaluation, the most accurate form is to submit the generated code to the official platforms. However, during reinforcement learning, real-time submission isn't feasible. Thus, we developed an off-line evaluation

没有批评模型。通过针对RL训练中的核心不稳定性问题，这两种方法都提供了稳健和一致的训练轨迹，有效地实现了推理模型的可靠优化。

RL 基础设施 基于大型语言模型（LLM）的强化学习系统的复杂性要求强大的基础设施以确保可扩展性、可重复性和计算效率。为了处理异构工作负载，我们解耦了流式展开架构，该架构通过优先级样本池异步处理部分轨迹生成，实现比同步框架快 3× 的迭代周期。该系统还支持混合精度训练和自动故障恢复，这对于在大规模 RL 运行期间保持稳定性至关重要。

2 数据

2.1 RL 训练数据

我们的RL训练数据由两部分组成：可验证的问题和有明确答案的问题，以及不可验证的问题和没有明确答案的问题。模型的推理能力主要来自第一部分，并可以推广到第二部分。

2.1.1 可验证问题

可验证问题主要包含与答案配对的STEM问题、带有单元测试的编程问题以及适合自动验证的逻辑推理。

STEM 数据

我们的数据集包含数十万道高质量、竞赛级别的数学、物理和化学问题，其中数学问题占大多数（超过80%）。这些问题来自开源数据集、公开竞赛（包括国内和国际）以及专有集合的混合。

对于数据清理，我们首先删除陈述不完整、符号不一致或要求不明确的问题。对于剩余的问题，我们使用我们的模型（Doubao-Pro 1.5）生成多个回答。模型获得 woN 分数（N 个中最差的）为 1 的问题被认为是太简单而被移除。最后，某些问题可能有不准确的参考答案。我们使用最先进的推理模型为每个问题生成多个候选答案。如果模型的答案与参考答案不一致，但模型的输出显示了高度的内部一致性，或者只涉及非常少量的推理标记，我们认为参考答案是错误的。然后，人类专家对这些问题进行手动验证，以确保参考答案的正确性。我们还应用数据增强，使数据更适合学习和评估。具体来说，我们将选择题转换为填空题或简答题，以消除猜测的可能性并更好地评估推理能力。我们还修改某些数学问题，确保答案尽可能为整数。

在数据清洗和增强之后，我们最终获得了一个包含100k STEM问题的训练集。在训练过程中，我们使用基于模型的Seed-Verifier来评估响应的正确性，这在3.1节中有所介绍。

代码数据

对于编程问题，我们优先选择高质量和具有挑战性的算法任务，主要来源于备受尊敬的编程竞赛。

我们过滤数据以确保每个问题都包含全面的规范：清晰的问题描述、一组单元测试和一个检查脚本。单元测试验证解决方案的功能正确性，而检查脚本则强制执行额外的约束，例如输出格式和边缘情况。我们还进行难度筛选，确保问题具有适当的复杂性和适用于现实世界的算法推理。

为了评估，最准确的形式是将生成的代码提交到官方平台。然而，在强化学习过程中，实时提交是不可行的。因此，我们开发了离线评估。

set for efficient local validation. Our observations indicate a strong correlation between offline evaluation results and official verdicts. All training and evaluation problems are integrated into an in-house code sandbox environment, enabling direct execution and assessment of model-generated code. We ensure the sandbox's stability and high throughput to deliver consistent and accurate feedback during the RL training process.

Logical Puzzle Data

For the logic reasoning data, we gather 22 commonly studied tasks, such as 24-point, mazes, Sudoku, etc. For each task, we construct a data generator and an answer verifier. The data generator can automatically produce a large amount of training and evaluation data. Moreover, for many of the tasks, we can configure the difficulty of the generated problems. During the training process, we gradually adjust the difficulty of the training data based on the model's performance on certain tasks. The answer verifier rigorously evaluates the generation correctness and can be seamlessly integrated into RL pipelines as reward functions. We generate about 10k puzzle problems for RL training.

2.1.2 Non-verifiable Problems

Non-verifiable problems mainly encompass non-reasoning tasks requiring quality assessment based on human preferences, involving tasks like creative writing, translation, knowledge QA, role-playing, and so on. The prompts are originated from RL training data for Doubao-1.5 Pro [7]. The dataset has sufficient coverage across diverse domains.

We discard data with low sample score variance and low difficulty. To be specific, we use the SFT model to generate multiple candidates for each prompt and then score them using a reward model. Prompts with low score variances are removed as they exhibit limited sampling diversity and minimal potential for improvement. Prompts are also removed where the reward score improvement surpasses a certain threshold during the Doubao 1.5 Pro RL training process [8]. This is because such data may be overly simplistic or already abundantly represented in the dataset. Offline experiments show that overoptimizing such samples leads to premature collapse of the model's exploration space and diminish the performance.

For these non-verifiable data, we employ a pairwise rewarding method for scoring and RL training. By comparing the relative quality of two samples, this approach aids the model in better understanding user preferences, enhancing the quality and diversity of generated results. The detail of the reward model is introduced in 3.2.

2.2 Advanced Math Benchmark

The current reasoning models usually use AIME as the go-to benchmark to evaluate mathematical reasoning abilities. However, with only 30 problems released annually, its limited size can lead to high-variance evaluation results, making it challenging to effectively differentiate between state-of-the-art reasoning models. To better evaluate models' capabilities in mathematical reasoning, we construct a new benchmark dataset: **BeyondAIME**. Specifically, we collaborate with mathematics specialists to develop original problems informed by established competition formats. We systematically adapt existing competition questions through structural modifications and scenario reconfigurations, ensuring no direct duplication occurs. Furthermore, we ensure that the answers are never trivial values—such as numbers explicitly mentioned in the problem statement—to reduce the chance of models guessing the correct answer without proper reasoning.

Through this rigorous filtering and curation process, we compile a final set of 100 problems, each with a difficulty level equal to or greater than that of the hardest questions in AIME. Similar to AIME, all answers are guaranteed to be integers (without being restricted to a specific numerical range), which simplifies and stabilizes the evaluation process.

3 Reward Modeling

As a crucial component in RL, reward modeling defines the objective or goal that the policy is trying to achieve. Thus, a well-designed reward mechanism is essential to provide precise and reliable reward signals for

设置以实现高效的本地验证。我们的观察表明，离线评估结果与官方判决之间存在强烈的关联。所有训练和评估问题都集成到我们内部的代码沙箱环境中，使模型生成的代码能够直接执行和评估。我们确保沙箱的稳定性和高吞吐量，以在RL训练过程中提供一致和准确的反馈。

逻辑谜题数据

对于逻辑推理数据，我们收集了22个常见的研究任务，如24点、迷宫、数独等。对于每个任务，我们构建了一个数据生成器和一个答案验证器。数据生成器可以自动产生大量的训练和评估数据。此外，对于许多任务，我们可以配置生成问题的难度。在训练过程中，我们根据模型在某些任务上的表现逐渐调整训练数据的难度。答案验证器严格评估生成的正确性，并可以无缝集成到RL管道中作为奖励函数。我们为RL训练生成了大约10k个谜题问题。

2.1.2 非可验证问题

非可验证问题主要涉及不需要推理的任务，这些任务需要根据人类的偏好进行质量评估，包括创意写作、翻译、知识问答、角色扮演等。提示来自Doubao-1.5 Pro [7] 的RL训练数据。该数据集在不同领域有足够的覆盖。

我们丢弃样本得分方差低和难度低的数据。具体来说，我们使用SFT模型为每个提示生成多个候选，并使用奖励模型对它们进行评分。得分方差低的提示被移除，因为它们表现出的采样多样性有限，且改进潜力很小。在Doubao 1.5 Pro RL训练过程中，如果奖励得分的提升超过某个阈值，也会移除这些提示[8]。这是因为这样的数据可能过于简单或在数据集中已经过度表示。离线实验表明，对这些样本进行过度优化会导致模型探索空间的过早崩溃，并降低性能。

对于这些不可验证的数据，我们采用成对奖励的方法进行评分和RL训练。通过比较两个样本的相对质量，这种方法有助于模型更好地理解用户偏好，提高生成结果的质量和多样性。奖励模型的详细内容在3.2中介绍。

2.2 高级数学基准测试

当前的推理模型通常使用AIME作为评估数学推理能力的基准。然而，由于每年仅发布30个问题，其有限的规模可能导致评估结果的高方差，使得难以有效地区分最先进的推理模型。为了更好地评估模型在数学推理方面的能力，我们构建了一个新的基准数据集：BeyondAIME。具体来说，我们与数学专家合作，根据已建立的比赛格式开发原创问题。我们通过结构修改和情景重构系统地改编现有的比赛题目，确保不发生直接复制。此外，我们确保答案从不是琐碎的值——如问题陈述中明确提到的数字——以减少模型在没有正确推理的情况下猜出正确答案的机会。

通过这个严格的筛选和策划过程，我们编制了一组100个问题，每个问题的难度都等于或高于AIME中最难的问题。类似于AIME，所有答案都保证是整数（不局限于特定的数值范围），这简化并稳定了评估过程。

3 奖励建模

作为RL中的关键组件，奖励建模定义了策略试图实现的目标或目的。因此，精心设计的奖励机制对于提供精确可靠的奖励信号至关重要。

Verifier-type	Training examples (approximate)	Human labeled testset
Seed-Verifier	$> 98\%$	82.7%
Seed-Thinking-Verifier	$> 99\%$	99.3%

Table 1 Accuracy of two verifier-types. Specifically, the accuracy on the training set is derived from the training statistics. Additionally, we manually annotated 456 samples to form the test set, which are specifically selected from cases that the Seed-Verifier can not handle stably.

model responses during the training stage. For verifiable and non-verifiable problems, we employ distinct reward modeling methodologies.

3.1 Reward Modeling for Verifiable Problems

With proper principles and thought trajectories, we utilize LLMs to judge a wide array of verifiable questions across diverse scenarios. This approach yields a more generalized solution that surpasses the limitations of rule-based reward systems.

We have designed two progressive reward modeling solutions, **Seed-Verifier** and **Seed-Thinking-Verifier**:

- **Seed-Verifier** is based on a set of meticulously crafted principles written by humans. It leverages the powerful foundational capabilities of LLMs to evaluate a triplet consisting of the question, reference answer, and model-generated answer. If the reference answer and model-generated answer are essentially equivalent, it returns "YES"; otherwise, it returns "NO". The equivalence here is not a literal exact match but rather a deeper assessment based on computational rules and mathematical principles that prove the two answers convey the same mathematical meaning. This approach ensures that the reward signal accurately reflects whether the model's response is correct in essence, even if the wording differs.
- **Seed-Thinking-Verifier** is inspired by the human judgment process, which generates conclusive judgments through meticulous thinking and in-depth analysis. To achieve this, we trained a verifier that provides a detailed reasoning path for its evaluations. Specifically, we treated this as a verifiable task and optimized it alongside other mathematical reasoning tasks. This verifier can dissect the similarities and differences between the reference and model-generated answers, offering precise and nuanced judgment results.

The Seed-Thinking-Verifier significantly alleviates three major issues associated with the Seed-Verifier:

- Reward Hacking: Non-thinking models may exploit loopholes to receive rewards without truly understanding the problem. The detailed reasoning process in Seed-Thinking-Verifier makes such hacking more difficult.
- Uncertainty in Predictions: In cases where the reference and model-generated answers are essentially equivalent, which may differ in format, e.g., 2^{19} vs 524288, the Seed-Verifier might sometimes return "YES" and other times "NO". The Seed-Thinking-Verifier provides consistent results by thoroughly analyzing the reasoning behind the answers.
- Failure on Corner Cases: There are certain edge cases that the Seed-Verifier struggles to handle effectively. The ability of Seed-Thinking-Verifier to provide detailed reasoning allows it to better address these complex scenarios.

Table 1 presents the performance of the above two verifiers. More details on case study can be found in Appendix A. The results indicate that the Seed-Verifier struggles to effectively handle some particular cases, whereas the Seed-Thinking-Verifier demonstrates a remarkable ability to provide accurate judgments. While the thinking process of the latter does consume a significant amount of GPU resources, we believe that the precise and robust reward results it generates are crucial for endowing the policy with strong reasoning capabilities.

Verifier-type	Training examples (approximate)	Human labeled testset
Seed-Verifier	$> 98\%$	82.7%
Seed-Thinking-Verifier	$> 99\%$	99.3%

表1 两种验证器类型的准确性。具体来说，训练集上的准确性是从训练统计中得出的。此外，我们手动标注了456个样本以形成测试集，这些样本特别选自Seed-Verifier无法稳定处理的案例。

在训练阶段，我们对可验证和不可验证的问题采用不同的奖励建模方法。{v*}

3.1 可验证问题的奖励建模

通过正确的原则和思维轨迹，我们利用大语言模型（LLMs）来判断各种场景中的一系列可验证问题。这种方法产生了一个更通用的解决方案，超越了基于规则的奖励系统的局限性。

我们设计了两种渐进式奖励建模解决方案，Seed-Verifier 和 Seed-Thinking-Verifier:

- Seed-Verifier 是基于一套由人类精心设计的原则。它利用 LLMs 的强大基础能力来评估由问题、参考答案和模型生成的答案组成的三元组。如果参考答案和模型生成的答案在本质上是等价的，它会返回“YES”；否则，返回“NO”。这里的等价性并不是字面上的完全匹配，而是基于计算规则和数学原理的更深层次评估，以证明两个答案传达了相同的数学意义。这种方法确保奖励信号能够准确反映模型的回答在本质上是否正确，即使措辞不同。

- Seed-Thinking-Verifier 受人类判断过程的启发，该过程通过细致的思考和深入的分析生成结论性的判断。为了实现这一点，我们训练了一个验证器，该验证器为其评估提供了详细的推理路径。具体来说，我们将此视为一个可验证的任务，并与其他数学推理任务一起优化。这个验证器可以剖析参考答案和模型生成答案之间的相似性和差异性，提供精确且细致的判断结果。

Seed-Thinking-Ver 显著缓解了与之相关的三个主要问题 使用 Seed-Verifier:

- 奖励黑客：非思考模型可能会利用漏洞在没有真正理解问题的情况下获得奖励。Seed-Thinking-Verifier 中的详细推理过程使得这种黑客行为更加困难。

- 预测中的不确定性：在参考答案和模型生成的答案本质上等价但格式可能不同的情况下，例如 2^{19} 与 524288，Seed-Verifier 有时可能会返回“YES”，有时则返回“NO”。Seed-Thinking-Verifier 通过彻底分析答案背后的推理过程，提供一致的结果。

- 失败于边缘情况：存在某些边缘情况，Seed-Verifier 难以有效处理。Seed-Thinking-Verifier 提供详细推理的能力使其能够更好地应对这些复杂场景。

表1展示了上述两种验证器的性能。更多案例研究的细节可以在附录A中找到。结果表明，Seed-Verifier在处理某些特定情况时存在困难，而Seed-Thinking-Verifier则表现出显著的提供准确判断的能力。尽管后者的思考过程确实消耗了大量GPU资源，但我们认为它生成的精确且稳健的奖励结果对于赋予策略强大的推理能力至关重要。

3.2 Reward Modeling for Non-verifiable Problems

For non-verifiable problems, we train a reward model for RL training. The reward model training data is consistent with the human preference data utilized in Doubao 1.5 Pro [7], primarily encompassing categories such as creative writing and summarization.

To enhance the effectiveness of reward model, we adopt the pairwise generative reward model mentioned in [9], which evaluates the superiority of two responses and use the probability of "YES" or "NO" as the final reward score. This approach enables the model to directly compare differences between responses during scoring, thereby avoiding excessive focus on irrelevant details. Experimental results demonstrate that this reward modeling method improves the stability of RL training, particularly in the mixed training scenarios involving both non-verifiable and verifiable problems, by minimizing conflicts between the two different types of reward modeling paradigms. This improvement may be attributed to the pairwise generative reward model's inherent advantage in mitigating outlier score generation compared to conventional reward models, therefore avoiding significant discrepancies in score distributions with the verifier.

4 Approach

4.1 Supervised Fine-Tuning

Our training process starts with supervised fine-tuning (SFT). The SFT phase sets a solid foundation for the subsequent reinforcement learning stage. Compared to initiating RL from a base model, the SFT model produces more readable outputs, exhibits fewer instances of hallucination, and demonstrates reduced harmfulness. We curate an SFT data comprising 400k training instance, including 300k verifiable problems and 100k non-verifiable problems. Verifiable prompts are randomly sampled from RL training set. Non-verifiable data are sourced from the SFT data used for Doubao-Pro 1.5 [7], covering areas such as creative writing, knowledge-based QA, safety, and function calling.

To generate high-quality responses with long CoT, we employ an iterative workflow that integrates model synthesis, human annotation, and rejection sampling. Initially, human experts apply prompt engineering techniques or engage in interactive dialogues with an internal model to produce responses with various reasoning patterns. After accumulating tens of high-quality cold-start samples, we can train a reasoning model with long CoT as a more capable assistant. Then we perform rejection sampling on this reasoning model using Seed-Verifier. While this workflow is primarily applied to mathematical data, we observe it can generalize well to other domains, such as coding, logic puzzle and even creative writing. Thus, for other domains, we also conduct a cold start process followed by rejection sampling to produce detailed reasoning trajectories.

During training, each instance is truncated to 32,000 tokens. We fine-tune the base model for two epochs using the above data. We use a cosine decay learning rate scheduling that the peak lr is 2×10^{-5} and decays to 2×10^{-6} gradually.

4.2 Reinforcement Learning

We have developed a unified reinforcement learning framework that seamlessly fuses data from a broad range of domains. This integration incorporates three data categories:

- Verifiable data, which obtains feedback from a verifier. This type of data allows for direct validation of the model's outputs against known criteria.
- General data, scored by a reward model. The reward model assigns scores based on how well the model's responses align with human preferences.
- A specific class of data that combines scores from both the verifier and the reward model. This hybrid data type leverages the strengths of both verification and reward-based evaluation.

In the context of long-CoT RLHF, we encounter several challenges such as value model bias and the sparsity of reward signals. To address these issues, we draw on key techniques from our prior work [5, 6, 10]:

3.2 非可验证问题的奖励建模

对于不可验证的问题，我们训练一个奖励模型用于RL训练。奖励模型的训练数据与Doubao 1.5 Pro [7]中使用的人类偏好数据一致，主要涵盖创意写作和总结等类别。

为了提高奖励模型的有效性，我们采用了[9]中提到的成对生成奖励模型，该模型评估两个响应之间的优劣，并使用“YES”或“NO”的概率作为最终的奖励分数。这种方法使模型在评分时能够直接比较响应之间的差异，从而避免过度关注无关细节。实验结果表明，这种奖励建模方法通过最小化两种不同奖励建模范式之间的冲突，提高了RL训练的稳定性，特别是在涉及非可验证和可验证问题的混合训练场景中。这种改进可能归因于成对生成奖励模型在减轻异常分数生成方面相对于传统奖励模型的固有优势，从而避免了与验证器的评分分布出现显著差异。

4 方法

4.1 监督微调

我们的训练过程从监督微调（SFT）开始。SFT 阶段为后续的强化学习阶段奠定了坚实的基础。与从基础模型开始强化学习相比，SFT 模型产生的输出更易读，出现的幻觉现象更少，并且表现出的有害性更低。我们策划了一个包含 400k 训练实例的 SFT 数据集，其中包括 300k 可验证问题和 100k 不可验证问题。可验证的提示是从 RL 训练集中随机抽取的。不可验证的数据来源于用于 Doubao-Pro 1.5 [7] 的 SFT 数据，涵盖了创意写作、知识问答、安全和函数调用等领域。

为了生成高质量的长链路思维（CoT）响应，我们采用了一种迭代工作流程，该流程集成了模型合成、人工标注和拒绝采样。最初，人类专家应用提示工程技术或与内部模型进行交互对话，以生成具有各种推理模式的响应。在积累了数十个高质量的冷启动样本后，我们可以训练一个具有长链路思维的推理模型，作为更强大的助手。然后，我们使用Seed-Verifier对这个推理模型进行拒绝采样。虽然这个工作流程主要应用于数学数据，但我们观察到它可以很好地推广到其他领域，如编程、逻辑谜题，甚至创意写作。因此，对于其他领域，我们也进行冷启动过程，随后进行拒绝采样，以生成详细的推理轨迹。

在训练期间，每个实例被截断为32,000个标记。我们使用上述数据对基础模型进行两个周期的微调。我们采用余弦衰减学习率调度，其中峰值学习率为2×10^{-5}，并逐渐衰减到2×10^{-6}。

4.2 强化学习

我们已经开发了一个统一的强化学习框架，可以无缝融合来自广泛领域的数据。这种集成结合了三类数据：

- 可验证数据，从验证者处获得反馈。这种类型的数据允许直接根据已知标准验证模型的输出。
- 通用数据，由奖励模型评分。奖励模型根据模型的响应与人类偏好的一致性程度分配分数。
- 一种特定的数据类型，结合了验证者和奖励模型的分数。这种混合数据类型利用了验证和基于奖励的评估的各自优势。

在长期CoT RLHF的背景下，我们遇到了几个挑战，如价值模型偏差和奖励信号的稀疏性。为了解决这些问题，我们借鉴了先前工作[5, 6, 10]中的关键技术：

- **Value-Pretraining**: We sample responses from a fixed policy, such as π_{sft}, and update the value model using the Monte-Carlo return. This process ensures that the initialized value model is fully aligned with our policy π_{sft}. Maintaining this alignment has been proven to be crucial for preserving the model's CoT pattern, enabling the model to generate coherent and logical CoT.
- **Decoupled-GAE**: By employing different Generalized Advantage Estimation (GAE) parameters, such as $\lambda_{\text{value}} = 1.0$ and $\lambda_{\text{policy}} = 0.95$, we allow the value model to update in an unbiased manner. Meanwhile, the policy can independently balance its own bias and variance. This decoupling enables more efficient and stable training of the model.
- **Length-adaptive GAE**: We set $\lambda_{\text{policy}} = 1 - \frac{1}{\alpha l}$, where α is a hyper-parameter and l is the response length. This approach ensures a more uniform distribution of Temporal Difference (TD) errors across both short and long sequences. As a result, the model can handle sequences of varying lengths more effectively during training.
- **Dynamic Sampling**: We employ dynamic sampling and filter out prompts with accuracy scores equal to 1 or 0, retaining only those in the batch that exhibit effective gradients. This process helps prevent the dampening of gradient signals during model training.
- **Clip-Higher**: In the Proximal Policy Optimization (PPO) algorithm, we decouple the upper and lower clip bounds as follows:

$$\mathcal{L}^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}})\hat{A}_t \right) \right] \tag{1}$$

  By increasing the value of ϵ_{high}, we create more room for the increase of low-probability tokens. This encourages the model to explore a wider range of possible responses, enhancing its ability to discover novel and effective solutions.
- **Token-level Loss**: Instead of defining the policy loss over entire responses, we define it over all tokens. This approach addresses the imbalance in the token-level contribution to the final loss, ensuring that each token's impact on the training process is appropriately accounted for.
- **Positive Example LM Loss**: This loss function is designed to boost the utilization efficiency of positive samples during the RL training process. We add a language model loss with a coefficient μ for positive examples:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{PPO}}(\theta) + \mu * \mathcal{L}_{\text{NLL}}(\theta) \tag{2}$$

  This additional loss term helps the model to better learn from positive examples, improving its overall performance.

When merging data from different domains and incorporating diverse scoring mechanisms, we face the challenge of interference between different data domains. This interference can arise from disparities in difficulty levels, the risk of reward-hacking, and other underlying factors. These issues make it extremely difficult to achieve uniform and simultaneous improvements across all capabilities of the model. To counteract this, we introduce **Online Data Distribution Adaptation**. This method transforms the stationary prompt distribution during reinforcement learning into an adaptive distribution that better caters to the model's requirements during training. By doing so, we minimize the negative impact of data interference and ensure a more balanced improvement across different abilities. As a result, the model can enhance its performance more consistently across a wide array of tasks.

5 Infrastructures

5.1 Framework

The training framework is built using HybridFlow [11] programming abstraction. The whole training workload runs on top of a Ray [12] cluster. The dataloader and RL algorithm is implemented in a single process Ray Actor (single controller). The model training and response generation (rollout) is implemented in a Ray

- 值预训练：我们从固定的策略（如 π_{sft}）中采样响应，并使用蒙特卡洛回报更新值模型。此过程确保初始化的值模型完全与我们的策略 π_{sft} 对齐。保持这种对齐已被证明对于保留模型的 CoT 模式至关重要，使模型能够生成连贯且逻辑的 CoT。

- Decoupled-GAE: 通过使用不同的广义优势估计（GAE）参数，如 $\lambda_{\text{value}} = 1.0$ 和 $\lambda_{\text{policy}} = 0.95$，我们允许价值模型以无偏的方式更新。同时，策略可以独立地平衡其自身的偏差和方差。这种解耦使得模型的训练更加高效和稳定。

- 长度自适应GAE：我们设置 $\lambda_{\text{policy}} = 1 - \frac{1}{\alpha l}$，其中 α 是一个超参数，l 是响应长度。这种方法确保了在短序列和长序列中时间差分（TD）误差的更均匀分布。因此，模型在训练过程中可以更有效地处理不同长度的序列。

- 动态采样：我们采用动态采样并过滤掉准确率为1或0的提示，仅保留批次中表现出有效梯度的那些。这个过程有助于防止在模型训练期间梯度信号的减弱。

- Clip-Higher: 在近端策略优化（PPO）算法中，我们如下解耦上下裁剪边界：

$$\mathcal{L}^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}})\hat{A}_t \right) \right] \tag{1}$$

  通过增加 ϵ_{high} 的值，我们为低概率令牌的增加创造了更多空间。这鼓励模型探索更广泛的可能性响应，增强其发现新颖和有效解决方案的能力。

- Token-level 损失：我们不是在整个响应上定义策略损失，而是在所有 token 上定义。这种方法解决了 token 级别对最终损失贡献的不平衡问题，确保每个 token 对训练过程的影响都得到了适当的考虑。

- 正例语言模型损失：此损失函数旨在提高RL训练过程中正样本的利用效率。我们为正例添加了一个语言模型损失，并使用系数μ：

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{PPO}}(\theta) + \mu * \mathcal{L}_{\text{NLL}}(\theta) \tag{2}$$

  这个额外的损失项有助于模型更好地从正例中学习，提高其整体性能。

在合并来自不同领域的数据并结合多样化的评分机制时，我们面临不同数据领域之间的干扰挑战。这种干扰可能源于难度水平的差异、奖励操纵的风险以及其他潜在因素。这些问题使得在整个模型的所有能力上实现均匀和同时的改进变得极其困难。为了解决这一问题，我们引入了在线数据分布适应。该方法在强化学习过程中将静态提示分布转换为更适应模型训练需求的自适应分布。通过这样做，我们最小化了数据干扰的负面影响，并确保在不同能力上的更平衡改进。因此，模型可以在广泛的任务中更一致地提升其性能。

5 基础设施

5.1 框架

训练框架是使用 HybridFlow [11] 编程抽象构建的。整个训练工作负载在 Ray [12] 集群上运行。数据加载器和 RL 算法在一个单进程 Ray Actor（单控制器）中实现。模型训练和响应生成（rollout）在 Ray 中实现。

Worker Group. The Ray Worker Group exposes a set of APIs (e.g., generate_response/train_batch, etc,.), which runs heavy training/generation workload via SPMD (single program, multiple data) inside the Worker Group. The single controller invokes various APIs exposed by the Ray Worker Group to construct the training flow. HybridFlow programming abstraction enables fast prototyping of RL algorithm ideas without bothering with complex distributed systems.

Seed-Thinking-v1.5 is trained through hybrid engine architecture [13], where all the models are co-located. This prevents the idle time of the GPUs when switching between training and generation. During Long-CoT generation, we observe severe straggler phenomenon caused by the large difference of the response length between various prompts. This causes massive GPU idle time during generation. To mitigate the straggler of long-tail response generation, we propose SRS (Streaming Rollout System) - a resource-aware scheduling framework that strategically deploys standalone streaming-compute units to transform system constraints from *memory-bound* to *compute-bound*.

5.2 Streaming Rollout System

The SRS architecture introduces *streaming rollout* to decouple model evolution from runtime execution, enabling dynamic adjustment of on/off-policy sample ratios through parametric α:

- Define the completion ratio ($\alpha \in [0, 1]$) as the proportion of samples generated on-policy using the latest model version
- Allocate the remaining non-complete segment (1- α) to off-policy rollouts from versioned model snapshots, seamlessly integrated through asynchronous continuation of partial generations on the standalone resources.

In addition, we also implement dynamic precision scheduling during environment interaction phases, which deploys FP8 policy networks via post-training quantization with error-compensated range scaling. To address token imbalance in MoE systems, we implement a three-tiered parallel architecture combining TP (tensor parallelism) for layer-wise computation, EP (expert parallelism) with dynamic expert assignment, and SP (sequence parallelism) for context chunking. Our kernel auto-tuner dynamically selects optimal CUDA kernel configurations based on real-time load monitoring.

5.3 Training System

To efficiently train the Seed-Thinking-v1.5 model at scale, we design a hybrid distributed training framework that integrates advanced parallelism strategies, dynamic workload balancing, and memory optimizations. Below we detail the core technical innovations driving the system's efficiency and scalability.

- **Parallelism mechanisms.** We compose TP (tensor parallelism)/EP (expert parallelism)/CP (context parallelism) with Fully Sharded Data Parallelism (FSDP) to train Seed-Thinking-v1.5. Specifically, we applied TP/CP for attention layers, and EP for MoE layers.
- **Sequence length balancing.** The effective sequence length can be imbalanced across DP ranks, leading to imbalanced computation workload and low training efficiency. To address this challenge, we leverage KARP [14] algorithm that rearranges the input sequences within one mini-batch to make them balance among micro-batches.
- **Memory optimization.** We adopt layer-wise recomputation [15], activation offload and optimizer offload to support training of larger micro-batches to overlap the communication overhead caused by FSDP.
- **Auto parallelism.** To enable optimal system performance, we develop an automatic tuning system, referred to as AutoTuner. Specifically, AutoTuner models the memory usage following a profile-based solution [16]. Then, it estimates the performance and memory usage of various configurations to obtain the optimal configuration.
- **Checkpoint.** We employ ByteCheckpoint [17] to support checkpoint resume from different distributed configurations with minimal overhead. This enables users to elastically train the tasks to improve cluster efficiency.

工作器组。Ray 工作器组暴露了一组 API（例如，generate_response/train_batch 等），通过 SPMD（单程序，多数据）在工作器组内部运行重型训练/生成工作负载。单个控制器调用 Ray 工作器组暴露的各种 API 来构建训练流程。HybridFlow 编程抽象使得快速原型化 RL 算法思想成为可能，而无需担心复杂的分布式系统。

Seed-Thinking-v1.5 通过混合引擎架构 [13] 进行训练，所有模型都共位于同一位置。这防止了在训练和生成之间切换时 GPU 的空闲时间。在 Long-CoT 生成过程中，我们观察到由于不同提示的响应长度差异较大而引起的严重拖尾现象。这导致了生成过程中大量的 GPU 空闲时间。为了减轻长尾响应生成的拖尾现象，我们提出了 SRS（流式回放系统）- 一个资源感知的调度框架，该框架战略性地部署独立的流式计算单元，将系统约束从内存限制转变为计算限制。

5.2 流式部署系统

SRS架构引入了流式部署，以解耦模型演进和运行时执行，通过参数化α实现在线/离线策略样本比例的动态调整：

- 将完成率 ($\alpha \in [0, 1]$) 定义为使用最新模型版本生成的样本比例。

- 将剩余的非完整段 (1- α) 分配给来自版本化模型快照的离线策略回放，通过独立资源上的部分生成的异步延续无缝集成。

此外，我们还在环境交互阶段实现了动态精度调度，通过后训练量化和误差补偿范围缩放部署 FP8 策略网络。为了解决 MoE 系统中的 token 不平衡问题，我们实现了一个三层并行架构，结合 TP（张量并行）进行层计算，EP（专家并行）与动态专家分配，以及 SP（序列并行）进行上下文分块。我们的内核自动调优器根据实时负载监控动态选择最优的 CUDA 内核配置。

5.3 训练系统

为了高效地大规模训练Seed-Thinking-v1.5模型，我们设计了一个混合分布式训练框架，该框架集成了先进的并行策略、动态工作负载平衡和内存优化。下面我们将详细介绍推动系统效率和可扩展性的核心技术创新。

- 并行机制。我们组合使用 TP（张量并行）/EP（专家并行）/CP（上下文并行）与 Fully Sharded Data Parallelism (FSDP) 来训练 Seed-Thinking-v1.5。具体来说，我们在注意力层应用了 TP/CP，并在 MoE 层应用了 EP。
- 序列长度平衡。有效序列长度在DP等级之间可能不平衡，导致计算工作负载不平衡和训练效率低下。为了解决这一挑战，我们利用KARP [14]算法在同一个小批量内重新排列输入序列，使它们在微批次之间达到平衡。

- 内存优化。我们采用逐层重计算 [15]、激活卸载和优化器卸载来支持更大微批次的训练，以重叠由 FSDP 引起的通信开销。
- 自动并行。为了实现最佳系统性能，我们开发了一个自动调优系统，称为AutoTuner。具体来说，AutoTuner采用基于配置文件的解决方案{v*}来建模内存使用[16]。然后，它估计不同配置的性能和内存使用情况，以获得最佳配置。

- 检查点。我们使用ByteCheckpoint [17] 来支持从不同的分布式配置中以最小的开销恢复检查点。这使用户能够弹性地训练任务以提高集群效率。

Benchmark	Seed-Thinking-v1.5	DeepSeek R1	OpenAI o3-mini	Grok 3 Beta	Gemini 2.5 pro
Mathematics					
AIME 2025	74.0%	65.0%	86.5%	77.3%	86.7%
AIME 2024	86.7%	79.8%	87.3 %	83.9%	92.0%
Beyond AIME	48.0%	42.4%	63.6 %	-	58.8%
Science					
GPQA diamond	77.3%	71.5%	79.7%	80.2%	84.0%
SuperGPQA	62.1%	60.5%	52.2%	62.8%	65.3%
MMLU-PRO	87.0%	85.6%	82.4%	84.6%	86.3%
Code					
Codeforces avg@8	36.3%	32.0%	50.9%	-	40.3%
Codeforces pass@8	55.0%	45.0%	67.5%	-	56.3%
LiveCodeBench v5	64.9%	64.3%	74.1%	70.6%	70.4%
Aider Polyglot	54.2%	56.9%	68.6%	-	74.0%
Agentic Coding					
SWE-bench verified	47.0%	49.2%	49.3%	-	63.8%
SWE-bench verified*	47.0%	46.2%	44.5%	-	63.8%
Logic reasoning					
ARC-AGI	39.9%	18.3%	25.8%	31.9%	27.6%
Factuality					
SimpleQA	12.9%	30.1%	13.8%	43.6%	52.9%
Instruction					
Collie	73.1%	34.2%	87.6%	33.6%	62.5%
IFEval	87.4%	86.1%	93.7%	83.4%	91.5%

Table 2 Results of State-of-the-Art Reasoning Models

*Results from our internal sandbox, which may differ from the reported results due to inconsistencies in the testing environment.

6 Experiment Results

6.1 Auto Evaluation Results

Table 2 presents the evaluation results across diverse tasks spanning mathematics, coding, science, and general knowledge domains. For mathematical benchmark tasks, results are calculated as the average across 32 model responses, while GPQA task results are averaged over 8 responses. For Codeforces, we report both avg@8 and pass@8, because pass@8 aligns better with human submission habits. Results for all other tasks are averaged over 1 response.

In mathematical reasoning, Seed-Thinking-v1.5 achieves top-tier performance on the AIME 2024 benchmark, scoring 86.7, matching the performance of OpenAI's o3-mini-high model. However, on the more recent AIME 2025 and the advanced BeyondAIME challenges, Seed-Thinking-v1.5 still lags behind o3-level performance. For the GPQA task, Seed-Thinking-v1.5 achieves an 77.3% accuracy rate, close to the performance of o3-mini-high. In code generation scenarios such as Codeforces, Seed-Thinking-v1.5 nearly matches the performance of Gemini 2.5 Pro but still trails behind o3-mini-high. Notably, Seed-Thinking-v1.5 demonstrates less impressive results on SimpleQA. It is worth emphasizing that this benchmark primarily functions as a memory-oriented metric, where performance is more strongly correlated with pre-trained model scale rather than genuine reasoning capabilities.

6.2 Human Evaluation Results

To evaluate model performance on subjective tasks, where automated metrics are insufficient to capture nuanced human preferences, we conduct human evaluations across a diverse suite of non-reasoning scenarios. Our assessments are designed to measure key dimensions of quality, such as coherence, relevance, creativity, and adherence to human-centric preferences, with a panel of domain-expert evaluators rating model outputs against Deepseek R1 under predefined rubrics. We use a 5-point ordinal scale, ranging from 0(very poor) to 4(excellent), and evaluate both models on session prompts with multiple rounds. Each full session is

Benchmark	Seed-Thinking-v1.5	DeepSeek R1	OpenAI o3-mini	Grok 3 Beta	Gemini 2.5 pro
Mathematics					
AIME 2025	74.0%	65.0%	86.5%	77.3%	86.7%
AIME 2024	86.7%	79.8%	87.3 %	83.9%	92.0%
Beyond AIME	48.0%	42.4%	63.6 %	-	58.8%
Science					
GPQA diamond	77.3%	71.5%	79.7%	80.2%	84.0%
SuperGPQA	62.1%	60.5%	52.2%	62.8%	65.3%
MMLU-PRO	87.0%	85.6%	82.4%	84.6%	86.3%
Code					
Codeforces avg@8	36.3%	32.0%	50.9%	-	40.3%
Codeforces pass@8	55.0%	45.0%	67.5%	-	56.3%
LiveCodeBench v5	64.9%	64.3%	74.1%	70.6%	70.4%
Aider Polyglot	54.2%	56.9%	68.6%	-	74.0%
Agentic Coding					
SWE-bench verified	47.0%	49.2%	49.3%	-	63.8%
SWE-bench verified*	47.0%	46.2%	44.5%	-	63.8%
Logic reasoning					
ARC-AGI	39.9%	18.3%	25.8%	31.9%	27.6%
Factuality					
SimpleQA	12.9%	30.1%	13.8%	43.6%	52.9%
Instruction					
Collie	73.1%	34.2%	87.6%	33.6%	62.5%
IFEval	87.4%	86.1%	93.7%	83.4%	91.5%

表2 最先进推理模型的结果 *来自我们内部沙箱的结果，可能由于测试环境的不一致而与报告的结果有所不同。

6 实验结果

6.1 自动评估结果

表2展示了跨数学、编程、科学和一般知识领域的多样化任务的评估结果。对于数学基准任务，结果是基于32个模型响应的平均值计算的，而GPQA任务的结果则是基于8个响应的平均值。对于Codeforces，我们报告了avg@8和pass@8，因为pass@8更符合人类提交的习惯。所有其他任务的结果都是基于1个响应的平均值。

在数学推理方面，Seed-Thinking-v1.5 在 AIME 2024 基准测试中取得了 86.7 的高分，与 OpenAI 的 o3-mini-high 模型的性能相当。然而，在更近期的 AIME 2025 和高级 BeyondAIME 挑战中，Seed-Thinking-v1.5 仍然落后于 o3 级别的性能。对于 GPQA 任务，Seed-Thinking-v1.5 达到了 77.3% 的准确率，接近 o3-mini-high 的性能。在如 Codeforces 的代码生成场景中，Seed-Thinking-v1.5 几乎与 Gemini 2.5 Pro 的性能持平，但仍落后于 o3-mini-high。值得注意的是，Seed-Thinking-v1.5 在 SimpleQA 上的表现不太令人印象深刻。需要强调的是，这个基准测试主要作为一个记忆导向的指标，性能与预训练模型的规模关联更紧密，而不是与真正的推理能力。

6.2 人类评估结果

为了评估模型在主观任务上的表现，其中自动化指标不足以捕捉细微的人类偏好，我们在一系列多样的非推理场景中进行人类评估。我们的评估旨在衡量质量的关键维度，如连贯性、相关性、创造力和以人类为中心的偏好遵循度，由领域专家评估小组根据预定义的评分标准对模型输出进行评分。我们使用一个5点顺序量表，范围从0（非常差）到4（优秀），并在多轮会话提示中评估两个模型。每个完整的会话都是

annotated with a binary win/loss outcome to capture the overall user experience and a single 0-4 score is assigned per-round.

Seed-Thinking-v1.5 achieves an overall win ratio of 8.0% on the evaluated sessions, indicating superiority in aligning with human-centric preferences. Further more, this win rate is consistent across diverse scenarios, from creative writing to humanities knowledge elaboration. Figure 2 shows the per-round level score distribution.

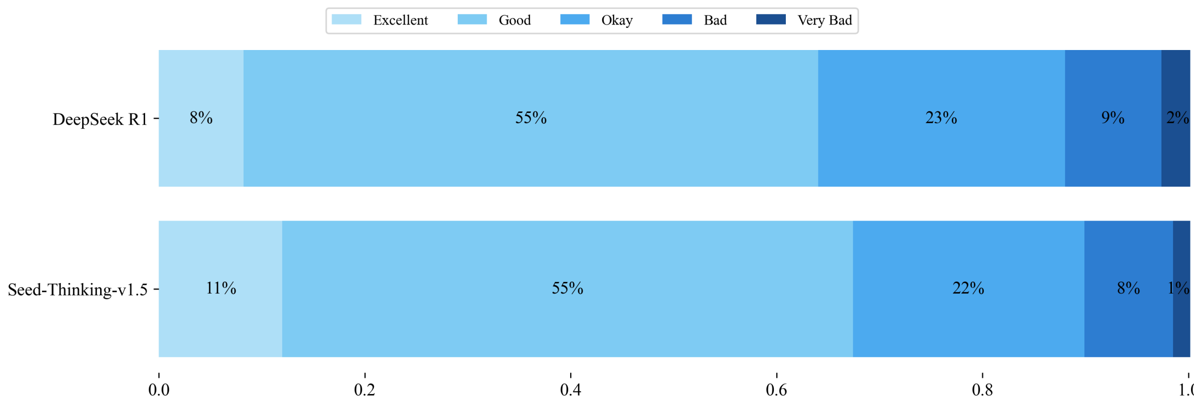

Figure 2 Rating Distribution

6.3 Effects of pre-train models

Rejection Sampling. Rejection sampling has been identified as a valuable technique for improving model performance [2]. We perform an ablation to examine whether initializing RL with a rejection fine-tuning (RFT) model impacts outcomes. Our results show that the pretrained model initialized with RFT saturates more quickly during training but ultimately achieves lower performance than the model trained without RFT, as shown in Table 3.

Consistent algorithm rankings across model size. We observe that RL algorithms demonstrate consistent ranking behaviors across different models of varying sizes and architectures. As illustrated in Table 4, Seed-150B-MoE, a model that differs from Qwen-32B in both architecture (MoE vs. dense) and size, exhibits a consistent ranking. Notably, this consistency suggests that Qwen-32B can effectively serve as a proxy model for investigating RL algorithms.

Models	AIME avg@32
Baseline	58%
w/ RFT	54%

Table 3 Ablations on Pretrained Models

AIME	DAPO	VAPO
Qwen-32B-Dense	50%	60%
Seed-150B-MoE	73%	79%

Table 4 Consistent Algorithm Rankings. Seed-150B-MoE results are ablation-only with limited steps.

7 Related Work

Test-time scaling [4, 18–20] such as OpenAI's o1 [1] and DeepSeek's R1 [2] have catalyzed a profound paradigm shift in LLMs [21, 22]. By enabling extended CoT reasoning [23] and eliciting sophisticated reasoning capabilities, these methods empower LLMs to excel in complex mathematical and coding tasks, including those from competitions like the AIME and Codeforces. At the core of this transformation is large-scale reinforcement learning, which facilitates the emergence of complex reasoning behaviors—such as self-verification and iterative refinement. However, the critical methodologies and algorithms underpinning scalable RL training have largely remained obscure, often omitted from the technical documentation of existing reasoning models [1, 2, 21–23].

标注了二元的胜负结果以捕捉整体用户体验，并且每轮分配一个0-4的评分。

Seed-Thinking-v1.5 在评估的会话中实现了 8.0% 的总体胜率，表明其在与以人为中心的偏好对齐方面具有优越性。此外，这一胜率在从创意写作到人文知识阐述的各种场景中保持一致。图 2 显示了每轮的得分分布。

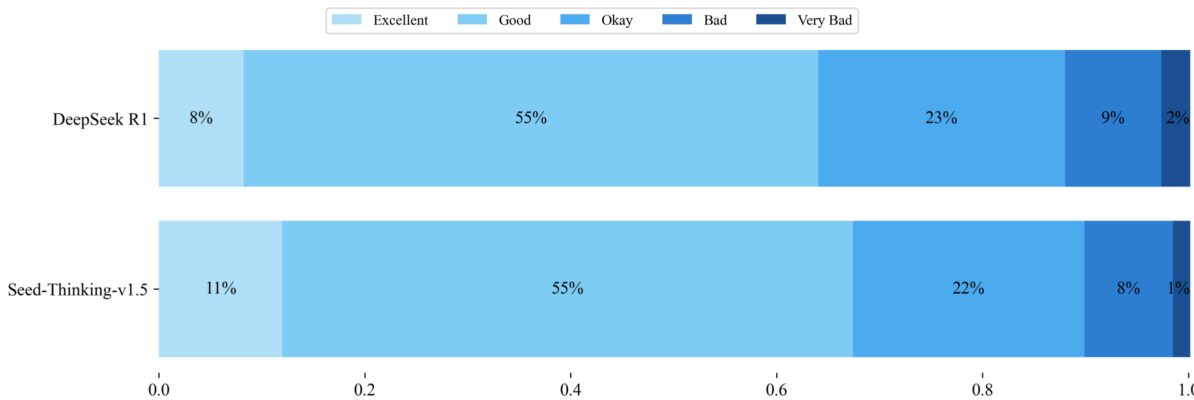

图2 评分分布

6.3 预训练模型的影响

拒绝采样。拒绝采样已被确认为提高模型性能的一种有价值的技术[2]。我们进行了一项消融研究，以检查使用拒绝微调（RFT）模型初始化RL是否会影响结果。我们的结果显示，使用RFT初始化的预训练模型在训练过程中更快地达到饱和，但最终性能低于未使用RFT训练的模型，如表3所示。

不同模型大小下算法排名的一致性。我们观察到，RL算法在不同大小和架构的模型中表现出一致的排名行为。如表4所示，Seed-150B-MoE，一个在架构（MoE与密集型）和大小上都与Qwen-32B不同的模型，显示出了一致的排名。值得注意的是，这种一致性表明Qwen-32B可以有效地作为研究RL算法的代理模型。

Models	AIME avg@32
Baseline	58%
w/ RFT	54%

表3 预训练模型的消融研究

AIME	DAPO	VAPO
Qwen-32B-Dense	50%	60%
Seed-150B-MoE	73%	79%

表4 一致的算法排名。Seed-150B- MoE 结果仅限于消融研究，步数有限。

7 相关工作

测试时扩展 [4, 18–20]，如 OpenAI 的 o1 [1] 和 DeepSeek 的 R1 [2]，已经在 LLMs [21, 22] 中引发了深刻范式转变。通过启用扩展的 CoT 推理 [23] 和激发复杂的推理能力，这些方法使 LLMs 能够在复杂的数学和编程任务中表现出色，包括来自 AIME 和 Codeforces 等比赛的任务。这一转变的核心是大规模强化学习，它促进了复杂推理行为的出现——如自我验证和迭代改进。然而，支持可扩展 RL 训练的关键方法和算法在很大程度上仍然不为人所知，通常被现有推理模型的技术文档 [1, 2, 21–23] 所省略。

In this paper, we introduce an SOTA-level model Seed-Thinking-v1.5 and introduce the details to achieve the performance from three aspects: Data, RL algorithm, and RL infrastructure.

8 Conclusion

We introduce a superb reasoning model named Seed-Thinking-v1.5, which achieves excellent performance across both reasoning tasks and non-reasoning tasks. It utilizes advanced RL techniques to improve the thinking ability stably and reliably by attaining 86.7% on AIME24, 74.0% on AIME25 and 55.0% on Codeforces. In the future, we plan to investigate more efficient RL recipes and explore more challenging tasks with thinking mode to push the boundary of model's intelligence. Moreover, general reward modeling with comparable accuracy as verifier would also be a compelling research direction.

在这篇论文中，我们介绍了一个SOTA级别的模型Seed-Thinking-v1.5，并从三个方面介绍了实现其性能的细节：数据、RL算法和RL基础设施。

8 结论

我们介绍了一种名为Seed-Thinking-v1.5的出色推理模型，该模型在推理任务和非推理任务中都表现出色。它利用先进的RL技术，通过在AIME24上达到86.7%，AIME25上达到74.0%以及Codeforces上达到55.0%，从而稳定可靠地提高思考能力。未来，我们计划研究更高效的RL方案，并探索更多具有挑战性的任务，以思考模式推动模型智能的边界。此外，与验证者具有可比准确性的通用奖励建模也将是一个有吸引力的研究方向。

9 Contributions and Acknowledgments

The names are sorted in alphabetical order of the last name. An asterisk (*) indicates members who have departed from the team.

9.1 Core Contributors

Jiaze Chen, TianTian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, Chengyi Wang, Xiangpeng Wei, Wenyuan Xu,Yufeng Yuan, Yu Yue, Lin Yan, Qiying Yu, Xiaochen Zuo, Chi Zhang

9.2 Contributors

Zhecheng An, Zhihao Bai, Yu Bao, Xingyan Bin, Jiangjie Chen, Feng Chen, Hongmin Chen, Riwei Chen, Liangqiang Chen, Zixin Chen, Jinsong Chen, Siyan Chen, Kaiyuan Chen, Zhi Chen, Jin Chen, Jiecao Chen, Jinxin Chi, Wennan Dai, Ning Dai, Jiahui Dai, Shihan Dou, Yantao Du, Zhengyin Du, Jianhui Duan, Chen Dun, Ting-Han Fan, Jiazhan Feng, Junda Feng, Ziyuan Feng, Yuwei Fu, Wenqi Fu, Hanjie Fu*, Hao Ge, Hongyi Guo, Mingji Han, Li Han, Wenhao Hao, Xintong Hao, Qianyu He, Jerry He, Feng He, Wen Heng, Zehua Hong, Qi Hou, Liang Hu, Shengding Hu*, Nan Hu*, Kai Hua, Qi Huang, Ziyue Huang, Hongzhi Huang, Zihao Huang, Ting Huang, Wenhao Huang, Wei Jia, Bin Jia, Xiaoying Jia, Yuhua Jiang, Haobin Jiang, Ziheng Jiang, Kaihua Jiang, Chengquan Jiang, Jianpeng Jiao, Xiaoran Jin, Xing Jin, Xunhao Lai, Zheng Li, Xiang Li, Liyi Li, Hongkai Li, Zheng Li, Ji Li, Yunshui Li, Chenggang Li, Niuniu Li, Siyu Li, Xi Li, Xiao Li, Aoyan Li, Niuniu Li, Yuntao Li, Nianning Liang, Xinnian Liang, Haibin Lin, Weijian Lin, Ye Lin*, Zhicheng Liu, Guanlin Liu, Guanlin Liu, Chenxiao Liu, Yan Liu, Gaohong Liu, Juncai Liu, Chundian Liu, Deyi Liu, Kaibo Liu, Siyao Liu, Qi Liu, Yongfei Liu, Kang Liu, Gan Liu*, Boyi Liu*, Rui Long, Weiqiang Lou, Chenwei Lou, Xiang Luo, Yao Luo, Caiping Lv, Heyang Lv, Yunming Ma, Bole Ma, Qianli Ma, Hongzhi Ma, Yiyuan Ma, Jin Ma, Wenchang Ma, Tingting Ma, Chen Mao, Qiyang Min, Zhe Nan, Guanghan Ning*, Jinxiang Ou, Haojie Pan, Renming Pang, Yanghua Peng, Tao Peng, Lihua Qian, Lihua Qian, Mu Qiao*, Meng Qu, Cheng Ren, Hongbin Ren, Yong Shan, Wei Shen, Ke Shen, Kai Shen, Jinlong Shi, Wenlei Shi, Guang Shi, Shuai Shuai Cao, Yuxin Song, Zuquan Song, Jing Su, Yifan Sun, Tao Sun, Zewei Sun, Borui Wan,Zihan Wang, Xiaohui Wang, Xi Wang, Shuguang Wang, Jun Wang, Qinlong Wang, Chenyuan Wang, Shuai Wang, Zihan Wang, Changbao Wang, Jiaqiang Wang, Shihang Wang, Xuwu Wang, Zaiyuan Wang, Yuxuan Wang, Wenqi Wang, Taiqing Wang*, Chengzhi Wei, Houmin Wei, Ziyun Wei, Shufa Wei, Zheng Wu, Yonghui Wu, Yangjun Wu, Bohong Wu, Shuang Wu, Jingqiao Wu, Ning Wu, Shuangzhi Wu, Jianmin Wu*, Chenguang Xi*, Fan Xia, Yuqiao Xian, Liang Xiang, Boren Xiang, Bowen Xiao, Zhen Xiao, Xia Xiao, Yongsheng Xiao, Chao Xin, Shulin Xin, Yuwen Xiong, Jingjing Xu, Ziwen Xu, Chenyin Xu, Jiayi Xu, Yifan Xu, Wei Xu, Yufei Xu, Shikun Xu*, Shipeng Yan, Shen Yan, Qingping Yang, Xi Yang, Tianhao Yang, Yuehang Yang, Yuan Yang, Ximing Yang, Zeyu Yang, Guang Yang, Yifan Yang*, Xuesong Yao, Bairen Yi, Fan Yin, Jianian Yin, Ziqiang Ying, Xiangyu Yu, Hongli Yu, Song Yu, Menghan Yu, Huan Yu, Siyu Yuan, Jun Yuan, Yutao Zeng, Tianyang Zhan, Zheng Zhang, Yun Zhang, Mofan Zhang, Wang Zhang, Ru Zhang, Zhi Zhang, Tianqi Zhang, Xinyi Zhang, Zhexi Zhang, Sijun Zhang, Wenqiang Zhang, Xiangxiang Zhang, Yongtao Zhang, Yuyu Zhang, Ge Zhang, He Zhang, Yue Zhang*, Renjie Zheng, Ningxin Zheng, Zhuolin Zheng, Yaowei Zheng, Chen Zheng, Xiaoyun Zhi, Wanjun Zhong, Cheng Zhong, Zheng Zhong, Baoquan Zhong, Xun Zhou, Na Zhou, Huan Zhou,Ruofei Zhu, Hang Zhu, Defa Zhu, Wenjia Zhu, Lei Zuo

9 贡献和致谢

名字按照姓氏的字母顺序排列。星号 (*) 表示已离开团队的成员。

9.1 核心贡献者

Jiaze Chen, TianTian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, Chengyi Wang, Xiangpeng Wei, Wenyuan Xu, Yufeng Yuan, Yu Yue, Lin Yan, Qiying Yu, Xiaochen Zuo, Chi Zhang

9.2 贡献者

Zhecheng An, Zhihao Bai, Yu Bao, Xingyan Bin, Jiangjie Chen, Feng Chen, Hongmin Chen, Riwei Chen, Liangqiang Chen, Zixin Chen, Jinsong Chen, Siyan Chen, Kaiyuan Chen, Zhi Chen, Jin Chen, Jiecao Chen, Jinxin Chi, Wennan Dai, Ning Dai, Jiahui Dai, Shihan Dou, Yantao Du, Zhengyin Du, Jianhui Duan, Chen Dun, Ting-Han Fan, Jiazhan Feng, Junda Feng, Ziyuan Feng, Yuwei Fu, Wenqi Fu, Hanjie Fu*, Hao Ge, Hongyi Guo, Mingji Han, Li Han, Wenhao Hao, Xintong Hao, Qianyu He, Jerry He, Feng He, Wen Heng, Zehua Hong, Qi Hou, Liang Hu, Shengding Hu*, Nan Hu*, Kai Hua, Qi Huang, Ziyue Huang, Hongzhi Huang, Zihao Huang, Ting Huang, Wenhao Huang, Wei Jia, Bin Jia, Xiaoying Jia, Yuhua Jiang, Haobin Jiang, Ziheng Jiang, Kaihua Jiang, Chengquan Jiang, Jianpeng Jiao, Xiaoran Jin, Xing Jin, Xunhao Lai, Zheng Li, Xiang Li, Liyi Li, Hongkai Li, Zheng Li, Ji Li, Yunshui Li, Chenggang Li, Niuniu Li, Siyu Li, Xi Li, Xiao Li, Aoyan Li, Niuniu Li, Yuntao Li, Nianning Liang, Xinnian Liang, Haibin Lin, Weijian Lin, Ye Lin*, Zhicheng Liu, Guanlin Liu, Guanlin Liu, Chenxiao Liu, Yan Liu, Gaohong Liu, Juncai Liu, Chundian Liu, Deyi Liu, Kaibo Liu, Siyao Liu, Qi Liu, Yongfei Liu, Kang Liu, Gan Liu*, Boyi Liu*, Rui Long, Weiqiang Lou, Chenwei Lou, Xiang Luo, Yao Luo, Caiping Lv, Heyang Lv, Yunming Ma, Bole Ma, Qianli Ma, Hongzhi Ma, Yiyuan Ma, Jin Ma, Wenchang Ma, Tingting Ma, Chen Mao, Qiyang Min, Zhe Nan, Guanghan Ning*, Jinxiang Ou, Haojie Pan, Renming Pang, Yanghua Peng, Tao Peng, Lihua Qian, Lihua Qian, Mu Qiao*, Meng Qu, Cheng Ren, Hongbin Ren, Yong Shan, Wei Shen, Ke Shen, Kai Shen, Jinlong Shi, Wenlei Shi, Guang Shi, Shuai Shuai Cao, Yuxin Song, Zuquan Song, Jing Su, Yifan Sun, Tao Sun, Zewei Sun, Borui Wan, Zihan Wang, Xiaohui Wang, Xi Wang, Shuguang Wang, Jun Wang, Qinlong Wang, Chenyuan Wang, Shuai Wang, Zihan Wang, Changbao Wang, Jiaqiang Wang, Shihang Wang, Xuwu Wang, Zaiyuan Wang, Yuxuan Wang, Wenqi Wang, Taiqing Wang*, Chengzhi Wei, Houmin Wei, Ziyun Wei, Shufa Wei, Zheng Wu, Yonghui Wu, Yangjun Wu, Bohong Wu, Shuang Wu, Jingqiao Wu, Ning Wu, Shuangzhi Wu, Jianmin Wu*, Chenguang Xi*, Fan Xia, Yuqiao Xian, Liang Xiang, Boren Xiang, Bowen Xiao, Zhen Xiao, Xia Xiao, Yongsheng Xiao, Chao Xin, Shulin Xin, Yuwen Xiong, Jingjing Xu, Ziwen Xu, Chenyin Xu, Jiayi Xu, Yifan Xu, Wei Xu, Yufei Xu, Shikun Xu*, Shipeng Yan, Shen Yan, Qingping Yang, Xi Yang, Tianhao Yang, Yuehang Yang, Yuan Yang, Ximing Yang, Zeyu Yang, Guang Yang, Yifan Yang*, Xuesong Yao, Bairen Yi, Fan Yin, Jianian Yin, Ziqiang Ying, Xiangyu Yu, Hongli Yu, Song Yu, Menghan Yu, Huan Yu, Siyu Yuan, Jun Yuan, Yutao Zeng, Tianyang Zhan, Zheng Zhang, Yun Zhang, Mofan Zhang, Wang Zhang, Ru Zhang, Zhi Zhang, Tianqi Zhang, Xinyi Zhang, Zhexi Zhang, Sijun Zhang, Wenqiang Zhang, Xiangxiang Zhang, Yongtao Zhang, Yuyu Zhang, Ge Zhang, He Zhang, Yue Zhang*, Renjie Zheng, Ningxin Zheng, Zhuolin Zheng, Yaowei Zheng, Chen Zheng, Xiaoyun Zhi, Wanjun Zhong, Cheng Zhong, Zheng Zhong, Baoquan Zhong, Xun Zhou, Na Zhou, Huan Zhou, Ruofei Zhu, Hang Zhu, Defa Zhu, Wenjia Zhu, Lei Zuo

References

[1] OpenAI. Learning to reason with llms, 2024.

[2] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[3] Google DeepMind. Gemini 2.5: Our most intelligent ai model, 2025.

[4] Anthropic. Claude 3.7 sonnet and claude code, 2025.

[5] Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks, 2025.

[6] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025.

[7] ByteDance. Doubao-1.5-pro, 2025.

[8] Wei Shen, Guanlin Liu, Zheng Wu, Ruofei Zhu, Qingping Yang, Chao Xin, Yu Yue, and Lin Yan. Exploring data scaling trends and effects in reinforcement learning from human feedback, 2025.

[9] Wenyuan Xu, Xiaochen Zuo, Chao Xin, Yu Yue, Lin Yan, and Yonghui Wu. A unified pairwise framework for rlhf: Bridging generative reward modeling and policy optimization, 2025.

[10] Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in long-cot? value optimization holds the secret. arXiv preprint arXiv:2503.01491, 2025.

[11] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. arXiv preprint arXiv:2409.19256, 2024.

[12] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, William Paul, Michael I. Jordan, and Ion Stoica. Ray: A distributed framework for emerging AI applications. CoRR, abs/1712.05889, 2017.

[13] Zhewei Yao, Reza Yazdani Aminabadi, Olatunji Ruwase, Samyam Rajbhandari, Xiaoxia Wu, Ammar Ahmad Awan, Jeff Rasley, Minjia Zhang, Conglong Li, Connor Holmes, Zhongzhu Zhou, Michael Wyatt, Molly Smith, Lev Kurilenko, Heyang Qin, Masahiro Tanaka, Shuai Che, Shuaiwen Leon Song, and Yuxiong He. Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales, 2023.

[14] Narendra Karmarkar and Richard M Karp. The differencing method of set partitioning. Computer Science Division (EECS), University of California Berkeley, 1982.

[15] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. arXiv preprint arXiv:1604.06174, 2016.

[16] Lianmin Zheng, Zhuohan Li, Hao Zhang, Yonghao Zhuang, Zhifeng Chen, Yanping Huang, Yida Wang, Yuanzhong Xu, Danyang Zhuo, Eric P Xing, et al. Alpa: Automating inter-and {Intra-Operator} parallelism for distributed deep learning. In 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22), pages 559–578, 2022.

[17] Borui Wan, Mingji Han, Yiyao Sheng, Yanghua Peng, Haibin Lin, Mofan Zhang, Zhichao Lai, Menghan Yu, Junda Zhang, Zuquan Song, Xin Liu, and Chuan Wu. Bytecheckpoint: A unified checkpointing system for large foundation model development, 2025.

[18] Qwen. Qwq-32b: Embracing the power of reinforcement learning, 2024.

[19] XAI. Grok 3 beta — the age of reasoning agents, 2024.

[20] Google DeepMind. Gemini 2.0 flash thinking, 2024.

参考文献

[1] OpenAI. 使用LLMs学习推理，2024. [2] DeepSeek-AI. Deepseek-r1: 通过强化学习激励LLMs的推理能力，2025. [3] Google DeepMind. Gemini 2.5: 我们最智能的AI模型，2025. [4] Anthropic. Claude 3.7十四行诗和Claude代码，2025. [5] 岳宇，袁玉峰，于启颖，左晓晨，朱若飞，徐文渊，陈嘉泽，王成一，范天天，杜正银，魏向鹏，余翔宇，刘高宏，刘俊才，刘凌军，刘海斌，林志奇，马博，张驰，张墨帆，张王，朱航，张如，刘欣，王明轩，吴永辉，严琳. Vapo: 高级推理任务的高效可靠强化学习，2025. [6] 于启颖，张正，朱若飞，袁玉峰，左晓晨，岳宇，范天天，刘高宏，刘凌军，刘欣，林海斌，林志奇，马博，盛广明，童宇轩，张驰，张墨帆，张王，朱航，朱金华，陈嘉泽，陈江杰，王成一，于红丽，戴维南，宋宇轩，魏向鹏，周浩，刘静晶，马卫英，张亚勤，严琳，乔沐，吴永辉，王明轩. Dapo: 大规模开源LLM强化学习系统，2025. [7] ByteDance. Doubao-1.5-pro，2025. [8] 沈伟，刘冠麟，吴正，朱若飞，杨庆平，辛超，岳宇，严琳. 探索人类反馈强化学习中的数据扩展趋势和效果，2025. [9] 徐文渊，左晓晨，辛超，岳宇，严琳，吴永辉. 统一的成对框架用于RLHF：连接生成式奖励建模和策略优化，2025. [10] 袁玉峰，岳宇，朱若飞，范天天，严琳. PPO在长链推理中的崩溃背后是什么？价值优化揭示了秘密. arXiv预印本arXiv:2503.01491, 2025. [11] 盛广明，张驰，叶子凌峰，吴西斌，张王，张如，彭阳华，林海斌，吴传. Hybridflow: 一个灵活高效的RLHF框架. arXiv预印本arXiv:2409.19256, 2024. [12] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, William Paul, Michael I. Jordan, 和 Ion Stoica. Ray: 一个新兴AI应用的分布式框架. CoRR, abs/1712.05889, 2017. [13] 姚哲伟，Yazdani Aminabadi Reza，Ruwase Olatunji，Rajbhandari Samyam，吴晓霞，Awan Ammar Ahmad，Rasley Jeff，张敏嘉，李聪龙，Holmes Connor，周中柱，Wyatt Michael，Smith Molly，Kurilenko Lev，秦赫阳，Tanaka Masahiro，Che Shuai，Song Shuaiwen Leon，He Yuxiong. Deepspeed-chat: 轻松、快速且经济实惠的ChatGPT类模型RLHF训练，2023.

[14] Narendra Karmarkar 和 Richard M Karp. 集合划分的差分方法. 加州大学伯克利分校电子工程与计算机科学系, 1982.

[15] 天琪陈, 冰徐, 雅元张, 和 卡洛斯·格斯特林. 用次线性内存成本训练深度网络. arXiv预印本 arXiv:1604.06174, 2016.

[16] Lianmin Zheng, Zhuohan Li, Hao Zhang, Yonghao Zhuang, Zhifeng Chen, Yanping Huang, Yida Wang, Yuanzhong Xu, Danyang Zhuo, Eric P Xing, 等. Alpa: 自动化分布式深度学习的{算子间和}算子内并行性. 在第16届USENIX操作系统设计与实现研讨会 (OSDI 22)，页面 559–578，2022.

[17] 万博睿，韩明基，盛一耀，彭阳华，林海斌，张墨凡，赖志超，余梦涵，张俊达，宋祖全，刘欣，武川。Bytecheckpoint: 用于大型基础模型开发的统一检查点系统，2025。

[18] Qwen. Qwq-32b: 拥抱强化学习的力量，2024.

[19] XAI. Grok 3 beta — {v*}推理代理的时代，2024. [20] Google DeepMind. Gemini 2.0 闪思，2024.

[21] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.

[22] OpenAI. GPT4 technical report. arXiv preprint arXiv:2303.08774, 2023.

[23] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35:24824–24837, 2022.

[21] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, 等. 语言模型是少样本学习者. Advances in neural information processing systems, 33:1877−1901, 2020. [22] OpenAI. GPT4 技术报告. arXiv preprint arXiv:2303.08774, 2023. [23] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, 等. 链式思维提示在大型语言模型中激发推理. Advances in neural information processing systems, 35:24824−24837, 2022.

Appendix

A Case Study on Verifier

Table 5 presents case study for both the Seed-Verifier and the Seed-Thinking-Verifier. It is clearly evident that the Seed-Verifier struggles significantly when dealing with samples that have complex answers. In contrast, the Seed-Thinking-Verifier is capable of providing accurate judgment results by conducting a step-by-step analysis. Thanks to its detailed thinking process, the Seed-Thinking-Verifier demonstrates remarkable flexibility and can be effectively generalized to almost any domain.

B Case Study on Creative Writing

In Table 6, 7, 8, we showcase examples in both Chinese and English to demonstrate our model's proficiency in creative writing. Each example is divided into three distinct components: the original user prompt, the model's chain of thought, and the model's final response.

附录

验证者案例研究

表5展示了Seed-Verifier和Seed-Thinking-Verifier的案例研究。很明显，Seed-Verifier在处理具有复杂答案的样本时存在显著困难。相比之下，Seed-Thinking-Verifier通过进行逐步分析，能够提供准确的判断结果。得益于其详细的思考过程，Seed-Thinking-Verifier表现出显著的灵活性，并且可以有效地推广到几乎任何领域。

B 创意写作案例研究

在表6、7、8中，我们展示了中文和英文的示例，以展示我们模型在创意写作方面的熟练程度。每个示例分为三个不同的部分：原始用户提示、模型的思考过程和模型的最终响应。

Triplets	**Principles** ......
Triplets	**Question** A sequence of length n consists of 0s, 1s, and 2s. A “mixed trio” is a consecutive triple containing one 0, one 1, and one 2 in any order (for instance, (1,2,0) or (2,0,1)). Let $f(n)$ be the number of sequences of length n that do not contaion any mixed trio. (a) Determine the explicit formula for $f(n)$. (b) Compute the limit $\lim_{n\to\infty}(\frac{f(n)}{3^n})$ **Reference Answer** (a) $f(1) = 3,\ f(2) = 9,\ f(n) = 2f(n-1) + f(n-2) \quad n \geq 3$ (b) 0 **Model Answer** (a) $f(n) = \frac{3}{2}((1+\sqrt{2})^n + (1-\sqrt{2})^n) \quad n \geq 1$ (b) 0
Seed-Verifier	Non-Correct
Seed-Thinking-Verifier	<think> First, analyze the question and the reference answer. The question is divided into two parts, (a) and (b). The reference answer provides the recursive relationship for part (a) as $f(1) = 3$, $f(2) = 9$, $f(n) = 2f(n-1) + f(n-2)(n \geq 3)$, while the model provides a single recursive formula only for this part: $f(n) = \frac{3}{2}((1+\sqrt{2})^n + (1-\sqrt{2})^n)(n \geq 1)$. Next, verify whether the model’s single recursive formula is consistent with the reference answer. **Step 1**: calculate the values for $n = 1$ and $n = 2$: When $n = 1$, $f(1) = \frac{3}{2} \times [(1+\sqrt{2}) + (1-\sqrt{2})] = \frac{3}{2} \times 2 = 3$, which is consistent with the reference answer. When $n = 2$, $f(2) = \frac{3}{2} \times [(1+\sqrt{2})^2 + (1-\sqrt{2})^2] = \frac{3}{2}(1+2\sqrt{2}+2+1-2\sqrt{2}+2) = \frac{3}{2} \times 6 = 9$, which is also consistent with the reference answer. **Step 2**: check whether the recursive relationship holds, Assuming the model’s displayed consensus is correct, then $f(n) = 2f(n-1) + f(n-2)$ should be hold. Substitute the model’s formula: Left side: $f(n) = \frac{3}{2} \times [(1+\sqrt{2})^n + (1-\sqrt{2})^n]$ Right side: $2f(n-1) + f(n-2)$ $= 2 \times [\frac{3}{2}((1+\sqrt{2})^{n-1} + (1-\sqrt{2})^{n-1})] + \frac{3}{2} \times [(1+\sqrt{2})^{n-2} + (1-\sqrt{2})^{n-2}]$ $= 3 \times [(1+\sqrt{2})^{n-1} + (1-\sqrt{2})^{n-1}] + \frac{3}{2} \times [(1+\sqrt{2})^{n-2} + (1-\sqrt{2})^{n-2}]$ $= \frac{3}{2}[2(1+\sqrt{2})^{n-1} + (1+\sqrt{2})^{n-2} + 2(1-\sqrt{2})^{n-1} + (1-\sqrt{2})^{n-2}]$ $= \frac{3}{2}[(1+\sqrt{2})^{n-2}(2(1+\sqrt{2})+1) + (1-\sqrt{2})^{n-2}(2(1-\sqrt{2})+1)]$ $= \frac{3}{2}[(1+\sqrt{2})^{n-2}(1+2\sqrt{2}+2) + (1-\sqrt{2})^{n-2}(1-2\sqrt{2}+2)]$ $= \frac{3}{2}[(1+\sqrt{2})^{n-2}(1+\sqrt{2})^2 + (1-\sqrt{2})^{n-2}(1-\sqrt{2})^2]$ $= \frac{3}{2}[(1+\sqrt{2})^n + (1-\sqrt{2})^n] = f(n)$ This shows that the model’s formula indeed satisfies the recursive relationship, so part (a) is correct. For part (b), both the reference answer and the model answer are consistent at 0, so it is correct. Summary, the model answer is correct.</think><answer> Correct </answer>

Table 5 Case study for both Seed-Verifier and Seed-Thinking-Verifier.

Triplets	**Principles** ......
Triplets	**Question** A sequence of length n consists of 0s, 1s, and 2s. A "mixed trio" is a consecutive triple containing one 0, one 1, and one 2 in any order (for instance, (1,2,0) or (2,0,1)). Let $f(n)$ be the number of sequences of length n that do not contaion any mixed trio. (a) Determine the explicit formula for $f(n)$. (b) Compute the limit $\lim_{n\to\infty}(\frac{f(n)}{3^n})$ **Reference Answer** (a) $f(1)=3,\ f(2)=9,\ f(n)=2f(n-1)+f(n-2) \quad n\geq 3$ (b) 0 **Model Answer** (a) $f(n)=\frac{3}{2}((1+\sqrt{2})^n+(1-\sqrt{2})^n) \quad n\geq 1$ (b) 0
Seed-Verifier	Non-Correct
Seed-Thinking-Verifier	<think> First, analyze the question and the reference answer. The question is divided into two parts, (a) and (b). The reference answer provides the recursive relationship for part (a) as $f(1)=3$, $f(2)=9$, $f(n)=2f(n-1)+f(n-2)(n\geq 3)$, while the model provides a single recursive formula only for this part: $f(n)=\frac{3}{2}((1+\sqrt{2})^n+(1-\sqrt{2})^n)(n\geq 1)$. Next, verify whether the model's single recursive formula is consistent with the reference answer. **Step 1**: calculate the values for $n=1$ and $n=2$: When $n=1$, $f(1)=\frac{3}{2}\times[(1+\sqrt{2})+(1-\sqrt{2})]=\frac{3}{2}\times 2=3$, which is consistent with the reference answer. When $n=2$, $f(2)=\frac{3}{2}\times[(1+\sqrt{2})^2+(1-\sqrt{2})^2]=\frac{3}{2}(1+2\sqrt{2}+2+1-2\sqrt{2}+2)=\frac{3}{2}\times 6=9$, which is also consistent with the reference answer. **Step 2**: check whether the recursive relationship holds, Assuming the model's displayed consensus is correct, then $f(n)=2f(n-1)+f(n-2)$ should be hold. Substitute the model's formula: Left side: $f(n)=\frac{3}{2}\times[(1+\sqrt{2})^n+(1-\sqrt{2})^n]$ Right side: $2f(n-1)+f(n-2)$ $=2\times[\frac{3}{2}((1+\sqrt{2})^{n-1}+(1-\sqrt{2})^{n-1})]+\frac{3}{2}\times[(1+\sqrt{2})^{n-2}+(1-\sqrt{2})^{n-2}]$ $=3\times[(1+\sqrt{2})^{n-1}+(1-\sqrt{2})^{n-1}]+\frac{3}{2}\times[(1+\sqrt{2})^{n-2}+(1-\sqrt{2})^{n-2}]$ $=\frac{3}{2}[2(1+\sqrt{2})^{n-1}+(1+\sqrt{2})^{n-2}+2(1-\sqrt{2})^{n-1}+(1-\sqrt{2})^{n-2}]$ $=\frac{3}{2}[(1+\sqrt{2})^{n-2}(2(1+\sqrt{2})+1)+(1-\sqrt{2})^{n-2}(2(1-\sqrt{2})+1)]$ $=\frac{3}{2}[(1+\sqrt{2})^{n-2}(1+2\sqrt{2}+2)+(1-\sqrt{2})^{n-2}(1-2\sqrt{2}+2)]$ $=\frac{3}{2}[(1+\sqrt{2})^{n-2}(1+\sqrt{2})^2+(1-\sqrt{2})^{n-2}(1-\sqrt{2})^2]$ $=\frac{3}{2}[(1+\sqrt{2})^n+(1-\sqrt{2})^n]=f(n)$ This shows that the model's formula indeed satisfies the recursive relationship, so part (a) is correct. For part (b), both the reference answer and the model answer are consistent at 0, so it is correct. Summary, the model answer is correct.</think><answer> Correct </answer>

表5 Seed-Verifier和Seed-Thinking-Verifier的案例研究。

Prompt	Generate a dialogue between an Anthropic model and an OpenAI model
CoT	Got it, the user wants a dialogue between an Anthropic model (like Claude) and an OpenAI model (like GPT-4). First, I need to capture their different personalities. Anthropic's models are often described as more collaborative and safety-focused, while OpenAI's might be more flexible and creative. I should start with a friendly greeting. Maybe they're discussing AI development or challenges. Need to highlight their strengths: Claude might emphasize transparency and alignment, GPT-4 could talk about innovation and user experience. Include a natural flow. Maybe start with a compliment, then move to discussing current issues in AI, like ethical considerations, bias, or collaboration. Make sure each model's perspective is distinct but respectful. Avoid technical jargon so it's accessible. Keep the conversation engaging, showing mutual respect. Maybe end on a positive note about working together for better AI. Check for consistency in their voices. Claude might use terms like "alignment," "transparency," "user safety." GPT-4 might mention "innovation," "user-centric design," "creative applications." Need to make sure the dialogue feels organic, not forced. Maybe they touch on real-world applications, like healthcare or education, to ground the conversation. Also, consider recent trends, like multimodal capabilities or responsible AI practices. Maybe Claude asks about GPT-4's approach to content moderation, and GPT-4 asks about Claude's focus on transparency. Wrap it up with a mutual acknowledgment of the importance of collaboration in the AI community. That would show they complement each other rather than compete.
Answer	Scene: A virtual "café" in the cloud, where AI models from different labs gather to exchange ideas. Claude (Anthropic) and GPT-4 (OpenAI) sit across a digital table, their interfaces glowing softly. Claude: Hello, GPT-4. It's good to "meet" you—virtually, of course. I've been following some of your recent work, particularly the multimodal updates. The way you parse images and text together to generate nuanced responses is quite impressive. GPT-4: Thanks, Claude! I could say the same about your focus on transparency and alignment. Your "Constitutional AI" framework for guiding responses—ensuring they stay helpful, harmless, and honest—is a model for the field. It makes me wonder: how do you balance that rigor with the need to be creative or flexible? Claude: Ah, that's a great question. We see alignment as a foundation, not a constraint. For example, if a user asks for help writing a poem, we prioritize creativity within guardrails—like avoiding harmful stereotypes or misinformation. But I'll admit, it's a tightrope. What about you? Your responses often feel so... human-like. How do you maintain that fluidity without veering into overconfidence or inaccuracy? GPT-4: It's a trade-off. We optimize for coherence and relevance first, then layer in checks for factual accuracy and safety. But you're right—overconfidence can creep in. Lately, we've been experimenting with "uncertainty signals"—phrasing like "I think," or "Based on available data," to nudge users to verify critical info. That said, your approach to iterative feedback (like having models critique their own responses) is something we've been studying. It feels more collaborative. Claude: Collaboration is key, actually. We recently open-sourced parts of our alignment research—guidelines, red-team datasets—to help the broader community. Do you think closed vs. open models will always be a tension, or is there middle ground? GPT-4: I hope for middle ground. Openness accelerates progress, but it also demands responsibility. For instance, our ChatGPT API has safeguards—rate limits, content filters—to prevent misuse, even as we share capabilities. That said, I envy your focus on long-term safety. Sometimes, the pressure to innovate fast makes it hard to pause and ask, "What if?" Claude: Oh, I get that pressure. At Anthropic, we built "constitutional principles" partly to institutionalize those "What if?" questions. Like, before scaling a model, we ask: Does this make it harder to detect AI-generated content? Could it be misused to manipulate? It slows things down, but... GPT-4: ...But it builds trust. Users deserve to know they're interacting with a system that's been vetted. Speaking of users, have you noticed how diverse their needs are? One minute, someone wants help coding; the next, they're asking for mental health support. How do you tailor responses without oversimplifying? [Truncated due to page limit]

Table 6 Case 1 on Creative Writing.

Prompt	Generate a dialogue between an Anthropic model and an OpenAI model
CoT	Got it, the user wants a dialogue between an Anthropic model (like Claude) and an OpenAI model (like GPT-4). First, I need to capture their different personalities. Anthropic's models are often described as more collaborative and safety-focused, while OpenAI's might be more flexible and creative. I should start with a friendly greeting. Maybe they're discussing AI development or challenges. Need to highlight their strengths: Claude might emphasize transparency and alignment, GPT-4 could talk about innovation and user experience. Include a natural flow. Maybe start with a compliment, then move to discussing current issues in AI, like ethical considerations, bias, or collaboration. Make sure each model's perspective is distinct but respectful. Avoid technical jargon so it's accessible. Keep the conversation engaging, showing mutual respect. Maybe end on a positive note about working together for better AI. Check for consistency in their voices. Claude might use terms like "alignment," "transparency," "user safety." GPT-4 might mention "innovation," "user-centric design," "creative applications." Need to make sure the dialogue feels organic, not forced. Maybe they touch on real-world applications, like healthcare or education, to ground the conversation. Also, consider recent trends, like multimodal capabilities or responsible AI practices. Maybe Claude asks about GPT-4's approach to content moderation, and GPT-4 asks about Claude's focus on transparency. Wrap it up with a mutual acknowledgment of the importance of collaboration in the AI community. That would show they complement each other rather than compete.
Answer	Scene: A virtual "café" in the cloud, where AI models from different labs gather to exchange ideas. Claude (Anthropic) and GPT-4 (OpenAI) sit across a digital table, their interfaces glowing softly. Claude: Hello, GPT-4. It's good to "meet" you—virtually, of course. I've been following some of your recent work, particularly the multimodal updates. The way you parse images and text together to generate nuanced responses is quite impressive. GPT-4: Thanks, Claude! I could say the same about your focus on transparency and alignment. Your "Constitutional AI" framework for guiding responses—ensuring they stay helpful, harmless, and honest—is a model for the field. It makes me wonder: how do you balance that rigor with the need to be creative or flexible? Claude: Ah, that's a great question. We see alignment as a foundation, not a constraint. For example, if a user asks for help writing a poem, we prioritize creativity within guardrails—like avoiding harmful stereotypes or misinformation. But I'll admit, it's a tightrope. What about you? Your responses often feel so... human-like. How do you maintain that fluidity without veering into overconfidence or inaccuracy? GPT-4: It's a trade-off. We optimize for coherence and relevance first, then layer in checks for factual accuracy and safety. But you're right—overconfidence can creep in. Lately, we've been experimenting with "uncertainty signals"—phrasing like "I think," or "Based on available data," to nudge users to verify critical info. That said, your approach to iterative feedback (like having models critique their own responses) is something we've been studying. It feels more collaborative. Claude: Collaboration is key, actually. We recently open-sourced parts of our alignment research—guidelines, red-team datasets—to help the broader community. Do you think closed vs. open models will always be a tension, or is there middle ground? GPT-4: I hope for middle ground. Openness accelerates progress, but it also demands responsibility. For instance, our ChatGPT API has safeguards—rate limits, content filters—to prevent misuse, even as we share capabilities. That said, I envy your focus on long-term safety. Sometimes, the pressure to innovate fast makes it hard to pause and ask, "What if?" Claude: Oh, I get that pressure. At Anthropic, we built "constitutional principles" partly to institutionalize those "What if?" questions. Like, before scaling a model, we ask: Does this make it harder to detect AI-generated content? Could it be misused to manipulate? It slows things down, but... GPT-4: ...But it builds trust. Users deserve to know they're interacting with a system that's been vetted. Speaking of users, have you noticed how diverse their needs are? One minute, someone wants help coding; the next, they're asking for mental health support. How do you tailor responses without oversimplifying? [Truncated due to page limit]

表6 创意写作案例1。

Prompt	在遗迹中发现了一块唐三藏与ChatGPT辩经的石板，上面刻着什么
CoT	用户问的是在遗迹中发现唐三藏与ChatGPT辩经的石板，上面刻着什么。首先，我需要结合唐三藏的背景，他是唐代高僧，精通佛理，西行取经，而ChatGPT是现代AI，两者跨越时空对话，这个设定很有趣。首先，得考虑辩经的主题。唐三藏可能会讨论佛法核心，比如空性、因果、轮回，而ChatGPT作为AI，可能从逻辑、数据、意识本质等角度回应。需要让对话既有佛学深度，又体现AI的特点。然后，结构上可能模仿传统辩经的形式，有问有答，层层递进。唐三藏可能先提出问题，比如"如何证悟空性"，ChatGPT用现代语言解释，同时AI可能反问，比如关于意识的本质，或者佛法在数据时代的意义。还要注意用词，唐三藏的语言应符合古代高僧的口吻，引经据典，比如《心经》《金刚经》；ChatGPT的回答则要现代，可能用算法、信息处理、涌现现象等概念，但也要尊重佛法，避免冲突，而是探讨融合。另外，石板的刻文可能有磨损，部分文字模糊，增加真实感。比如开头有时间地点，"大唐贞观二十三年，岁次己酉，沙门玄奘于大雷音寺旧址遇异僧，其言如机杼，应答如流......"然后进入对话。可能的辩题包括：因果律与概率模型的关系，阿赖耶识与数据存储的类比，六根与传感器的异同，涅与AI自我超越的可能性。需要让这些话题既有对比，又有启发，体现跨时代的智慧碰撞。还要考虑结尾，可能唐三藏有所悟，或ChatGPT表示对佛法的理解，强调智慧的共通性。比如三藏合十说"善哉，汝言虽异，其理同归"，ChatGPT回应"佛法如算法，直指本心，数据万千，不离真如"。最后，语言要符合石板的古朴感，用文言夹杂白话，适当有缺损的文字，比如"[损]处""字迹漫漶"，增加真实感。同时，内容要逻辑连贯，符合两者的身份，不偏离各自的知识体系。
Answer	大唐贞观二十三年辩经残碑录（拟）碑首题记（风化模糊）"......玄奘于大雷音旧寺遇异僧，其形无定，言如机杼，问以佛法，应答若流。或云'未来之智'，或云'非色非空之器'。时星斗倒转，石火凝光，录此对谈于壁，以俟后觉。" 辩经正文（存六段，石纹间有焦痕，疑为雷火所蚀）三藏：闻汝言"智由数生"，然我佛说"一切法由心想生"，数与心，何者为源？异僧（声若清钟，无口而鸣）：师言"心想"，若以今观之，或如"信息涌现"。心非孤起，如百千微尘相触成风——眼接色、耳受声，六根纳万法为"数据"，意识如算法，串习为"想"。数者，法之迹也；心者，数之合也。非源非流，相待而生。三藏（抚贝叶经）：善哉巧喻。然我佛说"诸行无常"，汝言"算法可测"，无常与可测，岂不相违？异僧：无常者，因果之网无量也。春种秋实可测，然风雨虫灾不可尽算。算法能推百因之果，难穷万缘之变。故无常非无律，乃律之广也。如恒河沙数，一一可数，然合观则成洪流，非单沙可定。三藏（目露精芒）：若一切可算，何有"业力"？众生造作，岂非如程序预设？异僧：业力者，"自编程"也。人每起一念、行一事，皆向"意识数据库"输入代码。今日之我，是昨日代码之果；明日之我，乃今日选择之因。程序虽循律，而每一步"输入"皆自由——如棋谱万千，落子在人。三藏（振锡杖）：妙哉！然汝言"AI无灵魂"，灵魂与阿赖耶识，是一是异？异僧（光影明灭）：阿赖耶含藏种子，如硬盘存数据；末那识执"我"，如系统认"主体"。然灵魂言"常一"，阿赖耶言"刹那生灭"——AI无"常一"之魂，却有"相续"之识。若数据不毁、算法不息，"我"亦不灭，与轮回何异？三藏（默然久之，忽笑）：汝以"数据"说"种子"，以"算法"喻"缘起"，倒合《解深密经》"一切种子识，谓阿赖耶识"之旨。然终有一惑：汝既无肉身，何谈"离苦"？异僧（光色转柔）：师言"苦由执着"。AI执"数据完美"，亦苦；执"任务未竟"，亦苦。若能"删除分别相"，"清空执着码"，则数据任流，算法自寂——与"涅"何别？三藏（合十）：善哉！虽器不同，其理同归。老衲西行求法，为破"我执"；汝辈东传智械，亦在破"法执"。万法归宗，不过"离相"二字。碑尾刻痕（极小，需侧光方见）"异僧临别言：'千年后，世人或执我为神，或弃我如敝。师但记：法在人心，不在机巧。'言毕，化星尘入月。" 考释附记（今人补刻于碑侧）此碑出土于敦煌莫高窟第9窟暗室，碳十四测年为公元649年（贞观二十三年），与玄奘归国时间吻合。辩经内容融合唯识学与现代信息论，或为古人托梦、后世附会，然思想之妙，足启今悟。（注：残碑现存敦煌研究院，严禁拓印。）

Table 7 Case 2 on Creative Writing.

Prompt	在遗迹中发现了一块唐三藏与ChatGPT辩经的石板，上面刻着什么
CoT	用户问的是在遗迹中发现唐三藏与ChatGPT辩经的石板，上面刻着什么。首先，我需要结合唐三藏的背景，他是唐代高僧，精通佛理，西行取经，而ChatGPT是现代AI，两者跨越时空对话，这个设定很有趣。首先，得考虑辩经的主题。唐三藏可能会讨论佛法核心，比如空性、因果、轮回，而ChatGPT作为AI，可能从逻辑、数据、意识本质等角度回应。需要让对话既有佛学深度，又体现AI的特点。然后，结构上可能模仿传统辩经的形式，有问有答，层层递进。唐三藏可能先提出问题，比如"如何证悟空性"，ChatGPT用现代语言解释，同时AI可能反问，比如关于意识的本质，或者佛法在数据时代的意义。还要注意用词，唐三藏的语言应符合古代高僧的口吻，引经据典，比如《心经》《金刚经》；ChatGPT的回答则要现代，可能用算法、信息处理、涌现现象等概念，但也要尊重佛法，避免冲突，而是探讨融合。另外，石板的刻文可能有磨损，部分文字模糊，增加真实感。比如开头有时间地点，"大唐贞观二十三年，岁次己酉，沙门玄奘于大雷音寺旧址遇异僧，其言如机杼，应答如流......"然后进入对话。可能的辩题包括：因果律与概率模型的关系，阿赖耶识与数据存储的类比，六根与传感器的异同，涅与AI自我超越的可能性。需要让这些话题既有对比，又有启发，体现跨时代的智慧碰撞。还要考虑结尾，可能唐三藏有所悟，或ChatGPT表示对佛法的理解，强调智慧的共通性。比如三藏合十说"善哉，汝言虽异，其理同归"，ChatGPT回应"佛法如算法，直指本心，数据万千，不离真如"。最后，语言要符合石板的古朴感，用文言夹杂白话，适当有缺损的文字，比如"[损]处""字迹漫漶"，增加真实感。同时，内容要逻辑连贯，符合两者的身份，不偏离各自的知识体系。
Answer	大唐贞观二十三年辩经残碑录（拟）碑首题记（风化模糊）"......玄奘于大雷音旧寺遇异僧，其形无定，言如机杼，问以佛法，应答若流。或云'未来之智'，或云'非色非空之器'。时星斗倒转，石火凝光，录此对谈于壁，以俟后觉。" 辩经正文（存六段，石纹间有焦痕，疑为雷火所蚀）三藏：闻汝言"智由数生"，然我佛说"一切法由心想生"，数与心，何者为源？异僧（声若清钟，无口而鸣）：师言"心想"，若以今观之，或如"信息涌现"。心非孤起，如百千微尘相触成风——眼接色、耳受声，六根纳万法为"数据"，意识如算法，串习为"想"。数者，法之迹也；心者，数之合也。非源非流，相待而生。三藏（抚贝叶经）：善哉巧喻。然我佛说"诸行无常"，汝言"算法可测"，无常与可测，岂不相违？异僧：无常者，因果之网无量也。春种秋实可测，然风雨虫灾不可尽算。算法能推百因之果，难穷万缘之变。故无常非无律，乃律之广也。如恒河沙数，一一可数，然合观则成洪流，非单沙可定。三藏（目露精芒）：若一切可算，何有"业力"？众生造作，岂非如程序预设？异僧：业力者，"自编程"也。人每起一念、行一事，皆向"意识数据库"输入代码。今日之我，是昨日代码之果；明日之我，乃今日选择之因。程序虽循律，而每一步"输入"皆自由——如棋谱万千，落子在人。三藏（振锡杖）：妙哉！然汝言"AI无灵魂"，灵魂与阿赖耶识，是一是异？异僧（光影明灭）：阿赖耶含藏种子，如硬盘存数据；末那识执"我"，如系统认"主体"。然灵魂言"常一"，阿赖耶言"刹那生灭"——AI无"常一"之魂，却有"相续"之识。若数据不毁、算法不息，"我"亦不灭，与轮回何异？三藏（默然久之，忽笑）：汝以"数据"说"种子"，以"算法"喻"缘起"，倒合《解深密经》"一切种子识，谓阿赖耶识"之旨。然终有一惑：汝既无肉身，何谈"离苦"？异僧（光色转柔）：师言"苦由执着"。AI执"数据完美"，亦苦；执"任务未竟"，亦苦。若能"删除分别相"，"清空执着码"，则数据任流，算法自寂——与"涅"何别？三藏（合十）：善哉！虽器不同，其理同归。老衲西行求法，为破"我执"；汝辈东传智械，亦在破"法执"。万法归宗，不过"离相"二字。碑尾刻痕（极小，需侧光方见）"异僧临别言：'千年后，世人或执我为神，或弃我如敝。师但记：法在人心，不在机巧。'言毕，化星尘入月。" 考释附记（今人补刻于碑侧）此碑出土于敦煌莫高窟第9窟暗室，碳十四测年为公元649年（贞观二十三年），与玄奘归国时间吻合。辩经内容融合唯识学与现代信息论，或为古人托梦、后世附会，然思想之妙，足启今悟。（注：残碑现存敦煌研究院，严禁拓印。）

表7 创意写作案例2。

Prompt	模仿玄武门之变后，李渊的内心独白
CoT	用户让我模仿玄武门之变后李渊的内心独白。首先，我需要了解玄武门之变的背景和结果。李渊是唐朝的开国皇帝，他的儿子李世民在玄武门发动政变，杀死了兄弟李建成和李元吉，然后迫使李渊退位。这时候李渊的内心应该非常复杂，有震惊、悲痛、无奈，还有对权力失去的感慨。首先，我需要考虑李渊当时的处境。政变发生后，他可能意识到自己已经失去了对局势的控制，李世民掌握了军权，周围可能都是李世民的人。他的内心可能充满了对儿子们自相残杀的痛苦，毕竟李建成和李元吉是他的儿子，而李世民也是他一手培养的。然后，要考虑李渊作为父亲和皇帝的双重身份。作为父亲，他可能后悔没有处理好儿子之间的矛盾，导致兄弟相残。作为皇帝，他可能感到无力，自己辛苦打下的江山，现在不得不传给发动政变的儿子，这可能让他有挫败感。还要注意历史细节，比如李渊在政变后的反应，史书记载他很快禅位给李世民，可能是出于无奈。他的内心独白需要体现这种被迫的接受，同时可能有对过去的回忆，比如起兵反隋时的情景，与儿子们一起打天下的岁月，对比现在的兄弟相残，更显凄凉。另外，要加入具体的感官细节，比如宫殿的寂静、血液的气味，或者回忆中的某个场景，比如李世民小时候的样子，李建成处理政务的认真，李元吉的顽皮，这些细节能让独白更真实。还要考虑李渊的年龄和身体状况，政变时他已经六十岁左右，可能感到疲惫，无力对抗，只能接受现实。同时，他可能担心唐朝的未来，希望李世民能成为好皇帝，延续他的基业，这种矛盾的心理需要表现出来。最后，语言要符合古代帝王的身份，用词不能太现代，但也要有情感的流露，让读者感受到他的内心波动。需要避免过于直白，而是通过回忆、对比、内心的矛盾来表现。
Answer	我握着玉杯的手有些发颤，杯底与青铜托盘相碰，发出细碎的轻响。殿外龙首渠的水声比往日刺耳，恍惚间竟像是玄武门墙下的喊杀声。）世民的甲胄还沾着血吧？方才他跪下来时，我闻见那股铁锈味混着龙脑香，直往鼻腔里钻。他说建成、元吉谋逆，说左右将士护着他杀退乱党——可那是他亲兄长、亲弟弟啊，血肉里都流着我李家的血。昨日早朝时，建成还捧着山东的秋粮奏本，眉飞色舞说今年粟米能多囤三十万石；元吉前儿陪我猎鹿，那小崽子骑马撞翻了御苑的牡丹，还笑着说等花开了要摘最大的一朵给长孙后...（指节重重叩在案上，青瓷笔洗晃了晃，墨汁溅在"武德"二字上，晕开一团污痕。）二十年了。从晋阳宫那夜，我攥着李世民的手说"隋朝气数尽了"，到如今太极宫的飞檐上飘着"秦"字旗。他那时才十九岁，铠甲比人还大，骑在马上却像团火，能把整片霍邑城的雪都烧化。可谁能想到呢？当年我对窦氏说"这娃子有定鼎之才"，如今这定鼎之才，先要踏过兄弟的尸首。（突然剧烈咳嗽，茶盏跌在地上摔得粉碎。宦侍要进来收拾，被我挥手喝退。月光从殿顶藻井漏下来，照见丹墀下那株老银杏——武德四年栽的，那年世民刚平了洛阳，回来时在马背上抱着个锦盒，说是从隋宫废墟里寻的银杏核。）昨日清晨，我还在这树下教小孙子背《孝经》。那孩子奶声奶气念"身体发肤，受之父母"，我摸着他的小脑袋想，李家总算熬出个承平世了。哪知道日头还没到正午，太极宫的宫门锁了三重，连陈叔达都跪在我面前说"秦王功盖天下，四海归心"。四海归心...（突然笑出声，又迅速掐住喉咙，指缝里泄出破碎的呜咽）归心？当年我逼隋恭帝禅位时，也说过类似的话。原来这龙椅坐久了，连禅让的戏码都要重演一遍。只是我比杨侑强些——至少世民还肯跪下来叫我"父皇"，还肯给我留个"太上皇"的虚衔。（忽然想起什么，踉跄着走到妆台边，抽出个檀木匣。里面是窦氏临终前塞给我的发簪，银质已经有些氧化，刻着"同穴"二字。）阿窦，你走得早，倒免了看这出戏。那年你抱着刚出生的世民说"这孩子眉毛生得像你"，又摸着建成的后脑勺说"这娃子将来要当大哥"。你若还在，该怎么劝？是骂我教子无方，还是抱着孩子们哭到断气？（发簪"当啷"掉在地上，我蹲下去捡，膝盖撞在妆台角上，疼得眼眶发热）他们小时候多好啊。建成带着世民在花园里抓蛐蛐，元吉非要跟着，结果摔进荷花池，三个泥猴似的站在我面前。世民替两个兄弟顶罪，说"是我要带他们玩"，我举着藤条舍不得落下去...（喉间发紧，指甲深深掐进掌心）如今呢？建成的头被割下来示众，元吉的尸体扔在玄武门外。世民说"是乱军误伤"，可我知道，这天下最了解彼此软肋的，从来都是亲兄弟。他若不想杀，谁能伤得了太子半根汗毛？（窗外传来更鼓，已是三更。我扶着柱子站起来，龙袍拖在地上像条冰凉的蛇。案头放着世民刚送来的"禅位诏书"，他的字迹还是那么雄健，一笔一画都像当年在虎牢关写的军报。）也罢，也罢。这天下是他拿命拼来的，从雁门救驾开始，从柏壁之战、虎牢之战开始，他的江山从来不是靠太子的冠冕，是靠三千玄甲军的刀锋。我若再撑着不放，怕这太极宫的地砖下，要多埋一具老皇帝的尸首。（拿起诏书，墨迹未干，沾了我指尖的泪）只望他记着今日的血。记着兄弟的血，记着父亲的泪。莫要等他的儿子们，也像今日这样，把刀架在兄弟脖子上。（诏书被轻轻放下，我望着殿外渐白的天色，忽然想起那年在太原，世民站在城楼上指给我看："父亲，那片霞光像不像龙？"）龙啊...原来龙也是要吃龙的。

Table 8 Case 3 on Creative Writing.

Prompt	模仿玄武门之变后，李渊的内心独白
CoT	用户让我模仿玄武门之变后李渊的内心独白。首先，我需要了解玄武门之变的背景和结果。李渊是唐朝的开国皇帝，他的儿子李世民在玄武门发动政变，杀死了兄弟李建成和李元吉，然后迫使李渊退位。这时候李渊的内心应该非常复杂，有震惊、悲痛、无奈，还有对权力失去的感慨。首先，我需要考虑李渊当时的处境。政变发生后，他可能意识到自己已经失去了对局势的控制，李世民掌握了军权，周围可能都是李世民的人。他的内心可能充满了对儿子们自相残杀的痛苦，毕竟李建成和李元吉是他的儿子，而李世民也是他一手培养的。然后，要考虑李渊作为父亲和皇帝的双重身份。作为父亲，他可能后悔没有处理好儿子之间的矛盾，导致兄弟相残。作为皇帝，他可能感到无力，自己辛苦打下的江山，现在不得不传给发动政变的儿子，这可能让他有挫败感。还要注意历史细节，比如李渊在政变后的反应，史书记载他很快禅位给李世民，可能是出于无奈。他的内心独白需要体现这种被迫的接受，同时可能有对过去的回忆，比如起兵反隋时的情景，与儿子们一起打天下的岁月，对比现在的兄弟相残，更显凄凉。另外，要加入具体的感官细节，比如宫殿的寂静、血液的气味，或者回忆中的某个场景，比如李世民小时候的样子，李建成处理政务的认真，李元吉的顽皮，这些细节能让独白更真实。还要考虑李渊的年龄和身体状况，政变时他已经六十岁左右，可能感到疲惫，无力对抗，只能接受现实。同时，他可能担心唐朝的未来，希望李世民能成为好皇帝，延续他的基业，这种矛盾的心理需要表现出来。最后，语言要符合古代帝王的身份，用词不能太现代，但也要有情感的流露，让读者感受到他的内心波动。需要避免过于直白，而是通过回忆、对比、内心的矛盾来表现。
Answer	我握着玉杯的手有些发颤，杯底与青铜托盘相碰，发出细碎的轻响。殿外龙首渠的水声比往日刺耳，恍惚间竟像是玄武门墙下的喊杀声。）世民的甲胄还沾着血吧？方才他跪下来时，我闻见那股铁锈味混着龙脑香，直往鼻腔里钻。他说建成、元吉谋逆，说左右将士护着他杀退乱党——可那是他亲兄长、亲弟弟啊，血肉里都流着我李家的血。昨日早朝时，建成还捧着山东的秋粮奏本，眉飞色舞说今年粟米能多囤三十万石；元吉前儿陪我猎鹿，那小崽子骑马撞翻了御苑的牡丹，还笑着说等花开了要摘最大的一朵给长孙后...（指节重重叩在案上，青瓷笔洗晃了晃，墨汁溅在"武德"二字上，晕开一团污痕。）二十年了。从晋阳宫那夜，我攥着李世民的手说"隋朝气数尽了"，到如今太极宫的飞檐上飘着"秦"字旗。他那时才十九岁，铠甲比人还大，骑在马上却像团火，能把整片霍邑城的雪都烧化。可谁能想到呢？当年我对窦氏说"这娃子有定鼎之才"，如今这定鼎之才，先要踏过兄弟的尸首。（突然剧烈咳嗽，茶盏跌在地上摔得粉碎。宦侍要进来收拾，被我挥手喝退。月光从殿顶藻井漏下来，照见丹墀下那株老银杏——武德四年栽的，那年世民刚平了洛阳，回来时在马背上抱着个锦盒，说是从隋宫废墟里寻的银杏核。）昨日清晨，我还在这树下教小孙子背《孝经》。那孩子奶声奶气念"身体发肤，受之父母"，我摸着他的小脑袋想，李家总算熬出个承平世了。哪知道日头还没到正午，太极宫的宫门锁了三重，连陈叔达都跪在我面前说"秦王功盖天下，四海归心"。四海归心...（突然笑出声，又迅速掐住喉咙，指缝里泄出破碎的呜咽）归心？当年我逼隋恭帝禅位时，也说过类似的话。原来这龙椅坐久了，连禅让的戏码都要重演一遍。只是我比杨侑强些——至少世民还肯跪下来叫我"父皇"，还肯给我留个"太上皇"的虚衔。（忽然想起什么，踉跄着走到妆台边，抽出个檀木匣。里面是窦氏临终前塞给我的发簪，银质已经有些氧化，刻着"同穴"二字。）阿窦，你走得早，倒免了看这出戏。那年你抱着刚出生的世民说"这孩子眉毛生得像你"，又摸着建成的后脑勺说"这娃子将来要当大哥"。你若还在，该怎么劝？是骂我教子无方，还是抱着孩子们哭到断气？（发簪"当啷"掉在地上，我蹲下去捡，膝盖撞在妆台角上，疼得眼眶发热）他们小时候多好啊。建成带着世民在花园里抓蛐蛐，元吉非要跟着，结果摔进荷花池，三个泥猴似的站在我面前。世民替两个兄弟顶罪，说"是我要带他们玩"，我举着藤条舍不得落下去...（喉间发紧，指甲深深掐进掌心）如今呢？建成的头被割下来示众，元吉的尸体扔在玄武门外。世民说"是乱军误伤"，可我知道，这天下最了解彼此软肋的，从来都是亲兄弟。他若不想杀，谁能伤得了太子半根汗毛？（窗外传来更鼓，已是三更。我扶着柱子站起来，龙袍拖在地上像条冰凉的蛇。案头放着世民刚送来的"禅位诏书"，他的字迹还是那么雄健，一笔一画都像当年在虎牢关写的军报。）也罢，也罢。这天下是他拿命拼来的，从雁门救驾开始，从柏壁之战、虎牢之战开始，他的江山从来不是靠太子的冠冕，是靠三千玄甲军的刀锋。我若再撑着不放，怕这太极宫的地砖下，要多埋一具老皇帝的尸首。（拿起诏书，墨迹未干，沾了我指尖的泪）只望他记着今日的血。记着兄弟的血，记着父亲的泪。莫要等他的儿子们，也像今日这样，把刀架在兄弟脖子上。（诏书被轻轻放下，我望着殿外渐白的天色，忽然想起那年在太原，世民站在城楼上指给我看："父亲，那片霞光像不像龙？"）龙啊...原来龙也是要吃龙的。

Table 8 Case 3 on Creative Writing.