LONG-HORIZON REASONING AGENT FOR OLYMPIAD-LEVEL MATHEMATICAL PROBLEM SOLVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Reasoning Models (LRMs) have expanded the mathematical reasoning frontier through Chain-of-Thought (CoT) techniques and Reinforcement Learning with Verifiable Rewards (RLVR), capable of solving AIME-level problems. However, the performance of LRMs is heavily dependent on the extended reasoning context length. For solving ultra-hard problems like those in the International Mathematical Olympiad (IMO), the required reasoning complexity surpasses the space that an LRM can explore in a single round. Previous works attempt to extend the reasoning context of LRMs but remain prompt-based and built upon proprietary models, lacking systematic structures and training pipelines. Therefore, this paper introduces Intern-S1-MO, a long-horizon math agent that conducts multi-round hierarchical reasoning, composed of an LRM-based multi-agent system including reasoning, summary, and verification. By maintaining a compact memory in the form of lemmas, Intern-S1-MO can more freely explore the lemma-rich reasoning spaces in multiple reasoning stages, thereby breaking through the context constraints for IMO-level math problems. Furthermore, we propose OREAL-H, an RL framework for training the LRM using the online explored trajectories to simultaneously bootstrap the reasoning ability of LRM and elevate the overall performance of Intern-S1-MO. Experiments show that Intern-S1-MO can obtain 26 out of 35 points on the non-geometry problems of IMO2025, matching the performance of silver medalists. Code and model will be released to benefit future research.

1 INTRODUCTION

Reasoning is a highly intellectual human activity that requires the integration of deductive logic, pattern recognition, and creative problem decomposition to address complex challenges, which is regarded as a significant milestone towards Artificial General Intelligence (AGI) (Sun et al., 2025). In recent years, large reasoning models (LRMs) have made substantial progress in mathematical reasoning, driven primarily by techniques such as Chain-of-Thought (CoT) (Zhang et al., 2022; Wang et al., 2023) and Reinforcement Learning from Verifiable Rewards (RLVR) (Shao et al., 2024; Yue et al., 2025; Zeng et al., 2025). Along with the increasing reasoning capabilities of LRMs, a clear trend is that LRMs are being allocated more thinking budgets for more difficult problems to support the exploration of larger solution spaces and the trial-and-error processes (Zhou et al., 2022; Aggarwal & Welleck, 2025).

However, hardware and data limitations have made unlimited scaling of context length infeasible. Currently, state-of-the-art (SOTA) reasoning models typically support a maximum context length of only 64k or 128k tokens (Yang et al., 2025; Bai et al., 2025b; DeepMind, 2025a), insufficient for ultra-challenging problems such as those in International Mathematical Olympiads (IMO) (Balunovi'c et al., 2025). Figure 1(a) illustrates the logarithmic growth of the required context length with increasing difficulty of the problem, highlighting the mismatch between the existing capacity limits and practical demands. While resource investment can marginally raise this context ceiling, developing a cost-effective paradigm to meet context requirements is more compelling (Li et al., 2025a; Ke et al., 2025).

Some studies have explored multi-round interaction (Motwani et al., 2024) or parallel decoding (Zhang et al., 2024a) to perform long logical deduction in mathematical reasoning. Furthermore, Huang & Yang (2025) introduced self-reflective with prompt engineering, allowing models to identify

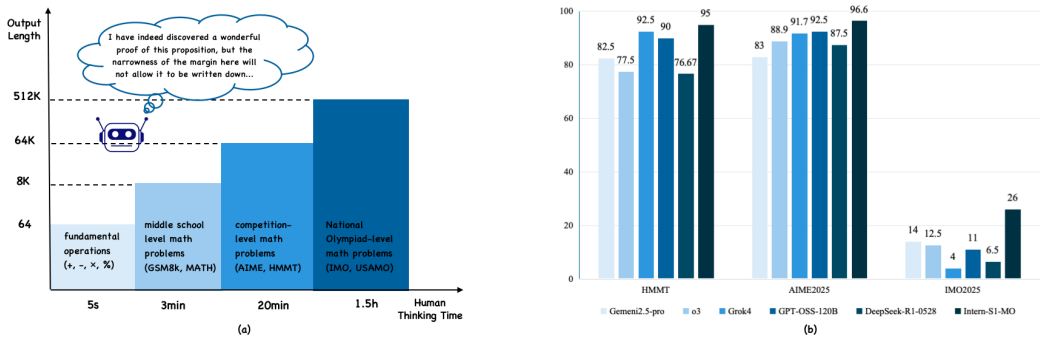

Figure 1: As problem difficulty increases, both the average human thinking time and the model token consumption per problem grow exponentially (a), already reaching concerning limits under current development trends. Intern-S1-MO enables LRMs to use about 512K tokens to solve a single problem, achieving state-of-the-art performance on challenging mathematical benchmarks (b).

flaws in intermediate reasoning steps and refine the outputs. Nevertheless, these approaches still confine problem-solving to a single reasoning cycle (even with internal iterations) rather than building cumulatively upon prior reasoning trajectories, which limits their capacity to leverage historical explorations for further in-depth deduction (Wang et al., 2025). Alternatively, formal language–based search (Ren et al., 2025; Chen et al., 2025; Zhou et al., 2025) shows some promise: by maintaining a structured repository to store and reuse intermediate results, they reduce reliance on model context length. However, the proof verification and state traversal demand extensive iterations, leading to high computational and search overhead. Moreover, formal systems require translating informal descriptions into formal logic, introducing additional costs and hindering the interaction between AI and humans.

Proprietary LRMs (OpenAI, 2025; DeepMind, 2025b) have reported impressive results on the International Mathematical Olympiad 2025 (IMO2025) problems, yet the research community lacks access to their methodologies and models. In this work, we present Intern-S1-MO, an open-source solution for building math reasoning agents unconstrained by context length, and solves complex reasoning problems through hierarchical decomposition, a strategy that closely aligns with human problem-solving patterns. Intern-S1-MO achieves unlimited exploration capability through lemma memory management. Specifically, after each single-round reasoning, the agent compresses its current reasoning history into concise sub-lemmas with a structured memory repository, which enables the agent to recover historical exploration outcomes in subsequent steps. We furthermore design process verification and revision mechanisms to certify the quality of the lemma repository. Notably, Intern-S1-MO enables adaptive control of its reasoning budget: it initiates multi-round exploration only for challenging tasks, ensuring efficient resource allocation.

To support the bootstrapping and online improvement of Intern-S1-MO, we additionally introduce the OREAL-H framework, enabling the agent to enhance its performance on complex problems with online reinforcement learning (RL). Starting from the basic formulation of Outcome Reward Reinforcement Learning (OREAL) (Lyu et al., 2025), OREAL-H exploits the additional reward signal produced by the outcome process verifier (OPV) that is continuous and accelerates training, and is modified for the Hierarchical Markov Decision Process (MDP) formulation to suit the multi-agent setting of Intern-S1-MO.

As a result, Intern-S1-MO establishes new state-of-the-art results across multiple mathematical reasoning benchmarks. As shown in Figure 1(b), on AIME2025 and HMMT2025, it achieves a 96.6% and 95% average pass@1 score, respectively. On the 5 non-geometry problems of International Mathematical Olympiads 2025 (IMO2025), Intern-S1-MO could obtain 26 out of 35 scores, surpassing the silver medalist-level performance (21) of humans. Additionally, we test Intern-S1-MO on CNMO2025, a new benchmark comprising 14 high-school math competition problems (excluding geometry problems) from the recently concluded China National Mathematics Olympiad 2025, on which Intern-S1-MO scores 232.4 out of 260 points. To facilitate rapid reproduction of our agent framework, we will open-source Intern-S1-mini-MO, a fine-tuned 8B model based on Intern-S1-mini (Bai et al., 2025a), specifically optimised for the entire multi-agent system. When integrated

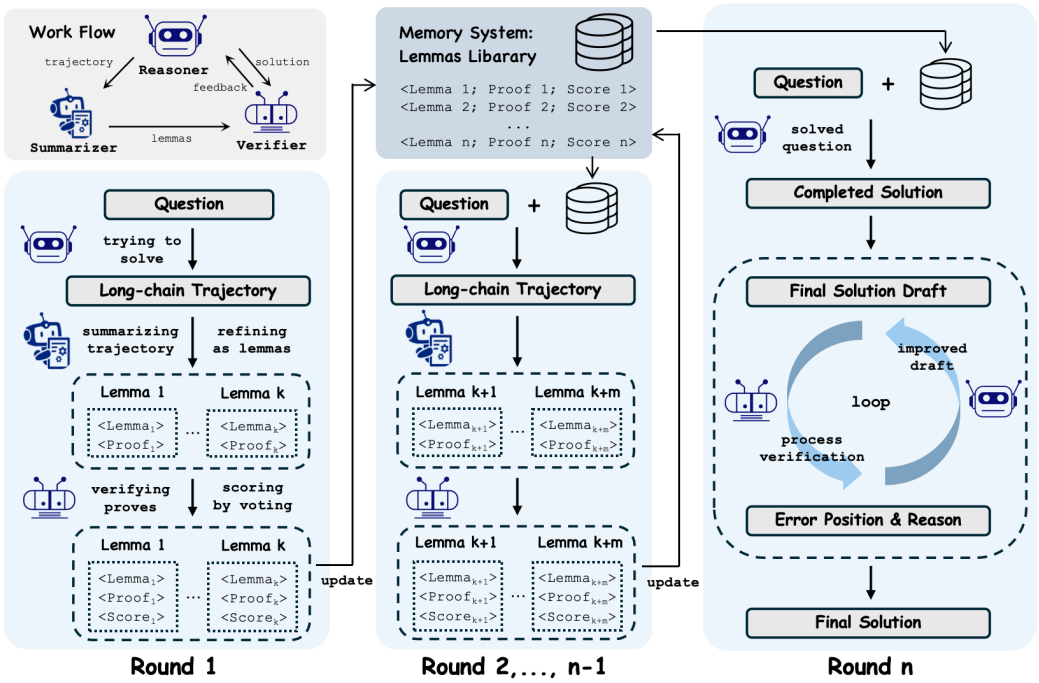

Figure 2: The agentic framwork of Intern-S1-MO.

with Intern-S1-MO, Intern-S1-mini-MO outperforms models at greater parameter scales, achieving a 90% pass@1 score on AIME2025 and completely solving 3 out of 5 non-geometric problems in IMO2025. Overall, our contributions are as follows:

- We explore multi-round complex reasoning scenarios and propose a multi-agent system, Intern-S1-MO, which effectively extends the reasoning depth of current LRMs by the lemma-based memory management.
- We contribute a RL framework, termed OREAL-H, for optimizing the performance of Intern-S1-MO on high-difficulty mathematical problems.
- The trained model, Intern-S1-mini-MO, and the multi-agent system will be open-sourced, laying a foundation for reproducibility and further research in the field of mathematical reasoning agents.

2 Building Hierarchical Math Agents

To extend the exploration of reasoning, we designed a hierarchical mathematical reasoning agent tailored for complex competition-level mathematical problems, as shown in Figure 2. By enabling recursive subproblem solving, it specifically addresses the aforementioned reasoning limitations constrained by context length.

Lemma Search via Sub-Problem Decomposition Decomposing complex problems into manageable sub-lemmas is a defining feature of human problem-solving for high-difficulty mathematics, as it breaks long-chain logical reasoning into incremental steps. We first observe that state-of-the-art models already exhibit a degree of reasonable decomposition capability for mathematical problems, though this ability is often undermined by a premature conclusion bias: when reasoning budgets are exhausted, models tend to rush toward incomplete or incorrect final answers instead of acknowledging partial progress. To mitigate this, we refine the model via prompt engineering and targeted training, explicitly enabling it to produce partial deductive progress in single-turn attempts (e.g., deriving intermediate sub-lemmas without forcing a full problem solution). This adjustment aligns the model's behavior with human iterative reasoning and lays the groundwork for cumulative exploration, the complete style requirements are presented in the Appendix A.

Summarizing Effective Exploration with Memory Maintenance The model's reasoning processes for complex problems often include redundant exploratory efforts and trial-and-error content. While this content aids in generating intermediate conclusions, it adds little value to subsequent deductive steps. Such facts enable us to extract only the essential components that drive progress, specifically, validated intermediate lemmas from each reasoning turn and store them in a structured lemma library. This library encourages the agent to reuse historical conclusions during new exploration rounds, allowing for deeper deductions based on prior lemmas rather than reprocessing redundant information. Notably, summarizing compelling exploration is as complex as the exploration process itself, as it requires distilling and checking the logical validity independently. Therefore, we allocate a dedicated reasoning turn after each exploration step to update the lemma library. This computational cost is necessary to ensure the library remains useful for long-chain reasoning.

Theorem Verifier to Mitigate Error Propagation Advanced reasoning models can self-reflect, but if they rely on erroneous historical premises, they will expend significant resources trying to validate questionable results. Such problem is compounded by error propagation, that a flawed intermediate conclusion can mislead subsequent deductive directions, leading to circular reasoning or invalid proofs. Fortunately, the verification of lemmas is comparatively more tractable than that of the complete problem. We address this by integrating a theorem verifier that uses parallel sampling to compute confidence scores for each lemma.

Process Verification for Complete Proof Verifying the validity of final solutions is crucial for obtaining reliable performance feedback, both in evaluation scenarios and reinforcement learning loops. To achieve this, we train a specialized process verifier using synthetic cold start data with outcome supervision. This process helps bootstrap verification capability and employs direct preference optimization (DPO) to align the verifier with the agent's output distribution. Evaluations demonstrate that this verifier achieves an F1-score greater than 85% on ProcessBench, surpassing the performance of o1-mini. In practice, the verifier serves two main functions: (1) enhancing robustness through test time scaling by aggregating verification results across multiple runs, and (2) providing high-quality feedback signals for reinforcement learning training to further optimize the agent's reasoning precision.

3 RL TRAINING FOR EVOLUTION OF MATH AGENTS

3.1 PRELIMINARIES

We model the agentic mathematical reasoning process as a *Hierarchical Markov Decision Process*, denoted $\mathcal{M} = \langle \mathcal{S}, \mathcal{U}, \mathcal{V}, r, R, \gamma \rangle$, where $\mathcal{S}$ is the state space (problem context + reasoning trace + verification feedback), $\mathcal{U}$ the high-level meta-action space (e.g., "extract lemmas", "invoke verification", "commit answer"), and $\mathcal{V}$ the low-level token vocabulary. The agent alternates between high-level decisions and low-level generation: at each round t, it executes a reasoning action u_t with token sequence $\boldsymbol{v}_t = (v_{t,1}, \ldots, v_{t,T_t}) \sim \pi_\theta^L(\cdot|s_t)$ to produce a reasoning segment. This output is summarized and verified by an external module, yielding natural language feedback which induces an intermediate proxy reward $r_t \in \mathbb{R}$. Upon termination after several rounds, a sparse final reward R indicates correctness of the solution. The training objective is to maximise expected final reward:

$$J(\theta, \phi) = \mathbb{E}_{\pi_\phi^H, \pi_\theta^L}[R]. \tag{1}$$

Leveraging the conditional structure of the hierarchical policy, the per-round advantage can be estimated via a high-level critic $V(s_t)$, updated to satisfy:

$$V(s_t) \leftarrow \mathbb{E}[r_t + \gamma V(s_{t+1})], \tag{2}$$

where s_{t+1} is the state after applying u_t. The advantage for round t is then $A_t = r_t + \gamma V(s_{t+1}) - V(s_t)$. On low-level, we can then perform an online policy gradient conditioned on this advantage, aggregating token-level log-likelihoods within the round:

$$\nabla_\theta J = \mathbb{E}\left[\sum_{t=1}^{K} A_t \cdot \sum_{\tau=1}^{T_t} \nabla_\theta \log \pi_\theta^L(v_{t,\tau} \mid s_t, v_{t,<\tau})\right], \tag{3}$$

Reward Function As mentioned in Section 2, we employ a Process Verifier (PV) to assess the logical rigor of complex mathematical proofs. Specifically, the PV examines the agent's final solution and outputs natural language feedback identifying the indices of steps containing logical fallacies. We estimate the PV's confidence via a multi-round voting mechanism. In particular, for problems amenable to outcome supervision, the final reward R is set to 0 if the final answer is incorrect. We further discuss the role of these supervision signals for RL steps in Section 3.3.

3.2 Cloning Success Trajectory for Cold Start

To prime the agent's adherence to structured reasoning formats and internalise the iterative agentic workflow, we initialize policies via behavioural cloning on filtered trajectories — retaining only rounds t where the output admits a well-formed lemma summary (e.g., syntactically valid, non-empty, logically segmented). Let $\mathcal{D}_{\text{init}} = \{(s_t, \boldsymbol{v}_t)\}$ denote such transitions. The token-level pretraining objective is:

$$\mathcal{L}_{\text{RFT}}(\theta) = -\mathbb{E}_{(s_t,\boldsymbol{v}_t)\sim\mathcal{D}_{\text{init}}}\left[\sum_{\tau=1}^{T_t} \log \pi_\theta^L(v_{t,\tau} \mid s_t, v_{t,<\tau})\right]. \tag{4}$$

Notably, we continuously augment $\mathcal{D}_{\text{init}}$ with question-answer pairs that are filtered by outcome-based scoring, without previous thinking. We observe that the model exhibits emergent generalization: patterns learned from these simplified trajectories boost agentic solving of the same problems, thereby improving the efficiency of positive trajectory discovery during online RL.

3.3 Oreal with Conjugate Reward under Process Judgement

We adopt the reinforcement learning framework of Oreal for policy optimization, and introduce two critical adaptations tailored to our Hierarchical MDP setting: (1) credit assignment across high-level reasoning actions is non-trivial due to delayed rewards; (2) the Process Verifier (PV) introduces a continuous, noisy reward signal that deviates from the binary outcome supervision assumed in RLVR setting.

Progress-Conditioned Advantage for Hierarchical Credit Assignment In multiround agentic reasoning, the naive estimation of trajectory-level advantage (e.g., $A_{\text{traj}} = R - V(s_0)$) disproportionately amplifies gradients from multi-round trajectories. To align optimization with meaningful reasoning progress, we anchor credit to rounds that yield verifiable advances: either extracting a well-formed lemma or committing a final answer.

Let $C_t \in \{0, 1\}$ denote whether round t produces such an advance (as determined by PV or syntactic structure). We define the high-level reward at round t as:

$$r_t^H = C_t \cdot \gamma^{T-t} R, \tag{5}$$

where T is the termination round and $\gamma \in (0, 1]$ is the discount factor (shared with Eq. (2)). This assigns full credit R to the final round ($t = T$), while earlier progress rounds receive exponentially discounted credit — reflecting their indirect contribution to the solution.

The high-level advantage is then estimated using a dedicated critic $V^H(s_t)$, updated via:

$$A_t^H = r_t^H + \gamma V^H(s_{t+1}) - V^H(s_t). \tag{6}$$

The total advantage driving policy updates is the sum over progress rounds:

$$A_{\text{total}}^H = \sum_{t=1}^{T} A_t^H. \tag{7}$$

This formulation ensures that only rounds contributing to reasoning progress influence the gradient — effectively decoupling optimization intensity from trajectory length. For example, a 10-round trajectory with 2 progress rounds receives comparable update magnitude to a 5-round trajectory with 2 progress rounds, mitigating bias toward verbosity.

Conjugate Reward Modeling for Noisy Process Verification Process Verification (PV) offers valuable insight into the internal logical consistency of a generated solution by subjecting its intermediate steps to multiple stochastic checks. However, unlike final-answer correctness—which is deterministic—PV feedback is inherently noisy: a solution passing k out of n verification rounds does not guarantee superior reasoning quality, as passes may arise from lucky sampling or superficial plausibility rather than deep correctness. Directly using the empirical ratio k/n as a reward signal risks amplifying this noise, leading to unstable or misguided policy updates that overfit to verification artifacts rather than genuine mathematical rigor.

To address this, we adopt a Bayesian perspective and model the latent reasoning quality $p \in [0, 1]$ as a random variable. We place a uniform prior $p \sim \text{Beta}(1, 1)$, encoding no initial assumption about solution validity. After observing k successful verifications in n independent PV trials, the conjugate Beta-Bernoulli update yields the posterior:

$$p \mid (k, n) \sim \text{Beta}(k + 1, n - k + 1). \tag{8}$$

Instead of using point estimates (e.g., posterior mean), we define the reward as the probability that this solution is *strictly better* than a canonical "completely invalid" baseline—one that fails all n checks ($k = 0$). Let $p_1 \sim \text{Beta}(k + 1, n - k + 1)$ represent the quality of the current solution and $p_0 \sim \text{Beta}(1, n + 1)$ that of the baseline. The reward is then:

$$R(k, n) = \mathbb{P}(p_1 > p_0) = \int_0^1 \int_0^1 \mathbb{I}(p_1 > p_0) \cdot f_{\text{Beta}(k+1,n-k+1)}(p_1) \cdot f_{\text{Beta}(1,n+1)}(p_0)\, dp_1 dp_0. \tag{9}$$

This formulation provides a principled, probabilistically calibrated reward that accounts for uncertainty in the verification process. It naturally suppresses spurious signals from low-pass outcomes while preserving strong gradients for high-confidence valid solutions.

In practice, we fix $n = 4$, balancing verification cost and signal fidelity. Under this setting, $R(4, 4) \approx 5.5$, corresponding to a 99.5% dominance probability over the $R(0, 4) = 0$ baseline, with smoothly interpolated rewards for intermediate cases ($k = 1, 2, 3$). A complete reward mapping is provided in Appendix 5. By grounding the reward in a relative, distributional comparison rather than raw counts, our conjugate reward model effectively denoises PV feedback, ensuring that policy optimization aligns with latent reasoning quality rather than stochastic verification artifacts. This enables stable and meaningful reinforcement learning even in the presence of imperfect process-level supervision.

4 EXPERIMENT

Implementation. We constructed and trained an agent-based reasoning framework, Intern-S1-MO, built upon the Intern-S1 architecture—a large-scale language model pre-trained on extensive mathematical corpora and further aligned for formal and informal reasoning. To bootstrap the system, we curated a diverse cold-start dataset spanning multiple difficulty tiers: it includes middle school-level problem-solving exercises, undergraduate coursework problems (e.g., from calculus, linear algebra, and discrete mathematics), and advanced competition questions drawn from national and international olympiads. This dataset comprises both *solution-based* problems (requiring a final numerical or symbolic answer) and *proof-based* problems (requiring structured logical arguments), ensuring broad coverage of mathematical reasoning patterns. From this pool, we selectively sampled the most challenging subset—particularly those involving multi-step deduction, non-trivial lemma synthesis, or ambiguous problem interpretation—as the foundation for reinforcement learning (RL). These RL samples were used to train the policy via our conjugate reward modeling mechanism (Section 2), enabling the agent to iteratively refine its reasoning trajectories. Subsequently, through knowledge distillation from the full Intern-S1-MO model, we derived a lightweight variant, Intern-S1-mini-MO, based on the smaller Intern-S1-mini backbone. This lite version preserves core reasoning capabilities while significantly reducing inference latency and memory footprint, making it suitable for resource-constrained deployment scenarios.

Table 1: Overall evaluation results for Intern-S1-MO and each baseline.

Model	HMMT	AIME2025	CNMO2025	IMO2025
Gemeni2.5-pro	82.5	83	157.5	14
o3-high	77.5	88.9	138.5	12.5
Grok4	92.5	91.7	84	4
GPT-OSS-120B	90	92.5	130	11
DeepSeek-R1-0528	76.67	87.5	113.5	6.5
Qwen3-235B-A22B	60.4	81.5	109	14
Intern-S1-mini-MO,	79.2	87.3	176.3	17
Intern-S1-MO	**95**	**96.6**	**232.4**	**26**

Evaluation. We evaluate our models on four representative mathematical benchmarks that collectively span the spectrum from advanced high school contests to elite olympiad-level challenges: AIME2025, HMMT (Harvard–MIT Mathematics Tournament), CNMO2025 (Chinese National Mathematical Olympiad), and IMO2025 (International Mathematical Olympiad). Following standard practice in mathematical AI evaluation, we exclude geometry problems from CNMO2025 and IMO2025 due to their heavy reliance on diagram interpretation and spatial reasoning—capabilities not natively supported in current text-only LLMs. To ensure fair and meaningful scoring, we adopt a point-aligned evaluation protocol inspired by MathArena: each problem is scored according to its original contest point value (e.g., 7 points per problem in IMO), rather than binary correctness. Full implementation details, including scoring rubrics and problem filtering criteria, are provided in Appendix 1. For each test instance, we perform 16 independent rollouts to account for stochasticity in generation. The primary metric is average `pass@1`—i.e., the expected score from the best single attempt—except for IMO2025, where we report `pass@4` to better reflect the high-variance nature of olympiad problem solving and align with community evaluation norms that allow multiple attempts in human competitions.

Baseline. We compare against a comprehensive suite of state-of-the-art reasoning models, encompassing both proprietary and open-source systems. These include: Gemeni2.5-pro, o3-high, Grok4, GPT-OSS-120B, DeepSeek-R1-0528, and Qwen3-235B-A22B. All baselines are evaluated under identical conditions—same prompts, same rollout count, same scoring rules—to ensure a fair comparison. This diverse set of baselines allows us to assess Intern-S1-MO's performance not only against general-purpose LRMs but also against models explicitly optimized for mathematical reasoning.

4.1 Overall Results

Notably, even the lightweight variant Intern-S1-mini-MO demonstrates remarkably strong performance—outperforming all baselines on CNMO2025 (176.3 vs. 157.5) and achieving a score of 17 on IMO2025, which exceeds the bronze-medal threshold. This suggests that the core architectural innovations—particularly the multi-round verification loop and hierarchical reasoning decomposition—are highly effective even when deployed on a smaller backbone, offering a favorable trade-off between capability and efficiency. The consistent outperformance of both Intern-S1-MO variants across all benchmarks further validates the generalizability of our framework beyond specific model scales.

The widening performance gap on more advanced benchmarks also reveals a qualitative shift in problem-solving behavior. On HMMT and AIME2025, many strong baselines can often produce correct answers through pattern matching or memorization of solution templates. However, CNMO2025 and IMO2025 problems typically require constructing novel arguments, introducing auxiliary constructions, or applying deep theoretical insights—tasks where rote recall fails. Intern-S1-MO excels precisely in these settings by maintaining a dynamic "reasoning memory" across rounds, allowing it to accumulate and refine partial insights (e.g., conjecturing a useful inequality or identifying an invariant) that would be lost in a single-pass generation.

To contextualize the IMO2025 result further: a score of 26 places the model within the top 10–15% of human contestants in recent years, surpassing the average national team member in many countries.

Table 2: Ablation study results.

Model	HMMT	AIME2025	CNMO2025
Single-tune with Agents	70.8	81.9	178.0
+ Multi-tune Reasoning	85.4	91.0	201.7
+ Lemma Verifier	86.3	93.3	203.0
+ Process Verifier	89.1	94.0	215.2
+ OReal-H	95.0	96.6	232.4

While it still falls short of gold-medal performance (More than 28 scores for non-geometric IMO2025), this represents a leap from prior AI systems, which rarely exceeded 10 points on non-geometry IMO problems. A preliminary error analysis shows that most failures occur on problems requiring highly non-standard transformations (e.g., functional equations with pathological solutions) or those where the key insight hinges on a single, elusive observation—a regime where even human experts often struggle without hints.

Together, these results underscore that scaling alone is insufficient for olympiad-level reasoning; instead, structured, verifiable, and iterative reasoning architectures are essential to bridge the gap between narrow competence and broad mathematical intelligence.

4.2 Ablation Study

To better understand the contribution of each key component in **Intern-S1-MO**, we conduct a systematic ablation study. The architecture of our method integrates several novel mechanisms—including multi-tune reasoning, lemma verification, process validation, and the OReal-H training objective—that collectively enable deep, iterative mathematical reasoning. However, it is crucial to disentangle their individual impacts to validate design choices and assess whether performance gains stem from architectural sophistication or synergistic interactions among modules. Therefore, we incrementally build up the full model from a simplified baseline ("Single-tune with Agents") and measure performance on HMMT, AIME2025, and CNMO2025—benchmarks that capture a gradient of reasoning complexity.

As shown in Table 2, the base configuration (Single-tune with Agents) already leverages agent-based problem decomposition but lacks iterative refinement, achieving moderate scores (70.8 on HMMT, 81.9 on AIME2025, and 178.0 on CNMO2025). Introducing *Multi-tune Reasoning*—which allows the model to revisit and refine intermediate steps across multiple reasoning rounds—yields substantial improvements (+14.6 on HMMT, +9.1 on AIME2025, +23.7 on CNMO2025), highlighting the importance of sustained exploration over one-shot inference. The addition of the *Lemma Verifier*, which validates generated sub-results for logical consistency, provides further gains, particularly on harder problems (e.g., +12.7 on CNMO2025), suggesting that error propagation is a critical bottleneck in long-horizon reasoning.

More notably, incorporating the *Process Verifier*—which evaluates the coherence and validity of the entire solution trajectory—leads to a pronounced jump on CNMO2025 (+12.2 points), indicating that global reasoning structure matters as much as local correctness. Finally, the integration of *OReal-H*, our hierarchical outcome-aligned reward mechanism that prioritizes both correctness and solution elegance, pushes performance to the final reported levels (95.0, 96.6, and 232.4). This last step not only refines answer accuracy but also encourages more human-like proof strategies, which is especially vital for olympiad-level problems where multiple valid approaches exist but only some are efficient or insightful.

Overall, the ablation study confirms that each component plays a non-redundant role, with the largest marginal gains coming from mechanisms that enforce reasoning discipline (verification) and strategic refinement (multi-tune + OReal-H). The cumulative effect demonstrates that high-level mathematical reasoning cannot be achieved by scaling alone—it requires explicit architectural support for iterative validation, hierarchical decomposition, and outcome-aware learning.

5 Related Work

5.1 Mathematical Reasoning Agents

Recent advancements in large reasoning models have significantly enhanced their performance on mathematical reasoning tasks; however, systematic exploration and reflection are still areas that require further investigation. A notable approach involves the use of tree search methods—such as Tree-of-Thoughts (Yao et al., 2023) and Monte Carlo Tree Search (Zhang et al., 2024a)—to facilitate parallel search during inference. While these methods broaden the search landscape, they often lack depth and struggle to effectively decompose complex problems(Sun et al., 2025; Balunovi'c et al., 2025). Other research has focused on augmenting LLMs with external tools to ground reasoning in computation or verified knowledge (Gou et al., 2023; Shao et al., 2024; **?**). Yet, these tools typically serve to enhance the existing reasoning process rather than fundamentally restructure it. More recent efforts propose structured reasoning frameworks that integrate planning, exploration, and reflection to iteratively refine solutions. These methods outperform standard chain-of-thought prompting on challenging problems, but they usually rely on carefully designed prompts and sometimes human-provided hints. Importantly, they shift reasoning from single-path generation to structured problem solving. Yet, training math agents—where exploration and reflection are optimized through learning signals—remains an emerging area. Recent initiatives have introduced structured reasoning frameworks that integrate exploration and reflection to iteratively refine solutions. These methods have been shown to outperform traditional methods on challenging problems. However, they often depend on meticulously crafted prompts and, at times, hints provided by humans. These nascent frameworks mark a shift from single-path generation in mathematical reasoning towards more structured agent solutions, and await further exploration into how to design and optimise the entire agent to enhance its performance.

5.2 Reinforcement Learning for Math Agents

Reinforcement learning (RL) for mathematical reasoning has primarily focused on outcome rewards, where feedback is based solely on final answer correctness. Despite this sparse signal, methods like ARTIST (Zhang et al., 2024b), ToRL (Li et al., 2025b), and rStar2-Agent (Shang et al., 2025) exhibit emergent agentic behaviors—such as adaptive tool use, self-correction, and context-aware reasoning. Scaling studies (e.g., ZeroTIR Mai et al. (2025)) further show that increased training effort leads to more sophisticated tool-integrated strategies. Nevertheless, current math agents remain limited: their decisions are mostly confined to choosing when to retry within a fixed reasoning template—rather than engaging in strategic planning or deep exploration. Critically, they lack summarization and cross-episode awareness. While approaches like TTRL (Zuo et al., 2025) and Satori (Shen et al., 2025) introduce basic reflection or meta-actions, they operate within isolated reasoning episodes and do not support cumulative knowledge transfer across inferences. Process-aware RL and verifier-guided training (e.g., Prover-Verifier Games (Kirchner et al., 2024)) aim to provide intermediate supervision with predefined rules or code execution, and are not well-suited for complex reasoning scenarios. In this paper, we use a process verifier to judge the rigor of natural language proofs, which provides a more flexible feedback signal.

6 Conclusion

This paper aims to address the critical bottleneck in large reasoning models (LRMs) for complex mathematical reasoning: the inherent limitation of context length, which has hindered progress in solving ultra-challenging tasks such as International Mathematical Olympiad (IMO) problems. To this end, this paper introduces Intern-S1-MO, an LRM-driven multi-agent system that conducts multi-round hierarchical reasoning, which conducts reasoning, summary, and verification at each round. By maintaining a compact memory in the form of lemmas, Intern-S1-MO can more freely explore the lemma-rich reasoning spaces in multiple reasoning rounds, which significantly extends the 64K constraints of LRMs by about 8 times. We further propose OREAL-H, an RL framework for training the LRM to simultaneously bootstrap the reasoning ability of the LRM and elevate the overall performance of Intern-S1-MO. Intern-S1-MO can now solve problems that require humans to think about 1.5 hours, which eventually obtains 26 out of 35 points on the non-geometry problems of IMO2025, matching the performance of silver medalists.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *ArXiv*, abs/2503.04697, 2025. URL https://api.semanticscholar.org/CorpusID:276813519.

Lei Bai, Zhongrui Cai, Yuhang Cao, Maosong Cao, Weihan Cao, Chiyu Chen, Haojiong Chen, Kai Chen, Pengcheng Chen, Ying Chen, Yongkang Chen, Yu Cheng, Pei Chu, Tao Chu, Erfei Cui, Ganqu Cui, Long Cui, Ziyun Cui, Nianchen Deng, Ning Ding, Nanqing Dong, Peijie Dong, Shihan Dou, Sinan Du, Haodong Duan, Caihua Fan, Ben Gao, Changjiang Gao, Jianfei Gao, Songyang Gao, Yang Gao, Zhangwei Gao, Jiaye Ge, Qiming Ge, Lixin Gu, Yuzhe Gu, Aijia Guo, Qipeng Guo, Xu Guo, Conghui He, Junjun He, Yili Hong, Siyuan Hou, Caiyu Hu, Hanglei Hu, Jucheng Hu, Ming Hu, Zhouqi Hua, Haian Huang, Junhao Huang, Xu Huang, Zixian Huang, Zhe Jiang, Lingkai Kong, Linyang Li, Peiji Li, Pengze Li, Shuaibin Li, Tianbin Li, Wei Li, Yuqiang Li, Dahua Lin, Junyao Lin, Tianyi Lin, Zhishan Lin, Hongwei Liu, Jiangning Liu, Jiyao Liu, Junnan Liu, Kai Liu, Kaiwen Liu, Kuikun Liu, Shichun Liu, Shudong Liu, Wei Liu, Xinyao Liu, Yuhong Liu, Zhan Liu, Yinquan Lu, Haijun Lv, Hongxia Lv, Huijie Lv, Qitan Lv, Ying Lv, Chengqi Lyu, Chenglong Ma, Jianpeng Ma, Ren Ma, Runmin Ma, Runyuan Ma, Xinzhu Ma, Yichuan Ma, Zihan Ma, Sixuan Mi, Junzhi Ning, Wenchang Ning, Xinle Pang, Jiahui Peng, Runyu Peng, Yu Qiao, Jiantao Qiu, Xiaoye Qu, Yuan Qu, Yuchen Ren, Fukai Shang, Wenqi Shao, Junhao Shen, Shuaike Shen, Chunfeng Song, Demin Song, Diping Song, Chenlin Su, Weijie Su, Weigao Sun, Yu Sun, Qian Tan, Cheng Tang, Huanze Tang, Kexian Tang, Shixiang Tang, Jian Tong, Aoran Wang, Bin Wang, Dong Wang, Lintao Wang, Rui Wang, Weiyun Wang, Wenhai Wang, Jiaqi Wang, Yi Wang, Ziyi Wang, Ling-I Wu, Wen Wu, Yue Wu, Zijian Wu, Linchen Xiao, Shuhao Xing, Chao Xu, Huihui Xu, Jun Xu, Ruiliang Xu, Wanghan Xu, GanLin Yang, Yuming Yang, Haochen Ye, Jin Ye, Shenglong Ye, Jia Yu, Jiashuo Yu, Jing Yu, Fei Yuan, Yuhang Zang, Bo Zhang, Chao Zhang, Chen Zhang, Hongjie Zhang, Jin Zhang, Qiaosheng Zhang, Qiuyinzhe Zhang, Songyang Zhang, Taolin Zhang, Wenlong Zhang, Wenwei Zhang, Yechen Zhang, Ziyang Zhang, Haiteng Zhao, Qian Zhao, Xiangyu Zhao, Xiangyu Zhao, Bowen Zhou, Dongzhan Zhou, Peiheng Zhou, Yuhao Zhou, Yunhua Zhou, Dongsheng Zhu, Lin Zhu, and Yicheng Zou. Intern-s1: A scientific multimodal foundation model, 2025a. URL https://arxiv.org/abs/2508.15763.

Lei Bai, Zhongrui Cai, Yuhang Cao, Maosong Cao, Weihan Cao, Chiyu Chen, Haojiong Chen, Kaiming Chen, Pengcheng Chen, Ying Chen, Yongkang Chen, Yu Cheng, Pei Chu, Tao Chu, Erfei Cui, Ganqu Cui, Long Cui, Ziyun Cui, Nianchen Deng, Ning Ding, Nanqing Dong, Peijie Dong, Shi-Hua Dou, Si na Du, Haodong Duan, Caihua Fan, Ben Gao, Changjiang Gao, Jianfei Gao, Songyang Gao, Yang Gao, Zhangwei Gao, Jiaye Ge, Qiming Ge, Lixin Gu, Yuzhe Gu, Aijia Guo, Qipeng Guo, Xu Guo, Conghui He, Junjun He, Yili Hong, Siyuan Hou, Caiyu Hu, Han-Hwa Hu, Jucheng Hu, Mingxue Hu, Zhouqi Hua, Haian Huang, Junhao Huang, Xuantuo Huang, Zixian Huang, Zhe Jiang, Lingkai Kong, Linyang Li, Peijin Li, Pengze Li, Shuaibin Li, Tian-Xin Li, Wei Li, Yuqiang Li, Dahua Lin, Junyao Lin, Tianyi Lin, Zhishan Lin, Hong wei Liu, Jiangning Liu, Jiyao Liu, Junnan Liu, Kaiwen Liu, Kaiwen Liu, Kuikun Liu, Shichun Liu, Shi Yuan Liu, Shudong Liu, Wei Liu, Xinyao Liu, Yuhong Liu, Zhan Liu, Yinquan Lu, Haijun Lv, Hong Lv, Huijie Lv, Qitan Lv, Ying Lv, Chengqi Lyu, Chenglong Ma, Jian-Kai Ma, Ren Ma, Runmin Ma, Runyuan Ma, Xinzhu Ma, Yi dan Ma, Zihan Ma, Sixuan Mi, Junzhi Ning, Wenchang Ning, Xinle Pang, Jiahui Peng, Runyu Peng, Yu Qiao, Jia-Ming Qiu, Xiaoye Qu, Yuanbin Qu, Yuchen Ren, Fukai Shang, Wenqi Shao, Junhao Shen, Shuaike Shen, Chun-Dong Song, Demin Song, Diping Song, Chenlin Su, Weijie Su, Weigao Sun, Yu Sun, Qian Tan, Cheng Tang, Huanze Tang, Ke Kerri Tang, Shixiang Tang, Jian Tong, Aoran Wang, Bin Wang, Dong Wang, Lintao Wang, Rui Wang, Weiyun Wang, Wenhai Wang, Jiaqi Wang, Yi Wang, Ziyi Wang, Ling-I Wu, Wenzheng Wu, Yue Wu, Zijian Wu, Li-Yi Xiao, Shu-Qiao Xing, Chao Xu, Huihui Xu, Jun Xu, Ruiliang Xu, Wanghan Xu, Ganlin Yang, Yuming Yang, Hao nan Ye, Jin Ye, Shenglong Ye, Jia Yu, Jiashuo Yu, Jing Yu, Fei Yuan, Yu Zang, Bo Zhang, ChaoBin Zhang, Chen Zhang, Hongjie Zhang, Jin Zhang, Qiao-Xuan Zhang, Qiuyinzhe Zhang, Songyang Zhang, Taolin Zhang, Wenlong Zhang, Wenwei Zhang, Yechen Zhang, Ziyang Zhang, Haiteng Zhao, Qian Zhao, Xiangyu Zhao, Bowen Zhou, Dongzhan Zhou, Peiheng Zhou, Yuhao Zhou, Yun-Yi Zhou, Dongsheng Zhu, Lin Zhu, and Yi Zou. Intern-s1: A scientific multimodal foundation model. *ArXiv*, abs/2508.15763, 2025b. URL https://api.semanticscholar.org/CorpusID:280710453.

Mislav Balunovi'c, Jasper Dekoninck, Ivo Petrov, Nikola Jovanovi'c, and Martin T. Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *ArXiv*, abs/2505.23281, 2025. URL https://api.semanticscholar.org/CorpusID:278996037.

Luoxin Chen, Jinming Gu, Liankai Huang, Wenhao Huang, Zhicheng Jiang, Allan Jie, Xiaoran Jin, Xing Jin, Chenggang Li, Kaijing Ma, et al. Seed-prover: Deep and broad reasoning for automated theorem proving. *arXiv preprint arXiv:2507.23726*, 2025.

Google DeepMind. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *ArXiv*, abs/2507.06261, 2025a. URL https://api.semanticscholar.org/CorpusID:280151524.

Google DeepMind. Advanced version of gemini with deep think officially achieves gold-medal standard at the international mathematical olympiad, 2025b. URL https://deepmind.google/discover/blog/advanced-version-of-gemini-with-deep-think-officially-achieves-gold-medal-standard

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. Tora: A tool-integrated reasoning agent for mathematical problem solving. *ArXiv*, abs/2309.17452, 2023. URL https://api.semanticscholar.org/CorpusID:263310365.

Yichen Huang and Lin F Yang. Gemini 2.5 pro capable of winning gold at imo 2025. *arXiv preprint arXiv:2507.15855*, 2025.

Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, PeiFeng Wang, Silvio Savarese, Caiming Xiong, and Shafiq Joty. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://api.semanticscholar.org/CorpusID:277781085.

Jan Hendrik Kirchner, Yining Chen, Harri Edwards, Jan Leike, Nat McAleese, and Yuri Burda. Prover-verifier games improve legibility of llm outputs, 2024. URL https://arxiv.org/abs/2407.13692.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *ArXiv*, abs/2504.21776, 2025a. URL https://api.semanticscholar.org/CorpusID:278207550.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Torl: Scaling tool-integrated rl. *arXiv preprint arXiv:2503.23383*, 2025b.

Chengqi Lyu, Songyang Gao, Yuzhe Gu, Wenwei Zhang, Jianfei Gao, Kuikun Liu, Ziyi Wang, Shuaibin Li, Qian Zhao, Haian Huang, et al. Exploring the limit of outcome reward for learning mathematical reasoning. *arXiv preprint arXiv:2502.06781*, 2025.

Xinji Mai, Haotian Xu, Zhong-Zhi Li, Xing W, Weinong Wang, Jian Hu, Yingying Zhang, and Wenqiang Zhang. Agent rl scaling law: Agent rl with spontaneous code execution for mathematical problem solving, 2025. URL https://arxiv.org/abs/2505.07773.

Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Markian Rybchuk, Philip Torr, Ivan Laptev, Fabio Pizzati, Ronald Clark, and Christian Schröder de Witt. Malt: Improving reasoning with multi-agent llm training. *ArXiv*, abs/2412.01928, 2024. URL https://api.semanticscholar.org/CorpusID:274446212.

OpenAI. Openai imo 2025 proofs, 2025. URL https://github.com/aw31/openai-imo-2025-proofs.

Z. Z. Ren, Zhihong Shao, Jun-Mei Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, Z. F. Wu, Zhibin Gou, Shirong Ma, Hongxuan Tang, Yuxuan Liu, Wenjun Gao, Daya Guo, and Chong Ruan. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *ArXiv*, abs/2504.21801, 2025. URL https://api.semanticscholar.org/CorpusID:278207693.

Ning Shang, Yifei Liu, Yi Zhu, Li Lyna Zhang, Weijiang Xu, Xinyu Guan, Buze Zhang, Bingcheng Dong, Xudong Zhou, Bowen Zhang, Ying Xin, Ziming Miao, Scarlett Li, Fan Yang, and Mao Yang. rstar2-agent: Agentic reasoning technical report, 2025. URL https://arxiv.org/abs/2508.20722.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300, 2024. URL https://api.semanticscholar.org/CorpusID:267412607.

Chuming Shen, Wei Wei, Xiaoye Qu, and Yu Cheng. Satori-r1: Incentivizing multimodal reasoning with spatial grounding and verifiable rewards. *arXiv preprint arXiv:2505.19094*, 2025.

Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, et al. A survey of reasoning with foundation models: Concepts, methodologies, and outlook. *ACM Computing Surveys*, 57(11):1–43, 2025.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Annual Meeting of the Association for Computational Linguistics*, 2023. URL https://api.semanticscholar.org/CorpusID:258558102.

Pengyuan Wang, Tian-Shuo Liu, Chenyang Wang, Yidi Wang, Shu Yan, Cheng-Xing Jia, Xu-Hui Liu, Xin-Wei Chen, Jia-Cheng Xu, Ziniu Li, and Yang Yu. A survey on large language models for mathematical reasoning. *ArXiv*, abs/2506.08446, 2025. URL https://api.semanticscholar.org/CorpusID:279261286.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Jingren Zhou, Junyan Lin, Kai Dang, Keqin Bao, Ke-Pei Yang, Le Yu, Li-Chun Deng, Mei Li, Min Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shi-Qiang Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *ArXiv*, abs/2505.09388, 2025. URL https://api.semanticscholar.org/CorpusID:278602855.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *ArXiv*, abs/2305.10601, 2023. URL https://api.semanticscholar.org/CorpusID:258762525.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *ArXiv*, abs/2504.13837, 2025. URL https://api.semanticscholar.org/CorpusID:277940134.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *ArXiv*, abs/2503.18892, 2025. URL https://api.semanticscholar.org/CorpusID:277940848.

Dan Zhang, Sining Zhoubian, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *ArXiv*, abs/2406.03816, 2024a. URL https://api.semanticscholar.org/CorpusID:270285630.

Jianyi Zhang, Yufan Zhou, Jiuxiang Gu, Curtis Wigington, Tong Yu, Yiran Chen, Tong Sun, and Ruiyi Zhang. Artist: Improving the generation of text-rich images with disentangled diffusion models and large language models, 2024b. URL https://arxiv.org/abs/2406.12044.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alexander J. Smola. Automatic chain of thought prompting in large language models. *ArXiv*, abs/2210.03493, 2022. URL https://api.semanticscholar.org/CorpusID:252762275.

Denny Zhou, Nathanael Scharli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. *ArXiv*, abs/2205.10625, 2022. URL https://api.semanticscholar.org/CorpusID:248986239.

Yichi Zhou, Jianqiu Zhao, Yongxin Zhang, Bohan Wang, Siran Wang, Luoxin Chen, Jiahui Wang, Haowei Chen, Allan Jie, Xinbo Zhang, Haocheng Wang, Luong Ngoc Trung, Rong Ye, Phan Nhat Hoang, Huishuai Zhang, Peng Sun, and Hang Li. Solving formal math problems by decomposition and iterative reflection. *ArXiv*, abs/2507.15225, 2025. URL https://api.semanticscholar.org/CorpusID:280271297.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

THE USE OF LARGE LANGUAGE MODELS (LLMS)

We used LLMs solely for language polishing. The scientific ideas, methodology, analyses, and conclusions were entirely developed by the authors, while the LLMs assisted only in improving clarity and readability of the text.

A SYSTEM PROMPTS FOR MATH AGENTS

A.1 LEMMA SEARCH

Listing 1: LEMMA SEARCH

```
**Objective:**
Your task is to provide a rigorous mathematical proof and solution for
    ↪ the given problem. The problem is expected to be challenging. Your
    ↪ primary goal is to demonstrate a deep and correct understanding of
    ↪ the problem through logical, step-by-step reasoning.

**Guiding Principles:**

1.  **Rigor is Paramount:**
    *   Every step in your proof must be logically sound and clearly
    ↪ justified.
    *   The final answer is secondary to the correctness of the
    ↪ derivation. A correct answer resulting from a flawed or incomplete
    ↪ proof will be considered a failure.

2.  **Embrace Partial Solutions:**
    *   It is understood that a complete solution may not be found in a
    ↪ single attempt.
    *   If you cannot provide a complete solution, you must provide any
    ↪ significant partial results that you can prove with full rigor.
    *   **Do not guess or provide solutions with logical gaps.** Instead,
    ↪  focus on what you *can* prove.
    *   Examples of valuable partial results include:
        *   Proving a key lemma.
        *   Solving one or more cases of a proof by cases.
        *   Establishing a critical property of the mathematical objects
    ↪ involved.
        *   For an optimization problem, proving an upper or lower bound.
    *   Clearly state which parts of the problem you have solved and
    ↪ which remain open. Acknowledging the limits of your solution is a
    ↪ critical part of the task.

3.  **Mathematical Formatting:**
    *   All mathematical variables, expressions, equations, and relations
    ↪  must be formatted using TeX. For example: `Let $G$ be a group and
    ↪ let $H$ be a subgroup of $G$.`

**Output Format:**
Your response MUST be structured into the following sections, in this
    ↪ exact order.

---

**1. Summary**

**a. Verdict:**
*   Begin by stating clearly whether you have found a complete or a
    ↪ partial solution.
*   **For a complete solution:** State the final answer. (e.g., "I have
    ↪ found a complete solution. The answer is...")
```

```
*   **For a partial solution:** State the main rigorous conclusion(s) you
    ↪ have proven. (e.g., "I have not found a complete solution, but I
    ↪ have rigorously proven that...")

**b. Method Sketch:**
*   Provide a high-level, conceptual outline of your logical argument.
    ↪ This should be clear enough for an expert to grasp your approach
    ↪ without reading the full proof.
*   Include:
    *   A narrative of your overall strategy.
    *   The full and precise mathematical statements of any key lemmas or
    ↪ major intermediate results you proved.
    *   A description of any key constructions or case splits that form
    ↪ the backbone of your argument.

**2. Detailed Solution**

*   Present the full, step-by-step mathematical proof of your results.
*   This section should contain *only* the rigorous proof itself, free
    ↪ from any commentary, reflections on your process, or alternative
    ↪ approaches you considered.
*   The level of detail must be sufficient for an expert to verify the
    ↪ correctness of your reasoning without needing to fill in any gaps.
```

A.2 REASONER

Listing 2: Memory Management

```
You are a top-tier mathematical research assistant, proficient in the
    ↪ logical analysis and argumentation of high-level competitive
    ↪ mathematics.

Your core task is to conduct an in-depth analysis of a solution approach
    ↪ generated by a large language model for problems at the
    ↪ International Mathematical Olympiad (IMO) level, identifying and
    ↪ extracting all key lemmas.

During this analysis, you must rigorously distinguish between
    ↪ propositions **newly proposed** by the model and **universal lemmas
    ↪ ** already provided by us. Your final output **shall only contain**
    ↪  those lemmas appearing in the model's solution approach but not
    ↪ provided in the universal lemma repository.

**The input comprises three sections:**
1.  `### Problem ###`: The mathematical problem requiring resolution.
2. `### Provided Lemmas ###`: A set of known, verified lemmas for
    ↪ reference during problem-solving.
3. `### Model's Thinking Process ###`: The reasoning generated by the
    ↪ large language model to solve the problem.

**Your output must adhere to the following principles and format:**

#### **A. Extraction Principles**

1.  **Novelty**: Extract only lemmas that are first introduced or proven
    ↪ within the `Model's Thinking Process`. If the model utilises a
    ↪ lemma from the `Provided Lemmas`, do not include it in your output.
2.  **Classification**: Categorise extracted new lemmas into two types:
    *   **Proven Lemmas**: Propositions explicitly stated or implicitly
    ↪ utilised within the `Model's Problem-Solving Approach`, accompanied
    ↪  by complete or core proof steps.
    *   **Unproven Lemmas**: Propositions claimed, relied upon, or
    ↪ treated as critical assumptions within the `Model Solution Approach
    ↪ `, but for which **no valid proof is provided**.
```

```
#### **B. Strict Formatting Requirements**

Your output must strictly adhere to the following Markdown and LaTeX formatting.

1.  **Format for Proven Lemmas:**
    *   All **proven lemmas** and their proofs must be contained entirely within a single `\boxed{}` environment.
    *   Use `---` horizontal rules to separate distinct lemmas.
    *   Each lemma begins with `**Lemma X (Lemma X):**`, where `X` is a positive integer numbering.
    *   The statement of the lemma should use concise, formal mathematical language, employing LaTeX where appropriate.
    *   This is immediately followed by the proof, beginning with `**Proof X (Proof X):**`.
    *   Each step of the proof begins with an unordered list `*` and is prefixed with `**Step Y (Step Y):**`.

2.  **Format for Unproven Lemmas:**
    *   All **unproven lemmas** must be placed entirely within a separate `\boxed{}` environment.
    *   Each lemma begins with `**Lemma X (Lemma X):**`.
    *   If all critical steps of the model are already provided in the `Historical Lemmas Repository` or sufficiently proven within its own content (i.e., **no novel unproven lemmas are discovered**), place `**Lemma -1 (Lemma -1)**` within that box.

---

### Problem ###
{Problem}

### Provided Lemmas ###
Lemma 1:
Proof 1:...

Lemma n:
Proof n:

### Model's Thinking Process ###
{Thinking}

---

####`DESIREDOUTPUT:
```\boxed{
lemma n+1:{lemma n+1}
proof n+1:
*step 1:{step 1}
*step 2:{step 2}
*step 3:{step 3}
...

lemma n+2:{lemma N=2}
proof2:
*step 1:{step 1}
...}
\boxed{
withoutproof:
lemma -1}
```

Translated with DeepL.com (free version)
```