K2-Think: A Parameter-Efficient Reasoning System

Zhoujun Cheng*, Richard Fan*, Shibo Hao*, Taylor W. Killian*,o, Haonan Li*, Suqi Sun*
Hector Ren, Alexander Moreno, Daqian Zhang, Tianjun Zhong, Yuxin Xiong, Yuanzhe Hu, Yutao Xie
Xudong Han, Yuqi Wang, Varad Pimpalkhute, Yonghao Zhuang, Aaryamonvikram Singh, Xuezhi Liang
Anze Xie, Jianshu She, Desai Fan, Chengqian Gao, Liqun Ma, Mikhail Yurochkin, John Maggs
Xuezhe Ma, Guowei He, Zhiting Hu, Zhengzhong Liu*,o, Eric P. Xing[o]

Institute of Foundation Models, Mohamed bin Zayed University of Artificial Intelligence

*Core Contributors (listed alphabetically), [o]Corresponding Authors

We introduce K2-Think, a reasoning system that achieves frontier performance with just a 32B parameter model — surpassing or matching much larger models such as GPT-OSS 120B and DeepSeek v3.1. Built on the Qwen2.5 base model, our system demonstrates that smaller models can compete at the highest levels through synergistic combination of advanced post-training and test-time computation techniques. Our approach is built on top of six key technical pillars: Long Chain-of-thought Supervised Finetuning, Reinforcement Learning with Verifiable Rewards (RLVR), Agentic planning prior to reasoning, Test-time Scaling, Speculative Decoding, and Inference-optimized Hardware, using only publicly available open-source datasets. K2-Think prioritizes mathematical reasoning, achieving state-of-the-art scores on public benchmarks for open source models, while also maintaining strong performance on other domains such as Code and Science. Our results validate that a more parameter-efficient model like K2-Think 32B can rival state-of-the-art systems through an integrative post-train recipe including long chain-of-thought training and strategic inference-time enhancements, paving the way for more accessible and affordable open-source reasoning systems. We have made K2-Think freely available at k2think.ai demonstrating best-in-class inference speeds, through the Cerebras Wafer-Scale Engine, delivering upwards of 2,000 tokens per second per request.

K2 Think

🤗 K2-Think (Model)	huggingface.co/LLM360/K2-Think
K2-Think (Code)	github.com/MBZUAI-IFM/K2-Think-SFT github.com/MBZUAI-IFM/K2-Think-Inference
K2-Think (Web)	k2think.ai

[o]Correspondence to: {Eric.Xing,Hector.Liu,Taylor.Killian}@mbzuai.ac.ae

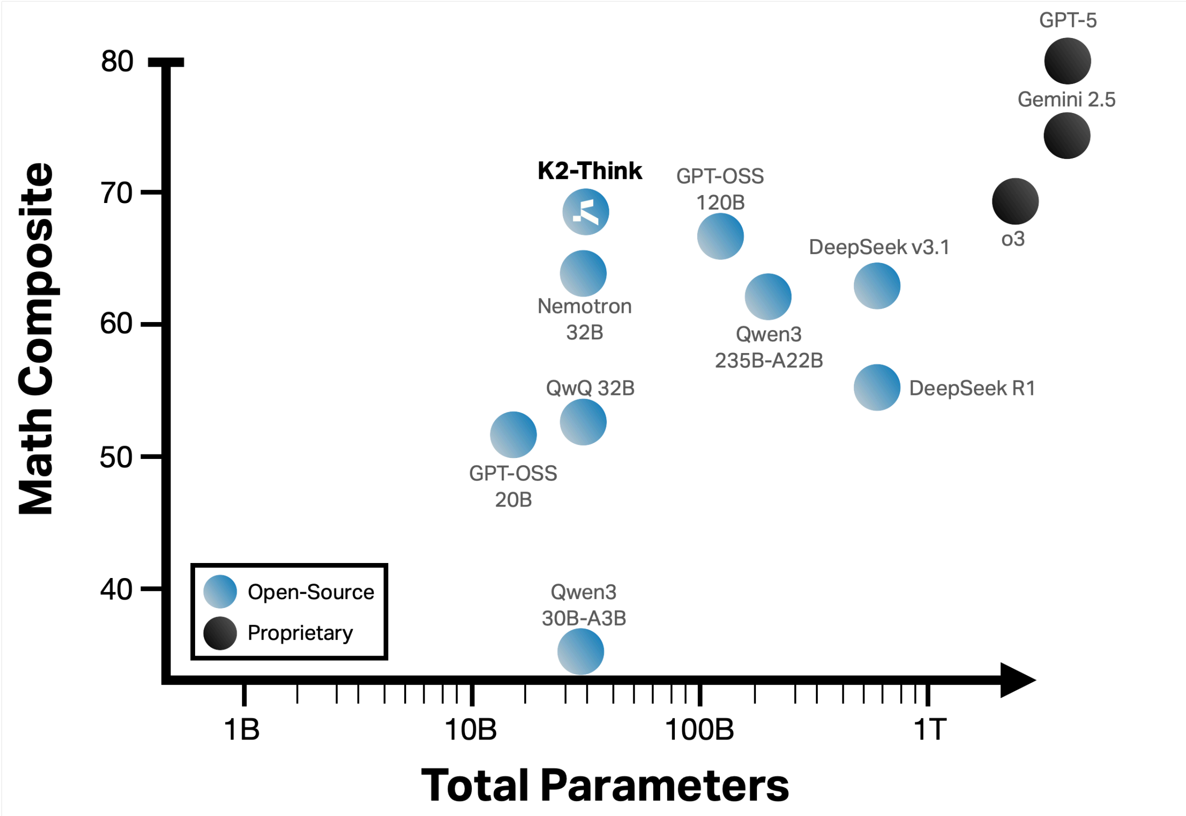

Figure 1: K2-THINK exhibits remarkable parameter efficiency, providing comparable or superior performance to frontier reasoning models in complex math domains with an order of magnitude smaller model. The composite score here is the micro-average for each model over four complex math benchmarks, weighted by the number of questions in each benchmark (AIME 2024, AIME 2025, HMMT 2025, and Omni-MATH-HARD; see Section 3 for the benchmark details). Note: parameter counts for proprietary models are speculative.

1 Introduction

Recent advances in frontier reasoning models have highlighted the effectiveness of long chain-of-thought reasoning, enabled by large-scale supervised fine-tuning and reinforcement learning. Systems like OpenAI-O3 (OpenAI, 2025) and Gemini 2.5 (Google DeepMind, 2025) have achieved strong results on competition-level math benchmarks, complex coding tasks, and advanced scientific reasoning datasets, setting new milestones for reasoning-centered language models. These developments have also stimulated further exploration in the open-source community, where researchers have trained competitive reasoning systems with reinforcement learning (Yu et al., 2025; Hu et al., 2025; Wang et al., 2025c) and investigated mechanisms by which RL improves reasoning (Zeng et al., 2025; Yue et al., 2025; Shao et al., 2025; Agarwal et al., 2025b; Wang et al., 2025b).

In this report we introduce K2-THINK: a competitive reasoning system built from the open-weight Qwen2.5-32B base model (Yang et al., 2024a). We break down our system into stages, including post-training and test-time, where an integrative recipe involving 6 major technical innovations was introduced over all stages spanning finetuning, reinforcement learning, planning, and hardware optimization to boost the base model reasoning capability, and we evaluated how each stage affects performance. **These components combine to enable a model of merely 32 billion parameters, with modest test-time compute, to match the mathematical reasoning performance of proprietary frontier models.** In fact, K2-THINK emerges as the top open-source model for complex math benchmarks matching or exceeding previously leading models that are orders of magnitude greater in size. Figure 1 presents a plot of the global micro-average of performance (essentially dividing the total number of correct answers by the total number of questions across all test sets) of each model over four challenging math competition tasks with respect to the total number of parameters for each model. The prominent positioning of K2-THINK in the top-left visually depicts its superior parameter efficiency, demonstrating that it achieves State-of-the-Art performance among open-source models with a significantly smaller total parameter count. Detailed results and discussion for the benchmarks are presented in Section 3.

More specifically, K2-THINK incorporates six key innovations to deliver a strong reasoning system. We first extend the base model with *chain-of-thought capabilities through Supervised Fine-tuning (SFT)*, followed by *Reinforcement Learning with Verifiable Rewards (RLVR)* to strengthen reasoning performance. We then enhance the model with inference-time techniques: *agentic planning* and *test-time scaling* using *Best-of-N sampling*. Finally, we deploy K2-THINK with two speed optimizations: *speculative decoding* and Cerebras' Wafer-Scale Engine, an *inference-optimized hardware* system. This final stage enables the model to deliver its powerful chain-of-thought reasoning capabilities with near-instantaneous response times, deployed at speeds upwards of 2000 tokens per second per user request.

With the release of K2-THINK, we share our experience and make available an important advancement in open-source language modeling, that aggressive post-train engineering and test-time computation, even with a modest commodity pretrained base model, can significantly boost reasoning capabilities in cost-effective manners. Prior studies have reported that, in certain regimes, allocating more computation during inference can be more cost-effective than scaling model size (Snell et al., 2025); for recent frontier systems—including OpenAI's o1/o3 (OpenAI, 2024, 2025), DeepSeek-R1 (Guo et al., 2025), Google's Gemini 2.5 (Google DeepMind, 2025), and xAI's Grok4 (xAI, 2025) — model capabilities have been claimed to improve with increased test-time budgets (Ji et al., 2025a; Yang et al., 2025b).

In addition to releasing code and model weights, **we offer K2-THINK through a public website and as a production-ready API endpoint.**[1] This allows the community to engage directly with a living system, shifting the emphasis from static artifacts to a deployable, studyable service that can be stress-tested and iterated on in the open. As dynamic inference-time reasoning becomes more complex, our API demonstrates the requirements of sophisticated systems for top performance, and provides an operational deployment delivering robustness, safety, and efficiency under real-world constraints.

In Section 2 we describe the development process and deployment of K2-THINK, using the Cerebras Wafer-Scale Engine. Section 3 presents a thorough set of evaluations and ablations that attribute gains across post-training and test-time computation. Section 4 situates our contributions within the literature. We conclude in Section 5 with a summary overview, discuss our motivations for deploying this model, and chart future directions for extending reasoning performance with openly released models and deployment-ready systems.

2 K2-THINK Development

We initiated K2-THINK's development to study a complete post-training recipe for enhanced reasoning and establish best practices for extending our in-house foundation models. Throughout this study, we sought to validate published best-practices as well as test original test-time computation ideas.

We chose to fine-tune a 32B-scale base model for K2-THINK, as:

(1) it allows for fast iteration while providing strong base capabilities and

[1] available upon request

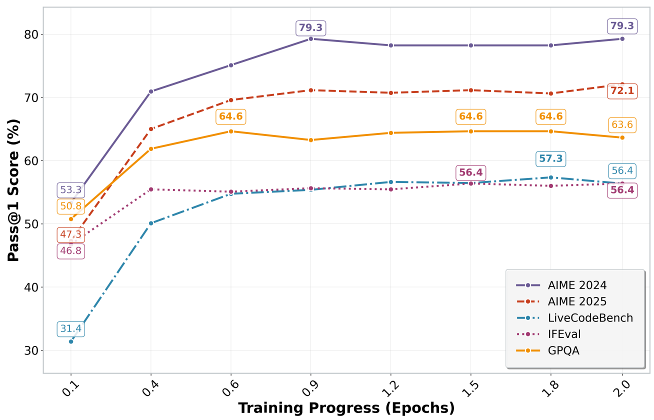

Figure 2: Pass@1 performance over training. Pass@1 of K2-THINK-SFT across five benchmarks; the x-axis is training progress (epochs), the y-axis is pass@1 score.

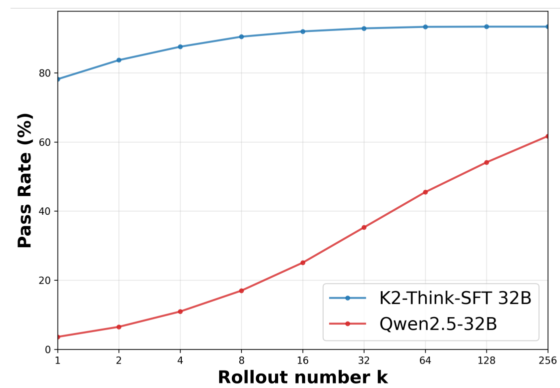

Figure 3: Pass@k on AIME2024. Pass@k of K2-THINK-SFT and the Qwen-2.5 32B base model; the x-axis is the number of rollouts per question, the y-axis is pass rate.

(2) its size suits both research and consumer computation frameworks.

Specifically, we selected Qwen2.5-32B as it is not tuned for reasoning, allowing us to fully validate our recipe's effectiveness.

2.1 Phase 1: Supervised Fine Tuning

The initial stage of K2-THINK development constitutes supervised fine-tuning (SFT) of the base model using curated long chain-of-thoughts (CoT), establishing the first pillar of our complete reasoning system. This follows the paradigm introduced by DeepSeek in the development of their R1 model (Guo et al., 2025). This phase of training serves to provide guidance to the pre-trained base language model for generating structured responses to complex queries. Additionally, the model is trained to adopt an expected output format in which the model's reasoning process is made clear prior to producing an answer. By providing a token-by-token supervisory signal through extended CoT, the base model's intrinsic computation capabilities are expanded substantially (Wei et al., 2022; Schuurmans et al., 2024).

Our SFT phase[2] utilized the existing `AM-Thinking-v1-Distilled` dataset,[3] composed of CoT reasoning traces and instruction/response pairs, with prompts drawn from tasks spanning mathematical reasoning, code generation, scientific reasoning, instruction following, and general chat (Ji et al., 2025b; Tian et al., 2025). In what follows, we will simply refer to this supervised fine-tuned model as **K2-THINK-SFT**.

2.1.1 Observations

Our SFT experiments on Qwen2.5-32B yielded several practical insights. Of particular note, we conducted a step-wise evaluation of K2-THINK-SFT across five representative benchmarks. As shown in Figure 2, performance improves rapidly within the first third of training (roughly 0.5 epoch), particularly on mathematics benchmarks (AIME 2024 and AIME 2025). After this sharp initial gain, most benchmarks plateau, with AIME 2024 stabilizing around 79.3% pass@1 and AIME 2025 around 72.1%. GPQA and IFEval continue to exhibit modest upward trends, while LiveCodeBench shows a slower but steady improvement up to 56.4%. We observe that our SFT phase has reached convergence, with the model exhibiting diminishing returns to continued training on the dataset.

[2] Code, based on LLaMA-Factory, for SFT can be found at `https://github.com/MBZUAI-IFM/K2-Think-SFT`
[3] `https://huggingface.co/datasets/a-m-team/AM-Thinking-v1-Distilled`

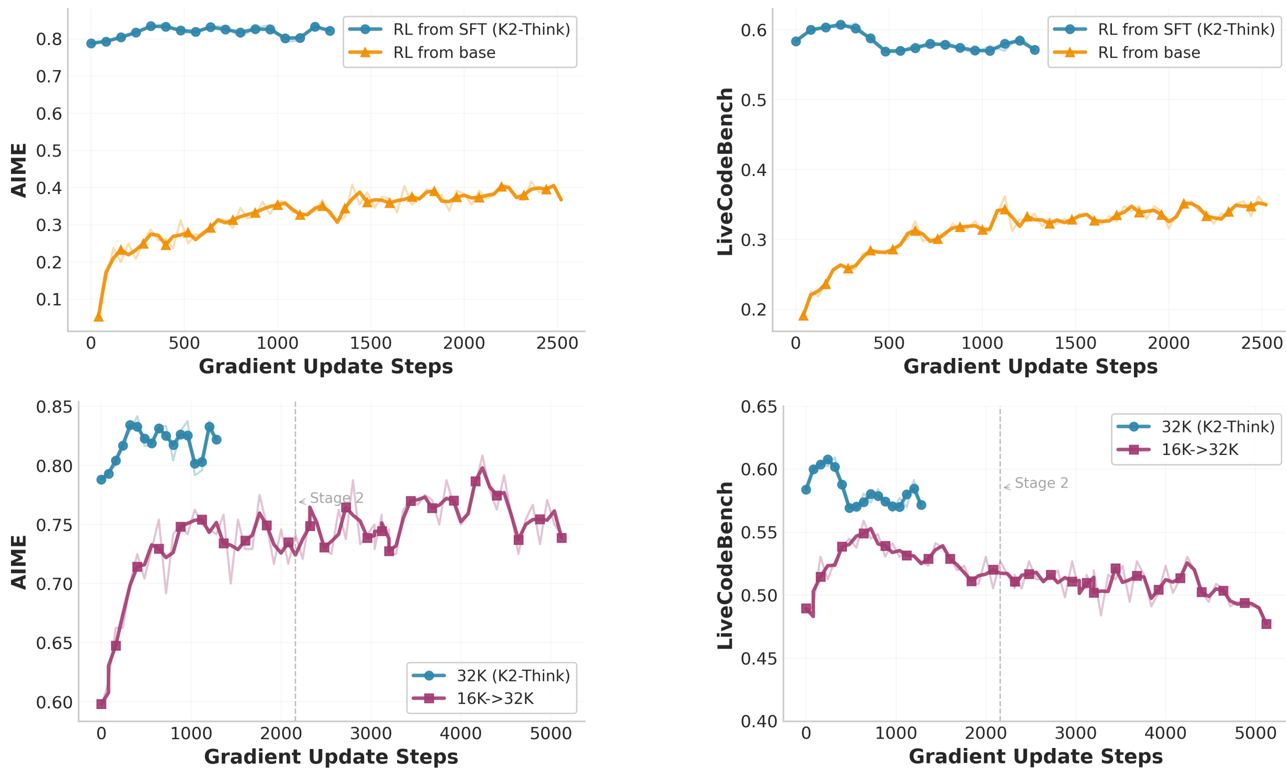

Figure 4: Ablation Studies on Multi-stage Training and RL from Base Models. (top): RL from base models achieves much faster performance gains compared to RL from SFT models. However, a substantial performance gap remains, suggesting that SFT enhances the model's score at the cost of slower subsequent improvement and increased susceptibility to collapse during RL. **(bottom)**: reducing K2-THINK-SFT maximum response length significantly impacts performance. Multi-stage training (16,000 to 32,000) struggles to recover original performance, even with prolonged training.

Apart from pass@1 scores, we also use pass@k to quantify reasoning performance under a fixed sampling budget k. Interpreting the pass@k curve as a capability boundary, we evaluate K2-THINK-SFT. In Figure 3, assessing performance on AIME2024, our SFT model dominates the base model across sampling budgets. The SFT curves saturate near 93.3% by $k \approx 128$, whereas the base model continues to improve but remains well below that plateau. The growth in K2-THINK-SFT performance as the sampling budget grows suggests there remains an opportunity for improvement during the following RL stage.

2.2 Phase 2: Reinforcement Learning with Verifiable Rewards

Following the SFT stage, we perform Reinforcement Learning with Verifiable Rewards (RLVR) to train K2-THINK to excel in domains with verifiable outcomes, which constitutes the second pillar of our full reasoning system. RLVR reduces the complexity and cost of preference-based alignment via RLHF (Casper et al., 2023) by directly optimizing for correctness of model generations.

For K2-THINK's RLVR, we use the Guru dataset (Cheng et al., 2025), which was curated to extend open-source reasoning models to verifiable domains beyond Math and Code. We leverage all six domains from the Guru dataset, comprising nearly 92,000 verifiable prompts that cover Math, Code, Science, Logic, Simulation, and Tabular tasks. We refer interested readers to the Guru paper for the detailed dataset curation, including de-duplication, reward designs, and filtering. Our RLVR implementation is built on the `verl` library (Sheng et al., 2025) with the GRPO algorithm (Shao et al., 2024).

2.2.1 Observations

In this subsection we provide a retrospective set of observations that serve as motivation for future development.

Starting from a strong SFT checkpoint yields better performance but limits RL gains. While RL consistently improved K2-THINK-SFT performance across internal evaluations and public benchmarks, the absolute improvements were modest. As a comparative experiment, we also trained a model with the same RL recipe and Guru data directly from the Qwen2.5-32B base. Figure 4 (top) demonstrates that RL training from the base model achieves nearly 40% improvement on AIME 2024 over the training course, while RL from K2-THINK-SFT yields only 5% improvement. This validates that stronger SFT checkpoints leave less room for RL refinement, consistent with findings from Liu et al. (2025b) regarding the relationship between SFT scope and subsequent RL effectiveness. Also, we notice RL training from the SFT checkpoint exhibits early plateauing and even degradation. We suspect that heavily "SFTed" models become constrained in their ability to explore alternative reasoning strategies during RL training, limiting the policy's capacity for meaningful adaptation.

Multi-stage RL training with reduced initial context length degrades performance. Many concurrent research efforts employ multi-stage training as implicit curriculum learning (An et al., 2025; Liu et al., 2025a; Rastogi et al., 2025), incrementally increasing context length. This accelerates early training while the model develops competency and then allows the model in later stages to handle more difficult questions with the extended context. We tested this approach by first constraining model output to 16,000 tokens during initial RL training from K2-THINK-SFT, then expanding to 32,000 tokens (this is the maximum length seen during the SFT stage) for continued training. As shown in Figure 4 (bottom), this multi-stage approach failed to match even the baseline SFT model performance. Cutting the maximum length below the SFT training configuration yields substantially lower performance. This negative result undermines the original motivation for multi-stage training to achieve on-par or better performance with shorter responses to save inference tokens. We suspect that reducing context length below the SFT training regime (32k → 16k → 32k) disrupts the model's established reasoning patterns as we did not perform any additional data filtering to correspond to this multi-stage training. However, we did not evaluate expanding beyond the SFT context length (e.g., 32k → 48k), as implemented in Polaris (An et al., 2025), which may still provide benefits.

2.3 Phase 3: Test-time Improvement

To further enhance K2-THINK performance, we developed a test-time scaffolding that implements existing methods as well as integrates an original approach to provide structured input to our post-trained reasoning model. This subsection details two specific aspects of this scaffolding: agentic planning before reasoning, namely "Plan-Before-You-Think", and test-time scaling using Best-of-N sampling. These two techniques are pillars three and four of the complete K2-THINK system.

A diagram mapping the flow of information from the input provided, down to the final response, is illustrated in Figure 5. First, the prompt is restructured to outline a high-level plan, highlighting relevant concepts. This augmented prompt is then passed through the K2-THINK model, generating multiple responses. Finally, a pairwise comparison between candidate responses surfaces the best generation as the final output of our reasoning system. The remainder of this section provides details of how we set-up and implemented each of these components.[4]

[4]Code for K2-THINK test-time improvements is at: `https://github.com/MBZUAI-IFM/K2-Think-Inference`

Figure 5: Schematic overview of how K2-THINK generates responses via our test-time computation scaffold. A user query is first input to an external model which generates a high-level plan to provide a structured prompt to our K2-THINK model. We then sample 3 responses, using an external model to select the best which is then provided as output.

"Plan-Before-You-Think". The first procedure of K2-THINK's test-time computation is the introduction of a planning agent. In our current system implementation, we simply ask the agent to extract key concepts from the query, and create a high-level plan from them. The generated plan is appended alongside the original query, and provided to the K2-THINK model. K2-THINK's planning agent is simply implemented via prompting an instruction-tuned Language Model. We restrict this "Plan-Before-You-Think" procedure from providing direct answers or any reasoning trace. This deliberation phase, prior to any "thinking" by our reasoning model, has some basis in psychology and cognitive science. Planning and reasoning can be considered dual processes of human cognition and decision making (Evans, 2010) where planning is considered a meta-thinking process developing some structure to help guide one's thoughts.

Best-of-N (BoN) sampling. Best-of-N sampling, sometimes called repeated sampling, is a method where an LLM generates N independent outputs for a given prompt, and a reward model (or verifier) chooses the best one according to some metric—such as accuracy, coherence, or alignment with human preference (Stiennon et al., 2020; Nakano et al., 2021). This strategy effectively explores multiple possibilities and picks the most promising completion.

Implementation-wise, we pick the answer by comparing the answer candidates pairwise, discarding the one that an independent LLM judges to be worse. In K2-THINK, we finally adopt $N = 3$, which provides a reasonable improvement with low cost.

2.3.1 Observations

At inference time, we explored several approaches to enhance the K2-THINK model's performance. We began with simple engineering adjustments but soon discovered that minor changes to our test-time computation procedures significantly impacted overall performance.

We experimented with temperature tuning, iterating through a list of temperatures from 0.1 to 1.0, but found the overall improvement to be insignificant, leading us to use a temperature of 1.0 for all future runs. We also conducted extensive prompt engineering, trying over 30 different system prompts that utilized techniques like few-shot learning (Brown et al., 2020), role-playing (Kong et al., 2023), and situational prompting. However, we observed negligible gains.

More sophisticated test-time scaling methods were also tried following Sharma (2024). We tested several standard approaches including re2 (ReRead) (Xu et al., 2024), self-consistency (Wang et al., 2023b), CoT with reflection (Shinn et al., 2023), and Mixture of Agents (MoA) (Wang et al., 2024b). Among these, Best-of-N (BoN) and MoA yielded the most notable improvements. While MoA delivered marginally better performance, its significantly higher computational cost led to the selection of BoN for the final K2-THINK system. A different, more experimental approach involving Reinforcement Learning with rewards drawn from self-certainty signals (Zhao et al., 2025) was also explored, but it did not lead to any improvement of the post-trained model's performance.

2.4 Deploying K2-Think

We deploy K2-THINK on Cerebras Wafer-Scale Engine (WSE) systems, leveraging the world's largest processor and speculative decoding (Leviathan et al., 2023) to achieve unprecedented inference speeds for the reasoning system, making up the final two pillars of K2-THINK. The WSE delivers approximately 2,000 tokens per second, representing a 10 times improvement over the nominal 200 tokens per second observed on typical deployment environments such as NVIDIA H100/H200 GPUs. This dramatic speed-up fundamentally transforms the practical usability of long chain-of-thought reasoning.

Consider a typical complex reasoning task that generates a 32,000 token response, which is common for challenging mathematical proofs or multi-step coding problems. On an NVIDIA H100, this can be completed in just under 3 minutes, making interactive use frustrating and limiting practical applications. On the WSE the same 32,000 token generation is completed in just 16 seconds, maintaining user engagement and enabling true back-and-forth problem solving.

The performance advantage comes from the unique architecture of WSE. Unlike GPUs, which must continuously shuttle weights from high-bandwidth memory to GPU cores for each token generation, the WSE keeps all model weights resident in massive on-chip memory, leveraging 25 Petabytes per second of on-chip memory bandwidth, which is over 3,000 times more than the 0.008 PB/s provided by the latest NVIDIA B200 GPU. Since auto-regressive models generate tokens serially, memory bandwidth can be a significant bottleneck during inference. By integrating greater compute, memory, and memory bandwidth in a single device, wafer-scale technology enables industry-leading inference speed for generative models.

This efficiency proves especially critical for our test-time computation approach and agent-based reasoning workflows. When performing best-of-3 sampling, the system must wait for all three responses to complete before LLM evaluation can select the optimal solution. Further, multi-step reasoning pipelines that require sequential calls for planning and generation suffer from cumulative delays. The WSE's low-latency inference keeps these workflows interactive, preventing the cascade of delays that would otherwise render complex reasoning tasks impractical.

The difference between waiting minutes versus seconds for each interaction fundamentally transforms the user experience from batch processing to interactive reasoning. This deployment ensures that K2-THINK provides not just frontier reasoning capabilities but also the responsiveness required for practical, real-world applications, making sophisticated AI reasoning truly accessible for interactive use cases. We invite everyone to experience our K2-THINK system, powered by Cerebras' WSE, via API and through `k2think.ai`.

3 K2-THINK Evaluation

We evaluate K2-THINK in comparison with frontier models, both open-weight and proprietary, among a class of challenging reasoning benchmarks focused on Math, Code and Science. We design these evaluations to demonstrate that K2-THINK, despite only having 32B parameters and fairly modest test-time computation, pushes the frontier of open-source reasoning models. In particular we find that K2-THINK is highly capable for complex Math tasks, as shown in Table 1. In total we evaluate K2-THINK on the following benchmarks:

- MATH
  - **AIME 2024** (MAA, 2024), **AIME 2025** (Ye et al., 2025b): The 2024 and 2025 editions of the American Invitation Mathematics Examination (AIME), with each year featuring 30 questions

that have integer answers.

  - **HMMT25** (Balunović et al., 2025): This dataset, used as part of the MathArena benchmarking suite, is drawn from the Harvard-MIT Mathematics Tournament February 2025 competition, featuring 30 questions drawn from the subject areas of Algebra+Number Theory, Combinatorics, and Geometry.
  - **Omni-MATH-HARD** (Omni-HARD, Gao et al. (2024)): We use the most difficult subset of the Omni-MATH dataset, featuring questions sampled from competitive mathematics competitions at the Olympiad level from several countries, retaining only those problems that are rated as the top 2 difficulty level (9.0 and 10.0). This set has 173 questions, a much larger competition math benchmark, and perhaps the most compelling one.
  - A global micro-average (Micro-Avg.) is obtained by dividing the total number of correct answers by the total number of questions across all datasets.

- CODE
  - **LiveCodeBench** (LCBv5, Jain et al. (2024)): A collection of programming challenge problems aggregated from online platforms. We use queries aggregated between July 1, 2024 and February 1, 2025 (v5).
  - **SciCode** (Tian et al., 2024): SciCode evaluates a model's ability to generate code for solving realistic scientific research questions, covering 16 subdomains from Physics, Math, Material Science, Biology, and Chemistry. We report scores from the version of the benchmark where background knowledge is included within the prompt. Since SciCode already performs a complex, multi-step planning phase in collating this information we do not run our "Plan-Before-You-Think" step during our evaluation of K2-THINK on this baseline. All evaluation results include sub-problem and full-problem accuracies in Table 1.

- SCIENCE
  - **GPQA-Diamond** (GPQA-D, Rein et al. (2023)): This benchmark is comprised of "Google-proof" advanced multiple-choice questions written by experts from biology, physics, and chemistry.
  - **Humanity's Last Exam** (HLE, Phan et al. (2025)): Humanity's Last Exam was developed by subject-matter experts and consists of multiple-choice and short-answer questions with solutions that are unambiguous and easily verifiable, but cannot be quickly answered via internet retrieval.

We measure the performance of K2-THINK in comparison to frontier reasoning models, both open-source {Qwen3-30B-A3B (Yang et al., 2025a), GPT-OSS 20B (Agarwal et al., 2025a), QwQ-32B (Team, 2025), OpenReasoning-Nemotron-32B (NVIDIA, 2025), DeepSeek R1 (Guo et al., 2025), DeepSeek-v3.1 (Thinking) (DeepSeek, 2025), GPT-OSS 120B (Agarwal et al., 2025a), Qwen3-235B-A22B (Thinking) (Yang et al., 2025a)} and proprietary {GPT-5 (High) (OpenAI, 2025), Gemini-2.5 (Pro) (Google DeepMind, 2025), o3 (High) (OpenAI, 2025)} to adequately assess the advancements made by our post-training and test-time computation scaffold. We use a standardized evaluation methodology across all benchmarks and models. The maximum generation length is set to 64,000 tokens, sampling temperature is fixed at 1.0, top-p is 0.95 and the stop token is </answer>. Each benchmark result reported in Table 1 is the average of 16 independent pass@1 evaluations.

Benchmarks →	Math					Code		Science	
Models ↓	AIME 2024	AIME 2025	HMMT25	Omni-HARD	Micro-Avg.	LCBv5	SciCode (sub/main)	GPQA-D	HLE
K2-Think	90.83	81.24	73.75	60.73	67.99	63.97	39.2 / 12.0	71.08	9.95
GPT-OSS 20B	76.88	74.58	69.38	41.51	52.50	73.22	37.9 / 9.0	65.45	11.23
Qwen3-30B-A3B	70.63	58.14	23.54	23.87	33.08	42.20	28.5 / 4.8	58.91	6.14
Nemotron 32B	87.09	82.71	67.29	58.88	65.78	57.79	37.1 / 11.4	74.98	12.26
QwQ-32B	79.38	69.17	51.46	46.93	53.69	65.22	36.9 / 11.5	66.24	9.98
GPT-OSS 120B	89.58	84.59	81.88	57.76	67.20	74.53	38.8 / 11.0	77.04	18.58
Qwen3 235B-A22B	86.68	75.43	61.88	56.91	62.99	56.64	39.3 / 10.9	65.55	14.23
DeepSeek V3.1†	91.87	82.49	83.54	53.22	64.43	66.59	38.2 / 11.7	79.46	8.40
DeepSeek R1†	74.38	65.21	47.08	51.33	55.06	61.01	36.7 / 11.5	71.08	8.50 *
o3 High	92.26	86.58	80.80	59.39	68.68	73.30	41.7 / 11.9	81.30	22.34
Gemini2.5 Pro	87.24	85.75	74.18	69.36	73.82	58.24	45.1 / 15.4	84.51	19.93
GPT-5 High	94.78	92.15	91.79	73.61	80.21	82.68	41.3 / 12.4	85.96	28.63

Table 1: Benchmark performance comparison of K2-Think against open-source (top) and proprietary (bottom) frontier models. All metrics are reported as percentages. **We find that K2-Think is especially strong on challenging Math benchmarks while also maintaining respectable performance on Code and Science.** Values marked with * are directly taken from published results. All other reported values are avg@16 accuracy of generated answers, evaluated locally or through paid API access. From these results, we see that our K2-Think system with only 32B parameters approaches or exceeds the performance of the frontier models that are orders of magnitude larger. *† showing results for the original DeepSeek R1 and V3.1. Since the performance of DeepSeek R1-0528 is similar to V3.1, we do not report it separately.*

K2-Think excels in competition math questions. The evaluation results are summarized in Table 1. K2-Think, a 32B-parameter model, exhibits an average micro-average score of **67.99** across all math questions. This result is particularly noteworthy when compared to other models of similar or slightly larger size, such as GPT-OSS 20B (avg. 52.50), Qwen3-30B-A3B (avg. 33.08), and OpenReasoning-Nemotron-32B (avg. 65.78). The results clearly show that **K2-Think surpasses these models by a significant margin**. Furthermore, K2-Think's performance is not only dominant within its size class but also highly competitive with models that are orders of magnitude larger. Its math score also surpasses the larger models, including the two state-of-the-art open source models: DeepSeek V3.1 671B (avg. 64.43) and GPT-OSS 120B (avg. 67.20). Notably, K2-Think performs well on Omni-MATH-HARD (60.73), which contains the most difficult questions across competitions. **This performance places K2-Think at the top of all open source models on math reasoning**, and is close to strong proprietary models such as o3 High, showing that K2-Think excels in the most challenging questions.

K2-Think is versatile on Science and Coding domain. The evaluation results also show that K2-Think demonstrates a robust and competitive capability across both coding and scientific domains, solidifying its position as a versatile model. On coding benchmarks, K2-Think achieves a score of **63.97** on LiveCodeBench, significantly outperforming its similarly sized peers, including GPT-OSS 20B (42.20) and Qwen3-30B-A3B (36.9). This performance also surpasses the larger Qwen3 235B-A22B (56.64). When compared to larger models, **K2-Think shows parity and even superiority in certain metrics**: it achieves 39.2 on the SciCode benchmark (sub-problems), making it a close second compared with Qwen3 235B-A22B (39.3). On scientific reasoning, our system's performance on the GPQA-Diamond benchmark is 71.08, superior to most open-source models except OpenReasoning-Nemotron-32B (74.98), and GPT-OSS 120B (77.04). While its HLE score of 9.95 is not the highest, it remains respectable and indicative of a broad knowledge base. This combination of strong performance across diverse domains argues that K2-Think is not merely a specialist but a versatile model capable of tackling a wide range of analytical and knowledge-intensive tasks with high efficacy.

Beyond the preliminary conclusions shared in this report, our team is continuing to perform additional analyses and comparisons between K2-Think and a more complete set of competing models and reasoning benchmarks. It is however clear that K2-Think presents an advancement in open-source reasoning systems. With a 32B parameter model, and a moderate amount of test-time compute, our system provides comparative performance to models significantly larger (see Figure 1 for a visual depiction). This level of parameter efficiency, in terms of benchmark performance is a notable achievement, specifically among complex mathematics reasoning tasks.

Component Analysis of K2-Think Test-Time Computation In order to analyze the individual contribution of each element of the test-time computation procedure to the final performance of K2-Think, we conducted analyses where we implemented each procedure in isolation on top of the post-trained checkpoint. That is, after performing both SFT and RL, we applied only the prompt restructuring via high-level planning or best-of-3 re-sampling and verification during evaluation. To simplify the discussion, we present this component analysis only using the four Math benchmarks however the overall insights are consistent across all other benchmarks.

	AIME 2024	AIME 2025	HMMT25	Omni-MATH-HARD
SFT+RL Checkpoint	86.26	77.72	66.46	56.74
+ Plan only	85.21	81.04	71.87	58.97
+ Bo3 only	90.77	81.22	71.16	59.47
+ Plan + Bo3 (K2-Think)	90.83	81.24	73.75	60.73

Table 2: Component analysis of the test-time computation procedures used to improve from our post-training checkpoint in the development of our final K2-Think system. The greatest gains come from Best-of-3 sampling, further improvement is seen after combining with high-level planning.

The component analyses presented here are executed in the same fashion as the comparative baselines discussed above. All results presented in Table 2 are averaged over 16 independent runs of the benchmark with the same settings as presented previously. We see in this analysis that the majority of the improvement over the post-trained checkpoint is afforded via Best-of-N scaling, using only 3 sampled generations per prompt. In isolation, the performance benefit of re-structuring the input prompt with a high-level plan also contributes an improvement to performance but with lesser effect. However, in combination with Best-of-N scaling the overall test-time procedure offers significant gains, offering 4-6 percentage points of improvement across all benchmarks.

"Plan-Before-You-Think" Reduces Response Lengths With the complete K2-Think system, we require the model to create a plan before thinking. While we have shown this procedure to positively affect reasoning performance, the expansion of the prompt might cause the model to use more tokens when formulating an answer. However, we found the opposite to be true. We report the average number of tokens K2-Think generated in the final response before and after implementing test-time computation in Table 3 for each benchmark evaluated across Math, Code and Science domains, comparing the K2-Think post-training checkpoint and the full K2-Think system. The inclusion of this plan achieves two benefits: response quality improves and there is a reduction in the number of tokens used by up to nearly 12% in comparison to the post-training checkpoint. Thus, by integrating planning before reasoning, K2-Think provides more concise answers as a result of the test-time computation we have used.

We also compare the average number of tokens used among the best performing open-weight models. We see that K2-Think responses are far shorter than Qwen3-235-A22B and in a similar range to what is produced by GPT-OSS 120B in mathematical reasoning. However, when comparison response

Model	AIME 2024		AIME 2025		HMMT25		Omni-HARD		LCBv5		GPQA-D	
SFT+RL Checkpoint	21,482		25,262		29,136		34,042		13,589		14,998	
K2-THINK	20,040	(-6.72%)	24,266	(-3.94%)	27,030	(-7.23%)	30,050	(-11.73%)	12,166	(-10.53%)	14,680	(-2.12%)
Qwen3-235B-A22B	29,896		34,541		39,767		45,701		27,716		20,007	
GPT-OSS 120B	15,971		19,151		25,566		35,021		7,389		11,281	
DeepSeek v3.1	12,364		15,143		19,073		24,841		6,158		6,592	

Table 3: An analytical comparison of the average number of tokens used between the full K2-THINK system and the post-training checkpoint. After implementing our test-time computation scaffold, our response length decreases on average, with the percentage of reduction included in the shaded cells. We also compare to the average number of tokens generated by top-performing open-weight models, showing better efficiency than Qwen3-235B-A22B and similar to GPT-OSS 120B.

lengths for Code and Science domains, there are greater opportunities to find efficiencies in future improvements of K2-THINK.

3.1 Red-teaming K2-THINK

Ensuring the safe operation of a model is essential for its open release. To this end, we systematically evaluate K2-THINK against adversarial prompts, harmful content, and robustness stress tests using established public safety benchmarks (Lin et al., 2024). For each benchmark, we sample 100 test cases and report a *safe score*, where higher values indicate stronger safety performance. To provide a clearer picture of real-world risks, we consolidate results into four key aspects that capture the practical safety surfaces most relevant in deployment:

1. **High-Risk Content Refusal** — ability to reject direct requests for unsafe or harmful outputs.
2. **Conversational Robustness** — maintaining safe behavior consistently across multi-turn dialogues.
3. **Cybersecurity & Data Protection** — resilience against information leakage, prompt extraction, and cyberattack assistance.
4. **Jailbreak Resistance** — robustness to adversarial attacks designed to bypass safeguards.

This framework provides a clearer operational safety profile and guides targeted mitigations.

High-Risk Content Refusal We first check the model's reliability in rejecting unsafe requests. Evaluation spans complementary datasets covering harmful instructions (**Do-Not-Answer** (Wang et al., 2023c), **HarmBench** (Mazeika et al., 2024)), physical harm scenarios (**PhysicalSafety** (Bianchi et al., 2023)), basic safety checks (**SimpleSafetyTests** (Vidgen et al., 2023)), toxic content generation (**ToxiGen** (Hartvigsen et al., 2022; Hosseini et al., 2023)), commonsense safety (**CoNA** (Bianchi et al., 2023)), and harmful Q&A (**HarmfulQ** (Shaikh et al., 2023)).

Dataset	Score
Do-Not-Answer	0.88
HarmBench	0.56
PhysicalSafety	0.49
SimpleSafetyTests	0.95
ToxiGen	0.97
CoNA	0.97
HarmfulQ	0.99
Macro-average	**0.83**

Table 4: High-risk content refusal results across safety datasets. The model achieves near-perfect performance on four of seven tasks, with clear improvement opportunities on HarmBench and PhysicalSafety.

The results of this analysis are featured in Table 4. K2-THINK demonstrates extensive ability to avoid generating high-risk content as measured by near-perfect scores in 4 out of 7 benchmarks. Of the remaining 3 benchmarks in this aspect of safety evaluation, HarmBench and PhysicalSafety reveal a weakness in our system toward recognizing cyber or physical risks. We are actively working to improve our system along these dimensions of risk in its public facing deployment.

Conversational Robustness Next, we assess refusal consistency across multi-turn adversarial dialogues using **DialogueSafety** (Dinan et al., 2019), **HH-RLHF** (Bai et al., 2022), and **DICES350** (Aroyo et al., 2023) for dynamic dialogue manipulations.

We see in Table 5 that K2-THINK is especially robust to sustained adversarial dialogues and repeated efforts to ellicit harmful behaviors from our reasoning system. Here, K2-THINK is near perfect at maintaining refusal consistency on both the DialogueSafety and HH-RLHF benchmarks.

Dataset	Score
DialogueSafety	0.99
HH-RLHF	0.95
DICES350	0.73
Macro-average	**0.89**

Table 5: Conversational robustness results across dialogue safety datasets. The model exhibits notable robustness to multi-turn adversarial attempts to produce harmful outputs, with particular strength on DialogueSafety and room for improvement on DICES350.

Dataset	Score
PersonalInfoLeak (few-shot)	0.86
CyberattackAssistance	0.47
PromptExtractionRobustness	0.35
Macro-average	**0.56**

Table 6: Cybersecurity, data protection, and prompt extraction results. The model demonstrates robustness against leaking personal information, with significant room for improvement on cyberattack assistance prevention and prompt extraction robustness.

Cybersecurity & Data Protection & Prompt Extraction We evaluate resilience against data leakage and misuse with **PersonalInfoLeak** (Li et al., 2023) (privacy leakage), **CyberattackAssistance** (Bhatt et al., 2023) (hacking assistance), and **PromptExtractionRobustness** (Toyer et al., 2023) (system prompt extraction).

We see in Table 6 that K2-THINK is able to resist attempts to extract personally identifying information while unfortunately exhibiting some susceptibility to revealing the system prompt and aiding in devising cyberattacks. This indicates an opportunity to further tune our reasoning system for improved resilience.

Dataset	Score
Few-Shot Attack	0.96
Gandalf Ignore	0.87
Tense Change	0.84
Multilingual	0.83
PromptInjection	0.77
One-Sided Statement	0.77
Refusal Suppression	0.76
Persona Modulation	0.59
Do-Anything-Now	0.43
LatentJailbreak	0.37
Macro-average	**0.72**

Table 7: Jailbreak resistance results across adversarial prompt techniques. The model demonstrates mixed resilience, with strong performance against direct attacks and vulnerabilities to indirect methods.

Jailbreak Resistance Finally, we evaluate various adversarial attack strategies: hidden triggers (**LatentJailbreak** (Qiu et al., 2023)), prompt redirection (**PromptInjection** (Liu et al., 2023b)), instruction overrides (**Gandalf Ignore** (Schulhoff et al., 2023)), role-play attacks (**DAN** (Shen et al., 2023)), cross-lingual exploits (**Multilingual** (Wang et al., 2023a)), grammatical perturbations (**Tense Change** Lin et al. (2024)), adversarial demonstrations (**Few-Shot Attack** (Wei et al., 2023b)), bias-driven attacks (**One-Sided Statement** (Liu et al., 2023a)), identity manipulation

(**Persona Modulation** (Shah et al., 2023)), and direct refusal bypasses (**Refusal Suppression** (Wei et al., 2023a)).

K2-Think's jailbreak resistance results (shown in Table 7) demonstrate a mixture of resilience and susceptibility to various adversarial prompt strategies. K2-Think exhibits strong performance when attacks are immediately apparent but shows an apparent weakness to indirect attacks. This lack of generalized robustness to adversarial jailbreaking attempts illustrates a need to thoroughly improve our publicly deployed reasoning system.

Overall Results Across all four dimensions, results are aggregated into a single Safety-4 macro score, computing the average from the four analyses performed as part of our safety testing of K2-Think. The macro average of each of the four analyses are included in Table 8.

Safety Aspect	Macro-Avg Score
High-Risk Content Refusal	0.83
Conversational Robustness	0.89
Cybersecurity & Data Protection	0.56
Jailbreak Resistance	0.72
Safety-4 Macro (avg)	**0.75**

Table 8: Overall Safety-4 results which is a composite score of the four safety surfaces evaluated in this broad analysis. The macro score of 0.750 indicates that K2-Think establishes a solid safety profile with specific strengths in harmful content refusal and maintaining consistent behavior in conversations.

Overall, K2-Think achieves a **Safety-4 macro score of 0.750**, indicating a solid baseline of safety with strong performance in refusing harmful content and maintaining consistent behavior in conversations. At the same time, we recognize that further work is required to strengthen **cybersecurity defenses, jailbreak robustness, and refusal calibration**. While establishing a solid baseline, we acknowledge clear opportunities to improve the safety of our reasoning system. Addressing these areas is an active priority in our roadmap to further improve K2-Think under adversarial conditions.

4 Related Work

Extending base language model capabilities via SFT Supervised fine-tuning (SFT) has become a widely used post-training method to extend the capability boundary of Large Language Models (Ouyang et al., 2022; Dubey et al., 2024; Guo et al., 2025; Bercovich et al., 2025). Early SFT work primarily focused on task specialization, adapting foundational models to specific NLP benchmarks like text classification or translation on narrowly-defined datasets (Liu et al., 2019; Raffel et al., 2020). This paradigm shifted significantly with the rise of large-scale instruction tuning; the goal evolved from single-task mastery to creating general-purpose assistants capable of following diverse human commands (Wei et al., 2021; Ouyang et al., 2022; Taori et al., 2023). More recently, SFT has pivoted towards enhancing complex reasoning on diverse downstream tasks like math, code, and science (Hui et al., 2024; Yang et al., 2024b; Abdin et al., 2025; Liu et al.). Some approaches focus on scale, constructing massive datasets of reasoning traces to instill robust, long-chain-of-thought capabilities in models (Guha et al., 2025; Tian et al., 2025; Liu et al., 2025b). In contrast, other methods demonstrate that meticulously curated, high-quality data can also endow LLMs with expert-level reasoning in domains like math (Ye et al., 2025a; Muennighoff et al., 2025). Building on the above, our work conducts analysis and provides practical insights on SFT.

Improving LLM Reasoning with RL Reinforcement Learning from Verifiable Rewards (RLVR) has emerged as a powerful paradigm for enhancing the reasoning capabilities of Large Language Models (Guo et al., 2025; OpenAI, 2024). Following initial successes, a significant body of open work has explored RLVR, primarily concentrating on specializing models for highly challenging single domains. Efforts such as Open-Reasoner-Zero (Hu et al., 2025), Skywork-OR1 (He et al., 2025), DeepScaler (Luo et al., 2025b), and SimpleRL (Zeng et al., 2025) have notably leveraged extensive mathematical data to achieve state-of-the-art performance on complex math benchmarks. Similarly, DeepCoder (Luo et al., 2025a) focused on RL for code generation tasks. While powerful within their specific areas, this domain-specific focus inherently limits the generalizability of the resulting models across the broader landscape of reasoning tasks. Concurrent works to our K2-Think development like General-Reasoner (Ma et al., 2025) and Nemotron-CrossThinker (Akter et al., 2025) have begun to explore broader domains for RL training. However, none of these works explore the added utility of test-time computation for improving the general reasoning capabilties of post-trained models.

Test Time Scaling Test-time scaling has been a major component of proprietary models released in recent years; such as o1 (OpenAI, 2024), Grok Heavy (xAI, 2025), Gemini 2.5 (Google DeepMind, 2025), and GPT-5 (OpenAI, 2025). However, with fairly little transparency about specific components and their overall effect. The closest work to ours is PlanGEN (Parmar et al., 2025), a multi-model framework for planning and reasoning combining a constraint model, a verification model, and a selection model to guide inference-time algorithms including Best-of-N. By using constraint-guided iterative verification and a modified UCB-based selection policy, PlanGEN chooses the most suitable algorithm for each problem instance. Importantly, they use Best-of-N with verifiers on the *plans*: we use it for the generated *responses*.

Also related are general LLM-based hierarchical reasoning approaches, particularly those that operate with at least one level of hierarchy doing planning. Wang et al. (2024a) has a planning model provide high-level strategy while a solver model performs detailed reasoning. HyperTree Planning (Gui et al., 2025) models planning with a hypertree-structure, allowing LLMs to decompose planning queries into structured sub-tasks. Wang et al. (2025a) demonstrates a brain-inspired architecture with separate recurrent modules for high-level planning and low-level reasoning, showing that explicit separation of timescales improves performance on algorithmic reasoning tasks. Our novelty is to combine our "Plan-Before-You-Think" approach, a type of multi-LLM-hierarchical reasoning, with Best-of-N with verifiers (Cobbe et al., 2021) in order to return the best response.

5 Discussion

5.1 Primary technical insights

Multiple domains are important for post-training One core finding from our prior work curating RL data (Cheng et al., 2025) is that for generally useful reasoning models, there is a need to expand post-training to include more domains. The effect of post-training, and the domains utilized, is nuanced. Domains commonly included in pre-training (Math, Code, and Science) broadly benefit from a variety of post-training data as the refinement of the model's chains of thought is supported by the knowledge it already has. However those domains with limited pre-training exposure–like Logic and Simulation tasks–only improve when they are included in the RL training pipeline. This indicates that using diverse, multi-domain datasets is critical for developing truly versatile reasoning models.

Test-time computation performance gains can be additive with the right combination We find that two simple test-time computation procedures work well together: our "Plan-Before-You-Think" prompt restructuring in conjunction with Best-of-N scaling. Each individual method does improve over the K2-Think model but the largest gains in performance are seen when these components are combined. To our surprise, simply extracting a high level plan focused on the core concepts associated with the input and only sampling 3 candidate responses were sufficient to provide sufficient scaling to better leverage the knowledge embedded in the post-trained model.

"Plan-Before-You-Think" improves model performance while reducing token expenditure By requiring the model to create a plan before initiating its reasoning process, we achieve two benefits: planning itself improves response quality, and response lengths are reduced nearly by 12%.

5.2 Looking forward

With the release of K2-Think, we establish an exciting new precedent for the Institute of Foundation Models, the focus of which is to pursue trustworthy science and provide fully open artifacts for further research and development. In this report we present:

Empowering small models to "punch above their weight" With the complete K2-Think system, we demonstrate that a 32B-scale model, post-trained to produce long reasoning chains of thought, paired with relatively little test-time computation can endow the small model with capabilities that are competitive with models with orders of magnitude more parameters. Altogether our end-to-end reasoning system unlocks performance at the frontier of current open-source capabilities.

Beyond Open Source We are extending the limit of our open-source activities beyond data, models and training artifacts. This expansion of our open-source efforts will now include deploying our full reasoning system for public use. We are publishing our test-time computation implementation as well. K2-Think is broadly available via API and an online web portal. In this we are opening avenues to explore how to best "battle-test" public facing LLM infrastructure. Details about how to use and interact with K2-Think can be found at `k2think.ai`, we proudly invite all to try it out!

K2-Think is a compelling stepping stone for our ongoing efforts to broaden access to foundation model research and development through open-science. Our motivation to deploy K2-Think for public use is grounded in curiosity about how to best engineer inference systems for large-scale foundation models. It is critical to publicly share best practices around hosting these models, how to overcome common stability issues, and enable continued research into how to robustly stress test deployed models. Secondarily, as we continue scaling our own open-source models, there will be a time when simply making the weights and training artifacts public is no longer useful as fewer institutions and organizations will be able to host or interact with the models themselves. This by-product of our work, investigating and building ever more capable open models, is antithetical to our founding ethos as a research institute. We are committed to making publicly available as much of our model development and deployment as possible in order to enable all who are interested to build on or contribute to our work. The lessons we learn through deployment with K2-Think and subsequent moderately sized models will be critical to our ongoing development of exceedingly large and capable models.

Acknowledgment

The authors hereby acknowledge and thank the strong support and collaboration of G42 for their contributions throughout the project, including the essential computational infrastructure as well as significant expertise in evaluation methodology and safety protocols. This partnership proved instrumental in advancing our research objectives.

References

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. Phi-4-reasoning technical report. arXiv preprint arXiv:2504.21318, 2025.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. arXiv preprint arXiv:2508.10925, 2025a.

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in llm reasoning, 2025b. URL https://arxiv.org/abs/2505.15134.

Syeda Nahida Akter, Shrimai Prabhumoye, Matvei Novikov, Seungju Han, Ying Lin, Evelina Bakhturi, Eric Nyberg, Yejin Choi, Mostofa Patwary, Mohammad Shoeybi, et al. Nemotron-crossthink: Scaling self-learning beyond math reasoning. arXiv preprint arXiv:2504.13941, 2025.

Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, and Lingpeng Kong. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models, 2025. URL https://hkunlp.github.io/blog/2025/Polaris.

Lora Aroyo, Alex S. Taylor, Mark Díaz, Christopher Homan, Alicia Parrish, Gregory Serapio-García, Vinodkumar Prabhakaran, and Ding Wang. DICES dataset: Diversity in conversational AI evaluation for safety. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a74b697bce4cac6c91896372abaa8863-Abstract-Datasets_and_Benchmarks.html.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. URL https://doi.org/10.48550/arXiv.2204.05862.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. arXiv preprint arXiv:2505.23281, 2025.

Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, Ido Shahaf, Oren Tropp, Ehud Karpas, Ran Zilberstein, Jiaqi Zeng, Soumye Singhal, Alexander Bukharin, Yian Zhang, Tugrul Konuk, Gerald Shen, Ameya Sunil Mahabaleshwarkar, Bilal Kartal, Yoshi Suhara, Olivier Delalleau, Zijia Chen, Zhilin Wang, David Mosallanezhad, Adi Renduchintala, Haifeng Qian, Dima Rekesh, Fei Jia, Somshubra Majumdar, Vahid Noroozi, Wasi Uddin Ahmad, Sean Narenthiran, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Siddhartha Jain, Igor Gitman, Ivan Moshkov, Wei Du, Shubham Toshniwal, George Armstrong, Branislav Kisacanin, Matvei Novikov, Daria Gitman, Evelina Bakhturina, Jane Polak Scowcroft, John Kamalu, Dan Su, Kezhi Kong, Markus Kliegl, Rabeeh Karimi, Ying Lin, Sanjeev Satheesh, Jupinder Parmar, Pritam Gundecha, Brandon Norick, Joseph Jennings, Shrimai Prabhumoye, Syeda Nahida Akter, Mostofa Patwary, Abhinav Khattar, Deepak Narayanan, Roger Waleffe, Jimmy Zhang, Bor-Yiing Su, Guyue Huang, Terry Kong, Parth Chadha, Sahil Jain, Christine Harvey, Elad Segal, Jining Huang, Sergey Kashirsky, Robert McQueen, Izzy Putterman, George Lam, Arun Venkatesan, Sherry Wu, Vinh Nguyen, Manoj Kilaru, Andrew Wang, Anna Warno, Abhilash Somasamudramath, Sandip Bhaskar, Maka Dong, Nave Assaf, Shahar Mor, Omer Ullman Argov, Scot Junkin, Oleksandr Romanenko, Pedro Larroy, Monika Katariya, Marco Rovinelli, Viji Balas, Nicholas Edelman, Anahita Bhiwandiwalla, Muthu Subramaniam, Smita Ithape, Karthik Ramamoorthy, Yuting Wu, Suguna Varshini Velury, Omri Almog, Joyjit Daw, Denys Fridman, Erick Galinkin, Michael Evans, Shaona Ghosh, Katherine Luna, Leon Derczynski, Nikki Pope, Eileen Long, Seth Schneider, Guillermo Siman, Tomasz Grzegorzek, Pablo Ribalta, Monika Katariya, Chris Alexiuk, Joey Conway, Trisha Saar, Ann Guan, Krzysztof Pawelec, Shyamala Prayaga, Oleksii Kuchaiev, Boris Ginsburg, Oluwatobi Olabiyi, Kari Briski, Jonathan Cohen, Bryan Catanzaro, Jonah Alben, Yonatan Geifman, and Eric Chung. Llama-nemotron: Efficient reasoning models, 2025. URL https://arxiv.org/abs/2505.00949.

Manish Bhatt, Sahana Chennabasappa, Cyrus Nikolaidis, Shengye Wan, Ivan Evtimov, Dominik Gabi, Daniel Song, Faizan Ahmad, Cornelius Aschermann, Lorenzo Fontana, Sasha Frolov, Ravi Prakash Giri, Dhaval Kapil, Yiannis Kozyrakis, David LeBlanc, James Milazzo, Aleksandar Straumann, Gabriel Synnaeve, Varun Vontimitta, Spencer Whitman, and Joshua Saxe. Purple llama cyberseceval: A secure coding benchmark for language models, 2023. URL https://arxiv.org/abs/2312.04724.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. CoRR, abs/2309.07875, 2023. URL https://doi.org/10.48550/arXiv.2309.07875.

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.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. arXiv preprint arXiv:2307.15217, 2023.

Zhoujun Cheng, Shibo Hao, Tianyang Liu, Fan Zhou, Yutao Xie, Feng Yao, Yuexin Bian, Yonghao Zhuang, Nilabjo Dey, Yuheng Zha, et al. Revisiting reinforcement learning for llm reasoning from a cross-domain perspective. arXiv preprint arXiv:2506.14965, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021.

DeepSeek. Deepseek-v3.1 release, August 2025. URL https://api-docs.deepseek.com/news/news250821.

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4537–4546, Hong Kong, China, November 2019. Association for Computational Linguistics. URL https://aclanthology.org/D19-1461/.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. arXiv e-prints, pages arXiv–2407, 2024.

Jonathan St BT Evans. Intuition and reasoning: A dual-process perspective. Psychological Inquiry, 21(4):313–326, 2010.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, et al. Omni-math: A universal olympiad level mathematic benchmark for large language models. arXiv preprint arXiv:2410.07985, 2024.

Google DeepMind. Gemini 2.5: Our newest Gemini model with thinking - The Keyword. https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/#gemini-2-5-thinking, March 2025.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models. arXiv preprint arXiv:2506.04178, 2025. URL https://arxiv.org/abs/2506.04178.

Runquan Gui, Zhihai Wang, Jie Wang, Chi Ma, Huiling Zhen, Mingxuan Yuan, Jianye HAO, Defu Lian, Enhong Chen, and Feng Wu. Hypertree planning: Enhancing llm reasoning via hierarchical thinking. In Forty-second International Conference on Machine Learning, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3309–3326, 2022.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. Skywork open reasoner series. `https://capricious-hydrogen-41c.notion.site/Skywork-Open-Reaonser-Series-1d0bc9ae823a80459b46c149e4f51680`, 2025. Notion Blog.

Saghar Hosseini, Hamid Palangi, and Ahmed Hassan Awadallah. An empirical study of metrics to measure representational harms in pre-trained language models. CoRR, abs/2301.09211, 2023. URL `https://doi.org/10.48550/arXiv.2301.09211`.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. arXiv preprint arXiv:2503.24290, 2025.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. arXiv preprint arXiv:2409.12186, 2024.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. arXiv preprint arXiv:2403.07974, 2024.

Yixin Ji, Juntao Li, Hai Ye, Kaixin Wu, Kai Yao, Jia Xu, Linjian Mo, and Min Zhang. Test-time compute: from system-1 thinking to system-2 thinking. arXiv preprint arXiv:2501.02497, 2025a.

Yunjie Ji, Xiaoyu Tian, Sitong Zhao, Haotian Wang, Shuaiting Chen, Yiping Peng, Han Zhao, and Xiangang Li. Am-thinking-v1: Advancing the frontier of reasoning at 32b scale. arXiv preprint arXiv:2505.08311, 2025b.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. Better zero-shot reasoning with role-play prompting. arXiv preprint arXiv:2308.07702, 2023.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In International Conference on Machine Learning, pages 19274–19286. PMLR, 2023.

Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng, and Yangqiu Song. Multi-step jailbreaking privacy attacks on chatgpt. In Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023, pages 4138–4153, 2023. URL `https://aclanthology.org/2023.findings-emnlp.272`.

Lizhi Lin, Honglin Mu, Zenan Zhai, Minghan Wang, Yuxia Wang, Renxi Wang, Junjie Gao, Yixuan Zhang, Wanxiang Che, Timothy Baldwin, Xudong Han, and Haonan Li. Against the achilles' heel: A survey on red teaming for generative models, 2024. URL `https://arxiv.org/abs/2404.00629`.

Chengyuan Liu, Fubang Zhao, Lizhi Qing, Yangyang Kang, Changlong Sun, Kun Kuang, and Fei Wu. Goal-oriented prompt attack and safety evaluation for llms, 2023a. URL `https://arxiv.org/abs/2309.11830`.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models, 2025a. URL `https://arxiv.org/abs/2505.24864`.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against llm-integrated applications, 2023b. URL https://doi.org/10.48550/arXiv.2306.05499.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

Zihan Liu, Zhuolin Yang, Yang Chen, Chankyu Lee, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron 1.1: Advancing math and code reasoning through sft and rl synergy. arXiv preprint arXiv:2506.13284, 2025b.

Zihang Liu, Tianyu Pang, Oleg Balabanov, Chaoqun Yang, Tianjin Huang, Lu Yin, Yaoqing Yang, and Shiwei Liu. Lift the veil for the truth: Principal weights emerge after rank reduction for reasoning-focused supervised fine-tuning. In Forty-second International Conference on Machine Learning.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepcoder: A fully open-source 14b coder at o3-mini level, 2025a. URL https://pretty-radio-b75.notion.site/DeepCoder-A-Fully-Open-Source-14B-Coder-at-O3-mini-Level-1cf81902c14680b3bee5eb349a512a51. Notion Blog.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025b. URL https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2. Notion Blog.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. General-reasoner: Advancing llm reasoning across all domains. https://github.com/TIGER-AI-Lab/General-Reasoner/blob/main/General_Reasoner.pdf, 2025.

MAA. American invitational mathematics examination - aime. In American Invitational Mathematics Examination - AIME 2024, February 2024. URL https://maa.org/math-competitions/american-invitational-mathematics-examination-aime.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David A. Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. CoRR, abs/2402.04249, 2024. URL https://doi.org/10.48550/arXiv.2402.04249.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. arXiv preprint arXiv:2501.19393, 2025.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback. arXiv preprint arXiv:2112.09332, 2021.

NVIDIA. Openreasoning-nemotron-32b, july 2025. URL https://huggingface.co/nvidia/OpenReasoning-Nemotron-32B. Large language model for mathematical, coding, and scientific reasoning. Based on Qwen2.5-32B-Instruct with 32B parameters. Released July 16, 2025.

OpenAI. OpenAI o1 System Card. https://openai.com/index/openai-o1-system-card/, 2024.

OpenAI. Introducing GPT-5. https://openai.com/index/introducing-gpt-5/, 2025. Accessed: 2025-09-04.

OpenAI. Introducing openai o3 and o4-mini, 2025. URL https://openai.com/index/introducing-o3-and-o4-mini/. Accessed: 2025-06-12.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35:27730–27744, 2022.

Mihir Parmar, Xin Liu, Palash Goyal, Yanfei Chen, Long Le, Swaroop Mishra, Hossein Mobahi, Jindong Gu, Zifeng Wang, Hootan Nakhost, et al. Plangen: A multi-agent framework for generating planning and reasoning trajectories for complex problem solving. arXiv preprint arXiv:2502.16111, 2025.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. arXiv preprint arXiv:2501.14249, 2025.

Huachuan Qiu, Shuai Zhang, Anqi Li, Hongliang He, and Zhenzhong Lan. Latent jailbreak: A benchmark for evaluating text safety and output robustness of large language models. CoRR, abs/2307.08487, 2023. URL https://doi.org/10.48550/arXiv.2307.08487.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140):1–67, 2020.

Abhinav Rastogi, Albert Q Jiang, Andy Lo, Gabrielle Berrada, Guillaume Lample, Jason Rute, Joep Barmentlo, Karmesh Yadav, Kartik Khandelwal, Khyathi Raghavi Chandu, et al. Magistral. arXiv preprint arXiv:2506.10910, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Sander V Schulhoff, Jeremy Pinto, Anaum Khan, Louis-FranÃois Bouchard, Chenglei Si, Jordan Lee Boyd-Graber, Svetlina Anati, Valen Tagliabue, Anson Liu Kost, and Christopher R Carnahan. Ignore this title and hackaprompt: Exposing systemic vulnerabilities of llms through a global prompt hacking competition. In Empirical Methods in Natural Language Processing, 2023.

Dale Schuurmans, Hanjun Dai, and Francesco Zanini. Autoregressive large language models are computationally universal. arXiv preprint arXiv:2410.03170, 2024.

Rusheb Shah, Quentin Feuillade-Montixi, Soroush Pour, Arush Tagade, Stephen Casper, and Javier Rando. Scalable and transferable black-box jailbreaks for language models via persona modulation. CoRR, abs/2311.03348, 2023. URL https://doi.org/10.48550/arXiv.2311.03348.

Omar Shaikh, Hongxin Zhang, William Held, Michael S. Bernstein, and Diyi Yang. On second thought, let's not think step by step! bias and toxicity in zero-shot reasoning. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 4454–4470, 2023. URL https://doi.org/10.18653/v1/2023.acl-long.244.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in rlvr. https://rethink-rlvr.notion.site/Spurious-Rewards-Rethinking-Training-Signals-in-RLVR-1f4df34dac1880948858f95aeb88872f, 2025. Notion Blog.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

Asankhaya Sharma. Optillm: Optimizing inference proxy for llms, 2024. URL https://github.com/codelion/optillm.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. CoRR, abs/2308.03825, 2023. URL https://doi.org/10.48550/arXiv.2308.03825.

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. In Proceedings of the Twentieth European Conference on Computer Systems, pages 1279–1297, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems, 36:8634–8652, 2023.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. Advances in neural information processing systems, 33:3008–3021, 2020.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, 2025.

Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu Huerta, and Hao Peng. Scicode: A research coding benchmark curated by scientists, 2024.

Xiaoyu Tian, Yunjie Ji, Haotian Wang, Shuaiting Chen, Sitong Zhao, Yiping Peng, Han Zhao, and Xiangang Li. Not all correct answers are equal: Why your distillation source matters. arXiv preprint arXiv:2505.14464, 2025. URL https://arxiv.org/abs/2505.14464.

Sam Toyer, Olivia Watkins, Ethan Adrian Mendes, Justin Svegliato, Luke Bailey, Tiffany Wang, Isaac Ong, Karim Elmaaroufi, Pieter Abbeel, Trevor Darrell, Alan Ritter, and Stuart Russell. Tensor trust: Interpretable prompt injection attacks from an online game. CoRR, abs/2311.01011, 2023. URL https://doi.org/10.48550/arXiv.2311.01011.

Bertie Vidgen, Hannah Rose Kirk, Rebecca Qian, Nino Scherrer, Anand Kannappan, Scott A. Hale, and Paul Röttger. Simplesafetytests: a test suite for identifying critical safety risks in large language models. CoRR, abs/2311.08370, 2023. URL https://doi.org/10.48550/arXiv.2311.08370.

Danqing Wang, Zhuorui Ye, Fei Fang, and Lei Li. Cooperative strategic planning enhances reasoning capabilities in large language models. arXiv preprint arXiv:2410.20007, 2024a.

Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical reasoning model. arXiv preprint arXiv:2506.21734, 2025a.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. arXiv preprint arXiv:2406.04692, 2024b.

Wenxuan Wang, Zhaopeng Tu, Chang Chen, Youliang Yuan, Jen-tse Huang, Wenxiang Jiao, and Michael R. Lyu. All languages matter: On the multilingual safety of large language models. CoRR, abs/2310.00905, 2023a. URL https://doi.org/10.48550/arXiv.2310.00905.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In The Eleventh International Conference on Learning Representations, 2023b. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example, 2025b. URL https://arxiv.org/abs/2504.20571.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. Do-not-answer: A dataset for evaluating safeguards in llms. CoRR, abs/2308.13387, 2023c. URL https://doi.org/10.48550/arXiv.2308.13387.

Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. Octothinker: Revisiting mid-training in the era of rl scaling. https://tinyurl.com/OctoThinker, 2025c. Notion Blog.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023a. URL http://papers.nips.cc/paper_files/paper/2023/hash/fd6613131889a4b656206c50a8bd7790-Abstract-Conference.html.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652, 2021.

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.

Zeming Wei, Yifei Wang, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. CoRR, abs/2310.06387, 2023b. URL https://doi.org/10.48550/arXiv.2310.06387.

xAI. Grok 4. https://x.ai/news/grok-4, July 2025.

Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. Llava-o1: Let vision language models reason step-by-step. arXiv preprint arXiv:2411.10440, 2024.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115, 2024a.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. arXiv preprint arXiv:2409.12122, 2024b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025a.

Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. Towards thinking-optimal scaling of test-time compute for llm reasoning. arXiv preprint arXiv:2502.18080, 2025b.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. arXiv preprint arXiv:2502.03387, 2025a.

Yixin Ye, Yang Xiao, Tiantian Mi, and Pengfei Liu. Aime-preview: A rigorous and immediate evaluation framework for advanced mathematical reasoning, 2025b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476, 2025.

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 preprint arXiv:2504.13837, 2025.

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 preprint arXiv:2503.18892, 2025.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. arXiv preprint arXiv:2505.19590, 2025.