ARIA: Adaptive Reasoning through Difficulty-Graded Trace Compression

Gagan*
Eunice Labs
PES University, Bengaluru

March 2026

Abstract

Large reasoning models overthink. Hand one an olympiad problem and it does the work; hand it $24 \div 2 = 12$ and it still burns through thousands of reasoning tokens, padding a trivial answer with the same deliberation it would spend on something genuinely hard. The usual fixes bolt something on: a difficulty classifier, a reinforcement learning loop, an architectural change. We wondered whether the data alone could carry the signal. Can a model learn to calibrate how much it thinks purely from how its training set is distributed? **ARIA** (Adaptive Reasoning Intelligence Architecture) is our answer. We split reasoning traces by difficulty, compress the easy and medium ones by 5.9× with a language model, leave the hard ones alone, and fine-tune on the mix, with no difficulty labels and no RL. Training DeepSeek-R1-Distill-Qwen-7B on 5,993 such examples produces a clean adaptive gradient. Token reduction runs from 3.2× on Level 1 problems down to 1.5× on Level 5: brief when brevity is enough, deep when depth is required. GSM8K accuracy even climbs by 2.5% while thinking tokens fall by half. Model, dataset, and code at https://huggingface.co/Eunice-Labs.

1 Introduction

Modern reasoning models rest on one bet: let the model think before it answers. Chain-of-thought prompting [Wei et al., 2022] demonstrated that walking through intermediate steps lifts performance on hard tasks, and reinforcement learning later scaled the idea up. Today o1 [OpenAI, 2024], DeepSeek-R1 [Guo et al., 2025], and QwQ [Qwen Team, 2024] routinely spin out reasoning traces running to thousands of tokens before they commit to an answer, and they top the leaderboards on math, coding, and science.

What they can't do is stop. Depth stays fixed no matter what the question needs. Chen et al. [2024] measured this plainly: asking whether 0.9 or 0.11 is larger sets off 19–42 seconds of deliberation. Distilled models inherit the habit. In OpenThoughts-114k-math [OpenThoughts, 2025], the easiest item in the whole set, where Kiyana hands half of her 24 grapes to a friend, draws 285 tokens of reasoning to establish that $24 \div 2 = 12$.

This is more than an eyesore. It shows up as inflated inference bills, added latency, and a genuine obstacle to shipping these models anywhere the budget is tight.

*Correspondence to: gaganp0099@gmail.com. Model, dataset, and code available at https://huggingface.co/Eunice-Labs/aria-stage1.

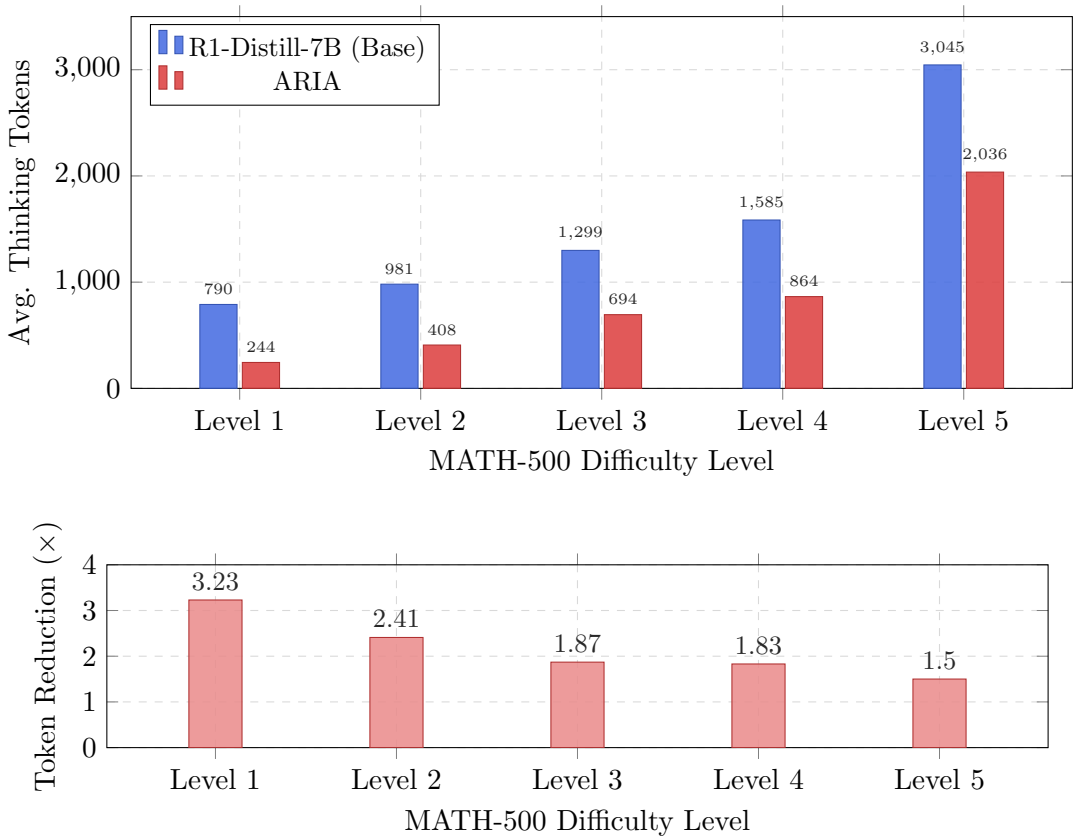

Figure 1: **ARIA learns adaptive reasoning depth.** *Top:* Average thinking tokens per difficulty level on MATH-500 for the base model (blue) and ARIA (red). *Bottom:* Token reduction ratio by difficulty level. ARIA compresses reasoning 3.2× on easy problems while preserving near-full reasoning depth on hard problems, demonstrating learned difficulty-awareness without explicit difficulty signals.

Existing approaches. CoT-Valve [Ma et al., 2025] locates a direction in LoRA parameter space that shortens the chain, but a human still has to dial in a scaling knob at inference. DiffAdapt [DiffAdapt, 2025] bolts on a separate MLP probe that grades difficulty before the model starts. Sonata [Sonata, 2025] leans on self-consistency as a stand-in for difficulty. Each works. Each also introduces a new moving part: a classifier here, an RL loop there, an extra component at inference.

Our approach. Our question was whether the model could absorb the mapping straight from data. ARIA builds a training set in which reasoning length tracks difficulty. Easy problems come with tight, compressed traces of roughly 164 words. Medium problems get a middle treatment, around 469 words. Hard problems keep their sprawling originals at about 7,640 words. Fine-tune on that distribution and the model picks up the correspondence on its own, with no difficulty labels and nothing extra fed in at inference.

Figure 1 shows the payoff: a 3.2× cut in tokens on Level 1 problems that tapers to 1.5× by Level 5. Brief where brief suffices, deep where the problem forces it.

Contributions.

1. A difficulty-graded compression pipeline that uses an LLM to rewrite traces so reasoning depth scales with problem difficulty.
2. ARIA, a 7B reasoning model that learns to modulate its thinking depth through SFT alone, with nothing added at inference.
3. Evidence that difficulty-adaptive reasoning can surface implicitly from the training distribution, with no difficulty tokens or classifiers in sight.
4. An open release of the model, dataset, and evaluation code.

2 Related Work

Reasoning in LLMs. Chain-of-thought prompting [Wei et al., 2022] established that spelling out intermediate steps helps on hard problems. Self-consistency [Wang et al., 2023] extended it by drawing several reasoning paths and voting for the majority answer. The larger leap arrived with RL-trained reasoning. DeepSeek-R1 [Guo et al., 2025] runs GRPO against rule-based rewards; o1 [OpenAI, 2024] uses RL at scale. Both make the same point: train a model to think longer and it reasons noticeably better.

Knowledge distillation for reasoning. Guo et al. [2025] found that pouring reasoning traces from a 671B MoE teacher into smaller students through SFT works remarkably well. There is a catch. The student soaks up the teacher's verbosity wholesale, learning not only how to reason but how to reason at one fixed length for everything it sees.

The overthinking problem. Chen et al. [2024] were the first to nail the phenomenon down systematically in o1-style models. The survey by Sui et al. [2025] maps the space of remedies and flags SFT on variable-length CoT data as underexplored given how promising it looks. ARIA aims squarely at that gap.

Efficient reasoning methods. CoT-Valve [Ma et al., 2025] isolates a LoRA-space direction that governs chain length and exposes it as an inference-time scale. DiffAdapt [DiffAdapt, 2025] trains an MLP probe to grade difficulty before a strategy gets chosen. TokenSkip [Xia et al., 2025] drops tokens ranked low by an importance score. C3oT [Kang et al., 2025] hands chains to GPT-4 for after-the-fact compression. CoThink [Xu et al., 2025] lets an instruct model sketch an outline that the reasoning model then completes.

Where ARIA fits. Every approach above pays one of three taxes: an inference-time component, an RL stage, or an external model in the loop. ARIA pays none. Its adaptivity falls out of the training distribution and nothing else. CoT-Valve is the nearest neighbor, same LoRA setup, same variable-length traces, but it still asks a human to set the scaling parameter at test time. ARIA simply runs.

3 Method

3.1 Overview

Three steps make up the pipeline: sort traces by difficulty, compress the easy and medium ones, and fine-tune on what comes out. The wager is straightforward. If the training data displays reasoning

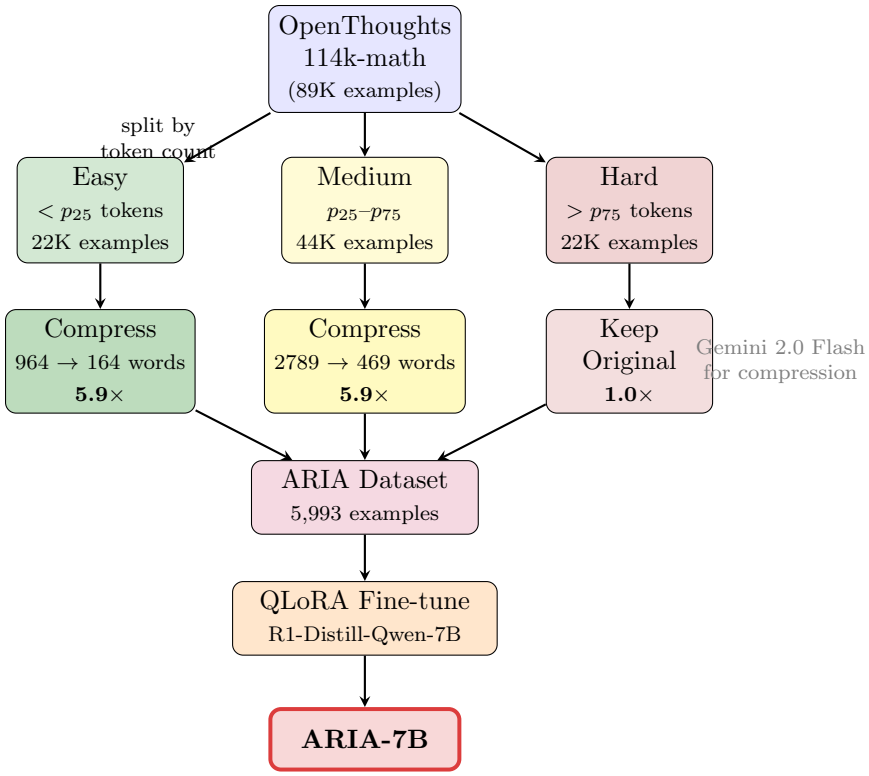

Figure 2: **ARIA method overview.** We partition reasoning traces by difficulty, compress easy/medium traces using Gemini 2.0 Flash while preserving hard traces, and fine-tune on the resulting difficulty-graded dataset. The model learns to calibrate reasoning depth from the data distribution alone.

depth in proportion to difficulty, the model should absorb that relationship without any explicit cue. Figure 2 lays out the whole pipeline.

3.2 Dataset Construction

Source data. Our source is OpenThoughts-114k-math [OpenThoughts, 2025], 89,120 mathematical reasoning traces produced by DeepSeek-R1. Every example pairs a problem with a tagged reasoning trace and a verified solution. Olympiad problems dominate at 79.6%; the remainder splits across competition forums (10.2%), standardized math (7.4%), and AMC/AIME (2.8%).

Difficulty partitioning. We treat the length of the original trace as a proxy for difficulty. It is a rough instrument but a defensible one, since tougher problems genuinely tend to draw out longer chains. Splitting at the 25th and 75th percentiles ($p_{25} = 2{,}820$, $p_{75} = 8{,}648$) carves the set into three tiers: Easy below p_{25} (22,272 examples), Medium between the two cutoffs (44,567 examples), and Hard above p_{75} (22,281 examples). From each tier we draw 2,000 at random.

Semantic compression. Easy and medium traces go through Gemini 2.0 Flash [Google, 2025], which compresses the reasoning while leaving the logical spine in place. This is rewriting, not truncation. Out goes the hedging (the "let me think," the "hmm"), the repeated verification passes, the abandoned branches, the steps too obvious to earn their space. What survives is the reasoning

itself, minus the noise. We target 150 words for easy traces and 400 for medium, and we leave hard traces exactly as they were. Table 1 reports the dataset after compression.

Table 1: Dataset statistics after compression. Both easy and medium traces achieve 5.9× compression.

Difficulty	Count	Orig. Avg (words)	Compressed Avg (words)	Ratio
Easy	1,993	964	164	5.9×
Medium	2,000	2,789	469	5.9×
Hard	2,000	7,640	7,640	1.0×
Total	**5,993**			

3.3 Fine-Tuning

The base model is DeepSeek-R1-Distill-Qwen-7B [Guo et al., 2025], 7.61B parameters distilled from DeepSeek-R1 into Qwen2.5-7B. We adapt it with QLoRA [Dettmers et al., 2023] at rank $r = 64$ and $\alpha = 32$, hitting all seven per-layer projection matrices (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`). That comes to roughly 160M trainable parameters, about 2.1% of the model. Training runs under 8-bit AdamW at a learning rate of 2×10^{-4}, an effective batch size of 16, three epochs, and sequence packing, all on one 24GB NVIDIA RTX 3090 rented through Vast.ai.

4 Experiments

4.1 Evaluation Setup

Benchmarks. We evaluate on two benchmarks:

- **GSM8K** [Cobbe et al., 2021]: first 200 examples of the test set (1,319 total), covering grade-school arithmetic and word problems.
- **MATH-500** [Lightman et al., 2023]: a stratified subset of 243 problems across all five difficulty levels (43/50/50/50/50 per level), covering competition-level mathematics.

Metrics. Three numbers matter to us. **Accuracy** is Pass@1 under greedy decoding. **Avg. Thinking Tokens** counts the tokens spent before the final answer lands. **Reasoning Efficiency Score** (RES = Accuracy/(Avg. Thinking Tokens/1000)) captures the accuracy bought per thousand thinking tokens.

Baseline. The comparison point is DeepSeek-R1-Distill-Qwen-7B left unmodified, run under settings identical to ARIA's down to the system prompt, the decoding parameters, and the generation cap.

4.2 Main Results

Table 2 presents the overall results on both benchmarks.

Table 2: Main results. ARIA improves both accuracy and efficiency on GSM8K, and substantially improves efficiency on MATH-500 with a modest accuracy tradeoff. RES (higher is better) measures accuracy per thousand thinking tokens.

	GSM8K (n=200)			**MATH-500** (n=243)		
Model	Acc (%)	Tokens	RES	Acc (%)	Tokens	RES
R1-Distill-7B	76.0	449.5	169.1	77.0	1552.8	49.6
ARIA	**78.5**	**203.7**	**385.4**	72.8	**847.5**	**85.9**
Δ	+2.5	2.2× ↓	2.3× ↑	−4.2	1.8× ↓	1.7× ↑

On GSM8K, ARIA picks up 2.5 points of accuracy on 2.2× fewer thinking tokens, more than doubling RES. The way we read it, verbose overthinking is actively harmful on easy problems. Every extra token is another chance to second-guess a step that was already right or to drift into a dead end. Short traces skip the temptation.

On MATH-500, tokens fall by 1.8× at a cost of 4.2 accuracy points. The headline figure misleads, though. The breakdown by difficulty tells the real story.

4.3 Per-Difficulty Analysis

Table 3 and Figure 1 carry the central result: ARIA's behavior shifts in step with difficulty.

Table 3: Per-difficulty results on MATH-500. ARIA achieves large token reductions on easy problems with no accuracy loss, while maintaining near-baseline accuracy on hard problems. The monotonically decreasing reduction ratio demonstrates learned adaptive reasoning.

Level	**n**	**Base Acc**	**ARIA Acc**	**Base Tok**	**ARIA Tok**	**Reduction**
Level 1 (easiest)	43	83.7%	83.7%	790.1	244.2	**3.23**×
Level 2	50	80.0%	78.0%	981.1	407.6	2.41×
Level 3	50	80.0%	74.0%	1299.4	693.5	1.87×
Level 4	50	74.0%	66.0%	1585.0	863.7	1.83×
Level 5 (hardest)	50	68.0%	64.0%	3045.1	2036.1	1.50×

A few patterns stand out:

Adaptive token reduction. The reduction ratio slides monotonically from 3.23× at Level 1 to 1.50× at Level 5. This is exactly the behavior we set out to find. The model learned to keep it short on easy problems and to stretch out on hard ones, all without seeing a single difficulty label at training or inference time.

Accuracy preservation on easy problems. At Level 1, ARIA matches the base model's accuracy to the decimal (83.7%) on 3.23× fewer thinking tokens. The compressed traces clearly kept every logical step that mattered for easy problems.

Accuracy-efficiency tradeoff on harder problems. Across Levels 3 and 4, ARIA gives up 6 to 8 accuracy points for a 1.8 to 1.9× token cut. The likely reading is that compression took out steps these medium-hard problems actually needed, which leaves room to improve.

Graceful degradation on hard problems. At Level 5, accuracy slips just 4 points (68.0% → 64.0%) even though the model carries a general bias toward compression. The modest 1.5× reduction says it correctly reads these problems as ones that warrant the long form.

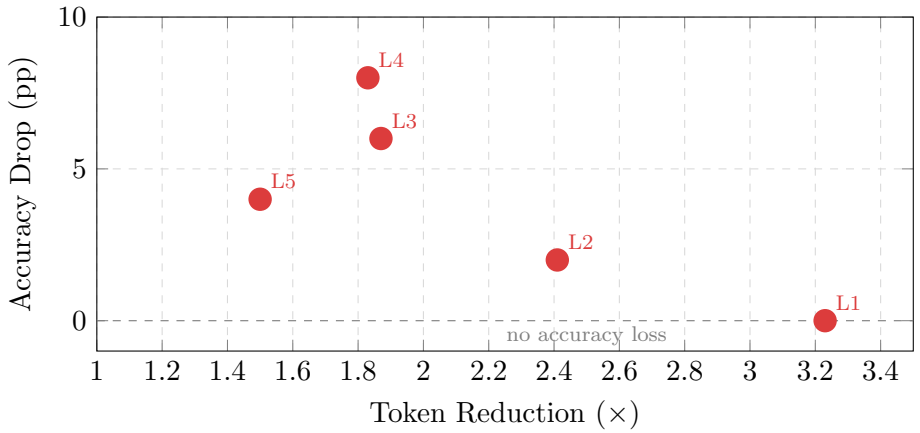

Figure 3: **Accuracy-efficiency tradeoff by difficulty level.** Each point represents one MATH-500 difficulty level. Level 1 achieves 3.23× reduction with zero accuracy loss (on the dashed line). Higher-level problems trade modest accuracy for token efficiency.

5 Analysis

5.1 Case Studies

Three examples bring ARIA's adaptive behavior into focus.

Easy problem: concise reasoning. On the Level 1 problem "What is the difference between the positive square root of 64 and the cube root of 64?", ARIA produces a 60-token thinking trace:

ARIA — Level 1 (60 thinking tokens)

The positive square root of 64 is 8 because $8^2 = 64$. The cube root of 64 is 4 because $4^3 = 64$. Therefore, the difference is $8 - 4 = 4$.

Answer: $\boxed{4}$ ✓

It spots the two computations that matter, runs them, and lands the answer with no hedging, no re-checking, and no detour through alternatives.

Hard problem: deep reasoning. Given a Level 5 geometry problem built around a hot-air balloon, a set of ropes, and an optimization (stated in full in the Appendix), ARIA writes a 5,615-token trace. It sets up coordinates, derives three simultaneous equations, solves them symbolically, parametrizes a line segment, computes a projection, and checks its work, arriving correctly at $\boxed{160}$ meters. The depth is still there when the problem calls for it.

Side-by-side comparison. On the Level 3 problem "Find the product of $6_8 \cdot 7_8$. Express your answer in base 8":

Base R1-Distill-7B — 4,926 thinking tokens, 130s

Converts to decimal (6, 7), multiplies to get 42, converts to base 8 (52_8). Then attempts binary multiplication as "cross-validation," makes errors, debugs for ∼2,000 tokens, tries three different addition alignments, eventually confirms the original answer. Concludes with reflections on "fundamental concepts across different numeral systems."

ARIA — 147 thinking tokens, 10s

Converts to decimal: $6_8 = 6_{10}$, $7_8 = 7_{10}$. Multiplies: $6 \times 7 = 42_{10}$. Converts back: $42 \div 8 = 5$ remainder 2, so $42_{10} = 52_8$.

Answer: $\boxed{52_8}$ ✓

Both land on the right answer. The base model spends 4,926 tokens and 130 seconds to get there; ARIA needs 147 tokens and 10 seconds, a **33.5**× cut in tokens and a **13**× speedup on this single problem.

5.2 Discussion

Why does GSM8K accuracy improve? The base model lands at 76.0% on GSM8K; ARIA reaches 78.5%. More tokens is not always more accuracy. On an easy problem a long chain just gives the model more room to talk itself out of a correct answer, re-open settled steps, or wander off. Trim the chain and that risk shrinks. The model commits sooner and is right more often.

The Level 3–4 accuracy gap. The accuracy hit is bigger at Levels 3 and 4 (6 to 8 points) than at Level 5 (4 points), which is the reverse of the naive expectation. Our best guess points to the compression boundary. Problems sitting near the medium-hard cutoff received compressed traces in training, and a few of the steps that got cut were load-bearing. Level 5 problems kept their full traces, so the model never lost the recipe for them. Finer tiers, or compression targets set per problem, would probably close the gap.

6 Limitations

A few points worth stating plainly.

The difficulty proxy is noisy. Trace length tracks difficulty on average, but a hard problem that happens to draw a short trace gets filed as easy and trained on compressed data it never should have seen.

Everything here is math. Whether any of this carries over to code, science, or commonsense reasoning is unknown. In principle it ought to generalize; we have not checked.

We only ran this at 7B. A smaller model may not have the capacity to absorb the difficulty mapping, and a larger one might behave differently. Open question.

The compression is only as good as Gemini 2.0 Flash. Some traces almost certainly lost useful steps in compression, which remains our leading explanation for the Level 3–4 gap.

The ablations are missing. We ran no comparison against random compression, uniform compression, or prompt-based baselines. Of everything we left undone, this is the one that matters most.

Evaluation scale is limited. Two hundred GSM8K problems and 243 from MATH-500 is not a lot. With only 43 to 50 problems per level, the per-level numbers carry wide confidence intervals.

7 Conclusion

ARIA rests on a simple claim: train a reasoning model on data where thinking depth matches difficulty, and it learns to match them too. The lever is the data distribution, nothing more.

Just 5,993 examples turned out to be enough for a clear adaptive gradient, a 3.2× token cut on easy problems with no loss in accuracy, easing to 1.5× on the hardest. GSM8K accuracy rose even as the token count halved.

The wider lesson is that overthinking is, in part, a data problem. Feed a model uniformly verbose traces and it concludes that every problem deserves a verbose answer. That conclusion is not baked into the architecture or the objective. The data taught it, and ARIA shows the lesson can be taught the other way.

Model, dataset, and code at https://huggingface.co/Eunice-Labs/aria-stage1.

References

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+3=? On the overthinking of o1-like LLMs. *arXiv preprint arXiv:2412.21187*, 2024.

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.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized language models. *Advances in Neural Information Processing Systems*, 2023.

DiffAdapt: Difficulty-adaptive reasoning for token-efficient LLM inference. *arXiv preprint arXiv:2510.19669*, 2025.

Google DeepMind. Gemini 2.0: Our next-generation model. Technical report, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Yifei Kang, Xiao Sun, Liangchen Chen, and Wei Zou. C3oT: Generating shorter chain-of-thought without compromising effectiveness. In *AAAI Conference on Artificial Intelligence*, 2025.

Hunter Lightman, Vineet Kosaraju, Yura Burda, et al. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. CoT-Valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025.

OpenAI. Learning to reason with LLMs. Technical report, 2024.

Open-R1 Project. OpenThoughts-114k: Open reasoning traces for language model training. https://huggingface.co/datasets/open-r1/OpenThoughts-114k-math, 2025.

Qwen Team. QwQ: Reflect deeply on the boundaries of the unknown. Technical report, 2024.

Adaptive thinking: Large language models learn when to think. *arXiv preprint*, 2025.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, et al. Stop overthinking: A survey on efficient reasoning for large language models. *Transactions on Machine Learning Research*, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, et al. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Heming Xia, Yongqi Li, Chak Tou Leong, Wenjie Wang, and Wenjie Li. TokenSkip: Controllable chain-of-thought compression in LLMs. *arXiv preprint arXiv:2502.12067*, 2025.

CoThink: Token-efficient reasoning via instruct models guiding reasoning models. *arXiv preprint arXiv:2505.22017*, 2025.