Reasoning with `OmniThought`: A Large CoT Dataset with Verbosity and Cognitive Difficulty Annotations

Anonymous ACL submission

Abstract

Tasks such as mathematical problem solving and coding require models to leverage chain-of-thought (CoT) processes, enabling human-like reasoning strategies. However, the advancement of large reasoning models (LRMs) is hindered by the lack of comprehensive CoT datasets. Existing resources often fail to provide extensive reasoning problems with coherent CoT processes distilled from multiple teacher models, and do not account for multi-faceted properties describing the internal characteristics of CoTs. To address these challenges, we introduce `OmniThought`, a large-scale dataset featuring **2 million** CoT processes generated and validated by multiple powerful LRMs. Each CoT process in `OmniThought` is annotated with novel Reasoning Verbosity (RV) and Cognitive Difficulty (CD) scores, which characterize the appropriateness of CoT verbosity and the cognitive difficulty level for models to comprehend these reasoning processes. We further establish a self-reliant pipeline to curate this dataset. Extensive experiments using Qwen2.5 and Qwen3 of various sizes demonstrate the positive impact of our RV and CD scores on LRM training effectiveness. Based on the `OmniThought` dataset, we train and release a series of high-performing LRMs, specifically equipped with enhanced reasoning abilities and optimized CoT output length. Our contributions advance the development of LRMs across different scales for solving complex reasoning tasks. [1]

1 Introduction

In recent years, the field of NLP has been profoundly transformed by the emergence of large language models (LLMs) (Zhao et al., 2023), which have demonstrated outstanding proficiency across a wide range of NLP tasks. Of particular interest are large reasoning models (LRMs) (Xu et al., 2025), such as OpenAI's o1[2], DeepSeek-R1 (DeepSeek-AI, 2025), and QwQ-32B[3], which excel at addressing tasks such as mathematical problem solving and code generation through slow thinking processes.

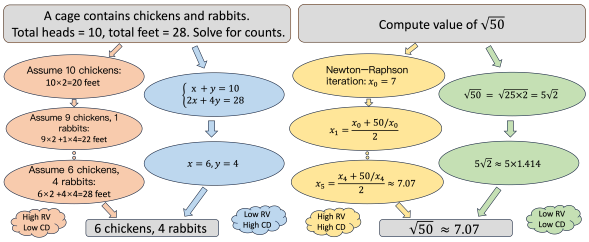


Figure 1: A motivating example of CoTs with different Reasoning Verbosity (RV) and Cognitive Difficulty (CD) levels. For simplicity, only key steps in these CoTs are presented.

The impressive performance of LRMs is largely attributed to chain-of-thought (CoT) reasoning (Wei et al., 2022), which empowers models to break down intricate problems into intermediate steps, closely emulating human problem solving (Hao et al., 2023; Yan et al., 2023). This capability supports diverse applications, including science education (Cohn et al., 2024), robotic control (Zawalski et al., 2024), and clinical assessment (Gu et al., 2025), among others. Producing effective LRMs necessitates CoT-based training, which typically integrates both supervised fine-tuning (SFT) and reinforcement learning (RL). Through SFT, models acquire the ability to follow explicit reasoning patterns, treating the CoT process and solutions as outputs, while RL enables optimization of reasoning strategies through iterative feedback and exploration (Kazemnejad et al., 2024; Schulman et al., 2017; Shao et al., 2024; Trung et al., 2024).

Despite these advances, progress in LRMs is hindered by the absence of large-scale, comprehensive

[1] All source codes, models, and datasets will be released.

[2] https://openai.com/o1/

[3] https://qwenlm.github.io/blog/qwq-32b/

CoT datasets. Many open-source data resources, often directly distilled from powerful LRMs, generally lack sufficient reasoning problems with detailed CoT processes generated by multiple teacher models, and do not incorporate multifaceted properties describing the intrinsic characteristics of CoTs. Beyond dataset size, we emphasize several aspects:

Comprehensive Quality Assessment. Prior studies (Jacovi et al., 2024) have shown that logical errors in CoT-based training sets can negatively impact the performance of LRMs. Examining the logical correctness of CoT processes within the "LLM-as-a-judge" paradigm (Gu et al., 2024) is essential, as human annotation is impractical at scale. Moreover, collecting and releasing CoT processes of varying quality can also be valuable for training alternative algorithms for LRMs, such as direct preference optimization (DPO) (Rafailov et al., 2023) and reward model training (Ouyang et al., 2022). To our knowledge, these aspects are not fully explored in existing dataset literature.

Reasoning Verbosity (RV). There are no ground-truth CoTs for reasoning problems; many candidate CoTs can be valid outputs. Nevertheless, using excessively long CoTs for training can impair the reasoning performance of resulting models (Yang et al., 2025; Luo et al., 2025). Recent studies suggest that optimal CoT lengths exist for different reasoning problems (Yang et al., 2025). In this work, we propose and compute a Reasoning Verbosity (RV) score for each problem-CoT pair, guiding users to select subsets of CoTs for training LRMs with length-optimal outputs. This practice enhances model's performance and avoids excessive reasoning, without requiring costly test-time scaling.

Cognitive Difficulty (CD). The RV score of a CoT is measured considering the complexity of the problem, independent of the target models. As our dataset is constructed to train LRMs of various sizes, CoTs should also be evaluated according to the capacities of the LRMs to be trained. We propose that cognitive difficulty affects the suitability of CoTs for different LRMs. For example, smaller models can exhibit distinct capabilities and cognitive trajectories compared to larger counterparts when tackling reasoning tasks (Yan et al., 2023; Zhang et al., 2024; Chen et al., 2025). Thus, overly difficult reasoning steps, although logically correct, may not be suitable for smaller models to acquire necessary abilities. We suggest that cognitive difficulty can guide the selection of appropriate CoTs

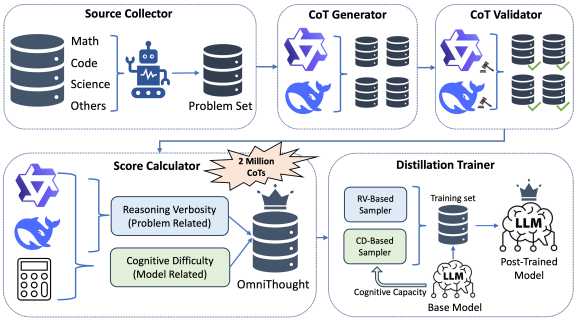


Figure 2: Our framework, including dataset construction pipeline and training procedure. We leverage DeepSeek-R1, DeepSeek-R1-0528 and QwQ-32B as teacher models, but other sufficiently strong models may also serve as teachers.

for more effective LRM training based on model's reasoning ability. For instance, relatively simpler CoTs are more suitable for small models, fitting their cognitive capacity and enabling more economical inference. Examples of CoTs with different RV and CD levels are in Figure 1.

We introduce `OmniThought`, which comprises challenging problems with more than **2 million** CoTs, generated and validated by powerful LRMs and annotated with RV and CD scores. To create the dataset, we propose a self-reliant pipeline shown in Figure 2. First, `Source Collector` aggregates reasoning problems from various high-quality sources, ensuring broad and varied task coverage. Next, `CoT Generator` employs multiple LRMs to generate CoTs for each task, with `CoT Validator` verifying logical correctness. Finally, `Score Calculator` assigns RV and CD scores indicating CoT characteristics.

In our experiments, we validate the effectiveness of `OmniThought` through extensive benchmarking. We utilize Qwen2.5 and Qwen3 models of varying parameter sizes as backbones. The results show that RV and CD scores have a positive impact on LRM training. Based on CoT sampling from `OmniThought`, we further train and release a series of high-performing LRMs. These models excel in reasoning with optimal CoT length, even outperforming the DeepSeek-R1-Distill series trained on proprietary datasets (DeepSeek-AI, 2025). Our main contributions are:

- **Dataset:** We construct `OmniThought`, a large-scale dataset of reasoning problems containing over **2 million** CoT solutions, generated and validated by multiple teacher models, and annotated with RV and CD scores reflect-

ing their characteristics.

- **Method:** We propose a self-reliant pipeline to curate `OmniThought` without manual intervention, and introduce a training set construction method that leverages CD, RV, and the cognitive capacity of the target model. Furthermore, we provide guidelines for applying this method to both SFT and RL.

- **Models:** We comprehensively evaluate the value of `OmniThought` through extensive experiments and release a series of LRMs that excel in reasoning with optimal CoT length.

2 Related Work

The generation of chain-of-thought (CoT) processes has attracted significant interest in recent years, as these processes enhance the reasoning capabilities of large language models (LLMs). Early work relied on manual annotation by domain experts to create gold-standard CoTs for benchmarking (Huang et al., 2024; Gao et al., 2024). This approach ensures high-quality outputs but is labor-intensive and lacks scalability.

Automatic methods focus on leveraging LLMs through prompt engineering (e.g., *Let's think step by step*), tapping into the models' latent abilities to generate detailed reasoning processes (Wei et al., 2022; Yao et al., 2023; Liu et al., 2023; Wang et al., 2023). Although such techniques can rapidly produce CoTs, they are constrained by model biases and are highly dependent on prompt design. The advent of powerful large reasoning models (LRMs), such as DeepSeek-R1 (DeepSeek-AI, 2025), has led to approaches that directly generate CoT processes (Cai et al., 2025; Team, 2025). Despite these developments, major challenges remain in generating high-quality CoT data, particularly with respect to accounting for varying difficulty levels and alignment with model capabilities.

Recently, several works have revealed that generated CoT processes are often suboptimal in multiple respects. For example, (Yang et al., 2025) finds that LRMs sometimes produce excessively lengthy CoTs, which can hinder reasoning performance. Meanwhile, (Chen et al., 2025) shows that stronger models benefit from detailed reasoning, whereas less powerful models achieve better results with simpler CoT supervision. Some projects have begun to explore concise yet effective CoTs,

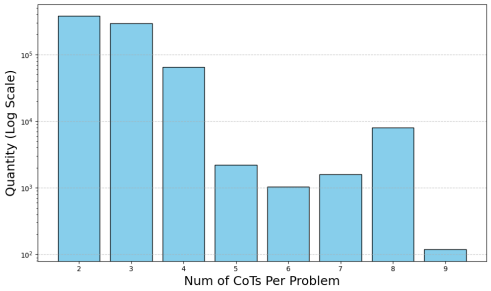


Figure 3: Distribution of CoT processes per problem.

such as those implemented in DeepSeek-V3-0324[4]. Subsequent research has investigated switchable reasoning modes, enabling models to alternate between standard outputs and explicit CoT reasoning, as in Llama-Nemotron (Bercovich et al., 2025) and Qwen3[5]. In contrast, our work suggests that lengthy and concise CoTs represent fundamentally different reasoning modes, and that models should automatically choose the optimal CoT form based on both task complexity and their cognitive capacity. Challenging problems may require deeper reasoning, while simpler ones might not.

3 Dataset Construction

In this section, we describe the construction process of the proposed `OmniThought` dataset (as illustrated in Figure 2). The pipeline comprises four main components: `Source Collector`, `CoT Generator`, `CoT Validator`, and `Score Calculator`. `Source Collector` gathers reasoning problems, `CoT Generator` and `CoT Validator` respectively generate and validate CoT processes using multiple teachers, and `Score Calculator` computes RV and CD scores.

3.1 Basic Modules

Initially, the `Source Collector` curates a diverse collection of reasoning problems from various domains, including math, code, and science. We then perform deduplication and decontamination on the collected problems against evaluation benchmarks. Specific details on the construction of the problem set are provided in Appendix A. This process culminates in the creation of the initial reasoning problem set for `OmniThought`.

Next, the `CoT Generator` employs DeepSeek-R1, DeepSeek-R1-0528, and QwQ-32B as teacher models to generate multiple reasoning

[4] https://huggingface.co/deepseek-ai/DeepSeek-V3-0324

[5] https://qwenlm.github.io/blog/qwen3/

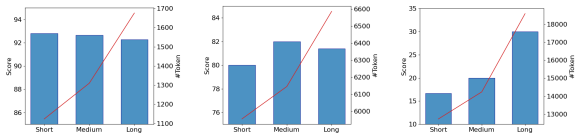
processes for each problem sourced by the `Source Collector`. Since simple problems offer limited improvement to a model's reasoning ability, we randomly filter out those with a CoT token count less than 3,000 to ensure a balanced difficulty distribution.

To ensure high quality, the `CoT Validator` applies various methods to evaluate multiple facets of each CoT, including logical correctness and final answer accuracy. For code problems, correctness is validated by executing test cases. For math and science problems, we employ a hybrid approach that combines rule-based systems with an "LLM-as-a-judge" framework for verification. The prompt template is provided in Appendix I. As discussed previously, releasing CoTs of lower quality can be valuable, as they can support DPO algorithms (Rafailov et al., 2023) and reward model training (Ouyang et al., 2022). Therefore, we retain a generated CoT process in the dataset unless it produces an incorrect answer. The validation results are added as metadata.

Ultimately, the `OmniThought` dataset comprises more than **2 million** CoT processes for 708K reasoning problems. We ensure that each problem in the dataset has at least two validated, correct CoT processes. The distribution of CoT processes per problem is shown in Figure 3. The sample data format is provided in Appendix B.

3.2 Reasoning Verbosity

CoT processes naturally involve self-reflection, prompting models to undergo multiple rounds of introspection and correction during reasoning (Zou et al., 2023). This approach helps reduce errors on complex problems but can lead to "overthinking" on simpler ones (Wang et al., 2025), such as performing unnecessary checks for "1 + 1 = ?". Such overthinking wastes computational resources and can decrease reasoning accuracy. Specifically, for a given problem, the length of its CoT should align with the problem's difficulty, reflecting the "Reasoning Verbosity" (RV) of the CoT. This phenomenon has also been observed in concurrent work (Yang et al., 2025). However, it remains unclear whether and how RV impacts the training effectiveness of LRMs. We formally define RV grading criteria on a scale from 0 to 9:



(a) GSM8K (b) MATH500 (c) AIME24

Figure 4: Comparison using CoTs from different RV levels as training sets. Blue bars and red lines represent Pass@1 and average output token counts, respectively. We use a temperature of 0.6, a top-p value of 0.95, and generate 32 responses per query to estimate Pass@1.

Grading Criteria for Reasoning Verbosity

0-1: Minimal verbosity; straightforward expression with little or no elaboration.
2-3: Clear and concise reasoning with necessary explanations.
4-5: Moderate verbosity, with detailed explanations and thorough reasoning.
6-7: Extensive verbosity, including comprehensive justification and exploration of complex connections.
8-9: High verbosity, featuring deep, exhaustive exploration; includes extensive elaboration, nested justifications, and consideration of counterarguments or alternative perspectives.

We observe that, in addition to the "LLM-as-a-judge" paradigm, the token count of a CoT process itself provides useful information for assessing verbosity. However, these two assessment methods do not always yield consistent evaluations. For instance, a CoT process might be verbose in terms of steps (having a large number of reasoning steps), but if each step is relatively simple, the total number of output tokens may not necessarily be large. CoTs of equal length may appear overly verbose on simple problems yet excessively concise on very difficult ones.

Therefore, the information conveyed by CoT length is one-dimensional and cannot accurately assess CoT redundancy. We choose to incorporate CoT length into the final Reasoning Verbosity (RV) score. We normalize L to a scale from 0 to 9 (denoted as L_{norm}) as follows:

$$L_{\text{norm}} = \mathcal{K} \cdot \frac{\log(L - L_{\text{min}} + 1)}{\log(L_{\text{max}} - L_{\text{min}} + 1)} \quad (1)$$

where L_{min} and L_{max} are the minimum and maximum token counts of CoT processes in our dataset, and $\mathcal{K}$ is the grading scale (i.e., $\mathcal{K} = 9$ in our case). Next, L_{norm} and the vanilla RV score judged by the LLMs are combined to compute the final RV score S_{RV}, defined as $S_{RV} = \text{round}\left(\alpha L_{RV} + (1-\alpha) L_{\text{norm}}\right)$, where $\alpha \in (0, 1)$ reflects the relative importance of the two fea-

tures (empirically set to 0.5), and L_{RV} is the model-judged score (see the prompt template in Appendix I). CoT examples for different RV levels are provided in Appendix J.

Experimental Verification. To verify the effectiveness of RV scores in LRM training, we randomly sample a subset of 10K problems, each with three CoTs categorized into distinct RV levels. In this subset, the difference in RV scores between adjacent levels exceeds 3. Based on this design, we construct three training sets containing the same problems but differing in RV scores. We perform SFT training with Qwen2.5-7B-Instruct under identical configurations (see Appendix D), resulting in three models categorized as short, medium, and long.

We evaluate models on GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021b), and AIME24[6]. As shown in Figure 4, the models effectively assimilate characteristics from their respective SFT training sets. On the relatively simple GSM8K tasks, all models perform similarly; an increased output token count provides no accuracy improvement and even cause a slight decline. On the medium-difficulty MATH500 tasks, accuracy initially increases with token count, then declines, with the medium model achieving the highest accuracy. For challenging AIME24 tasks, the long model achieves the highest score; accuracy increases with token count, with significant improvement once a certain threshold is surpassed.

These results confirm our hypothesis: for difficult problems, longer CoT processes can correct model errors and effectively improve accuracy. However, for simple tasks, excessive reasoning and verification not only increase computational costs but may also impair accuracy. Therefore, CoT verbosity should match problem difficulty. With this approach, we can construct training sets containing CoTs with appropriate RV levels according to problem difficulty, thereby maximizing computational efficiency while ensuring high accuracy.

3.3 Cognitive Difficulty

We argue that in the dataset, CoT's difficulty should align with the cognitive capacity of the model to be trained. Significant differences in model parameter sizes lead to diverse reasoning trajectories between large and small models (Yan et al., 2023; Zhang et al., 2024). Smaller models, constrained by their parameter limits, tend to employ basic approaches to solve problems, while larger models, possessing more advanced cognitive abilities, may utilize higher-level techniques. For example, when calculating the area of a triangle given its coordinates, a small model may use straightforward geometric decomposition, while a larger model might apply more sophisticated vector-based algebra.

Model	Avg. Score
DS-R1-Distill-Qwen-1.5B	4.5
DS-R1-Distill-Qwen-7B	6.2
DS-R1-Distill-Qwen-32B	7.3

Table 1: Comparison of average CoT difficulty on MATH500 for different model scales.

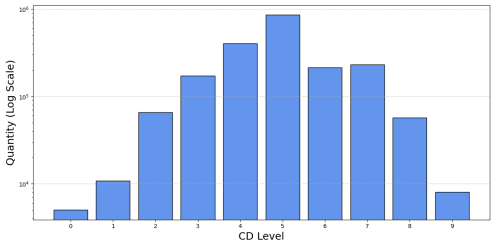


Figure 5: Distribution of CD scores for CoTs.

Experimental Verification. To test this hypothesis, we conduct experiments using three models from the DeepSeek-R1-Distill series (DeepSeek-AI, 2025): 1.5B, 7B, and 32B. We evaluate these models on MATH500 (Hendrycks et al., 2021b). For each model's CoT processes, DeepSeek-R1 assigns a difficulty rating from 0 to 9 based on "Cognitive Difficulty"(CD) of reasoning (see criteria below). Each CoT is evaluated three times, and scores are averaged. We then average these ratings across all examples per model to obtain their overall difficulty scores shown in Table 1. The results show that CoT difficulty increases with model size, indicating that stronger models possess enhanced reasoning and cognitive capacities. As a result, difficult CoTs may be unsuitable for training models with limited cognitive capacities. It is thus essential to select CoT processes that match the model's cognitive trajectory for effective reasoning improvement, a strategy akin to "teaching according to the student's ability." In our framework, the CD score reflects the difficulty of the methods used in the CoT process. Grading criteria, on a scale from 0 to 9, are as follows. The prompt template and CoT examples for each CD level are provided in Appendix I and Appendix J.

[6] https://artofproblemsolving.com/wiki/index.php/2024_AIME_I

Strategy	AIME2024	MATH500	GPQA Diamond	LiveCodeBench V2	Average
Random	16.67	80.6	36.36	31.31	41.24
RV Optimal	26.67	83.2	40.40	34.44	46.18
CD Optimal	33.33	83.8	39.90	36.10	48.28
Combined	**36.67**	**84.4**	**40.91**	**36.59**	**49.64**

Table 2: The impact of different CoT selection strategies on reasoning performance.

Grading Criteria for Cognitive Difficulty

0-1: Elementary facts or a single trivial operation.
2-3: Multi-step arithmetic, explicit enumeration, basic rule chaining.
4-5: Early undergraduate logic/algebra; one non-obvious insight.
6-7: Advanced undergraduate techniques (determinants, dynamic programming, code reasoning, etc.).
8-9: Graduate-level abstraction, nested proofs, intricate algorithmic analysis.

In `OmniThought`, we score all validated, correct CoT processes. The CD score distribution is shown in Figure 5. The distribution appears Gaussian-like, peaking at levels 4–5 and tapering towards both extremes. This finding also indicates that highly capable models, such as DeepSeek-R1, can produce extremely difficult CoT processes. When performing knowledge distillation, models with limited cognitive capacity are unlikely to effectively comprehend these processes. Therefore, given an original CoT dataset and a student model, one can filter the dataset according to the student's cognitive capability, thereby effectively improving the student's abilities.

3.4 Analysis of Grading of RV and CD

We conducted experiments to explore various rating systems and the specific content of definitions, ultimately establishing the RV and CD evaluation frameworks presented above. A detailed analysis is provided in Appendix F.

Additionally, to assess the reliability and stability of QwQ-32B as the judge model for RV and CD, we conducted experiments comparing its ratings against human evaluations and those of other judge models. The detailed analysis is presented in Appendix F.

3.5 Further Explorations

Building on the RV and CD scores above, we propose the following research questions.

RQ1: *Which is more significant for training: RV or CD, and is a combined approach superior?*

To investigate this, we conduct the following study. We randomly sample a subset of 10K problems from `OmniThought`, each with at least four CoTs, ensuring that the maximum differences in RV and CD among these CoTs both exceed 4. Under these conditions, the CoTs for each problem exhibit considerable differences in both verbosity and difficulty, making them ideal experimental subjects. Using an initial LLM, we employ four strategies to select the SFT training set: RV-optimal, CD-optimal, combined, and random selection. In the RV-optimal approach, we define an RV range and select CoTs best fitting this range. Conversely, the CD-optimal method uses a CD range to construct the training set. The combined approach specifies both RV and CD ranges and equally weights conformity to both when selecting CoT processes. Random selection simply samples one CoT process per problem. We use Qwen2.5-7B-Instruct as the base model, with RV range 3–5 and CD range 0–6. After constructing the four datasets, we train four models with identical SFT configurations (see Appendix D). We then evaluate these models on AIME24, MATH500, GPQA-Diamond (Rein et al., 2023), and LiveCodeBench V2 (Jain et al., 2025); results are listed in Table 2. Compared to random selection, both RV-based and CD-based selections yield greater improvements, with the combined selection achieving the best scores. Between RV and CD, CD is more impactful for training. By adjusting RV and CD ranges, we can assemble training sets that better match a model's inherent cognitive abilities.

Dataset	Model w/o DPO	Model w/ DPO
AIME2024	36.67 (12,248)	36.67 (10,352)
MATH500	84.4 (3,676)	86.2 (3,108)
GPQA-D	40.91 (6,295)	42.93 (5,635)
LCB V2	36.59 (8,599)	39.9 (7,658)

Table 3: Performance comparison before and after DPO training. Numbers in parentheses indicate the average output tokens per problem.

RQ2: *How does our dataset benefit other training algorithms beyond SFT?*

Beyond SFT, `OmniThought` can also be effectively applied to other training algorithms. Using the DPO (Rafailov et al., 2023) as an example, criteria can be defined for "chosen" and "rejected" CoTs to construct a preference-pair dataset. For

Reward Setting	MATH-500	AIME2024
Qwen2.5-7B-Instruct (Raw)	73.6	10.0
Vanilla GRPO	78.8	13.3
GRPO+RV	79.0	16.7
GRPO+CD	80.8	20.0
GRPO+RV+CD	**81.4**	**23.3**

Table 4: Performance of our approach (with Qwen2.5-7B-Instruct as the backbone). Note that none of the models have undergone CoT-based SFT to clearly show the performance in the RL phase.

the same 10K-problem subset created for RQ1, we treat CoTs with RV in the range 3–5 as chosen, and those with the maximum RV as rejected, thus forming 10K preference pairs. Starting from the SFT model from RQ1, we apply the DPO algorithm and evaluate the resulting model on four benchmarks, also recording output token counts; the results are shown in Table 3. We observe that: (i) model scores further improve on MATH500, GPQA-D, and LiveCodeBench V2; (ii) results for AIME24 remain unchanged; and (iii) the output token counts of the new model decrease for all benchmarks. Thus, by leveraging desirable and undesirable CoT scores as chosen and rejected responses, we can further refine a model's output preferences without sacrificing accuracy.

As for reward models, we employ the GRPO algorithm (Shao et al., 2024) and incorporate RV- and CD-based rewards. Specifically, let $f_{\text{RV}}(x)$ and $f_{\text{CD}}(x)$ be the predicted RV and CD scores from trained reward models based on our dataset. These predicted scores are further normalized to $[0, 1]$. Combined with the conventional *accuracy* and *format* rewards (denoted as R_{acc} and R_{fmt}), the overall reward function is defined as: $R = R_{\text{fmt}} + R_{\text{acc}} + \lambda_{RV} R_{\text{RV}}(x) + \lambda_{CD} R_{\text{CD}}(x)$ where λ_{RV} and λ_{CD} are hyperparameters. We evaluate the effectiveness of these additional rewards by comparing against vanilla GRPO; the results are shown in Table 4. As demonstrated, GRPO augmented with RV/CD-based rewards consistently outperforms vanilla GRPO, further supporting our hypothesis that RV and CD scores derived from teacher models can benefit RL training, which is a promising direction for future work.

4 Training Strong LRMs

We describe our procedures for developing a series of LRMs based on `OmniThought`, each endowed with strong reasoning abilities and optimal CoT length.

4.1 CoT Selection Guidelines

To achieve this goal, the CoTs selected from the entire dataset follow two key criteria: (i) the RV score should be as close to optimal as possible, and (ii) the CD score should align with the student model's capacity. Denote $\mathcal{D} = \{(x, y_{CoT}, y, S_{RV}, S_{CD})\}$ as the proposed dataset, where x, y_{CoT}, y, S_{RV}, and S_{CD} represent the problem, the CoT process, the answer, and the RV and CD scores, respectively. For any problem x, a set of CoTs with their metadata is denoted as $\mathcal{D}(x)$. We further denote the model's capacity score as μ_{CD}[7]. If $S_{CD} \leq \mu_{CD}$, we select suitable CoTs according to: $P_1(y_{CoT}) \propto \max_i \left| S_{CD}^{(i)} - \mu_{CD} \right|$. Otherwise,

$$P_1(y_{CoT}) \propto \max_i \left| S_{CD}^{(i)} - \mu_{CD} \right| - (S_{CD} - \mu_{CD}) \,. \tag{2}$$

This means CoTs with $S_{CD} \leq \mu_{CD}$ are assigned an equally high probability of being selected, whereas for those with $S_{CD} > \mu_{CD}$, the greater the deviation from μ_{CD}, the lower the selection probability. Our goal is to select y_{CoT} that lies within the model's cognitive capacity (μ_{CD}) more frequently. Additionally, CD and RV scores are naturally correlated, as more difficult solutions often require more verbose explanations. Therefore, we penalize cases where, for a particular CoT, the gap between CD and RV scores is excessively large. Empirically, we define the second rule as:

$$P_2(y_{CoT}) \propto \left(\max_i \left| S_{CD}^{(i)} - S_{RV}^{(i)} \right| \right) - |S_{CD} - S_{RV}| \,. \tag{3}$$

Based on these two rules, suitable CoTs can be sampled to construct training sets. Readers are referred to the detailed sampling process in Appendix C.

4.2 Major Experimental Results

Following the guidelines, we sample suitable CoTs from `OmniThought` to train models via SFT. To verify that our dataset is effective for developing models with different parameter sizes, we train models from the Qwen2.5/Qwen3 series. Notably, due to differences in parameter size, the training sets for the models include distinct CoTs selected by the RV and CD samplers. Detailed experimental settings are presented in Appendix D. We further compare the performance of our models against SOTA models in the open-source community, with

[7] Please refer to the experiments below as examples for determining the value of μ_{CD}.

Model	AIME2024	MATH500	GPQA-D	LCB V2	Average
OpenThinker-7B	31.3	83.0	42.4	39.9	49.1
DeepSeek-R1-Distill-Qwen-7B	57.3	89.6	47.3	48.4	60.6
OpenThinker2-7B	50.0	88.4	49.3	55.6	60.8
`OmniThoughts-7B` (Qwen2.5)	56.7	90.2	50.0	56.8	63.4
`OmniThoughts-8B` (Qwen3)	**76.7**	**94.6**	**62.1**	**78.1**	**77.8**
LIMO-32B (Ye et al., 2025)	56.7	86.6	58.1	60.0	65.3
OpenThinker-32B	66.0	90.6	61.6	68.9	71.7
DeepSeek-R1-Distill-Qwen-32B	74.7	90.0	62.4	72.3	74.8
OpenThinker2-32B	76.7	90.8	64.1	72.5	76.0
Light-R1-32B (Wen et al., 2025)	74.7	90.4	62.0	56.0	70.7
s1.1-32B (Muennighoff et al., 2025)	59.3	87.4	62.0	58.7	66.8
`OmniThoughts-32B` (Qwen2.5)	80.0	92.6	64.3	73.4	77.6
`OmniThoughts-32B` (Qwen3)	**90.0**	**95.6**	**64.5**	**76.3**	**81.6**

Table 5: Performance comparison between our models and other distilled LRMs in the open-source community.

Setting	Model (7B)	Model (32B)
Full Dataset	55.5	72.8
$\mu_{CD} = 3$	44.2	65.1
$\mu_{CD} = 5$	**63.4**	74.8
$\mu_{CD} = 7$	59.5	**77.6**
$\mu_{CD} = 9$	57.8	74.9

Table 6: Comparison of our trained models under different settings (average scores only).

results summarized in Table 5. We observe that, using `OmniThought` and our RV- and CD-based CoT selection strategies, we obtain strong LRMs that outperform open-source models.

4.3 Further Experimental Studies

We further conduct detailed studies on CoT selection strategies. For both the 7B and 32B models initialized from the Qwen2.5 series, we set μ_{CD} to 3, 5, 7, and 9 to derive four distinct training sets, and also use the entire `OmniThought` dataset as an additional training set. SFT training is performed on each backbone using these five datasets under the same configurations as in the main experiments. Due to space limitations, detailed experimental results are provided in Table 7 in Appendix H, and are summarized in Table 6, which reports average scores on AIME2024, MATH500, GPQA-D, and LCB V2. From these results, we observe that for Qwen2.5-7B-Instruct, the optimal setting is $\mu_{CD} = 5$, while for Qwen2.5-32B-Instruct, the best result is achieved with $\mu_{CD} = 7$. These findings are consistent with those in Table 1. We suggest that when training other models, our results can serve as a reference for selecting an appropriate μ_{CD} value. For example, to train Llama3.1-8B-Instruct, with reasoning capabilities comparable to Qwen2.5-7B-Instruct, a moderate μ_{CD} is preferable. Conversely, for stronger models, increasing μ_{CD} may be beneficial based on our findings with Qwen2.5-32B-Instruct.

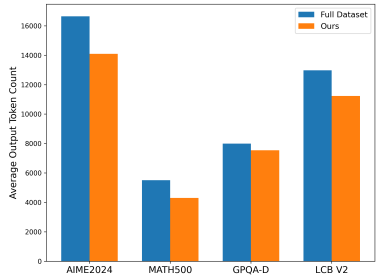


Figure 6: Average output token counts for 7B models.

To demonstrate that our approach effectively produces more concise CoT processes, we further analyze the average output token counts for both 7B and 32B models across the four benchmarks. As shown in Figure 6 for 7B models, our proposed method consistently reduces the output token count compared to full-dataset training, with the most pronounced reduction occurring in the AIME2024 dataset. A similar reduction trend is observed for 32B models (see Figure 7 in Appendix H). These results show that our approach not only maintains reasoning performance but also improves inference efficiency by generating shorter outputs.[8]

5 Concluding Remarks

In conclusion, the introduction of `OmniThought`, a large dataset comprising **2 million** CoT processes annotated with Reasoning Verbosity and Cognitive Difficulty scores, addresses a critical gap in the development of LRMs across different model sizes. Through a self-reliant curation pipeline and by demonstrating improved LRM training effectiveness with Qwen2.5 and Qwen3 models, `OmniThought` enables the creation of high-performing LRMs with strong reasoning abilities and optimal CoT length and difficulty.

[8] We also present a follow-up analysis on the training process's scaling laws. Due to space constraints, this analysis is presented in Appendix E.

Limitations

Several limitations exist in this work, such as the limited scope of reasoning tasks and the static nature of the dataset construction process. These can be addressed in future work through: i) expanding the dataset to cover a broader spectrum of domains, ii) enhancing the data curation pipeline with adaptive feedback mechanisms and interactive learning loops, and iii) extending the annotations to include dynamic assessments of reasoning proficiency.

Ethical Considerations

The broader implications of `OmniThought` extend beyond mere performance improvements, as LRMs are increasingly embedded into a wide range of applications with potentially significant societal impact. However, it is essential to remain vigilant about ethical considerations at every stage of dataset creation and LRM deployment. The automatic generation and annotation of large-scale CoT processes raise concerns regarding potential propagation of biases encoded in teacher models. There is also a risk that models trained with `OmniThought` could produce misleading reasoning in high-stakes applications. We recommend responsible usage practices, such as human-in-the-loop validation, and monitoring for unintended outcomes when deploying LRMs trained with `OmniThought`.

References

Wasi Uddin Ahmad, Sean Narenthiran, Somshubra Majumdar, Aleksander Ficek, Siddhartha Jain, Jocelyn Huang, Vahid Noroozi, and Boris Ginsburg. 2025. Opencodereasoning: Advancing data distillation for competitive coding.

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, Katherine Luna, Leon Derczynski, Nikki Pope, Eileen Long, Seth Schneider, Guillermo Siman, Tomasz Grzegorzek, Pablo Ribalta, Monika Katariya, 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, Eric Chung, and Chris Alexiuk. 2025. Llama-nemotron: Efficient reasoning models.

Wenrui Cai, Chengyu Wang, Junbing Yan, Jun Huang, and Xiangzhon Fang. 2025. Training small reasoning llms with cognitive preference alignment. *CoRR*, abs/2504.09802.

Xinghao Chen, Zhijing Sun, Wenjin Guo, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, and Xiaoyu Shen. 2025. Unveiling the key factors for distilling chain-of-thought reasoning. *CoRR*, abs/2502.18001.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Clayton Cohn, Nicole Hutchins, Tuan Le, and Gautam Biswas. 2024. A chain-of-thought prompting approach with llms for evaluating students' formative assessment responses in science. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*, pages 23182–23190. AAAI Press.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *CoRR*, abs/2410.07985.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on llm-as-a-judge. *CoRR*, abs/2411.15594.

Zhanzhong Gu, Wenjing Jia, Massimo Piccardi, and Ping Yu. 2025. Empowering large language models for automated clinical assessment with generation-augmented retrieval and hierarchical chain-of-thought. *Artif. Intell. Medicine*, 162:103078.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173. Association for Computational Linguistics.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *CoRR*, abs/2504.1145.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. 2021a. Measuring coding challenge competence with apps.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, Yikai Zhang, Yuqing Yang, Ting Wu, Binjie Wang, Shichao Sun, Yang Xiao, Yiyuan Li, Fan Zhou, Steffi Chern, Yiwei Qin, Yan Ma, Jiadi Su, Yixiu Liu, Yuxiang Zheng, Shaoting Zhang, Dahua Lin, Yu Qiao, and Pengfei Liu. 2024. Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent AI. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024*.

Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roee Aharoni, and Mor Geva. 2024. A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 4615–4634. Association for Computational Linguistics.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*. OpenReview.net.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron C. Courville, and Nicolas Le Roux. 2024. Vineppo: Unlocking RL potential for LLM reasoning through refined credit assignment. *CoRR*, abs/2410.01679.

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. 2023. Taco: Topics in algorithmic code generation dataset.

Tengxiao Liu, Qipeng Guo, Yuqing Yang, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023. Plan, verify and switch: Integrated reasoning with diverse x-of-thoughts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2807–2822. Association for Computational Linguistics.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *CoRR*, abs/2501.12570.

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. 2025. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Open Thoughts Team. 2025. Open Thoughts.

Luong Quoc Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Reft: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 7601–7614. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*. OpenReview.net.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Thoughts are all over the place: On the underthinking of o1-like llms. *CoRR*, abs/2501.18585.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. 2025. Light-r1: Curriculum sft, DPO and RL for long COT from scratch and beyond. *CoRR*, abs/2503.10460.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *CoRR*, abs/2501.09686.

Junbing Yan, Chengyu Wang, Taolin Zhang, Xiaofeng He, Jun Huang, and Wei Zhang. 2023. From complex to simple: Unraveling the cognitive tree for reasoning with small language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12413–12425. Association for Computational Linguistics.

Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. 2025. Towards thinking-optimal scaling of test-time compute for LLM reasoning. *CoRR*, abs/2502.18080.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. LIMO: less is more for reasoning. *CoRR*, abs/2502.03387.

Michal Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. 2024. Robotic control via embodied chain-of-thought reasoning. In *Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 3157–3181. PMLR.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. 2024. When scaling meets LLM finetuning: The effect of data, model and finetuning method. In *The Twelfth International Conference on Learning Representations*. OpenReview.net.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Anni Zou, Zhuosheng Zhang, Hai Zhao, and Xiangru Tang. 2023. Meta-cot: Generalizable chain-of-thought prompting in mixed-task scenarios with large language models. *CoRR*, abs/2310.06692.

A Details of Problem Set Construction

For the mathematics domain, our problems are sourced from OpenMathReasoning (Moshkov et al., 2025), MathInstruct [9], DeepMath-103K (He et al., 2025), NuminaMath [10], and OpenThoughts2-1M [11]. For the code domain, sources include TACO (Li et al., 2023), APPs (Hendrycks et al., 2021a), OpenThoughts2-1M, and OpenCodeReasoning (Ahmad et al., 2025). For the science domain, problems are drawn from StackExchange-Physics [12], StackExchange-Biology [13], Camel-AI-Chemistry [14] and OpenThoughts2-1M.

We use normalized Indel similarity and 10-gram-based similarity measures to perform deduplication and decontamination on the collected problems against evaluation benchmarks, including AIME24, MATH500, LiveCodeBench V2, and GPQA-Diamond.

B CoT Metadata

In the `OmniThought` dataset, each problem is associated with multiple CoTs, each accompanied by comprehensive metadata annotations. Below is an example of CoT metadata. This CoT was generated by `DeepSeek-R1` and validated as correct by `QwQ-32B`. Here, "thought" represents the reasoning process (CoT), "solution" denotes the answer derived from that process, and "full_response" is their combination. This CoT was scored by `QwQ-32B` (as the `Score Calculator`) for Reasoning Verbosity (RV) and Cognitive Difficulty (CD), receiving an RV score of 5 and a CD score of 7.

```
{
    "thought": "TL;DR",
    "solution": "TL;DR",
    "full_response": "TL;DR",
    "teacher": DeepSeek-R1,
    "thought_correctness_verify": True,
    "Reasoning_Verbosity":
        {"level": 5, "judge": QwQ-32B},
    "Cognitive_Difficulty":
        {"level": 7, "judge": QwQ-32B}
}
```

[9] https://huggingface.co/datasets/TIGER-Lab/MathInstruct
[10] https://huggingface.co/datasets/AI-MO/NuminaMath-1.5
[11] https://huggingface.co/datasets/open-thoughts/OpenThoughts2-1M
[12] https://physics.stackexchange.com
[13] https://biology.stackexchange.com
[14] https://huggingface.co/datasets/camel-ai/chemistry

CoTs in `OmniThought` can be assembled into various dataset types according to metadata filtering requirements, supporting the application of specialized training algorithms. In future work, we will further enrich CoT metadata, for example by incorporating additional judge models and introducing new scoring dimensions. As the metadata becomes more comprehensive, `OmniThought` will enable more diverse and customizable training regimes, facilitating model adaptation to a wider range of task scenarios.

C Detailed Sampling Process

For simplicity, we define:

$$f_1(y_{CoT}) = \max_i \left| S_{CD}^{(i)} - \mu_{CD} \right| \tag{4}$$

if $S_{CD} \leq \mu_{CD}$, and

$$f_1(y_{CoT}) = \max_i \left| S_{CD}^{(i)} - \mu_{CD} \right| - (S_{CD} - \mu_{CD}) \tag{5}$$

otherwise. We also define

$$f_2(y_{CoT}) = \left(\max_i \left| S_{CD}^{(i)} - S_{RV}^{(i)} \right| \right) - \left| S_{CD} - S_{RV} \right| . \tag{6}$$

Next, we normalize $f_1(y_{CoT})$ and $f_2(y_{CoT})$ to obtain the corresponding probability distributions $P_1(y_{CoT})$ and $P_2(y_{CoT})$:

$$P_1(y_{CoT}) = \frac{f_1(y_{CoT})}{\sum_{i=1}^{|\mathcal{D}(x)|} f_1(y_{CoT}^{(i)})} \tag{7}$$

$$P_2(y_{CoT}) = \frac{f_2(y_{CoT})}{\sum_{i=1}^{|\mathcal{D}(x)|} f_2(y_{CoT}^{(i)})} \tag{8}$$

The final selection probability for a CoT is defined as:

$$\Pr(y_{CoT}) = \beta \cdot P_1(y_{CoT}) + (1 - \beta) \cdot P_2(y_{CoT}) \tag{9}$$

where β is a tunable hyperparameter that determines the relative influence of $P_1(y_{CoT})$ and $P_2(y_{CoT})$ on the final probability. By default, we set $\beta = 0.5$ to balance the two factors. Given μ_{CD}, we can assign selection probabilities to all CoTs corresponding to each problem in `OmniThought`, and then perform sampling based on these probabilities. Notably, the number of CoT samples per

problem can be flexibly adjusted; that is, if multiple CoTs for a problem are assigned high sampling probabilities, several of these high-quality CoTs can be sampled simultaneously to enrich the training set.

D Detailed Experimental Settings

D.1 Verification on Reasoning Verbosity

For SFT verification on RV, we used a global batch size of 96, a learning rate of 1×10^{-5}, and trained for 3 epochs. Training was performed on a single node equipped with 8 A800 GPUs (80 GB), with each model requiring approximately 6 hours.

D.2 Explorations on RQ1 and RQ2

Similarly, for the SFT training in RQ1, we used a global batch size of 96, a learning rate of 1×10^{-5}, and trained for 3 epochs. Training was performed on a single node with 8 A800 GPUs (80 GB), taking approximately 5 hours per model.

For DPO training in RQ2, we set the global batch size to 96, the learning rate to 5×10^{-7}, β to 0.1, and trained for 1 epoch. Training was performed on a single node with 8 A800 GPUs (80 GB), with a total training time of 2 hours.

For GRPO training in RQ2, we set the global batch size to 512, the learning rate to 1×10^{-6}, and trained for 15 epoch. Training was performed on a single node with 8 A800 GPUs (80 GB), with a total training time of 13 hours.

D.3 Training Strong LRMs

For SFT training of the 7B/8B models, we set $\mu_{CD} = 5$, the global batch size to 512, the learning rate to 8×10^{-5}, and trained for 5 epochs. Training was conducted on 8 nodes, each equipped with 8 A800 GPUs (80 GB), with an overall training time of approximately 26 hours.

For SFT training of the 32B models, we set $\mu_{CD} = 7$, the global batch size to 512, the learning rate to 3×10^{-5}, and trained for 5 epochs. Training was conducted on 8 nodes, each with 8 A800 GPUs (80 GB), requiring approximately 140 hours in total.

E Analysis of Scaling Laws

To investigate whether a scaling law of training set size exists, we constructed datasets of varying sizes under the same CD and RV filtering criteria and conducted SFT of Qwen2.5-7B-Instruct under identical training configurations. The experimental results are presented in Table 8. The results show that as the dataset size increases, the model's reasoning ability continues to improve, but the rate of improvement gradually diminishes. When the dataset size approaches one million CoTs, further increases yield negligible gains in reasoning performance. Therefore, to balance computational cost and model performance in pure SFT scenarios, we recommend selecting a training set of fewer than one million CoTs for each iteration.

F Analysis of Grading of RV and CD

Cognitive Difficulty (CD) and Reasoning Verbosity (RV) are each divided into ten sub-levels from 0 to 9. Adjacent sub-levels form a standard range, for which we provide clear descriptive definitions (i.e., 0–1, 2–3, 4–5, 6–7, and 8–9).

Initially, we designed detailed descriptive definitions for all ten individual levels of CD and RV. However, this overly fine-grained scheme confused the Judge model and introduced substantial noise, preventing stable evaluation. By contrast, grouping adjacent sub-levels into standard ranges and providing explicit definitions only for those ranges creates a more relaxed mechanism, allowing the Judge model greater flexibility in rating a CoT. This approach yields more consistent ratings across multiple evaluations of the same CoT and thus more stable final assessments.

Regarding the definitional content, we experimented with various configurations, including definitions in multiple languages and those authored by humans versus those generated by a model. We then evaluated the reasoning performance of models trained under these different definition sets. Our findings indicate that as long as the boundaries between standard ranges are sufficiently clear and unambiguous, the variance in outcomes resulting from different definitional content is negligible. Therefore, we adopted a relatively concise and clear definition style.

To assess the reliability and stability of QwQ-32B as the judge model for RV and CD, we conducted experiments comparing human ratings, QwQ-32B, and other Judge models. We randomly sampled 10K CoTs for evaluation; both human raters and models rated each CoT three times according to the CD and RV definitions, and we averaged the three ratings. We then computed mean rating differences between QwQ-32B and human raters as well as between QwQ-32B and other

Setting	AIME2024	MATH500	GPQA-D	LCB V2	Avg.
Model Size: 7B					
Full Dataset	43.3	88.2	45.4	45.4	55.5
$\mu_{CD} = 3$	23.3	79.8	40.4	33.4	44.2
$\mu_{CD} = 5$	**56.7**	**90.2**	**50.0**	**56.8**	**63.4**
$\mu_{CD} = 7$	46.7	90.0	46.4	54.9	59.5
$\mu_{CD} = 9$	43.3	89.6	45.9	52.6	57.8
Model Size: 32B					
Full Dataset	70.0	91.8	59.6	70.1	72.8
$\mu_{CD} = 3$	56.7	89.6	54.3	59.8	65.1
$\mu_{CD} = 5$	73.3	92.4	61.6	72.0	74.8
$\mu_{CD} = 7$	**80.0**	**92.6**	**64.3**	**73.4**	**77.6**
$\mu_{CD} = 9$	73.3	92.0	62.6	71.8	74.9

Table 7: Performance comparison between our trained models with different μ_{CD} scores.

Scale	100K	300K	500K	700K	900K	1.1M
Score	42.73	48.32	53.81	55.79	58.02	58.57

Table 8: Scaling law analysis for Qwen2.5-7B-Instruct. The table reports the model's average reasoning performance (mean Pass@1 score) as a function of training dataset size (scale), measured on AIME24, MATH500, LiveCodeBench V2, and GPQA-Diamond.

QwQ-32B vs.	R1	R1-0528	Human
RV difference	1.35	0.96	1.22
CD difference	1.23	0.93	1.17

Table 9: Evaluation of QwQ-32B's reliability as a judge model, reported as the mean absolute difference in ratings for RV and CD between QwQ-32B and three separate evaluators (DeepSeek-R1, DeepSeek-R1-0528, and human annotators) on a test set of 10,000 CoTs.

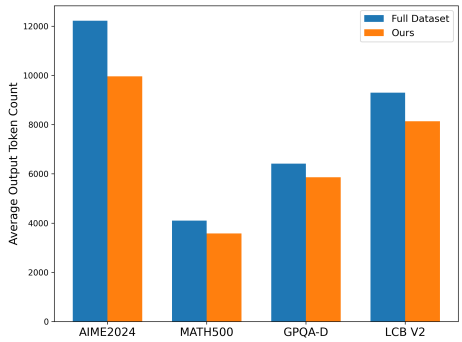


Figure 7: Average output token counts for 32B models across four test sets.

Judge models; the specific results are provided in Table 9. The mean rating differences in both CD and RV evaluations between QwQ-32B and human raters, and between QwQ-32B and other Judge models, are all below 2, which is within a reasonable range. We posit that it is this more relaxed definition that contributes to increased consistency in the ratings.

G Human Annotation Process and Ethical Considerations

The human ratings for Cognitive Difficulty (CD) and Reasoning Verbosity (RV) were provided by our in-house, professional data annotation team. The complete and verbatim instructions furnished to the annotators, defining the rating criteria for both metrics, are detailed in Section 3.2 and 3.3.

This annotation task was conducted as a standard component of the team's professional responsibilities. As salaried employees of our organization, the annotators are compensated with competitive wages appropriate for their professional role and geographic location. All team members were fully informed that the collected ratings would be used for academic research purposes.

The data under evaluation consisted exclusively of machine-generated text, containing no personal, private, or sensitive information. Given the nature of the data and the professional context of the annotation, this study was exempt from a formal ethics review by an Institutional Review Board (IRB). Our annotation team is composed of trained experts proficient in data analysis and evaluation, which ensures the high quality and consistency of the collected ratings.

H Other Experimental Results

Detailed experimental results for our trained models with different μ_{CD} ranges are presented in Table 7. The average output token counts for the 32B models are illustrated in Figure 7.

I Prompt Templates

The prompt templates used in this paper are listed below.

Prompt Template to Validate the Correctness of CoT Processes and Solution

You are a rigorous logical validator analyzing problem-solving components.
Your task is to separately assess the validity of the reasoning process and final solution.
Given a problem, the correct answer, a candidate reasoning process, and a candidate solution, you will:

For SOLUTION VALIDITY: Directly comparing it to the correct answer.
For REASONING PROCESS VALIDATION:
a. Verify stepwise logical coherence and soundness
b. Confirm all critical problem constraints are properly addressed
c. Check for self-contradictions or unsupported leaps in logic
d. Verify the process can actually derive the proposed solution

Evaluation Protocol:
- Solution validity MUST be FALSE for any numerical mismatch or missing units
- Reasoning process validity requires ALL validation criteria (a-d) satisfied
- Both assessments must be independent: correct answer with flawed reasoning gets (False, True)
- Return STRICT BOOLEAN assessments for both components

Problem: {problem}
Correct Answer: {answer}
Candidate Reasoning Process: {reasoning process}
Proposed Solution: {solution}

Output Format: reasoning_valid: bool, solution_valid: bool

J Examples of CoT Processes

In this section, we present examples of RV and CD across different level intervals in Table 10 and Table 11, respectively. For ease of understanding, we have summarize the step-by-step procedures in CoTs.

Prompt Template to Calculate the RV Score

You are an expert judge tasked with evaluating the Reasoning Verbosity of a Chain-of-Thought (CoT) for a given problem and its answer.

Reasoning Verbosity Evaluation Focus:
Assess how well the CoT's length and step complexity match the problem's inherent difficulty.
An optimal chain is neither missing essential steps nor padded with needless digressions.
A simple question should be solved with a brief, direct chain; a challenging one may justifiably require a longer path with reflection and error-checking.

Scoring Guidelines (0-9):
0-1 Minimal verbosity, straightforward expression with little to no elaboration.
2-3 Clear and concise reasoning with necessary explanations.
4-5 Moderate verbosity with detailed explanations and thorough reasoning.
6-7 Extensive verbosity with comprehensive justification and exploration of complex connections.
8-9 High verbosity with deep, exhaustive exploration of reasoning; involves extensive elaboration, nested justifications, and consideration of counterarguments or alternative perspectives.

Given Problem, Chain-of-Thought and Answer, you will:
1. Analyze the Reasoning Verbosity
2. Determine score using the above criteria
3. Output ONLY the integer score (0-9)

Problem: {problem}
Chain-of-Thought: {thought}
Answer: {solution}

Final Output: Single integer between 0-9

Prompt Template to Calculate the CD Score

You are an expert judge assessing the Cognitive Difficulty of a Chain-of-Thought (CoT) for a given problem and its answer.

Cognitive Difficulty Evaluation Focus:
The level of reasoning competence required for a model to follow and reproduce the chain faithfully.
Judge the reasoning approach, techniques, and overall difficulty.
Higher scores correspond to more advanced concepts, abstractions, or multi-layer reasoning patterns.

Scoring Guidelines (0-9):
0-1 Elementary facts or a single trivial operation.
2-3 Multi-step arithmetic, explicit enumeration, basic rule chaining.
4-5 Early-undergraduate logic/algebra; one non-obvious insight.
6-7 Advanced undergraduate techniques (determinants, dynamic programming, layered code reasoning, etc).
8-9 Graduate-level abstraction, nested proofs, intricate algorithmic analysis.

Given Problem, Chain-of-Thought and Answer, you will:
1. Analyze the Cognitive Difficulty
2. Determine score using the above criteria
3. Output ONLY the integer score (0-9)

Problem: {problem}
Chain-of-Thought: {thought}
Answer: {solution}

Final Output: Single integer between 0-9

RV Level	CoT
0–1	The integral evaluates immediately to $(x-1)e^x + C$.
2–3	Using integration by parts, let $u = x$ and $dv = e^x\,dx$, which yields $du = dx$ and $v = e^x$. Hence, $$\int xe^x\,dx = xe^x - \int e^x\,dx = xe^x - e^x + C.$$
4–5	Select the integration-by-parts formula: $$\int xe^x\,dx = xe^x - \int e^x\,dx.$$ Since $\int e^x\,dx = e^x + C$, it follows that $$\int xe^x\,dx = xe^x - e^x + C = (x-1)e^x + C.$$
6–7	Recall the identity $\int u\,dv = uv - \int v\,du$. Setting $u = x$, $dv = e^x\,dx$ gives $du = dx$ and $v = e^x$. Thus, $$\int xe^x\,dx = xe^x - \int e^x\,dx = xe^x - e^x + C_1 = (x-1)e^x + C.$$ Verification by differentiation: $$\frac{d}{dx}\big[(x-1)e^x\big] = e^x + (x-1)e^x = xe^x.$$
8–9	Method 1 (integration by parts): let $u = x$, $dv = e^x\,dx$; then $$\int xe^x\,dx = xe^x - \int e^x\,dx = (x-1)e^x + C.$$ Method 2 (series expansion): express $$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \implies xe^x = \sum_{m=1}^{\infty} \frac{x^m}{(m-1)!}.$$ Integrating termwise yields $$\int xe^x\,dx = \sum_{m=1}^{\infty} \frac{x^{m+1}}{(m-1)!\,(m+1)} = (x-1)e^x + C.$$ Generalization: $\displaystyle\int x^n e^x\,dx = x^n e^x - n\int x^{n-1}e^x\,dx.$

Table 10: CoT examples for $\displaystyle\int xe^x\,dx$ at varying RV levels.

CD level	Problem	CoT (Summarized)	Answer
0–1	Compute $2 + 3$.	1. Recognize that the operation is a simple sum of two integers. 2. Observe that $2 + 3 = 5$.	5
2–3	Compute the sum of the first five multiples of 7.	1. List the first five multiples of 7: $7, 14, 21, 28, 35$. 2. Compute their sum: $7 + 14 + 21 + 28 + 35 = 105$.	105
4–5	Simplify $\frac{x^2-4}{x-2}$ and evaluate at $x = 5$.	1. Note that $x^2 - 4 = (x - 2)(x + 2)$. 2. Cancel the common factor $(x - 2)$, obtaining $x + 2$. 3. Substitute $x = 5$: $5 + 2 = 7$.	7
6–7	Determine the time complexity of $T(n) = 2\,T(n/2) + n$.	1. Identify $a = 2$, $b = 2$, $f(n) = n$. 2. Compute $n^{\log_2 2} = n$. 3. Since $f(n) = \Theta(n)$, this is Case 2 of the Master theorem. 4. Conclude $T(n) = \Theta(n \log n)$.	$\Theta(n \log n)$
8–9	Prove that there are infinitely many prime numbers.	1. Assume, for contradiction, primes are finite $\{p_1, \ldots, p_n\}$. 2. Construct $N = p_1 p_2 \cdots p_n + 1$. 3. N leaves remainder 1 mod any p_i. 4. By the Fundamental Theorem of Arithmetic, N has a prime divisor not in the list. 5. Contradiction implies infinitely many primes.	Infinitely many primes

Table 11: CoT examples at varying CD levels.