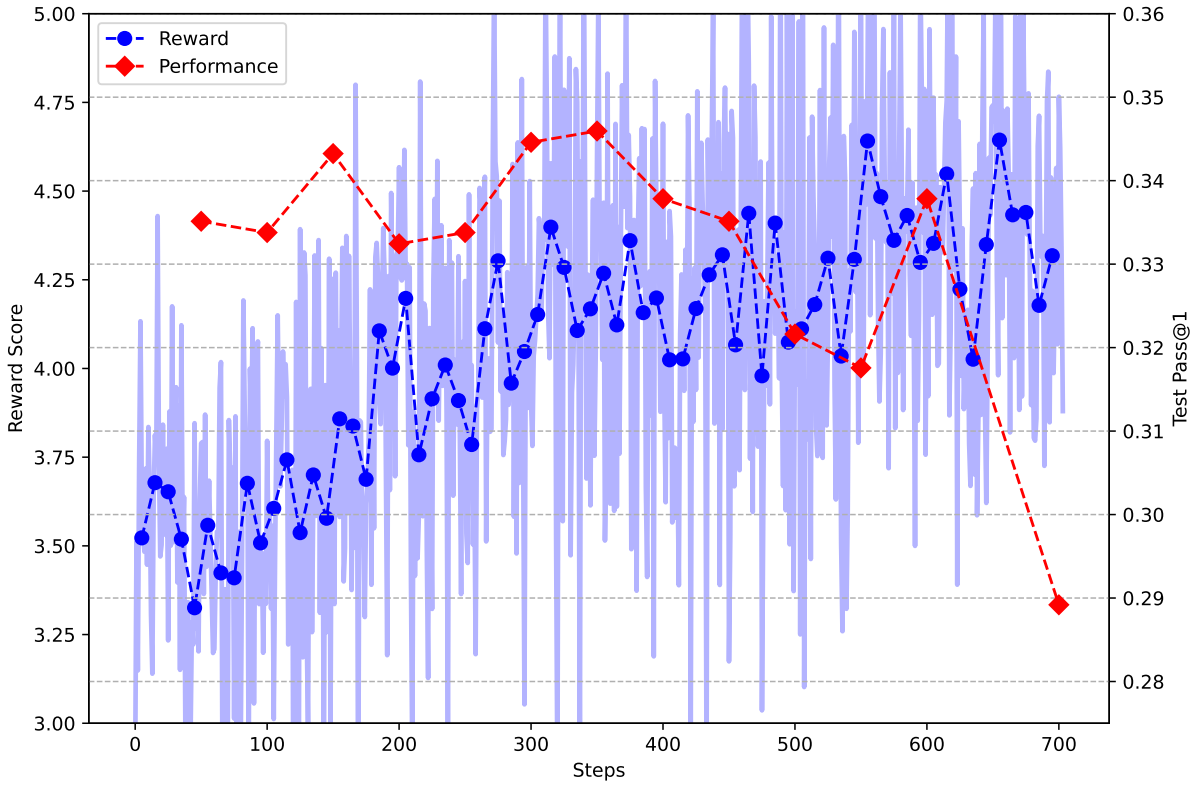

Figure 6 | Reward hacking: the reward exhibits an increasing trend as the performance on CodeForces decreases for training.

Table 7 | Training costs of DeepSeek-R1, assuming the rental price of H800 is $2 per GPU hour.

Training Costs	DeepSeek-R1-Zero	SFT data creation	DeepSeek-R1	Total
in H800 GPU Hours	101K	5K	41K	147K
in USD	$202K	$10K	$82K	$294K

and also exhibits language mixing during the RL process. The results are shown in Figure 7. As can be seen, without the LC reward, language consistency gradually deteriorates as training steps increase. However, when the LC reward is applied, stable language consistency is maintained throughout the training process. For benchmark performance, the model maintains comparable performance on the mathematical benchmark, while a slight degradation is observed on the coding benchmark. Although such alignment results in a slight degradation in model performance, this reward aligns with human preferences, making the output more readable.

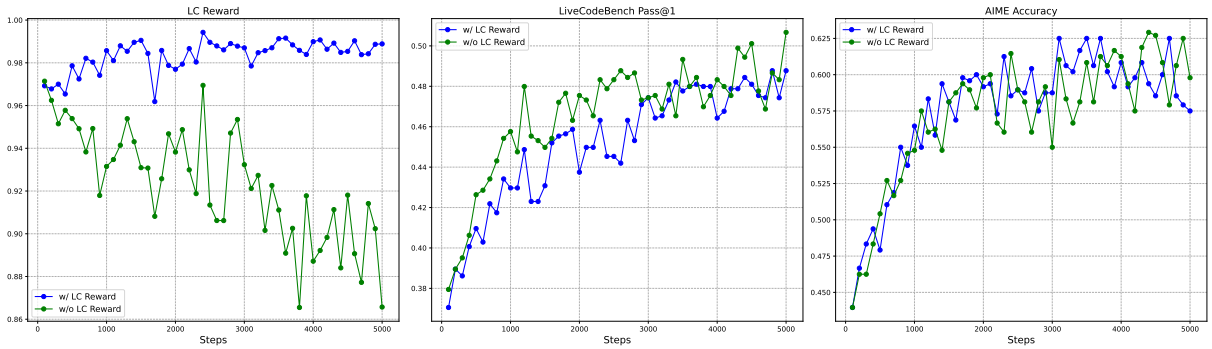

Figure 7 | The experiment results of Language Consistency (LC) Reward during reinforcement learning.

C. Self-Evolution of DeepSeek-R1-Zero

C.1. Evolution of Reasoning Capability in DeepSeek-R1-Zero during Training

We analyzed DeepSeek-R1-Zero's performance on the MATH dataset stratified by difficulty levels (1-5). Figure 8 reveals distinct learning patterns: easy problems (levels 1-3) quickly reach high accuracy (0.90-0.95) and remain stable throughout training, while difficult problems show remarkable improvement - level 4 problems improve from near 0.78 to 0.95, and the most challenging level 5 problems demonstrate the most dramatic improvement from near 0.55 to 0.90.

One may find it counterintuitive that the model's accuracy on harder questions (levels 3-4) occasionally surpasses its performance on easier questions (level 1) by a small margin. This apparent anomaly stems from several dataset characteristics. The MATH dataset is unevenly distributed, with level-1 questions comprising only 43 of 500 examples, while higher levels contain approximately 100 questions each. Consequently, the model's 95-97% accuracy on level-1 represents just 1-2 unsolved problems, primarily in geometry, where the model still struggles. Furthermore, the distribution of mathematical categories (geometry, algebra, etc.) varies across difficulty levels due to the dataset's construction methodology. It's also worth noting that these difficulty levels were annotated based on human perception of problem complexity rather than

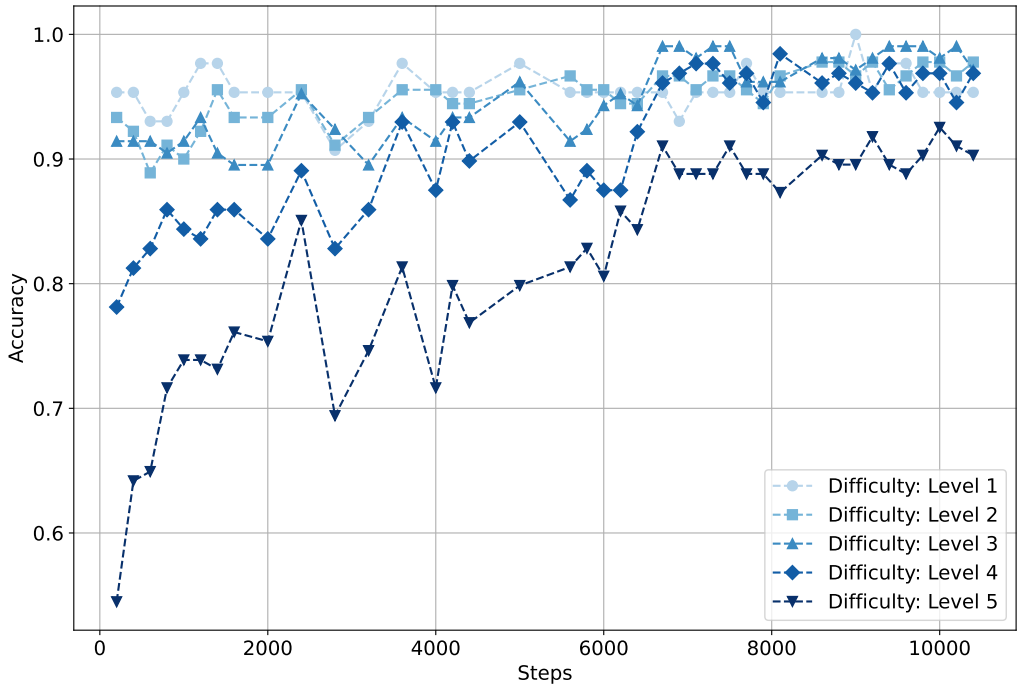

Figure 8 | Performance of DeepSeek-R1-Zero on problems with varying difficulty levels in the MATH dataset.

machine learning considerations.

Despite these nuances in comparing raw accuracy percentages across difficulty levels, the training trends still demonstrate that while simpler reasoning tasks (for humans) are mastered early in training, the model's capability on complex reasoning problems (level 3-5) significantly improves over time.

C.2. Evolution of Advanced Reasoning Behaviors in DeepSeek-R1-Zero during Training

We analyze the change in the reasoning behavior of the model during training.

First, as shown in Figure 9(a), we counted some representative reflective words, including "wait", "mistake", "however", "but", "retry", "error", "verify", "wrong", "evaluate", and "check" . These reflective words were selected by 3 human experts, who are asked to think of several reflective words and then merge them into a final word list. As is shown, there is a gradual increase in the frequency of reflective behaviors as training progresses. Specifically, the count of the reflective words rises 5- to 7-fold compared to the start of training, suggesting that RL plays a key role in generating long-chain intermediate tokens.

Second, specific reflective behaviors may appear at particular points in training. The analysis of the word "wait" (Figure 9(b)) demonstrates this clearly. This reflective strategy was nearly absent during early training, showed occasional usage between steps 4000-7000, and then exhibited significant spikes after step 8000. This suggests that the model learns different forms of reflection at specific stages of development.

In conclusion, we observe a gradual increase in the model's reflective behavior during training, while certain reflection patterns like the use of "wait" emerge at specific points in the training process.

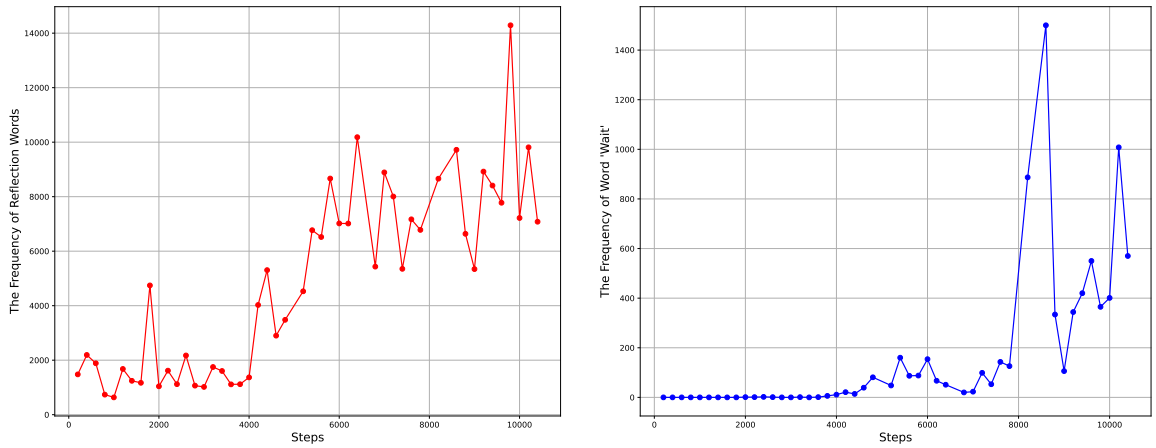

Figure 9 | Evolution of reasoning behaviors during training. (a) Frequency of representative reflective words during the training process; (b) Specific occurrence patterns of the word "wait" throughout the training process.

D. Evaluation of DeepSeek-R1

D.1. Experiment Setup

Benchmarks We evaluate models on MMLU (Hendrycks et al., 2021), MMLU-Redux (Gema et al., 2025), MMLU-Pro (Wang et al., 2024), C-Eval (Huang et al., 2023), IFEval (Zhou et al., 2023b), FRAMES (Krishna et al., 2024), GPQA Diamond (Rein et al., 2023), SimpleQA (OpenAI, 2024a), C-SimpleQA (He et al., 2024), SWE-Bench Verified (OpenAI, 2024b), Aider (Gauthier, 2025), LiveCodeBench (Jain et al., 2024) (2024-08 – 2025-01), Codeforces (Mirzayanov, 2025), Chinese National High School Mathematics Olympiad (CNMO 2024) (CMS, 2024), and American Invitational Mathematics Examination 2024 (AIME 2024) (MAA, 2024).

Specifically, MMLU, MMLU-Redux, MMLU-Pro, C-Eval, and CMMLU are multiple-choice benchmarks designed to assess model performance on general encyclopedic knowledge. Higher scores on these benchmarks indicate a broader understanding of world knowledge and the ability to correctly answer questions in a multiple-choice format. SimpleQA and C-SimpleQA evaluate model performance on long-tail knowledge, while GPQA assesses the ability to solve Ph.D.-level tasks in physics, chemistry, and biology. IFEval is designed to evaluate the model's capacity to generate outputs in a required format. FRAMES and DROP focus on assessing model performance in processing and reasoning over long documents. In addition to these standard benchmarks, we also evaluate our models on open-ended generation tasks, employing LLM as judges. We follow the original evaluation protocols of AlpacaEval 2.0 and Arena-Hard, utilizing GPT-4-Turbo-1106 for pairwise comparisons. To mitigate length bias, only the final summary is provided to the evaluation model.

LiveCodeBench and Codeforces are designed to measure model performance on algorithmic competition tasks, whereas SWE-Verified and Aider assess the model's capabilities on real-world software engineering problems. Finally, AIME, MATH-500, and CNMO 2024 comprise mathematics problems that test the model's reasoning abilities in mathematical domains.

For distilled models, we report representative results on AIME 2024, MATH-500, GPQA Diamond, Codeforces, and LiveCodeBench.

Decontamination To prevent benchmark contamination, we implemented comprehensive decontamination procedures for both pre-training and post-training data. DeepSeek-V3 base has a knowledge cutoff date of July 2024, predating evaluation benchmarks like CNMO 2024, and we filtered out any text segments (including web pages and GitHub files) that contained matching 10-gram sequences from evaluation questions or reference solutions. As one example of our decontamination efforts, in the mathematics domain alone, our decontamination process identified and removed approximately six million potential pre-training texts. For post-training, mathematical SFT data and RL training prompts were sourced exclusively from pre-2023 competitions and underwent the same n-gram filtering protocol used in pre-training, ensuring no overlap between training and evaluation data. These measures ensure our model evaluation results reflect genuine problem-solving capabilities rather than memorization of test data.

However, we acknowledge that the n-gram based decontamination method cannot prevent the paraphrase of testset. Therefore, it is possible that benchmarks released before 2024 may suffer from contamination issues.

Evaluation Prompts Following the setup in DeepSeek-V3, standard benchmarks such as MMLU, DROP, GPQA Diamond, and SimpleQA are evaluated using prompts from the simple-evals framework. For MMLU-Redux, we adopt the Zero-Eval prompt format (Lin, 2024) in a zero-shot setting. In terms of MMLU-Pro, C-Eval and CLUE-WSC, since the original prompts are few-shot, we slightly modify the prompt to the zero-shot setting. The CoT in few-shot may hurt the performance of DeepSeek-R1. Other datasets follow their original evaluation protocols with default prompts provided by their creators. For code and math benchmarks, the HumanEval-Mul dataset covers eight mainstream programming languages (Python, Java, C++, C#, JavaScript, TypeScript, PHP, and Bash). Model performance on LiveCodeBench is evaluated using CoT format, with data collected between August 2024 and January 2025. The Codeforces dataset is evaluated using problems from 10 Div.2 contests, along with expert-crafted test cases, after which the expected ratings and percentages of competitors are calculated. SWE-Bench verified results are obtained via the agentless framework (Xia et al., 2024). AIDER-related benchmarks are measured using a "diff" format. DeepSeek-R1 outputs are capped at a maximum of 32,768 tokens for each benchmark.

Table 18 to Table 32 present examples of our evaluation formats on different benchmarks. We also detail the specific capabilities of large language models assessed by each benchmark in the corresponding table captions.

Baselines We conduct comprehensive evaluations against several strong baselines, including DeepSeek-V3, Claude-Sonnet-3.5-1022, GPT-4o-0513, OpenAI-o1-mini, and OpenAI-o1-1217. Since accessing the OpenAI-o1-1217 API is challenging in mainland China, we report its performance based on official reports. For distilled models, we also compare the open-source model QwQ-32B-Preview (Qwen, 2024a).

We set the maximum generation length to 32,768 tokens for the models. We found that using greedy decoding to evaluate long-output reasoning models results in higher repetition rates and significant variability across different checkpoints. Therefore, we default to pass@k evaluation (Chen et al., 2021) and report pass@1 using a non-zero temperature. Specifically, we use a sampling temperature of 0.6 and a top-p value of 0.95 to generate k responses (typically between 4 and 64, depending on the test set size) for each question. Sepcifically, we use $k = 64$ for AIME and GPQA, $k = 16$ for MATH and CodeForces, and $k = 8$ for LCB. Pass@1 is then