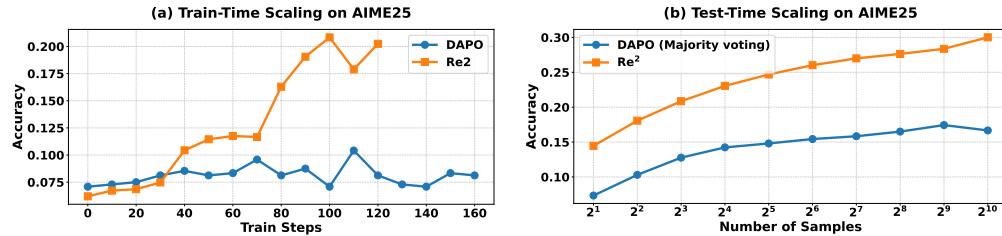


000 001 Re^2 : UNLOCKING LLM REASONING VIA REINFORCE- 002 MENT LEARNING WITH RE-SOLVING 003

004
005 **Anonymous authors**
006 Paper under double-blind review
007
008
009

ABSTRACT

011 Reinforcement learning with verifiable rewards (RLVR) has shown promise in en-
012 hancing the reasoning performance of large language models (LLMs) by increasing
013 test-time compute. However, even after extensive RLVR training, such models still
014 tend to generate unnecessary and low-quality steps in their chain-of-thought (CoT),
015 leading to inefficient overthinking and lower answer quality. We show that when the
016 initial direction or quality of the CoT is suboptimal, the model often fails to reach
017 the correct answer, even after generating several times more tokens than when the
018 initial CoT is well-initialized. To this end, we introduce *Reinforcement Learning*
019 with **Re^2** , in which LLMs learn to flexibly abandon unproductive
020 reasoning paths and restart the solution process when necessary, rather than always
021 committing to a final answer. Re^2 applies pure reinforcement learning without any
022 preliminary supervised fine-tuning, successfully amplifying the rare redo behavior
023 in vanilla models from only 0.5% to over 30%. This leads to substantial perfor-
024 mance gains over standard RLVR under the same training compute budget, and
025 also demonstrates notable improvements in test-time performance as the number
026 of samples increases.
027



036 Figure 1: **(a)** Accuracy improvements of **DAPO** and Re^2 on Qwen2.5-7B-Instruct at each training
037 step, with comparable numbers of generated and trained tokens per step. **(b)** Test-time scaling of
038 **DAPO** and Re^2 under the same training budget, as the number of samples increases.
039

040 1 INTRODUCTION

043 Recent studies have shown that large language models (LLMs) can achieve strong reasoning abilities
044 through scaling test-time compute (Snell et al., 2024; Wu et al., 2025). By generating longer chains of
045 thought (CoTs) that incorporate planning, reflection, and self-correction, LLMs attain higher accuracy
046 on complex reasoning tasks such as coding and mathematics (Yang et al., 2025a; Bercovich et al.,
047 2025; Team et al., 2025; Wu et al., 2024). To this end, state-of-the-art models adopt reinforcement
048 learning (RL) in post-training, which has proven effective in producing longer CoTs and strengthening
049 deep-thinking capabilities (Guo et al., 2025; Shao et al., 2024; Yu et al., 2025).

050 However, even with extensive RL training, LLMs still suffer from issues such as overthinking (Chen
051 et al., 2024; Cuadron et al., 2025) and underthinking (Wang et al., 2025; Cuesta-Ramirez et al., 2025;
052 Ding et al., 2025), generating unnecessary or low-quality reasoning steps that degrade both efficiency
053 and overall performance. In this paper, we investigate the limitations of test-time scaling in existing
LLMs by analyzing the correlation between CoT length and accuracy, together with the impact

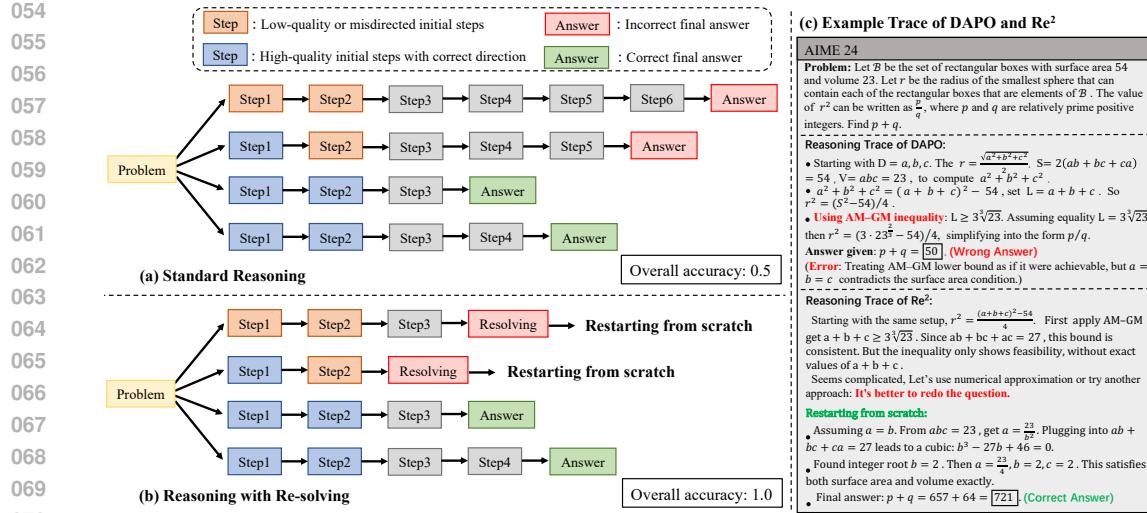


Figure 2: Illustration of reasoning with re-solving. (a) In standard reasoning, when the initial steps are suboptimal, the LLM struggles to reach the correct answer even after generating more reasoning steps and tokens. (b) Reasoning with re-solving, allowing the model to abandon an unpromising path and restart at any point, leads to a higher overall accuracy. (c) Example trace of DAPO and Re²: Both models initially attempt an incorrect approach using the AM-GM inequality; however, Re² detects the failure, restarts, and arrives at the correct answer.

of early reasoning quality on final outcomes. Our analysis reveals that when the initial reasoning steps are suboptimal, LLMs struggle to reach the correct answer, even after generating significantly more reasoning steps and tokens (Figure 2(a)). However, solving complex problems often requires early exploration, during which the model may pursue directions that initially appear promising but ultimately lead to dead ends or errors. Motivated by these findings, we aim to enable models to restart the reasoning process when the current path proves unproductive, thereby improving overall accuracy by escaping unpromising paths (Figure 2(b)).

To equip models with this capability, we introduce **Reinforcement Learning with Re-solving (Re²)**, a novel framework that equips models with the ability to either finalize an answer or re-solve the problem. During training, the model extends partial reasoning trajectories and learns to decide whether to continue or to restart from its current progress. If the model chooses to re-solve, its trajectory receives a reward equal to the expected success rate of solving the problem from scratch. If it instead produces a final answer, the reward is assigned as 1 for a correct solution and 0 otherwise. Under this scheme, when the current reasoning prefix is in the wrong direction or of low quality, abandoning it and re-solving the problem is more likely to yield higher accuracy and thus a larger reward. Conversely, when the reasoning trajectory is promising, directly producing the final answer leads to a higher expected reward. Notably, Re² requires no preliminary supervised fine-tuning. Through pure reinforcement learning alone, it amplifies the rare redo behavior in vanilla models from just 0.5% to over 30%, thereby enabling models to flexibly decide when to re-solve and ultimately leading to more accurate reasoning, as illustrated in Figure 2(c).

We evaluate Re² on a diverse set of reasoning benchmarks, including AIME 2024, AIME 2025 (MAA Committees), AMC 2023 (AI-MO, 2024), GSM8K (Cobbe et al., 2021), and GPQA-Diamond (Rein et al., 2021), covering a wide range of domains and difficulty levels. Our evaluation covers five models ranging from 3B to 14B parameters, including pre-trained, instruction-tuned, and reasoning models. Experimental results demonstrate that our approach achieves significant improvements in reasoning performance compared to recent RLVR methods such as DAPO (Yu et al., 2025) under the same training budget. Moreover, Re² yields a superior trade-off curve between test-time compute and performance compared to majority voting (Wang et al., 2022). We believe our work highlights the promise of integrating RL with a new paradigm of *re-solving* reasoning, which goes beyond the traditional single-chain approach and opens up new directions for developing more flexible and reliable reasoning in LLMs.

108
109
110
2 RELATED WORK111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127

LLM for reasoning. Existing approaches enhance the reasoning capabilities of LLMs through prompt engineering (Wei et al., 2022; Yang et al., 2023), supervised fine-tuning (Yang et al., 2024a; Qin et al., 2024), and reinforcement learning (Schulman et al., 2017; Shao et al., 2024; Zheng et al., 2025; Yue et al., 2025; Zhang et al., 2025; Wang et al., 2024). Among these, reinforcement learning with verifiable rewards (RLVR) has emerged as a mainstream paradigm for post-training optimization, encouraging models to produce longer CoTs with planning and self-reflection, thereby pushing the frontier of reasoning performance (OpenAI, 2024; Qu et al., 2024; Gandhi et al., 2024; Zeng et al., 2025). Nevertheless, even after extensive RLVR training, LLMs remain prone to overthinking and underthinking, leading to redundant or low-quality reasoning steps (Chen et al., 2024; Cuadron et al., 2025; Wang et al., 2025; Cuesta-Ramirez et al., 2025). This remains a fundamental limitation of the prevailing paradigm, in which the model generates a single CoT trajectory and ultimately derives its final answer within that trajectory (Wen et al., 2025; Shojaee et al., 2025). Recent studies (Yang et al., 2025b; Fu et al., 2025) have attempted to address this issue by backtracking to earlier steps or terminating low-confidence reasoning chains, but these methods are limited to supervised fine-tuning or decoding strategies and do not leverage the potential of RL. To the best of our knowledge, our work is the first to propose a reasoning paradigm that allows models to abandon unproductive reasoning paths and re-solve problems from scratch through reinforcement learning.

128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

Test-time scaling. Recent studies have shown that LLMs can effectively improve reasoning performance by increasing inference-time compute (Snell et al., 2024; Welleck et al., 2024; Wu et al., 2025; Muennighoff et al., 2025). Large reasoning models, as exemplified by OpenAI’s O1, learn to produce traces that are longer than the typical solutions via SFT or RLVR (OpenAI, 2024; Qin et al., 2024; Guo et al., 2025; Zhao et al., 2024). Some works improve performance by allowing models to iteratively revise their answers through multiple rounds of self-correction (Xiong et al., 2025; Zhao et al., 2025; Xi et al., 2024; Paul et al., 2024; Yang et al., 2024b). In addition, parallel sampling methods (e.g., majority voting (Wang et al., 2022; Wan et al., 2024) and tree search (Hao et al., 2023; Zhang et al., 2024)) further improve performance by increasing the number of samples. In contrast, our approach scales test-time compute by enabling the model to abandon unpromising reasoning trajectories and re-solve the problem when necessary, thereby unlocking the potential of test-time scaling for reasoning.

3 DIFFICULTY OF RECOVERING FROM SUBOPTIMAL EARLY REASONING

153
154
155
156
157
158
159
160
161

To gain a deeper understanding of the limitations of test-time scaling in existing LLMs, we first analyze the correlation between CoT length and reasoning accuracy (Section 3.1), and further investigate why LLMs fail even when they having sufficient capabilities (Section 3.2). In our experiments, we select Qwen2.5-7B-Instruct (Yang et al., 2024a) as a representative instruction-tuned LLM and DeepScaleR-1.5B-Preview (Luo et al., 2025) as a representative long-CoT reasoning model extensively trained with RLVR. We evaluate them on AMC23 and AIME25, respectively, which aligns the difficulty of datasets with the capabilities of each model.

3.1 CORRELATION BETWEEN RESPONSE LENGTH AND PERFORMANCE

153
154
155
156
157
158
159
160
161

Although training models to produce longer CoTs can significantly improve reasoning performance, we aim to examine whether, for the same problem across multiple samples, longer CoTs actually lead to higher accuracy. To this end, we analyze the correlation between response length and accuracy by sampling 128 responses per problem. As shown in Figure 3, our results reveal **a clear negative correlation between CoT length and accuracy**, both across the entire dataset and at the level of individual problems of varying difficulty. We further conduct case analysis of CoTs with different lengths for the same problem, which shows that longer responses are typically caused by early critical mistakes, such as following the wrong solution path or overanalyzing the problem’s assumptions, making recovery unlikely regardless of the number of additional tokens generated. Additional results are provided in Appendix E.

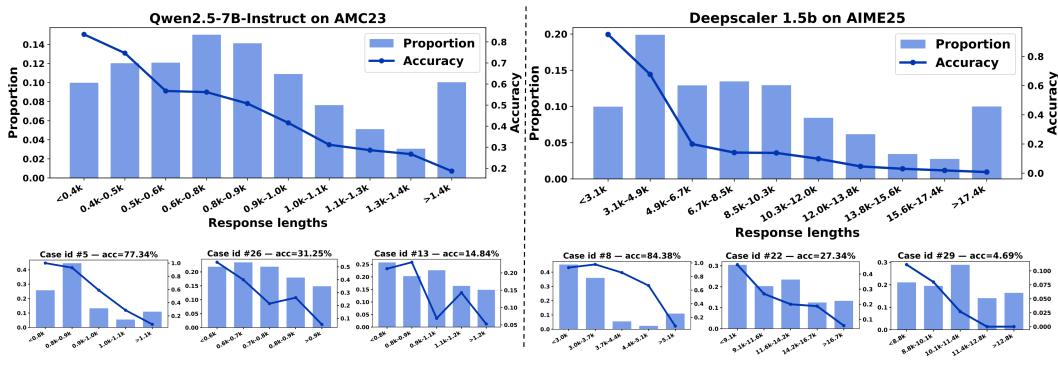


Figure 3: Correlation between CoT length and reasoning performance. The bars represent the proportion of cases within different CoT length intervals, while the line indicates the average accuracy for responses in each interval. The large plots in the top row show the correlation across the entire dataset. The small plots in the second row illustrate the correlation at the level of individual problems.

3.2 IMPACT OF INITIAL REASONING QUALITY ON FINAL ACCURACY

As shown in Section 3.1, shorter CoTs are typically more likely to be correct for a given problem. We hypothesize that this is because suboptimal early reasoning leads the model to generate longer responses, yet recovery is still challenging. To illustrate this phenomenon, we design an experiment in which we truncate different proportions (20%, 40%, 60%, 80%) of **incorrect responses** and prompt LLMs to continue reasoning from these prefixes. We then measure the average accuracy of completions from each truncated prefix and compute the relative drop in accuracy compared to reasoning from scratch on the original problem. As shown in Figure 4, for each prefix length we report the number of cases where the relative drop in accuracy exceeds the 25% or 75% threshold (“All Drops”). The results show that as prefix length increases, the relative drop becomes larger, indicating that **the longer a model continues along an incorrect trajectory, the more likely it is to fail**.

Furthermore, we investigate how early such performance degradation begins in an incorrect response. For each response, we record the shortest prefix at which the relative drop first exceeds the threshold (“First Drops”). The results reveal that for most incorrect responses, a significant drop in accuracy already occurs when only the first 20% of the response is used as the prefix. This demonstrates that **once early reasoning is misguided, the model rarely recovers and struggles to return to the correct reasoning path**.

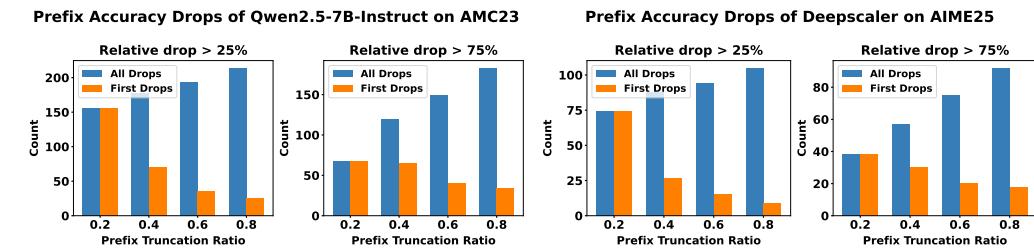


Figure 4: Number of cases where the relative drop exceeds the threshold when continuing from different proportions of incorrect response prefixes, compared to reasoning from scratch. “All Drops” counts all such cases at each prefix proportion, while “First Drops” records the earliest prefix for each response where the drop exceeds the threshold.

3.3 TAKEAWAYS

Based on the above analysis, we conclude that: (1) For a given problem, shorter responses following smoother reasoning tend to achieve higher accuracy, whereas longer responses are often associated with lower accuracy; (2) The quality of early reasoning process is crucial for the final accuracy.

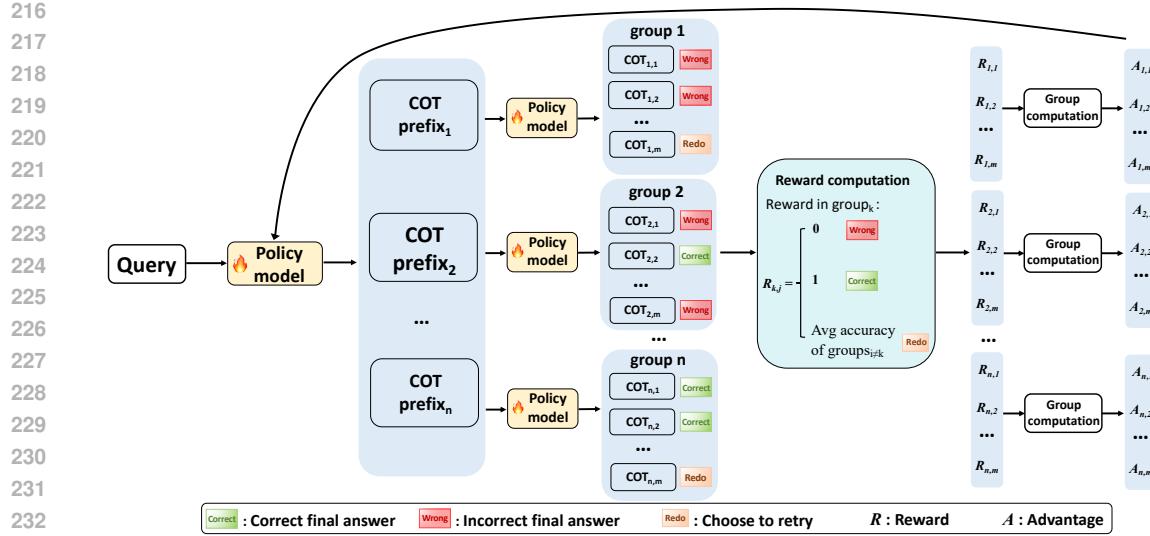


Figure 5: The framework of Re^2 . For each query, Re^2 samples multiple prefixes, then generates multiple continuations for each prefix. The advantage is calculated within each group, while the out-of-group accuracy is used as the reward for the redo action.

4 RE²: REINFORCEMENT LEARNING WITH RESOLVING

The above results highlight that a coherent reasoning process and high-quality early reasoning are crucial for model performance. However, when tackling challenging problems, it is often unavoidable for models to explore early reasoning paths that initially appear promising but ultimately fail to yield correct solutions. LLMs are also prone to generating low-quality content during the initial stages of reasoning. Similar to how humans may reconsider their strategy when solving difficult problems, we aim to equip models with the ability to restart the reasoning process when the current trajectory appears unpromising or leads to confusion. To this end, we propose Re^2 , which leverages reinforcement learning to train models to flexibly re-solve problems during reasoning.

4.1 PREFIX GROUP GENERATION

Recent RLVR methods such as GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025) improve pass@1 accuracy by sampling multiple reasoning trajectories in parallel for each query and rewarding only those that yield correct answers. In contrast, Re^2 aims to improve the final answer quality by allowing the model to rationally abandon an ongoing reasoning path and restart from scratch when the trajectory appears confusing or unpromising. Meanwhile, our method requires estimating the success rate of re-solving from scratch, which provides the reward signal that guides the model to learn whether to redo or to commit to a final answer. To this end, we first sample n full responses for each query. Each response is randomly truncated at a proportion uniformly drawn from $[0, 0.8]$, producing n diverse prefixes that serve as intermediate reasoning states. For each prefix, the model generates m CoT continuations, and all continuations derived from the same prefix are grouped together for subsequent advantage calculation, as illustrated in Figure 5. To incentivize the resolve capability of the base model, we design a specialized prompting strategy, described in Appendix A.

4.2 REWARD STRATEGY WITH RE-SOLVING

To encourage the model to rationally abandon unpromising reasoning paths and restart from scratch when necessary, we design a reward strategy that explicitly incorporates the option to re-solve. For the j -th CoT continuation of the i -th prefix Pre_i , denoted as $O_{i,j}$, there are three possible outcomes $C_{i,j}$: providing a correct final answer ($C_{i,j} = \text{correct}$), providing an incorrect final answer ($C_{i,j} = \text{incorrect}$), or choosing to re-solve the problem ($C_{i,j} = \text{resolve}$).

270 For continuations that yield final answers, the reward assignment follows standard RLVR: the model
 271 receives a reward of 1 for a correct answer and 0 otherwise. For continuations that choose to re-solve,
 272 the reward is given by the expected accuracy of re-solving from scratch, estimated using out-of-
 273 group CoT completions, i.e., completions whose prefix is not Pre_i . Specifically, let $P_{\neq i}(\text{correct})$,
 274 $P_{\neq i}(\text{incorrect})$, and $P_{\neq i}(\text{resolve})$ denote the empirical probabilities of correct, incorrect, and
 275 resolve outcomes among the $(n - 1) \times m$ out-of-group continuations. When at most R redo rounds
 276 are allowed, the reward is given by¹:

277

278

$$279 \quad r_{i,j} = \begin{cases} 1, & \text{if } C_{i,j} = \text{correct} \\ 0, & \text{if } C_{i,j} = \text{incorrect} \\ P_{\neq i}(\text{correct}) \cdot \frac{1 - P_{\neq i}(\text{resolve})^R}{1 - P_{\neq i}(\text{resolve})}, & \text{if } C_{i,j} = \text{resolve} \end{cases} \quad (1)$$

280

281

282

283

This three-way reward strategy encourages the model to continue reasoning when the current trajectory
 284 is promising, and to re-solve when the trajectory is confused or flawed, since the expected accuracy
 285 of re-solving exceeds that of continuation.

286

287

288 4.3 ADVANTAGE COMPUTATION AND PARAMETER UPDATE

289

290

291

Algorithm 1 Re^2 : Reinforcement Learning with Resolving

292 **Input** initial policy model π_θ ; task prompts \mathcal{D} ; maximum training steps s ; number of prefixes n ;
 293 number of continuations m ; max resolve rounds R ; clipping thresholds $\varepsilon_{\text{low}}, \varepsilon_{\text{high}}$; update steps per
 294 batch μ

295 **for** step = 1,...,s **do**

- 296 1: Sample a mini-batch \mathcal{D}_b from \mathcal{D}
- 297 2: For each question $q \in \mathcal{D}_b$, sample n responses from $\pi_{\theta_{\text{old}}}(\cdot | q)$ and truncate them at random
 ratios to form prefixes $\{\text{Pre}_i\}_{i=1}^n$
- 298 3: For each prefix Pre_i , sample m continuations $\{O_{i,j}\}_{j=1}^m \sim \pi_{\theta_{\text{old}}}(\cdot | q, \text{Prefix}_i)$
- 299 4: Compute rewards $\{r_{i,j}\}$ according to Eq. 1
- 300 5: Filter out degenerate groups if all $\{r_{i,j}\}_{j=1}^m$ are identical
- 301 6: Compute group-wise advantages $\hat{A}_{i,j}$ for each continuation according to Eq. 2
- 302 7: **for** iteration = 1,..., μ **do**
 Update the policy model π_θ by maximizing $\mathcal{J}_{\text{Re}^2}(\theta)$ (Eq. 3)
- 303 8: Update the old policy $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$

304 **Output** updated policy π_θ

305

306

307

308

309

310

311

312

313

After computing rewards under the re-solving strategy, we compute group-wise advantages and
 update the policy parameters following DAPO (Yu et al., 2025). Specifically, the advantage for the
 j -th continuation $O_{i,j}$ of prefix Pre_i is defined as the reward normalized by subtracting the group
 mean and dividing by the group standard deviation:

314

315

316

317

$$\hat{A}_{i,j} = \frac{r_{i,j} - \text{mean}(\{r_{i,j}\}_{j=1}^m)}{\text{std}(\{r_{i,j}\}_{j=1}^m)}. \quad (2)$$

318

319

320

321

322

323

If all continuations within a group yield the same outcome (all correct, all incorrect, or all choosing
 resolve), then $\hat{A}_{i,j} = 0$ and the gradients vanish. Such degenerate groups are filtered out during
 training. The computed advantage is then broadcast to all response tokens of the corresponding
 continuation.

¹The detailed derivation is provided in Appendix B.

324 Formally, for each query $q \in \mathcal{D}$, we first sample n prefixes $\{\text{Pre}_i\}_{i=1}^n \sim \pi_{\theta_{\text{old}}}(\cdot | q)$. For each prefix
 325 Pre_i , we then sample m continuations $\{O_{i,j}\}_{j=1}^m \sim \pi_{\theta_{\text{old}}}(\cdot | q, \text{Pre}_i)$. The optimization objective is:
 326

$$\begin{aligned} 327 \quad \mathcal{J}_{\text{Re}^2}(\theta) &= \mathbb{E}_{[q \sim \mathcal{D}, \{\text{Pre}_i\}_{i=1}^n \sim \pi_{\theta_{\text{old}}}(\cdot | q), \{O_{i,j}\}_{j=1}^m \sim \pi_{\theta_{\text{old}}}(\cdot | q, \text{Pre}_i)]} \\ 328 \quad &\left[\frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m \frac{1}{|O_{i,j}|} \sum_{t=1}^{|O_{i,j}|} \min \left(\frac{\pi_{\theta}^{i,j,t}}{\pi_{\theta_{\text{old}}}^{i,j,t}} \hat{A}_{i,j}, \text{clip} \left(\frac{\pi_{\theta}^{i,j,t}}{\pi_{\theta_{\text{old}}}^{i,j,t}}, 1 - \varepsilon_{\text{low}}, 1 + \varepsilon_{\text{high}} \right) \hat{A}_{i,j} \right) \right], \quad (3) \\ 329 \end{aligned}$$

332 where $\pi^{i,j,t} = \pi(O_{i,j,t} | q, \text{Pre}_i, O_{i,j,<t})$ denotes the conditional probability of the t -th token in
 333 continuation $O_{i,j}$ given the query q and the prefix Pre_i . The models π_{θ} and $\pi_{\theta_{\text{old}}}$ correspond to
 334 the training policy and the sampling policy, respectively. The clipping thresholds ε_{low} and $\varepsilon_{\text{high}}$ are
 335 hyperparameters used to bound the importance sampling ratio for stable optimization. The overall
 336 training algorithm is summarized in Algorithm 1.

337 5 EXPERIMENTS

339 5.1 EXPERIMENTAL SETUP

340 **Training datasets.** We construct our training set from the DAPO-Math-17K dataset (Yu et al.,
 341 2025), which is collected from AoPS² and official competition sources. The dataset covers a wide
 342 range of mathematical domains and contains 17K problems in total. To ensure reliable rule-based
 343 reward signals and minimize parsing errors, all answers are transformed into integers.
 344

345 **Baselines & Models.** We compare Re^2 against the vanilla model (before RL training) and the
 346 recent RLVR method DAPO (Yu et al., 2025), which we follow for advantage computation and
 347 parameter updates. To ensure a fair comparison, both methods are trained with the same amount of
 348 generated tokens during RL optimization. To evaluate the effectiveness of Re^2 across model types and
 349 scales, we conduct experiments on both base and instruction-tuned LLMs, including Qwen-7B-Base,
 350 Qwen-14B-Base, Llama-3.2-3B-Instruct, and Qwen2.5-7B-Instruct (Dubey et al., 2024; Yang et al.,
 351 2024a). We further evaluate on DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025), a reasoning model
 352 specifically finetuned to generate long chains of thought.
 353

354 **Benchmarks.** To comprehensively evaluate the reasoning ability of our model, We adopt five widely
 355 used benchmarks covering diverse difficulty levels and domains: **AIME 2024** (MAA Committees)
 356 contains 30 challenging problems from the 2024 American Invitational Mathematics Examination.
 357 The exam is designed to test advanced problem-solving skills across algebra, geometry, combinatorics,
 358 number theory, and probability, and is often used as a challenging benchmark for evaluating reasoning
 359 ability in language models. **AIME 2025** (MAA Committees) follows the same format as AIME
 360 2024, with 30 comparably difficult problems. As the most recent edition, it reduces the risk of
 361 contamination from pretraining or post-training data. **AMC 2023** (AI-MO, 2024) consists of 40
 362 problems covering algebra, geometry, number theory, and combinatorics. Compared to the AIME
 363 benchmarks, its difficulty level is relatively lower. **GSM8K** (Cobbe et al., 2021) is a curated dataset of
 364 1,319 elementary-level math word problems. Each problem typically requires two to eight reasoning
 365 steps, primarily involving multi-step arithmetic, making it a standard benchmark for assessing
 366 fundamental mathematical reasoning. **GPQA** (Rein et al., 2021) is a challenging dataset of graduate-
 367 level questions in physics, biology, and chemistry, where even PhD-level domain experts achieve only
 368 around 69.7% accuracy. In our experiments, we use the highest-quality subset, **GPQA-Diamond**,
 369 which consists of 198 carefully selected questions designed to provide a rigorous test of advanced
 370 scientific reasoning.

371 **Training and evaluation details.** For Re^2 , we use a learning rate of 1×10^{-6} . Each training step
 372 processes a batch of 32 queries, with $n = 8$ prefixes sampled per query and $m = 8$ continuations
 373 generated for each prefix. The maximum sequence length is set to 8192 tokens. The clipping
 374 parameters are fixed at $\varepsilon_{\text{low}} = 0.2$ and $\varepsilon_{\text{high}} = 0.28$, and the maximum number of redo rounds is
 375 $R = 5$. For DAPO, we adopt the same learning rate of 1×10^{-6} . To ensure comparable token budgets
 376 with Re^2 , each batch contains 128 queries with $n = 20$ samples per query. All other hyperparameters
 377

²<https://artofproblemsolving.com/>

378 Table 1: Experimental results on five reasoning benchmarks. Re^2 consistently improves the overall
 379 reasoning performance of each model over DAPO (p -value < 0.05). Red numbers in parentheses
 380 indicate performance gains relative to DAPO.

382	Models	Methods	AIME24	AIME25	AMC23	GSM8K	GPQA	Avg
383 Base Model								
385	Qwen2.5-7B Base	+ DAPO	11.9	10.3	64.7	91.8	29.7	41.7
386		+ Re^2	17.1	19.0	70.8	93.6	36.8	47.5 (+5.8)
387 Instruct Model								
390	Llama3.2-3B-Instruct	None	6.2	0.4	23.0	67.2	2.7	19.9
391		+ DAPO	15.0	0.5	32.3	80.4	20.7	29.8
392		+ Re^2	17.7	2.8	38.4	83.2	20.2	32.5 (+2.7)
393	Qwen2.5-7B-Instruct	None	11.4	7.5	51.4	85.3	33.4	37.8
394		+ DAPO	16.0	8.6	62.3	92.6	35.4	43.0
395		+ Re^2	18.6	21.2	64.7	94.1	38.4	47.4 (+4.4)
396 Reasoning Model								
398	DeepSeek-R1- Distill-Llama-8B	None	39.3	27.3	84.3	88.6	36.9	55.2
399		+ DAPO	38.4	26.5	86.9	89.6	38.4	55.9
400		+ Re^2	47.2	29.6	88.7	92.2	44.8	60.5 (+4.4)

402 are kept identical to those used in Re^2 . During evaluation, the maximum sequence length is increased
 403 to 16384 tokens, with sampling performed using a temperature of 0.6 and top- p of 0.95. For models
 404 trained with Re^2 , whenever a sampled completion produces a *redo* action, sampling is restarted until
 405 a final answer is generated, and the first valid final answer is taken as the model’s output.

407 5.2 MAIN RESULTS

409 As shown in Table 1, Re^2 improves reasoning performance across all five benchmark datasets and five
 410 model types, including base, instruction-tuned, and reasoning-optimized models ranging from 3B to
 411 14B parameters. On pretrained models such as Qwen2.5-7B and Qwen2.5-14B, our method achieves
 412 larger gains compared to DAPO. These consistent gains on in-domain mathematical benchmarks
 413 of varying difficulty (AIME24, AIME25, AMC, GSM8K) as well as the out-of-domain scientific
 414 reasoning benchmark (GPQA-Diamond) demonstrate the robustness of our approach. Moreover,
 415 since AIME25 was released after all the evaluated models were trained, it is free from potential
 416 data contamination, and Re^2 achieves superior performance on this benchmark, further validating its
 417 effectiveness. Notably, Re^2 achieves substantial improvements on AIME24 and AIME25, highlighting
 418 its effectiveness in tackling more challenging reasoning problems.

419 5.3 PERFORMANCE UNDER TEST-TIME SCALING

421 When tackling challenging problems (e.g., the AIME series), models trained with Re^2 may perform
 422 multiple redo attempts and generate several candidate solutions before producing a final answer,
 423 thereby consuming more tokens during inference. To fairly assess the effect of this additional token
 424 usage, we compare DAPO and Re^2 under the same number of sampled outputs, regardless of whether
 425 a sample corresponds to a direct final answer or a redo attempt. Accuracy is then measured using
 426 majority voting over these samples.

427 As shown in Figure 6, Re^2 fully exploits the benefits of test-time scaling: once the number of
 428 samples exceeds 64, they consistently surpass RLVR-trained models, whose performance has already
 429 saturated, and continue to improve as test-time compute increases. However, when the number
 430 of samples is small, Re^2 tends to trigger more redo actions on hard problems, which reduces the
 431 proportion of valid final answers within the sampled outputs and can lead to lower accuracy than
 RLVR under these settings.

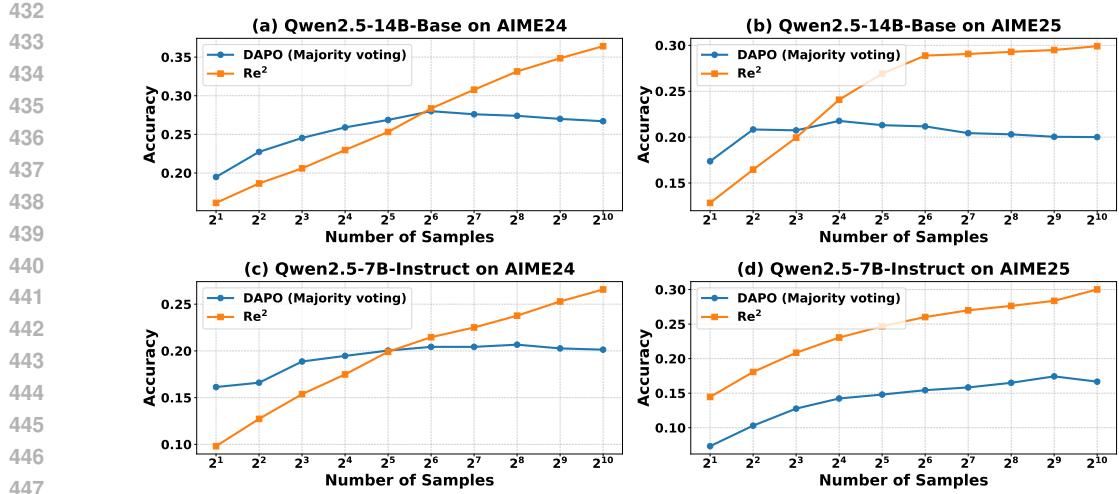


Figure 6: Test-time scaling of DAPO and Re^2 under the same training budget as the number of samples increases.

5.4 TRAINING DYNAMICS OF Re^2

Figure 7 shows the training dynamics of Re^2 , including the average reward, the reward under the resolve action, and the behavioral distribution over correct answers, incorrect answers, and re-solve actions. Both the average reward and the reward for resolving steadily increase as training progresses. In terms of behavior, the probability of choosing to re-solve rises sharply during the first 20 steps and then gradually decreases with further training. Meanwhile, the probability of producing incorrect answers drops substantially, while the probability of generating correct final answers increases slowly. These results suggest that Re^2 rapidly activates resolving behavior and then refines it, enabling the model to abandon unproductive reasoning paths while maintaining exploration of correct but more challenging trajectories.



Figure 7: The training progress of Re^2 on Qwen2.5-14B-Base.

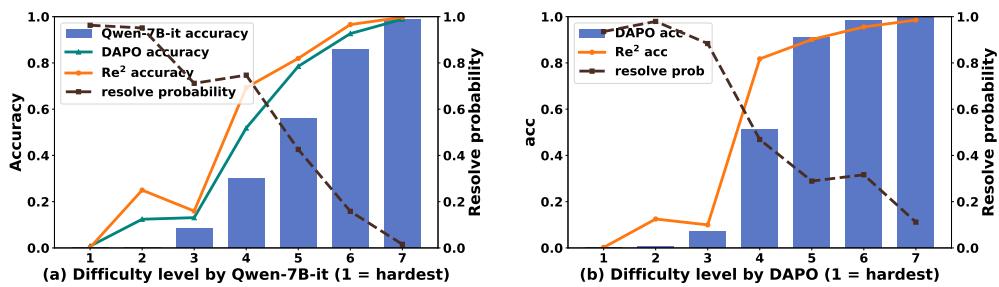
5.5 ANALYZING THE MECHANISMS BEHIND Re^2 ’S EFFECTIVENESS

To further investigate the underlying mechanisms behind the effectiveness of our method, we manually inspected sampled cases and analyzed accuracy improvements across different difficulty levels. Through manual inspection, we found that Re^2 produces reasoning chains of consistently higher quality than RLVR algorithms such as DAPO. This is largely because Re^2 allows the model to **restart reasoning when the current trajectory is unlikely to lead to the correct answer**, thereby **substantially reducing the tendency to force a final answer from flawed reasoning or nonexistent conditions**. We believe that the reward modeling in Re^2 is fundamentally more rational than the standard 0/1 end-reward paradigm in RLVR. Under a pure end-reward objective, the model is encouraged to always output a final answer—even when it is uncertain—often producing spurious steps and incoherent logic in an attempt to “guess correctly.” In contrast, Re^2 enables the model to output a final answer only when it is confident, and to honestly indicate the need to restart when the reasoning becomes unpromising. This more rational and self-aware behavior allows the model to

486 better recognize when its current chain of thought is unreliable and to avoid optimizing trajectories
 487 that accidentally guess the correct answer, which is a common issue in RLVR training.
 488

489 To illustrate these findings, we categorized problems by difficulty and measured the accuracy and
 490 resolving rate of Re^2 . We mixed AMC and AIME25 questions to create a test set with a balanced
 491 distribution of difficulty levels. In Fig. 8(a), we grouped questions into seven difficulty levels based
 492 on the accuracy of the Qwen2.5-7B-Instruct model obtained through multiple samples. We then
 493 evaluated DAPO and Re^2 on each difficulty group and recorded Re^2 ’s resolving rate. We observe
 494 that for questions the base model is completely unable to solve (Group 1), reinforcement learning
 495 cannot teach the model to solve them either. In such cases, DAPO often produces incorrect answers
 496 with unclear or erroneous reasoning chains, while Re^2 almost always refrains from giving a final
 497 answer and attempts to resolve the problem instead. For difficult but solvable questions (Group 2),
 498 Re^2 achieves more than twice the accuracy of DAPO due to its ability to restart from failed prefixes.
 499 Across all difficulty levels, Re^2 consistently outperforms DAPO, and its resolving rate decreases as
 500 question difficulty decreases.

500 In Fig. 8(b), we group questions by the difficulty estimated by DAPO, providing a more direct
 501 comparison against a standard RLVR method. We find that the largest improvement occurs on
 502 questions that RLVR occasionally solves (Group 4), where accuracy increases from 51.2% to 81.7%.
 503 These are questions that RLVR models are capable of solving, but their ability to answer correctly is
 504 highly unstable because they cannot discard unpromising prefixes. Re^2 overcomes this limitation:
 505 by allowing the model to restart, it substantially increases the probability of following a successful
 506 reasoning trajectory.



518 Figure 8: Analysis of accuracy and behavior across problem difficulty levels. (a) Accuracy improve-
 519 ments of DAPO and Re^2 , along with the resolving rate of Re^2 , when questions are grouped by the
 520 difficulty estimated by the base model. (b) Accuracy improvements and resolving rate of Re^2 when
 521 questions are grouped by the difficulty estimated by DAPO.

524 6 CONCLUSION

526 In this paper, we investigate the limitations of test-time scaling in existing LLMs and show that once
 527 early reasoning is misguided, the model rarely recovers and struggles to return to the correct reasoning
 528 path. To this end, we propose a new reasoning paradigm that allows language models to flexibly
 529 restart reasoning when the current trajectory appears unpromising. We introduce Reinforcement
 530 Learning with Re-solving (Re^2), which leverages pure reinforcement learning to encourage models
 531 to adopt re-solving behaviors. Empirical evaluations demonstrate that Re^2 consistently outperforms
 532 standard RLVR methods across benchmarks of varying domains and difficulty levels, while also
 533 raising the upper bound of performance achievable under test-time scaling.

535 ETHICS STATEMENT

538 This paper presents work aimed at advancing the field of reasoning with large language models. As
 539 with all research involving large language models, there are inherent risks, including the spread of
 misinformation and the propagation of societal biases.

540 REPRODUCIBILITY STATEMENT
541542 Our work is based on open-source models and datasets. In Section 5 and Appendix A, we provide
543 detailed descriptions of the prompt templates, method implementation, and experimental setups.
544545 REFERENCES
546547 AI-MO. Amc 2023, 2024. URL <https://huggingface.co/datasets/AI-MO/aimo-validation-amc>.
548549 Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil,
550 Zach Moshe, Tomer Ronen, Najeeb Nabwani, et al. Llama-nemotron: Efficient reasoning models.
551 *arXiv preprint arXiv:2505.00949*, 2025.
552553 Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu,
554 Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of
555 o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.
556557 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
558 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John
559 Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*,
560 2021.561 Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu
562 Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. The danger of overthinking: Examining
563 the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*, 2025.
564565 Jhouben Cuesta-Ramirez, Samuel Beaussant, and Mehdi Mounif. Large reasoning models are not
566 thinking straight: on the unreliability of thinking trajectories. *arXiv preprint arXiv:2507.00711*,
567 2025.568 Bowen Ding, Yuhang Chen, Futing Wang, Lingfeng Ming, and Tao Lin. Do thinking tokens help or
569 trap? towards more efficient large reasoning model. *arXiv preprint arXiv:2506.23840*, 2025.
570571 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha
572 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models.
573 *arXiv e-prints*, pp. arXiv–2407, 2024.574 Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. *arXiv
575 preprint arXiv:2508.15260*, 2025.
576577 Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and
578 Noah D Goodman. Stream of search (sos): Learning to search in language. *arXiv preprint
579 arXiv:2404.03683*, 2024.580 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
581 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
582 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
583584 Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning
585 with language model is planning with world model. In *NeurIPS 2023 Workshop on Generalization
586 in Planning*, 2023.587 Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Mingjie Liu, Min-Hung Chen, Hongxu Yin,
588 Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, et al. Dler: Doing length penalty
589 right-incentivizing more intelligence per token via reinforcement learning. *arXiv preprint
590 arXiv:2510.15110*, 2025.
591592 Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai,
593 Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview
with a 1.5b model by scaling rl, 2025. Notion Blog.

594 MAA Committees. Aime problems and solutions. [https://artofproblemsolving.com/
595 wiki/index.php/AIME_Problems_and_Solutions](https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions).

596

597 Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke
598 Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. s1: Simple test-time
599 scaling. In *Workshop on Reasoning and Planning for Large Language Models*, 2025.

600 OpenAI. Learning to reason with llms, 9 2024. URL [https://openai.com/index/
601 learning-to-reason-with-llms/](https://openai.com/index/learning-to-reason-with-llms/).

602

603 Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West,
604 and Boi Faltings. Refiner: Reasoning feedback on intermediate representations. In *Proceedings of
605 the 18th Conference of the European Chapter of the Association for Computational Linguistics
(Volume 1: Long Papers)*, pp. 1100–1126, 2024.

606

607 Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan,
608 Hector Liu, Yuanzhi Li, et al. O1 replication journey: A strategic progress report–part 1. *arXiv
609 preprint arXiv:2410.18982*, 2024.

610

611 Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. Recursive introspection: Teaching
612 language model agents how to self-improve. *Advances in Neural Information Processing Systems*,
613 37:55249–55285, 2024.

614

615 David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani,
616 Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In
617 *First Conference on Language Modeling*, 2021.

618

619 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy
620 optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

621

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

625

626 Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad
627 Farajtabar. The illusion of thinking: Understanding the strengths and limitations of reasoning
628 models via the lens of problem complexity. *arXiv preprint arXiv:2506.06941*, 2025.

629

630 Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally
631 can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

632

633 Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun
634 Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with
635 llms. *arXiv preprint arXiv:2501.12599*, 2025.

636

637 Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. Reasoning aware self-consistency: Leveraging
638 reasoning paths for efficient llm sampling. *arXiv preprint arXiv:2408.17017*, 2024.

639

640 Shuhe Wang, Shengyu Zhang, Jie Zhang, Runyi Hu, Xiaoya Li, Tianwei Zhang, Jiwei Li, Fei Wu,
641 Guoyin Wang, and Eduard Hovy. Reinforcement learning enhanced llms: A survey. *arXiv preprint
642 arXiv:2412.10400*, 2024.

643

644 Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha
645 Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language
646 models. In *The Eleventh International Conference on Learning Representations*, 2022.

647

648 Yue Wang, Qiuwei Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu,
649 Juntao Li, Zhuosheng Zhang, et al. Thoughts are all over the place: On the underthinking of
650 o1-like llms. *arXiv preprint arXiv:2501.18585*, 2025.

651

652 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny
653 Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in
654 neural information processing systems*, 35:24824–24837, 2022.

648 Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig,
 649 Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms
 650 for large language models. *Transactions on Machine Learning Research*, 2024.

651

652 Hao Wen, Yifan Su, Feifei Zhang, Yunxin Liu, Yunhao Liu, Ya-Qin Zhang, and Yuanchun Li.
 653 Parathinker: Native parallel thinking as a new paradigm to scale llm test-time compute. *arXiv*
 654 *preprint arXiv:2509.04475*, 2025.

655

656 Siwei Wu, Zhongyuan Peng, Xinrun Du, Tuney Zheng, Minghao Liu, Jialong Wu, Jiachen Ma, Yizhi
 657 Li, Jian Yang, Wangchunshu Zhou, et al. A comparative study on reasoning patterns of openai's o1
 658 model. *arXiv preprint arXiv:2410.13639*, 2024.

659

660 Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws:
 661 An empirical analysis of compute-optimal inference for llm problem-solving. In *The Thirteenth*
 662 *International Conference on Learning Representations*, 2025.

663

664 Zhiheng Xi, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang
 665 Hong, Shihan Do, Wenyu Zhan, et al. Enhancing llm reasoning via critique models with test-time
 666 and training-time supervision. *arXiv preprint arXiv:2411.16579*, 2024.

667

668 Wei Xiong, Hanning Zhang, Chenlu Ye, Lichang Chen, Nan Jiang, and Tong Zhang. Self-rewarding
 669 correction for mathematical reasoning. *arXiv preprint arXiv:2502.19613*, 2025.

670

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

674

675 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang
 676 Gao, Chengan Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*,
 677 2025a.

678

679 Chenguang Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun
 680 Chen. Large language models as optimizers. In *The Twelfth International Conference on Learning*
 681 *Representations*, 2023.

682

683 Ling Yang, Zhaochen Yu, Tianjun Zhang, Minkai Xu, Joseph E Gonzalez, Bin Cui, and Shuicheng
 684 Yan. Supercorrect: Advancing small llm reasoning with thought template distillation and self-
 685 correction. *arXiv preprint arXiv:2410.09008*, 2024b.

686

687 Xiao-Wen Yang, Xuan-Yi Zhu, Wen-Da Wei, Ding-Chu Zhang, Jie-Jing Shao, Zhi Zhou, Lan-Zhe
 688 Guo, and Yu-Feng Li. Step back to leap forward: Self-backtracking for boosting reasoning of
 689 language models. *arXiv preprint arXiv:2502.04404*, 2025b.

690

691 Qiyi Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian
 692 Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at
 693 scale. *arXiv preprint arXiv:2503.14476*, 2025.

694

695 Yu Yue, Yufeng Yuan, Qiyi Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi
 696 Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for
 697 advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

698

699 Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-
 700 zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025.
 701 URL <https://arxiv.org/abs/2503.18892>.

702

703 Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm
 704 self-training via process reward guided tree search. In *The Thirty-eighth Annual Conference on*
 705 *Neural Information Processing Systems*, 2024.

706

707 Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai
 708 Tian, Guoli Jia, Pengfei Li, et al. A survey of reinforcement learning for large reasoning models.
 709 *arXiv preprint arXiv:2509.08827*, 2025.

702 Xutong Zhao, Tengyu Xu, Xuewei Wang, Zhengxing Chen, Di Jin, Liang Tan, Zishun Yu, Zhuokai
 703 Zhao, Yun He, Sinong Wang, et al. Boosting llm reasoning via spontaneous self-correction. *arXiv*
 704 *preprint arXiv:2506.06923*, 2025.

706 Yu Zhao, Huirong Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo,
 707 and Kaifu Zhang. Marco-01: Towards open reasoning models for open-ended solutions. *arXiv*
 708 *preprint arXiv:2411.14405*, 2024.

710 Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang,
 711 Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint*
 712 *arXiv:2507.18071*, 2025.

716 A IMPLEMENTATION DETAILS

718 During training, Re^2 employs a specialized template, as illustrated in Figure 9, to elicit redo behavior
 719 from the vanilla model. The template is designed to encourage the model to explicitly indicate when
 720 it decides to abandon the current reasoning trajectory and restart the problem.

722 When performing reward assignment, if the model outputs a phrase such as “It’s better to
 723 redo the question.” before producing a boxed final answer, we truncate the response at
 724 that point and append a special `<eos>` token to mark the end of the sequence. This ensures that
 725 redo actions are clearly distinguished from final-answer completions and allows the policy to learn
 726 consistent behavior during reinforcement learning.

728 Re^2 template

729 **User:** Solve the following math problem step by step. If you obtain a final answer, enclose it in
 730 boxed{ }.

732 {QUESTION}

734 Note: As you work through the problem, continuously assess your solution path in real time. If
 735 you feel your current approach has become unclear or that rethinking the problem from scratch
 736 would be more productive, **simply say "It’s better to redo the question."** Once you say this,
 737 the answer should be immediately terminated (so do not use this phrase lightly—when reflecting
 738 to yourself, use "Do I need to redo the question?" instead). When faced with unclear or tangled
 739 reasoning, don’t hesitate to make use of the option to start over. Use this choice wisely for the
 740 best results.

741 **Assistant:**

743 Figure 9: The template of Re^2 .

748 B DERIVATION OF THE REWARD OF RE^2

750 We derive the expected reward for choosing `resolve` with at most R allowed rounds. For out-of-
 751 group completions (i.e., prefixes $\neq \text{Pre}_i$), let $P_{\neq i}(\text{correct})$, $P_{\neq i}(\text{incorrect})$, $P_{\neq i}(\text{resolve})$
 752 denote the empirical probabilities of the three outcomes, estimated from the $(n-1) \times m$ out-of-group
 753 CoTs. Each redo round draws one outcome from this distribution; a `correct` yields reward 1,
 754 an `incorrect` yields reward 0, and a `resolve` consumes one round and restarts from the same
 755 distribution. Hence, the expected reward of choosing `resolve` equals the probability that the first
 non-`resolve` outcome within the first R rounds is `correct`:

$$\begin{aligned}
756 \\
757 \\
758 \quad \mathbb{E}[r \mid \text{resolve}, R] &= \sum_{t=1}^R \underbrace{P_{\neq i}(\text{resolve})^{t-1}}_{\text{first } t-1 \text{ are resolve}} \cdot \underbrace{P_{\neq i}(\text{correct})}_{\text{the } t\text{-th is correct}} \\
759 \\
760 \\
761 \\
762 \\
763 \\
764 \quad &= P_{\neq i}(\text{correct}) \sum_{t=0}^{R-1} P_{\neq i}(\text{resolve})^t.
\end{aligned}$$

Evaluating the finite geometric series gives

$$\mathbb{E}[r \mid \text{resolve}, R] = P_{\neq i}(\text{correct}) \cdot \frac{1 - P_{\neq i}(\text{resolve})^R}{1 - P_{\neq i}(\text{resolve})}.$$

Therefore, for $O_{i,j}$ that chooses `resolve`, we set

$$r_{i,j} = P_{\neq i}(\text{correct}) \cdot \frac{1 - P_{\neq i}(\text{resolve})^R}{1 - P_{\neq i}(\text{resolve})}.$$

770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

810
811 Table 2: Accuracy with 95% confidence intervals on five reasoning benchmarks, confidence intervals
812 are given in parentheses.

813 Models	814 Methods	815 AIME24	816 AIME25	817 AMC23	818 GSM8K	819 GPQA
820 Base Model						
821 Qwen2.5-7B Base	822 + DAPO	823 11.9 (± 1.0)	824 10.3 (± 1.0)	825 64.7 (± 1.3)	826 91.8 (± 0.1)	827 29.7 (± 0.5)
828 + Re ²	829 17.1 (± 1.4)	830 19.0 (± 1.2)	831 70.8 (± 0.3)	832 93.6 (± 0.0)	833 36.8 (± 0.3)	
834 Instruct Model						
835 Llama3.2-3B-Instruct	836 + DAPO	837 15.0 (± 0.9)	838 0.5 (± 0.3)	839 32.3 (± 1.3)	840 80.4 (± 0.2)	841 20.7 (± 0.5)
842 + Re ²	843 17.7 (± 1.1)	844 2.8 (± 0.5)	845 38.4 (± 0.8)	846 83.2 (± 0.1)	847 20.2 (± 0.3)	
848 Qwen2.5-7B-Instruct	849 + DAPO	850 16.0 (± 1.1)	851 8.6 (± 0.9)	852 62.3 (± 1.3)	853 92.6 (± 0.1)	854 35.4 (± 0.6)
855 + Re ²	856 18.6 (± 1.6)	857 21.2 (± 1.1)	858 64.7 (± 0.4)	859 94.1 (± 0.0)	860 38.4 (± 0.4)	
861 Reasoning Model						
862 DeepSeek-R1-	863 + DAPO	864 38.4 (± 1.5)	865 26.5 (± 1.4)	866 86.9 (± 0.9)	867 89.6 (± 0.1)	868 38.4 (± 0.6)
869 Distill-Llama-8B	870 + Re ²	871 47.2 (± 0.7)	872 29.6 (± 0.8)	873 88.7 (± 0.2)	874 92.2 (± 0.0)	875 44.8 (± 0.3)

829 C EXPERIMENTS

830 C.1 MAIN RESULTS

831 We additionally provide a comparison between DAPO and Re² with confidence intervals, as shown
832 in Table 2.

833 C.2 PERFORMANCE UNDER TEST-TIME SCALING

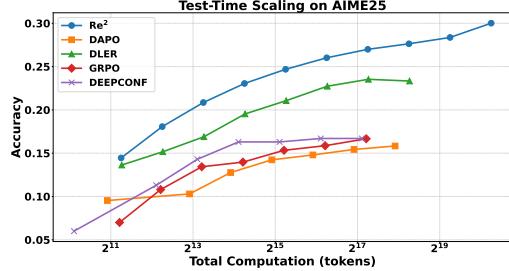
834 We further compare the test-time scaling performance of Re² with a broader set of baselines.
835 These include: **GRPO** (Shao et al., 2024), the
836 classical RLVR algorithm and the core technique
837 used in training DeepSeek-R1 (Guo et al., 2025);
838 **DLER** (Liu et al., 2025), the recent state-of-the-
839 art efficient reasoning method that reduces to-
840 ken consumption while maintaining performance
841 through truncated-length penalties and training-
842 stabilization strategies; and **DeepConf** (Fu et al.,
843 2025), which leverages internal confidence sig-
844 nals during decoding to dynamically terminate
845 low-quality reasoning traces.

846 We use the number of consumed tokens as the measure of computational cost and evaluate them on
847 the challenging AIME25 benchmark, which has no risk of data leakage. As shown in Figure 10, our
848 method achieves better test-time scaling than all competing approaches.

849 C.3 ANALYSIS OF THE RESOLVE REWARD ESTIMATOR

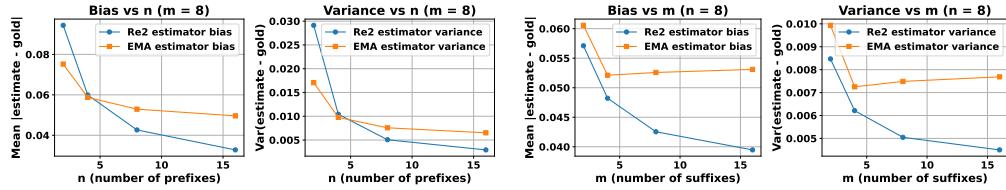
850 To better understand the performance of the resolve reward estimator in Re², we evaluate how
851 accurately it estimates a model’s resolving accuracy. For each training question, we draw 1024 inde-
852 pendent samples and treat the proportion of correct responses as the ground-truth resolving accuracy.
853 We then compare the bias and variance of the estimator under different sampling configurations.

854 Specifically, we fix the number of suffixes at $m = 8$ and vary the number of prefixes $n \in \{2, 4, 8, 16\}$,
855 and conversely fix $n = 8$ while varying $m \in \{2, 4, 8, 16\}$. As a baseline, we include an exponential
856 moving average (EMA) estimator with a decay rate of 0.9.



857 Figure 10: Test-time scaling of Re² compared
858 with additional baselines.

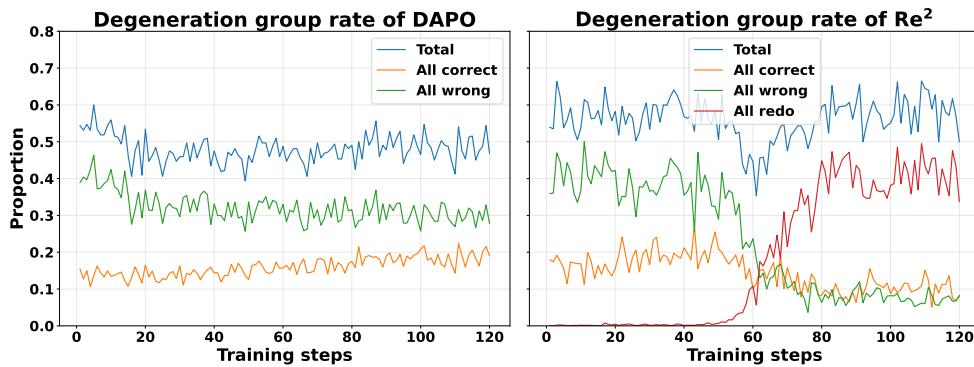
864 As shown in Fig. 11, the Re^2 reward estimator becomes increasingly accurate as either n or m
 865 increases, exhibiting consistently lower bias and variance. It also outperforms the EMA baseline
 866 across all settings, demonstrating the effectiveness of leveraging the naturally generated $n \times m$ suffix
 867 samples for estimating resolving accuracy.
 868



869
 870
 871
 872
 873
 874
 875
 876 Figure 11: Bias and variance of the estimated resolving accuracy under different values of n and m .
 877
 878

879 C.4 ANALYSIS OF THE DEGENERATION GROUP RATE

880 We report the degeneration group rate of DAPO and Re^2 , defined as the rate of groups in which all
 881 samples receive the same reward and therefore have zero advantage. As shown in Fig. 12, in Re^2 ,
 882 degeneration groups that are “all-wrong” during the later training stages gradually turn into “all-redo,”
 883 indicating that the model shifts from forcing an answer on unsolvable questions to choosing to redo
 884 them. The overall degeneration rate of Re^2 is about 10% higher than that of DAPO. However, as
 885 noted in DAPO (Yu et al., 2025), “*the filter strategy does not necessarily impede training efficiency,*
 886 *because the generation time is typically dominated by the generation of long-tail samples if the RL*
 887 *system is synchronized and the generation stage is not pipelined.*” Consistent with this observation,
 888 Re^2 does not incur additional training cost due to this effect.
 889



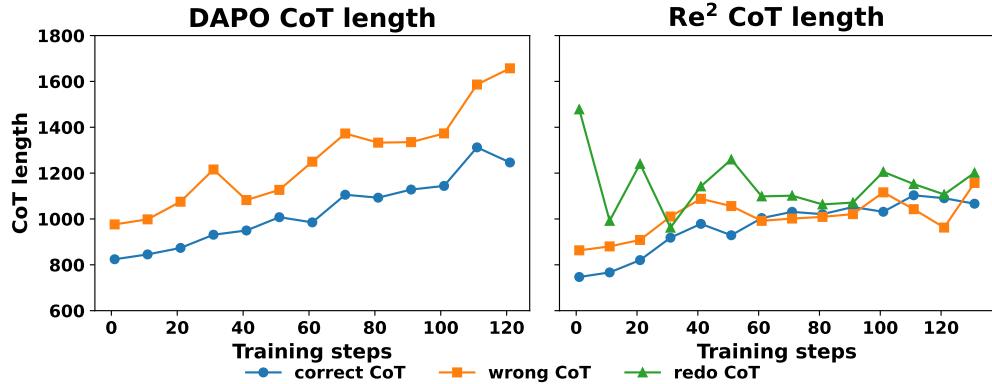
900
 901
 902
 903 Figure 12: Degeneration group rate during training for DAPO and Re^2 .
 904
 905

906 C.5 ANALYSIS OF THE REASONING LENGTH

907 We analyze the evolution of reasoning lengths for DAPO and Re^2 throughout training. As shown in
 908 Figure 13, combined with our inspection of a number of cases, DAPO exhibits steady growth in CoT
 909 length during training due to increasing amounts of self-reflection, error correction, and switching
 910 of reasoning strategies, similar to classical RLVR. Wrong CoTs are noticeably longer than correct
 911 ones because many failed trajectories attempt to extend or patch flawed prefixes. This represents
 912 the strategy optimization that occurs under the one-shot chain-of-thought paradigm. In contrast,
 913 Re^2 gradually stabilizes its CoT lengths, and the lengths of correct, wrong, and redo CoTs do not
 914 show substantial differences. Both the statistical results and our qualitative observations indicate
 915 that Re^2 behaves more rationally under the multi-chain, resolving reasoning paradigm: the model
 916 does not force itself into producing a strained and ultimately incorrect reasoning chain, nor does it
 917 over-commit to unpromising trajectories.

918 In addition, redo-CoTs in Re^2 tend to be longer at the beginning of training, and their lengths gradually
 919 approach those of final correct or incorrect CoTs as training progresses. Consistent with our manual

918 inspection of cases, early in training, the patterns that trigger a redo are relatively shallow—for
 919 CoTs becoming excessively long and close to the context window limit, frequent switching
 920 between reasoning threads, or resorting to brute-force enumeration on problems that actually require
 921 summarizing underlying patterns. In later stages, however, redo decisions become more closely tied
 922 to the intrinsic quality of the CoT, such as the effectiveness of the chosen approach or the soundness
 923 of the assumptions.



939 Figure 13: Reasoning lengths during training for DAPO and Re^2 .
 940
 941

942 D TRAINING COST ANALYSIS OF Re^2

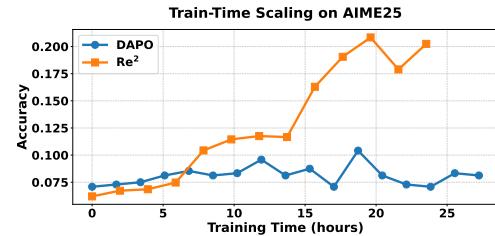
943 Training Re^2 involves a two-stage generation process: the model first produces n prefixes and then
 944 generates $n \times m$ suffixes. This sequential generation
 945 procedure can be more time-consuming than
 946 directly generating responses in parallel. To quanti-
 947 fy this overhead, we measure the rollout time of
 948 DAPO and Re^2 on an 8xA100-PCIE-40GB server.
 949

950 To produce the same number of rollout samples,
 951 DAPO uses a global batch size of 128, generat-
 952 ing 16 samples per question. Re^2 uses a global
 953 batch size of 32, generating 8 prefixes followed
 954 by 8 suffixes for each prefix. On average, DAPO
 955 requires 388 seconds per rollout step. In compari-
 956 son, Re^2 takes 89 seconds for prefix generation and
 957 342 seconds for suffix generation, resulting in a total of 431 seconds, which corresponds to an 11%
 958 increase in rollout time.
 959

960 Considering the performance improvements Re^2 provides during training, this additional cost is
 961 acceptable. As shown in Fig. 14, Re^2 outperforms DAPO under the same training time budget.
 962

963 E CASE ANALYSIS OF COTs WITH DIFFERENT LENGTHS

964 Here, we present a case study comparing short and long CoT responses obtained from multiple
 965 samples on the same problem in Figures 15, 16, 17, and 18. When the model follows a correct
 966 reasoning path with high-quality initial steps, it can often reach the correct answer within a relatively
 967 short CoT. In contrast, when the initial reasoning direction is misguided, the model tends to produce
 968 longer CoTs with many additional tokens, yet still fails to arrive at the correct solution. This illustrates
 969 the critical importance of early reasoning quality and highlights the inefficiency of over-extended
 970 reasoning when the initial trajectory is flawed.
 971



966 Figure 14: Accuracy improvements of DAPO
 967 and Re^2 under the same training time.
 968

972
973
974
975
976

977 longer and shorter response in AIME25

978
979 **Problem:** Let k be real numbers such that the system $|25+20i-z| = 5$ and $|z-4-k| = |z-3i-k|$
980 has exactly one complex solution z . The sum of all possible values of k can be written as $\frac{m}{n}$,
981 where m and n are relatively prime positive integers. Find $m+n$. Here $i = \sqrt{-1}$.982 **Shorter Response:** Okay, so I need to solve this problem where we have two complex equations:983 1. $|25 + 20i - z| = 5$
984 2. $|z - 4 - k| = |z - 3i - k|$ 985 And the system has exactly one complex solution z . We need to find the sum of all
986 possible values of k , express it as m/n where m and n are coprime, and then find
987 $m+n$. Alright, let's break this down step by step. First, let's understand what these equations
988 represent geometrically. Since z is a complex number, we can write $z = x + yi$. Then,
989 $25 + 20i - z = 25 + 20i - (x + yi) = (25 - x) + (20 - y)i$. The modulus of this is
990 $\sqrt{(25 - x)^2 + (20 - y)^2} = 5$. So, squaring both sides: $(25 - x)^2 + (20 - y)^2 = 25$. That's the
991 equation of a circle in the complex plane with center at $(25, 20)$ and radius 5. So, z lies on this
992 circle. Now, moving to the second equation: $|z - 4 - k| = |z - 3i - k|$. Let me write z as $x + yi$
993 again. Then Left side = $|(x + yi) - 4 - k| = |(x - 4 - k) + yi| = \sqrt{(x - 4 - k)^2 + y^2}$, Right
994 side = $|(x + yi) - 3i - k| = |(x - k) + (y - 3)i| = \sqrt{(x - k)^2 + (y - 3)^2}$. Setting
995 them equal: $\sqrt{(x - 4 - k)^2 + y^2} = \sqrt{(x - k)^2 + (y - 3)^2}$. Squaring both
996 sides: $(x - 4 - k)^2 + y^2 = (x - k)^2 + (y - 3)^2$. Let's expand both sides: Left side:
997 $(x - (4 + k))^2 + y^2 = (x - 4 - k)^2 + y^2 = (x^2 - 2(4 + k)x + (4 + k)^2) + y^2$.
998 Right side: $(x - k)^2 + (y - 3)^2 = (x^2 - 2kx + k^2) + (y^2 - 6y + 9)$. Set them
999 equal: $x^2 - 2(4 + k)x + (4 + k)^2 + y^2 = x^2 - 2kx + k^2 + y^2 - 6y + 9$. Simplify both sides by
1000 subtracting x^2 and y^2 : $-2(4 + k)x + (4 + k)^2 = -2kx + k^2 - 6y + 9$. Bring all terms to the
1001 left: $-2(4 + k)x + (4 + k)^2 + 2kx - k^2 + 6y - 9 = 0$. Simplify term by term: First, combine the
1002 x terms: $-2(4 + k)x + 2kx = [-8 - 2k + 2k]x = -8x$. Next, constants: $(4 + k)^2 - k^2 - 9$.
1003 Let's compute: $(16 + 8k + k^2) - k^2 - 9 = 16 + 8k + k^2 - k^2 - 9 = 7 + 8k$. So now,
1004 equation becomes: $-8x + 7 + 8k + 6y = 0$. So that's the equation of the second condition. So
1005 now, we have two conditions: 1. z lies on the circle with center $(25, 20)$ and radius 5;
1006 2. z lies on the line defined by $-8x + 6y + 7 + 8k = 0$. Since the system has exactly
1007 one solution, this implies that the line is tangent to the circle. So, the distance from the
1008 center of the circle to the line must be equal to the radius. So, the center of the circle is
1009 $(25, 20)$, and the line is $-8x + 6y + 7 + 8k = 0$. The distance from the center $(25, 20)$ to the
1010 line is equal to the radius 5. The formula for the distance from a point (x_0, y_0) to the line
1011 $ax + by + c = 0$ is $|ax_0 + by_0 + c|/\sqrt{a^2 + b^2}$. In our case, the line is $-8x + 6y + (7 + 8k) = 0$,
1012 so $a = -8$, $b = 6$, $c = 7 + 8k$. The distance is $|-8 * 25 + 6 * 20 + 7 + 8k|/\sqrt{((-8)^2 + 6^2)} =$
1013 $|-200 + 120 + 7 + 8k|/\sqrt{64 + 36} = |(-200 + 120 + 7) + 8k|/\sqrt{100} = |(-73) + 8k|/10$.
1014 Set this equal to the radius 5: $|-73 + 8k|/10 = 5 \Rightarrow |8k - 73| = 50$. Therefore, $8k - 73 = 50$
1015 or $8k - 73 = -50$. Solving: Case 1: $8k - 73 = 50 \Rightarrow 8k = 123 \Rightarrow k = 123/8$. Case
1016 2: $8k - 73 = -50 \Rightarrow 8k = 23 \Rightarrow k = 23/8$. Therefore, possible k are $123/8$ and $23/8$. Thus,
1017 the sum is $123/8 + 23/8 = (123 + 23)/8 = 146/8 = 73/4$. So $m = 73$, $n = 4$, which are
1018 coprime. Thus, $m + n = 73 + 4 = 77$. **Final Answer** 77 (Correct Answer)1019
1020
1021
1022
1023
1024
1025

Figure 15: Case Analysis of CoTs with Different Lengths in AIME25

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

longer and shorter response in AIME25

Longer Response: Okay, so I have this problem here where I need to find all real numbers k such that the system of equations:1. $|25 + 20i - z| = 5$;2. $|z - 4 - k| = |z - 3i - k|$ has exactly one complex solution z . Then, I need to find the sum of all possible values of k and express it as $\frac{m}{n}$ where m and n are coprime, and then find $m + n$.Alright, let's break this down step by step.First, let me understand what each equation represents geometrically.Starting with the first equation: $|25 + 20i - z| = 5$. Since z is a complex number, let's denote $z = x + yi$ where x and y are real numbers.So, substituting $z = x + yi$ into the first equation: $|25 + 20i - (x + yi)| = 5$.Simplify the expression inside the absolute value: $|(25 - x) + (20 - y)i| = 5$.The modulus of a complex number $a + bi$ is $\sqrt{a^2 + b^2}$, so: $\sqrt{(25 - x)^2 + (20 - y)^2} = 5$.Squaring both sides: $(25 - x)^2 + (20 - y)^2 = 25$.This is the equation of a circle in the complex plane with center at $(25, 20)$ and radius 5.So, the first equation represents a circle centered at $(25, 20)$ with radius 5.Now, moving on to the second equation: $|z - 4 - k| = |z - 3i - k|$.Again, substituting $z = x + yi$:Left side: $|(x - 4 - k) + (y - k)i|$.Right side: $|(x + yi) - (3i + k)| = |(x - k) + (y - 3)i|$.Therefore, the equation becomes: $|(x - 4 - k) + yi| = |(x - k) + (y - 3)i|$.Which is equivalent to: $\sqrt{(x - 4 - k)^2 + y^2} = \sqrt{(x - k)^2 + (y - 3)^2}$.Squaring both sides to remove the square roots: $(x - 4 - k)^2 + y^2 = (x - k)^2 + (y - 3)^2$.Let's expand both sides:Left side: $(x - (4 + k))^2 + y^2 = (x - (4 + k))^2 + y^2$.Which expands to: $x^2 - 2x(4 + k) + (4 + k)^2 + y^2$.Right side: $(x - k)^2 + (y - 3)^2$.Which expands to: $x^2 - 2kx + k^2 + y^2 - 6y + 9$.Now, set left side equal to right side: $x^2 - 2x(4 + k) + (4 + k)^2 + y^2 = x^2 - 2kx + k^2 + y^2 - 6y + 9$.Simplify both sides:First, notice that x^2 and y^2 appear on both sides, so they cancel out.Left side: $-2x(4 + k) + (4 + k)^2$.Right side: $-2kx + k^2 - 6y + 9$.So, bring all terms to one side: $-2x(4 + k) + (4 + k)^2 - (-2kx) - k^2 + 6y - 9 = 0$.Simplify term by term:First, expand $-2x(4 + k) + 2kx$:Which is $-8x - 2kx + 2kx = -8x$. Wait, perhaps let's do it step by step:After cancelling x^2 and y^2 , we have:Left: $-2x(4 + k) + (4 + k)^2$.Right: $-2kx + k^2 - 6y + 9$.So, subtract the right side from both sides: $-2x(4 + k) + (4 + k)^2 - (-2kx) - k^2 + 6y - 9 = 0$.Simplify each term:First term: $-2x(4 + k) + 2kx$.Second term: $(4 + k)^2 - k^2$.Third term: $6y - 9$.Compute each part:1. $-2x(4 + k) + 2kx = (-8x - 2kx) + 2kx = -8x$.2. $(4 + k)^2 - k^2 = (16 + 8k + k^2) - k^2 = 16 + 8k$.3. $6y - 9 = 9$.**(Error 1)Putting it all together:** $-8x + 16 + 8k + 6y - 9 = 0$.**Simplify constants:** $16 - 9 = 7$, so: $-8x + 8k + 7 + 6y = 0$.**Let's rewrite:** $-8x + 6y + (8k + 7) = 0$,We can divide the entire equation by -1 to make it look cleaner: $8x - 6y - (8k + 7) = 0$.**Alternatively, keep it as:** $-8x + 6y + 8k + 7 = 0$.This is the equation representing the set of points $z = x + yi$ such that $|z - 4 - k| = |z - 3i - k|$.This equation is actually the perpendicular bisector of the segment joining the points $4 + k$ and $3i + k$ in the complex plane.Wait, let's think in terms of coordinates.**(Error 2)In the complex plane, z is a point (x, y) . The equation $|z - (4 + k)| = |z - (3i + k)|$ can be rewritten as $|(x + yi) - (4 + k)| = |(x + yi) - (0 + (3 + k)i)|$. So, this is the set of points equidistant from $(4, k)$ and $(0, 3 + k)$.** Therefore, the perpendicular bisector of the segment connecting $(4, k)$ and $(0, 3 + k)$.Therefore, the second equation represents the perpendicular bisector of the segment between $(4, k)$ and $(0, 3 + k)$.So, now we have two conditions:1. z lies on the circle centered at $(25, 20)$ with radius 5;2. z lies on the perpendicular bisector of the segment between $(4, k)$ and $(0, 3 + k)$.We are told that the system has exactly one solution. That means that the line (perpendicular bisector) intersects the circle at exactly one point. Therefore, the line must be tangent to the circle.Therefore, the distance from the center of the circle to the line must be equal to the radius of the circle.So, perhaps the steps are:1. Find the equation of the perpendicular bisector of the segment between $(4, k)$ and $(0, 3 + k)$.2. Find the distance from the center of the circle $(25, 20)$ to this line.3. Set this distance equal to the radius 5, and solve for k .4. Sum all possible values of k .So, let's proceed step by step.First, let's find the equation of the perpendicular bisector.Given two points $A(4, k)$ and $B(0, 3 + k)$.The midpoint M of segment AB is: $M_x = \frac{4+0}{2} = 2$, $M_y = \frac{k+(3+k)}{2} = \frac{3+2k}{2}$ So, midpoint is $(2, (3 + 2k)/2)$.Next, the slope of AB is: $m_{AB} = \frac{(3+k)-k}{0-4} = \frac{3}{-4} = -3/4$.Therefore, the slope of the perpendicular bisector is the negative reciprocal, which is $4/3$.Therefore, the equation of the perpendicular bisector is: $y - M_y = (4/3)(x - M_x)$.Plugging in $M_x = 2$ and $M_y = (3 + 2k)/2$: $y - (3 + 2k)/2 = (4/3)(x - 2)$.

Figure 16: Case Analysis of CoTs with Different Lengths in AIME25

1080

1081

longer and shorter response in AIME25

1082

1083

Let's write this in standard form $ax + by + c = 0$. Multiply both sides by 3 to eliminate the fraction: $3(y - (3 + 2k)/2) = 4(x - 2)$. Simplify: $3y - (9 + 6k)/2 = 4x - 8$. Multiply all terms by 2 to eliminate denominator: $6y - 9 - 6k = 8x - 16$. Bring all terms to left side: $-8x + 6y - 9 - 6k + 16 = 0$. Simplify constants: $-9 + 16 = 7$. Thus: $-8x + 6y + 7 - 6k = 0$. Or, $-8x + 6y + (7 - 6k) = 0$. We can write this as: $8x - 6y + (6k - 7) = 0$. Wait, let me check. From above: After multiplying by 2: $6y - 9 - 6k = 8x - 16$. Then, moving all terms to left: $-8x + 6y + (-9 - 6k + 16) = 0$. Which is: $n - 8x + 6y + (7 - 6k) = 0$. So, if we want to write it as $ax + by + c = 0$, (Error 1) then: $-8x + 6y + (7 - 6k) = 0$. **Alternatively, multiply both sides by -1:** $8x - 6y - 7 + 6k = 0$, Which is: $8x - 6y + 6k - 7 = 0$. Thus, the equation is $8x - 6y + (6k - 7) = 0$. **Alternatively, let's verify.** Alternatively, perhaps it's better to keep the equation as: From above: $-8x + 6y + 7 - 6k = 0$. Which can be written as: $8x - 6y = 7 - 6k$. But perhaps let's consider the standard form as: $8x - 6y + (-7 + 6k) = 0$. Either way, perhaps it's better to find the distance from the center (25, 20) to this line. The distance from a point (x_0, y_0) to the line $ax + by + c = 0$ is $|ax_0 + by_0 + c|/\sqrt{a^2 + b^2}$. So, first, let's write the equation in standard form. From above, the equation is: $-8x + 6y + 7 - 6k = 0$. We can write it as: $-8x + 6y + (7 - 6k) = 0$. Thus, coefficients are $a = -8, b = 6, c = 7 - 6k$. Therefore, distance from (25, 20) is: $|-8*25 + 6*20 + (7 - 6k)|/\sqrt{((-8)^2 + 6^2)}$. Compute numerator: First compute each term: $-8 * 25 = -200, 6 * 20 = 120$. So, total so far: $-200 + 120 = -80$. Add $(7 - 6k)$: $-80 + 7 - 6k = -73 - 6k$. So numerator is $|-73 - 6k|$. Denominator is $\sqrt{64 + 36} = \sqrt{100} = 10$. Since the distance must be equal to radius 5, we have: $|-73 - 6k|/10 = 5$. Therefore, $|-73 - 6k| = 50$. Which implies: $-73 - 6k = 50$ or $-73 - 6k = -50$. Case 1: $-73 - 6k = 50, -6k = 50 + 73 = 123, k = -123/6 = -41/2$. Case 2: $-73 - 6k = -50, -6k = -50 + 73 = 23, k = -23/6$. Therefore, the possible values of k are $-41/2$ and $-23/6$. Therefore, the sum of all possible values is: $-41/2 + -23/6 =$ convert to common denominator: $-41/2 = -123/6$, so total is $-123/6 - 23/6 = (-146)/6 = -73/3$. (Error 3) **But wait, the question says "the sum of all possible values of k " so perhaps we need to check if both solutions are valid.** Wait, perhaps let's see if these k 's result in exactly one solution. Since each k gives a different line, each line can intersect the circle at two points, one point, or none. Since we have exactly one solution, it's when the line is tangent to the circle. Therefore, perhaps both k 's are valid, but perhaps we need to check if both lead to lines that are tangent to the circle. Alternatively, perhaps only one of them is valid because for some k , the line might not intersect the circle or something. But perhaps both are valid. Therefore, perhaps sum is $(-41/2) + (-23/6) =$ as above, which is $(-123/6 - 23/6) = (-146)/6 = (-73)/3$. But since the problem says "the sum of all possible values of k ", which could be negative or positive, **but perhaps we need to check if both are valid.** Alternatively, perhaps the sum is $(-41/2) + (-23/6) = (-123 - 23)/6 = (-146)/6 = -73/3$. **But since the problem says "sum of all possible values of k ", perhaps we can write it as a fraction.** **But perhaps let's verify.** First, let's check for $k = -41/2$. Compute the distance from center to the line: $|-73 - 6 * (-41/2)|/10$. Compute: $-73 - 6 * (-41/2) = -73 + (6 * 41)/2 = -73 + (246)/2 = -73 + 123 = 50$. Thus, $|50|/10 = 5$, which is the radius. So this is correct. Similarly, for $k = -23/6$: Compute $|-73 - 6 * (-23/6)| = |-73 + 23| = |-50| = 50$. Thus, same result, distance is 5. Therefore, both k 's are valid. Hence, the sum is $(-41/2) + (-23/6) = (-123/6 - 23/6) = (-146)/6 = (-73)/3$. Thus, $m = -73, n = 3$. Wait, but m and n must be positive integers. So perhaps take absolute values? **Wait, perhaps the problem expects the sum as positive. But perhaps not, since k can be negative. The problem says "sum of all possible values of k ", so perhaps we just add them as they are.** Thus, $(-41/2) + (-23/6) = (-41 * 3 - 23)/6 = (-123 - 23)/6 = (-146)/6 = (-73)/3$. So, the sum is $-73/3$, which can be written as $-73/3$. Since m and n are coprime, $m = 73, n = 3$, **but since the sum is negative, perhaps the question expects the absolute value? But no, because the problem says "the sum of all possible values of k can be written as m/n ".** Therefore, the sum is $-73/3$, so $m = 73, n = 3$, which are coprime. Thus, $m + n = 73 + 3 = 76$. **Final Answer** 76 (Wrong Answer)

1129

1130

(Error: 1. Repeated reformulation of the equation form. 2. Coordinate extraction error, renders all following calculations invalid. 3. Ineffective repeated verification of the answer)

1131

1132

1133

Figure 17: Case Analysis of CoTs with Different Lengths in AIME25

1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145

1146 longer and shorter response in AMC

1147
1148 **Problem:** How many positive perfect squares less than 2023 are divisible by 5?
1149 **Shorter Response:** To determine how many positive perfect squares less than 2023 are divisible
1150 by 5, we start by noting that a perfect square is divisible by 5 if and only if the integer itself
1151 is divisible by 5. This is because if n^2 is divisible by 5, then n must be divisible by 5 (since 5
1152 is a prime number). Let $n = 5k$ where k is an integer. Then, $n^2 = (5k)^2 = 25k^2$. We need
1153 $25k^2 < 2023$. Solving for k , we get: $25k^2 < 2023 \implies k^2 < \frac{2023}{25} \implies k^2 < 80.92$. The
1154 largest integer k that satisfies this inequality is 8, because $8^2 = 64$ and $9^2 = 81$ which is greater
1155 than 80.92. Therefore, k can take any integer value from 1 to 8. This gives us 8 possible values
1156 for k . Thus, there are 8 positive perfect squares less than 2023 that are divisible by 5. The answer
1157 is: 8 (Correct Answer)1158 **Longer Response:** To determine how many positive perfect squares less than 2023 are divisible
1159 by 5, we start by noting that a perfect square divisible by 5 must itself be a multiple of 25 (since
1160 the square of any integer that is not a multiple of 5 cannot be divisible by 5, and the square
1161 of an integer that is a multiple of 5 must be a multiple of $5^2 = 25$). We need to find the largest
1162 integer n such that $25n^2 < 2023$. This is equivalent to finding the largest integer n such that
1163 $n^2 < \frac{2023}{25} = 80.92$. Since n must be an integer, we take the greatest integer less than $\sqrt{80.92}$,
1164 which is 8 (since $\sqrt{80.92} \approx 8.996$). Thus, n can take any integer value from 1 to 8. (Error) We
1165 can verify this by calculating the squares of these integers and checking if they are less
1166 than 2023: $1^2 = 1, 2^2 = 4, 3^2 = 9, 4^2 = 16, 5^2 = 25, 6^2 = 36, 7^2 = 49, 8^2 = 64, 9^2 =$
1167 $81, 10^2 = 100, 11^2 = 121, 12^2 = 144, 13^2 = 169, 14^2 = 196, 15^2 = 225, 16^2 = 256, 17^2 =$
1168 $289, 18^2 = 324, 19^2 = 361, 20^2 = 400, 21^2 = 441, 22^2 = 484, 23^2 = 529, 24^2 = 576, 25^2 =$
1169 $625, 26^2 = 676, 27^2 = 729, 28^2 = 784, 29^2 = 841, 30^2 = 900, 31^2 = 961, 32^2 = 1024, 33^2 =$
1170 $1089, 34^2 = 1156, 35^2 = 1225, 36^2 = 1296, 37^2 = 1369, 38^2 = 1444, 39^2 = 1521, 40^2 =$
1171 $1600, 41^2 = 1681, 42^2 = 1764, 43^2 = 1849, 44^2 = 1936, 45^2 = 2025$. Since $45^2 = 2025$ is not
1172 less than 2023, the largest integer n is 44. Therefore, the positive perfect squares less than 2023
1173 that are divisible by 5 are the squares of the integers from 1 to 44, which are 44 in total. Thus, the
1174 number of positive perfect squares less than 2023 that are divisible by 5 is: 44. (Wrong Answer)
1175 **Error:** Overlooking the “divisible by 5” constraint during verification.)

Figure 18: Case Analysis of CoTs with Different Lengths in AMC

1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187

1188 F RE² EXAMPLES
11891190 We present examples of RE² in Figures 19, 20, 21, and 22.
11911192 Re² Examples 1
11931194 **Problem:** How many ordered pairs of positive real numbers (a, b) satisfy the
1195 equation $(1 + 2a)(2 + 2b)(2a + b) = 32ab$?1196 **Reasoning Trace of Re²:** To solve the equation $(1 + 2a)(2 + 2b)(2a + b) = 32ab$ for
1197 ordered pairs (a, b) of positive real numbers, we start by simplifying and analyzing the equa-
1198 tion. First, let's rewrite the equation for clarity: $(1 + 2a)(2 + 2b)(2a + b) = 32ab$. We can factor
1199 out the common terms in the first two factors: $(1 + 2a)2(1 + b)(2a + b) = 32ab$. This
1200 simplifies to: $2(1 + 2a)(1 + b)(2a + b) = 32ab$. Dividing both sides by 2, we
1201 get: $(1 + 2a)(1 + b)(2a + b) = 16ab$. Next, we will test some simple values for a and b
1202 to see if we can find any solutions. Let's start with $a = b$. Substituting $a = b$ into the equation, we
1203 get: $(1 + 2a)(1 + a)(2a + a) = 16a^2$. This simplifies to: $(1 + 2a)(1 + a)(3a) = 16a^2$. Expanding
1204 the left side, we get: $3a(1 + 2a)(1 + a) = 3a(1 + 3a + 2a^2) = 3a + 9a^2 + 6a^3$. So the equation
1205 becomes: $3a + 9a^2 + 6a^3 = 16a^2$. Rearranging terms, we get: $6a^3 + 9a^2 + 3a - 16a^2 = 0$, which
1206 simplifies to: $6a^3 - 7a^2 + 3a = 0$. Factoring out a , we get: $a(6a^2 - 7a + 3) = 0$. Since a is a positive
1207 real number, $a \neq 0$. Therefore, we need to solve the quadratic equation: $6a^2 - 7a + 3 = 0$. The
1208 discriminant of this quadratic equation is: $(-7)^2 - 4 \cdot 6 \cdot 3 = 49 - 72 = -23$. Since the
1209 discriminant is negative, the quadratic equation has no real solutions.
1210Therefore, there are no positive real solutions for a when $a = b$. **It's better to redo the question.****Restarting from scratch:**1211 Next, let's try another approach by assuming $a = \frac{1}{2}$ and solving for b : $(1 + 2 \cdot \frac{1}{2})(2 + 2b)(2 \cdot \frac{1}{2} + b) =$
1212 $32 \cdot \frac{1}{2} \cdot b$, which simplifies to: $(1 + 1)(2 + 2b)(1 + b) = 16b$ or $2(2 + 2b)(1 + b) = 16b$. Simplifying
1213 further, we get: $2(2 + 2b + 2b + 2b^2) = 16b$ or $2(2 + 4b + 2b^2) = 16b$, which simplifies
1214 to: $4 + 8b + 4b^2 = 16b$ or $4b^2 - 8b + 4 = 0$. Dividing by 4, we get: $b^2 - 2b + 1 = 0$, which factors
1215 to: $(b - 1)^2 = 0$. So, $b = 1$. Therefore, $(a, b) = (\frac{1}{2}, 1)$ is a solution. To confirm that there are no
1216 other solutions, we can use the symmetry and the nature of the equation. Given the complexity
1217 and the fact that we have found a solution, we can conclude that there are no other positive
1218 real solutions. Thus, the number of ordered pairs (a, b) that satisfy the equation is: **1**. **(Correct**
1219 **Answer)**1221
1222 Figure 19: Re² Examples 1
12231224 G LIMITATIONS AND FUTURE WORK
12251226 In this section, we outline several limitations of our study that highlight areas for future research and
1227 improvement: **(1)** Although our method improves performance on challenging problems through
1228 better test-time scaling, models may still require multiple re-solving rounds to reach the correct
1229 answer. At present, there is no mechanism to explicitly control the probability of invoking the re-solve
1230 action during inference, which may affect efficiency. **(2)** Our experiments primarily focus on large
1231 language models, leaving the potential of applying Re² to other generative modalities (e.g., vision or
1232 multi-modal reasoning) unexplored. **(3)** We mainly study reasoning benchmarks, while extending
1233 Re² to more complex tasks, such as tool use, or search-intensive problems could also provide benefits.
12341235 H USE OF LLMs
12361237 LLMs were used solely for language polishing of the manuscript.
12381239
1240
1241

1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257

1258 **Re² Examples 2**

1259 **Problem:**Ralph has a record store where people can trade their own records for new ones. People
1260 can trade 2 old records for 1 new one. 5 people come in with old records and leave with 7 new
1261 records between them. How many old records did the 5 people bring in?
1262

1263 **Reasoning Trace of Re²:**To determine how many old records the 5 people brought to
1264 Ralph's record store, we need to understand the trade ratio and the total number of new records
1265 they received.The trade ratio is 2 old records for 1 new record. So, if each person trades 2 old
1266 records for 1 new record, and 5 people are trading, we need to find out how many old records
1267 were traded in total to get 7 new records.Let's denote the number of new records each person
1268 receives as n . Since each new record is obtained by trading 2 old records, if each of the 5 people
1269 receives n new records, the total number of new records is $5n$. We know that the total number of
1270 new records they leave with is 7, so we have: $5n = 7$.Solving for n : $n = \frac{7}{5}$.This means each
1271 person receives $\frac{7}{5}$ new records.
1272 However, since each person can only receive a whole number of new records, we need to consider
1273 the total number of new records and the trade ratio in another way. **It's better to redo the question.**
1274

1275 **Restarting from scratch:**
1276 If 5 people together leave with 7 new records, and each new record is obtained by trading 2 old
1277 records, the total number of old records traded is: 7 new records \times 2 old records per new record =
1278 14 old records .Therefore, the 5 people brought in a total of 14 old records. So, the number of old
1279 records the 5 people brought in is: **14** (Correct Answer)

Figure 20: Re² Examples 2

1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295

1296

1297

Re² Examples 3

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

1349

Problem: Let x, y and z be positive real numbers that satisfy the following system of equations: $\log_2\left(\frac{x}{yz}\right) = \frac{1}{2}, \log_2\left(\frac{y}{xz}\right) = \frac{1}{3}, \log_2\left(\frac{z}{xy}\right) = \frac{1}{4}$. Then the value of $|\log_2(x^4y^3z^2)|$ is $\frac{m}{n}$ where m and n are relatively prime positive integers. Find $m + n$.

Reasoning Trace of Re²: Let's start by expressing the given logarithmic equations in terms of x, y , and z . The given equations are: $\log_2\left(\frac{x}{yz}\right) = \frac{1}{2}, \log_2\left(\frac{y}{xz}\right) = \frac{1}{3}, \log_2\left(\frac{z}{xy}\right) = \frac{1}{4}$. We can rewrite these equations as: $\log_2(x) - \log_2(yz) = \frac{1}{2}, \log_2(y) - \log_2(xz) = \frac{1}{3}, \log_2(z) - \log_2(xy) = \frac{1}{4}$. Using the properties of logarithms, we can further simplify these equations: $\log_2(x) - (\log_2(y) + \log_2(z)) = \frac{1}{2}, \log_2(y) - (\log_2(x) + \log_2(z)) = \frac{1}{3}, \log_2(z) - (\log_2(x) + \log_2(y)) = \frac{1}{4}$. Let's denote: $\log_2(x) = a, \log_2(y) = b, \log_2(z) = c$. So the equations become: $a - (b + c) = \frac{1}{2}, b - (a + c) = \frac{1}{3}, c - (a + b) = \frac{1}{4}$. Simplifying each equation, we get: $a - b - c = \frac{1}{2}$ (Equation 1), $b - a - c = \frac{1}{3}$ (Equation 2), $c - a - b = \frac{1}{4}$ (Equation 3). We now have a system of linear equations: $a - b - c = \frac{1}{2}$ (Equation 1), $-b + a - c = \frac{1}{3}$ (Equation 2), $-c - a + b = \frac{1}{4}$ (Equation 3). Let's add all three equations together: $(a - b - c) + (-b + a - c) + (-a - b + c) = \frac{1}{2} + \frac{1}{3} + \frac{1}{4}$. Simplifying the left side: $a - b - c - b + a - c - a - b + c = 2a - 3b - 2c = \frac{1}{2} + \frac{1}{3} + \frac{1}{4}$. The right side can be simplified as: $\frac{1}{2} + \frac{1}{3} + \frac{1}{4} = \frac{6}{12} + \frac{4}{12} + \frac{3}{12} = \frac{13}{12}$. So we have: $2a - 3b - 2c = \frac{13}{12}$. However, let's instead solve for a, b , and c by subtracting pairs of equations to find individual expressions. Subtract Equation 2 from Equation 1: $(a - b - c) - (-b + a - c) = \frac{1}{2} - \frac{1}{3}, a - b - c + b - a + c = \frac{1}{2} - \frac{1}{3}, 0 = \frac{1}{2} - \frac{1}{3} = \frac{3}{6} - \frac{2}{6} = \frac{1}{6}$. This simplifies to: $0 = \frac{1}{6}$. This is a contradiction, which indicates we need to recheck our method of solving. Instead, let's solve for a, b , and c by another method. **It's better to redo the question.**

Restarting from scratch:

Let's add Equation 1, Equation 2, and Equation 3 and then solve for a, b , and c individually. From the system: $a - b - c = \frac{1}{2}$ (Equation 1), $b - a - c = \frac{1}{3}$ (Equation 2), $c - a - b = \frac{1}{4}$ (Equation 3). We can add these equations: $(a - b - c) + (b - a - c) + (c - a - b) = \frac{1}{2} + \frac{1}{3} + \frac{1}{4}, a - b - c + b - a - c + c - a - b = \frac{13}{12}, -a - c - b = \frac{13}{12}, -(a + b + c) = \frac{13}{12}, a + b + c = -\frac{13}{12}$. This is incorrect, so let's re-evaluate by solving the system of linear equations. Instead, let's solve for a, b , and c directly. **It's better to redo the question.**

Restarting from scratch:

We can use a different approach by expressing a, b , and c in terms of a common variable. Let's add and subtract the equations in a different way. From: $a - b - c = \frac{1}{2}, b - a - c = \frac{1}{3}, c - a - b = \frac{1}{4}$. Let's solve these equations by eliminating variables step-by-step. We can start by expressing c in terms of a and b from the first equation: $c = a - b - \frac{1}{2}$. Substitute c into the second equation: $b - a - (a - b - \frac{1}{2}) = \frac{1}{3}, b - a - a + b + \frac{1}{2} = \frac{1}{3}, 2b - 2a + \frac{1}{2} = \frac{1}{3}, 2(b - a) = \frac{1}{3} - \frac{1}{2} = \frac{2}{6} - \frac{3}{6} = -\frac{1}{6}, 2(b - a) = -\frac{1}{6}, b - a = -\frac{1}{12}, b = a - \frac{1}{12}$. Now substitute $b = a - \frac{1}{12}$ and $c = a - b - \frac{1}{2}$ into the third equation: $c = a - (a - \frac{1}{12}) - \frac{1}{2} = a - a + \frac{1}{12} - \frac{1}{2} = \frac{1}{12} - \frac{1}{2} = \frac{1}{12} - \frac{6}{12} = -\frac{5}{12}$. Substitute $b = a - \frac{1}{12}$ and $c = -\frac{5}{12}$ into the third equation: $-\frac{5}{12} - a - (a - \frac{1}{12}) = \frac{1}{4}, -\frac{5}{12} - a - a + \frac{1}{12} = \frac{1}{4}, -\frac{5}{12} + \frac{1}{12} - 2a = \frac{1}{4}, -\frac{4}{12} - 2a = \frac{1}{4}, -\frac{1}{3} - 2a = \frac{1}{4}, -2a = \frac{1}{4} + \frac{1}{3} = \frac{3}{12} + \frac{4}{12} = \frac{7}{12}, -2a = \frac{7}{12}, a = -\frac{7}{24}$. Substitute $a = -\frac{7}{24}$ back to find b and c : $b = a - \frac{1}{12} = -\frac{7}{24} - \frac{1}{12} = -\frac{7}{24} - \frac{2}{24} = -\frac{9}{24} = -\frac{3}{8}, c = -\frac{5}{12}$. Now we have: $a = -\frac{7}{24}, b = -\frac{3}{8}, c = -\frac{5}{12}$. We need to find $(|\log_2(x^4y^3z^2)|)$. First, we express $x^4y^3z^2$ in terms of a, b , and c : $\log_2(x^4y^3z^2) = 4a + 3b + 2c$. Substituting a, b , and c : $4a + 3b + 2c = 4(-\frac{7}{24}) + 3(-\frac{3}{8}) + 2(-\frac{5}{12}) = -\frac{28}{24} - \frac{9}{8} - \frac{10}{12} = -\frac{28}{24} - \frac{27}{24} - \frac{20}{24} = -\frac{28+27+20}{24} = -\frac{75}{24} = -\frac{25}{8}$. Thus, $|\log_2(x^4y^3z^2)| = |-\frac{25}{8}| = \frac{25}{8}$. So, $|\log_2(x^4y^3z^2)| = \frac{25}{8}$. The fraction $\frac{25}{8}$ is in simplest form, so $m = 25$ and $n = 8$. Therefore, $m + n = 25 + 8 = 33$. The final answer is: **33 (Correct Answer)**

Figure 21: Re² Examples 3

1350
1351
1352
1353
13541355 Re² Examples 41356
1357 **Problem:** Let f be the unique function defined on the positive integers such that $\sum_{d|n} d \cdot f\left(\frac{n}{d}\right) = 1$
1358 for all positive integers n . What is $f(2023)$?1359
1360 **Reasoning Trace of Re²:** To determine the function f defined on the positive integers
1361 such that $\sum_{d|n} d \cdot f\left(\frac{n}{d}\right) = 1$ for all positive integers n , we start by analyzing the functional
1362 equation. We will use the properties of the function and the structure of the equation to find $f(n)$
1363 .Step-by-Step Analysis1364 1. Base Case: $n = 1$, $\sum_{d|1} d \cdot f\left(\frac{1}{d}\right) = 1$. The only divisor of 1 is 1 itself, so the sum reduces to:
1365 $1 \cdot f(1) = 1 \implies f(1) = 1$.1366 2. General Case: $n = p$ where p is a prime number, $\sum_{d|p} d \cdot f\left(\frac{p}{d}\right) = 1$. The divisors of p are 1
1367 and p , so the sum becomes: $1 \cdot f(p) + p \cdot f(1) = 1 \implies f(p) + p \cdot 1 = 1 \implies f(p) + p =$
1368 $1 \implies f(p) = 1 - p$.1369 3. General Case: $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ (where p_i are distinct primes), $\sum_{d|n} d \cdot f\left(\frac{n}{d}\right) = 1$. For
1370 $n = p_1^{e_1}$, we have: $\sum_{d|p_1^{e_1}} d \cdot f\left(\frac{p_1^{e_1}}{d}\right) = 1$. The divisors of $p_1^{e_1}$ are $(1, p_1, p_1^2, \dots, p_1^{e_1})$. So,
1371 the sum becomes: $\sum_{j=0}^{e_1} p_1^j \cdot f(p_1^{e_1-j}) = 1$. Substituting $f(p_1^m) = 1 - p_1^m$ for $m \geq 1$ and
1372 $f(1) = 1$, we get: $p_1^0 \cdot f(p_1^{e_1}) + p_1^1 \cdot f(p_1^{e_1-1}) + p_1^2 \cdot f(p_1^{e_1-2}) + \cdots + p_1^{e_1} \cdot f(1) = 1 \implies$
1373 $(1 - p_1^{e_1}) + p_1 \cdot (1 - p_1^{e_1-1}) + p_1^2 \cdot (1 - p_1^{e_1-2}) + \cdots + p_1^{e_1} \cdot 1 = 1 \implies (1 - p_1^{e_1}) + p_1(1 -$
1374 $p_1^{e_1-1}) + p_1^2(1 - p_1^{e_1-2}) + \cdots + p_1^{e_1} = 1 \implies (1 - p_1^{e_1}) + p_1 - p_1^{e_1} + p_1^2 - p_1^{e_1} + \cdots + p_1^{e_1} =$
1375 $1 \cdot (1 + p_1 + p_1^2 + \cdots + p_1^{e_1}) - (p_1^{e_1} + p_1^{e_1-1} + \cdots + p_1^0) = 1 \implies (1 + p_1 + p_1^2 + \cdots + p_1^{e_1}) - p_1^{e_1} \cdot e_1 = 1$.
1376 The sum $1 + p_1 + p_1^2 + \cdots + p_1^{e_1}$ is a geometric series: $(p_1^{e_1+1} - 1)/(p_1 - 1)$. So we
1377 get: $(p_1^{e_1+1} - 1)/(p_1 - 1) - p_1^{e_1} \cdot e_1 = 1$. Simplifying this, we need to check if this holds for
1378 general n . It's better to redo the question.1379 **Restarting from scratch:**1380 Let's hypothesize that $f(n) = \frac{1}{n}$. We substitute $f(n) = \frac{1}{n}$ into the original equation:1381 $\sum_{d|n} d \cdot f\left(\frac{n}{d}\right) = \sum_{d|n} d \cdot \frac{1}{\frac{n}{d}} = \sum_{d|n} d \cdot \frac{d}{n} = \sum_{d|n} \frac{d^2}{n} = \frac{1}{n} \sum_{d|n} d^2$. For this to equal 1, we
1382 need: $\frac{1}{n} \sum_{d|n} d^2 = 1 \implies \sum_{d|n} d^2 = n$. This is not generally true for all n , so we need to
1383 re-evaluate our hypothesis. It's better to redo the question.1384 **Restarting from scratch:**

1385 Instead, let's directly use the functional equation and the values we have derived.

1386 Determining $f(2023)$. First, factorize $2023 = 7 \times 17^2$. Using the functional equation and the
1387 derived values, we need to find $f(2023)$. We can use the multiplicative property of f if we assume
1388 f is multiplicative. That means if $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$, then $f(n) = f(p_1^{e_1}) f(p_2^{e_2}) \cdots f(p_k^{e_k})$. We
1389 already have: $f(p) = 1 - p$ for prime p . So: $f(7) = 1 - 7 = -6$, $f(17) = 1 - 17 = -16$. Since
1390 $2023 = 7 \times 17^2$, we have: $f(2023) = f(7) \cdot f(17^2)$. We need to determine
1391 $f(17^2)$: $\sum_{d|17^2} d \cdot f\left(\frac{17^2}{d}\right) = 1$. The divisors of 17^2 are $(1, 17, 17^2)$. So the sum
1392 is: $1 \cdot f(17^2) + 17 \cdot f(17) + 17^2 \cdot f(1) = 1$. Substituting $f(17^2) = 1 - 17^2$, $f(17) = -16$,
1393 and $f(1) = 1$: $f(17^2) + 17 \cdot (-16) + 17^2 \cdot 1 = 1$, $f(17^2) - 272 + 289 = 1$, $f(17^2) + 17 =$
1394 1 , $f(17^2) = 1 - 17 = -16$. So: $f(2023) = f(7) \cdot f(17^2) = -6 \cdot -16 = 96$. Thus, the value of
1395 $f(2023)$ is: 96. (Correct Answer)1396
13971398
1399
1400
1401
1402
1403Figure 22: Re² Examples 4