

# ENHANCING LANGUAGE MODEL REASONING WITH STRUCTURED MULTI-LEVEL MODELING

Anonymous authors

Paper under double-blind review

## ABSTRACT

Inference-time scaling enhances a model’s reasoning by extending its chain-of-thought (CoT). However, existing approaches typically rely on a single policy trained with outcome-reward reinforcement learning (RL), which often suffers from long-horizon plan failures, i.e., the implicit plan drifts away from any valid strategy. This problem is particularly severe for smaller language models (LMs) with long CoTs due to their limited capacity. To address this, we propose Multi-Level Reasoning (MLR), which reformulates long-CoT generation as a two-level stochastic process. Specifically, MLR employs two policies: a high-level planner that generates step descriptors (abstract subgoals) and a low-level executor that produces detailed content conditioned on these descriptors. The planner then generates the next subgoal based on the summarized current step, forming an alternating plan–execute loop. To maintain scalability, we adopt a minimal design, where the base model serves as the low-level policy and a lightweight LoRA module implements the high-level policy. For training, we observe that outcome-reward RL is inefficient and weakly informative for long trajectories (e.g., those exceeding 4K tokens). To overcome this, we introduce online Step-DPO, a process-level preference optimization scheme that leverages Twisted Sequential Monte Carlo (TSMC) to provide scalable stepwise supervision. This yields more effective training, improved stability, and higher accuracy. Extensive experiments on challenging math, science, and logical reasoning benchmarks show that, with only 10% SFT data and 5% of preference data, MLR outperforms both the DeepSeek-R1 distillation and the outcome-reward RL baselines across multiple base models and tasks. More importantly, MLR exhibits slower performance degradation on long-horizon reasoning, demonstrating stronger robustness under extended CoT generation.

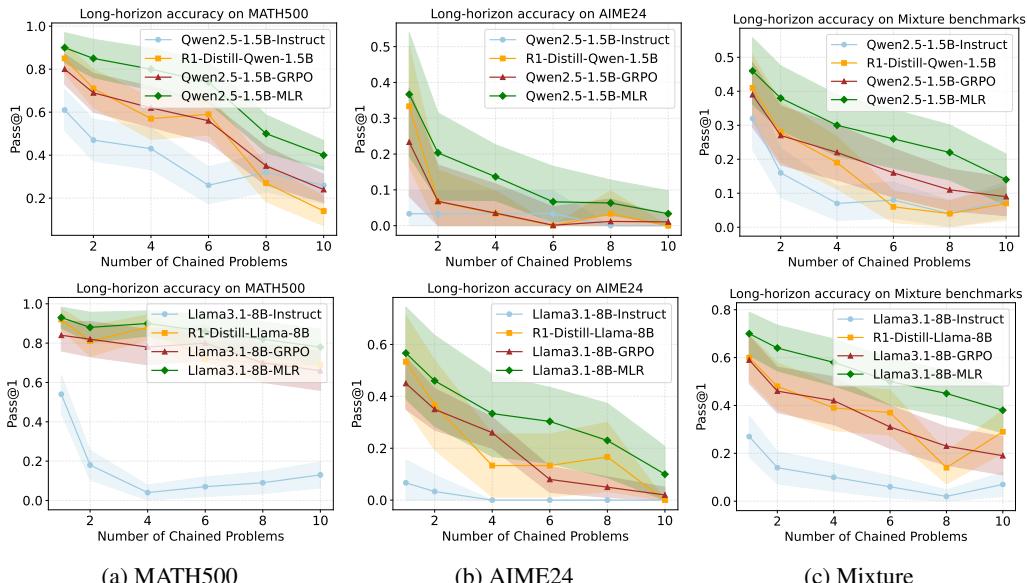


Figure 1: Long-horizon reasoning accuracy on (a) MATH500, (b) AIME24, and (c) Mixture benchmarks (MATH500, AIME24, GPQA, BoardGameQA). We simulate long-horizon reasoning by concatenating multiple problems in the prompt and report average accuracy. MLR consistently degrades more slowly than all baselines. Additional experimental details and statistics are provided in Section D.

054 **1 INTRODUCTION**

055

056 OpenAI’s o1 series (OpenAI, 2024) introduce inference-time scaling by increasing the length of  
 057 the Chain-of-Thought (CoT) (Wei et al., 2022) reasoning process. This design yields significant  
 058 improvements in complex reasoning tasks compared to non-reasoning models, marking a major step  
 059 forward in language model (LM) capabilities. Building on this idea, DeepSeek (Guo et al., 2025)  
 060 proposes a large-scale reinforcement learning (RL) pipeline that directly incentivizes the generation  
 061 of long CoTs through policy optimization. Despite the effectiveness of these methods, approaches  
 062 that rely on single-policy long CoTs for RL face important limitations, including long-horizon plan  
 063 failures and the inherent challenges of RL with sparse outcome rewards<sup>1</sup>. These issues are especially  
 064 pronounced for smaller LMs with long CoTs due to their limited capacity.

065 In reinforcement learning, long-horizon policy learning remains a fundamental challenge due to  
 066 the difficult credit assignment (Kaelbling et al., 1996). To address this, hierarchical reinforcement  
 067 learning (HRL) (Dietterich, 2000) has been proposed, where high-level and low-level policies are  
 068 learned to operate at different temporal abstractions. While HRL proven effective in domains such as  
 069 robotics (Nachum et al., 2018; Gupta et al., 2019), its application to LMs<sup>2</sup> presents several challenges:  
 070 1) Scalability. Modeling multiple policies, especially when implemented as separate LMs, can incur  
 071 significant computational overhead. Naïve multi-agent setups will suffer from high communication  
 072 and synchronization costs, making HRL computationally expensive to scale (Guo et al., 2024b). 2)  
 073 Flexibility. The existing full-plan-then-execute structure (Huang et al., 2022; Xu et al., 2023) is brittle  
 074 in LM-based reasoning tasks where new information or execution failures may require mid-course  
 075 corrections. Thus, it is desirable to allow the high-level plan to evolve dynamically based on the  
 076 ongoing progress of the low-level execution process. To address these challenges, we propose a  
 077 multi-level reasoning (MLR) strategy, where the model alternates between generating a step-level  
 078 descriptor and its corresponding detailed content, to enable efficient multi-policy modeling and  
 079 dynamic plan adaptation.

080 More importantly, we introduce an online Step-DPO pipeline for long CoT training, which signifi-  
 081 cantly accelerates training and improves reasoning performance. Existing RL fine-tuning frameworks  
 082 struggle to obtain effective process-level supervision (Guo et al., 2025). First, evaluating the correct-  
 083 ness of intermediate steps is inherently difficult. Automated annotation using LLMs (Wang et al.,  
 084 2023b) often yields unreliable or noisy signals, while manual annotation (Lightman et al., 2023) is  
 085 prohibitively expensive at scale. Second, introducing a separate process reward model (PRM) adds  
 086 complexity. It is vulnerable to reward hacking (Gao et al., 2023), requires substantial training data,  
 087 and complicates the pipeline by necessitating repeated retraining. To overcome these limitations,  
 088 we repurpose Twisted Sequential Monte Carlo (TSMC) (Doucet et al., 2001; Del Moral et al., 2006;  
 089 Briers et al., 2010) as a process-level supervision signal for Step-DPO. In the LM-based reasoning  
 090 setting, the importance weight in TSMC estimates how much more likely a partial trajectory is to  
 091 lead to a correct outcome under the target distribution compared to the current policy. We then define  
 092 the process preference between two candidate continuations at the same step by comparing their  
 093 incremental log-weights. This formulation has two key advantages: 1) it converts the multiplicative  
 094 nature of sequential importance weights into an additive form, improving numerical stability; 2) it  
 095 aligns naturally with the pairwise preference structure of DPO training. Empirically, our approach  
 096 provides stable and informative step-level preferences, leading to more efficient training and stronger  
 097 performance on complex reasoning tasks.

098 We summarize our key contributions as follows:  
 099

100 • We propose a novel multi-level reasoning (MLR) framework that directly addresses the  
 101 limitations of single-policy long-CoT approaches, such as long-horizon plan failures and  
 102 inefficiency. MLR decomposes reasoning into alternating high-level step descriptors and  
 103 low-level detailed content, enabling structured abstraction, dynamic plan adjustment, and  
 104 more reliable long-horizon reasoning.

105 • We repurpose Twisted Sequential Monte Carlo (TSMC) to provide process-level preferences  
 106 for Step-DPO training. This eliminates the need for a separate process reward model,

107 <sup>1</sup>See Section 2 for a detailed discussion.

108 <sup>2</sup>Note that HRL differs from prompting-based approaches that decompose tasks in CoTs. Instead, it treats  
 109 high- and low-level actions as separate distributions with distinct objectives and temporal scopes.

108 reducing overhead while supplying stable and informative supervision throughout long  
 109 reasoning trajectories.

110

- 111 We perform extensive experiments on challenging benchmarks in math, science, and logical  
 112 reasoning. Results show that our approach consistently outperforms both distillation-based  
 113 long-CoT methods and RL methods that rely solely on outcome rewards.

114

## 2 INFERENCE-TIME SCALING VIA LONG CHAIN-OF-THOUGHT

115

116

**Formulation.** Consider a query  $q$ , reasoning models generate a CoT  $c$  before producing the final  
 117 response  $a$ , where  $q, c, a$  are all sequences of tokens, i.e.,  $c = (c[1], c[2], \dots, c[L])$ . To improve  
 118 model performance, these models extend the length of  $c$  by incorporating human-like reasoning  
 119 behaviors such as exploration, self-verification and reflection. The generation of long CoTs follows  
 120 the standard autoregressive modeling: the probability of each token  $c[l]$  depends only on its preceding  
 121 tokens  $(c[1 : l - 1])$ , which enables the factorization of the joint likelihood of the entire sequence as:  
 122

123

$$p_\theta(c[1 : L]) = \prod_{l=1}^L p_\theta(c[l] | c[1 : l - 1]). \quad (1)$$

124

125

Note that, for notational simplicity, we omit the conditioning on  $q$  in Eq. 1 and in the following  
 126 derivations. Training the model  $p_\theta$  involves maximizing the likelihood of each token conditioned on  
 127 its prefix, i.e., optimizing  $p_\theta(c[l] | c[1 : l - 1])$  over the training data.

128

129

**Post-training.** Guo et al. (2025) detail how they incentivize the long CoT generation from a base  
 130 model through large-scale RL without relying on SFT. Specifically, they employ GRPO guided by  
 131 rule-based outcome reward. For each query  $q$ , GRPO samples a group of outputs  $\{o_1, o_2, \dots, o_G\}$   
 132 from the old policy  $\pi_{\theta_{\text{old}}}$ , where each output is composed of a CoT followed by the final response, i.e.,  
 133  $o_i = [c_i, a_i]$ , and then optimizes the policy  $\pi_\theta$  by maximizing the corresponding objective.

134

135

**Discussion on the weakness of single-policy long CoT.** The above approach of using single-policy  
 136 long CoT enables inference-time scaling with LMs, but introduces several issues:

137

138

- 139 1) **Long-horizon plan failures.** In single-policy long CoT generation, the same policy is responsible  
 140 for both planning and execution. Without guidance or structure, errors can accumulate and cause  
 141 the implicit plan drifts away from any valid strategy (see examples in Section D).
- 142 2) **Long-horizon RL with sparse outcome reward.** Long CoTs involve thousands of token-level  
 143 actions before receiving a reward, which hinders effective credit assignment. As shown in  
 144 Figure 2, these trajectories can be extremely long, with errors occurring at widely varying  
 145 positions, which undermines the effectiveness of outcome-based fine-tuning. Moreover, Figure 3  
 146 shows that latency and memory usage grow rapidly with trajectory length, while outcome-based  
 147 supervision requires the entire trajectory to finish before feedback is provided. Consequently,  
 148 learning is slow and unstable, especially in the early stages when the model rarely produces  
 149 correct trajectories.

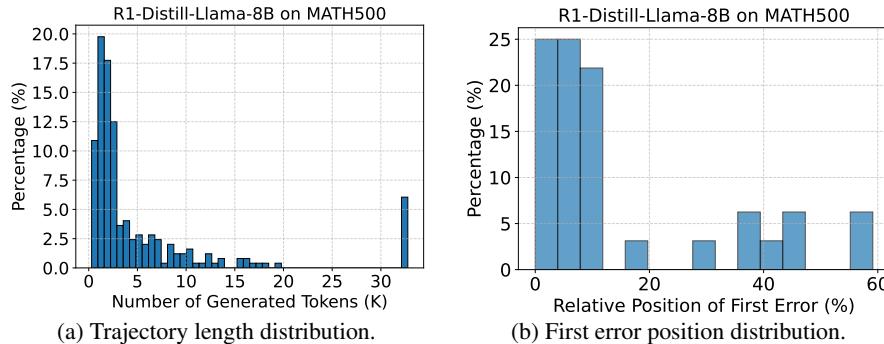


Figure 2: The Chain-of-Thought trajectories can be lengthy and the positions of the first error vary considerably, making outcome-based RL fine-tuning inefficient. The statistics in (b) are based on 100 trajectories with incorrect final answers, where the first error was manually identified.

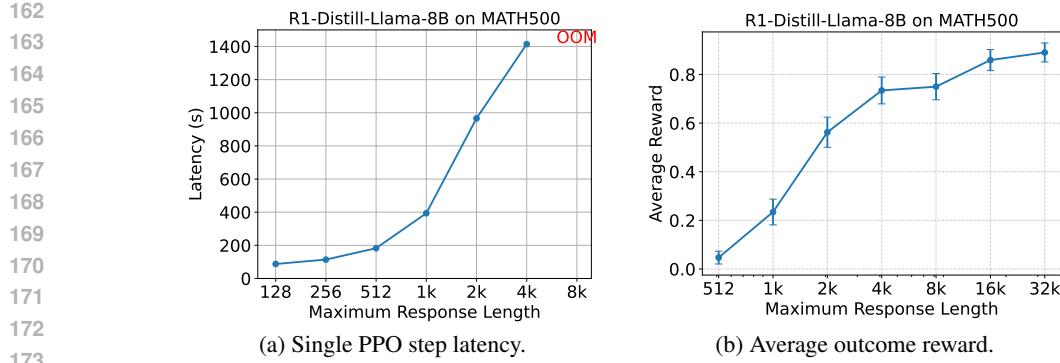


Figure 3: Training long trajectories with outcome rewards is highly inefficient. Both latency and memory usage increase rapidly with trajectory length, and outcome-based supervision requires the entire trajectory to complete before providing feedback. All measurements are obtained using Verl (Sheng et al., 2024) with vLLM (Kwon et al., 2023) on a single A100 node (batch size = 32).

### 3 METHODOLOGY

#### 3.1 ENHANCING INFERENCE-TIME SCALING WITH MULTI-LEVEL REASONING

**Reformulation as MLR.** We reconceptualize inference-time scaling by shifting from “single-policy long CoT” to “multi-level reasoning” strategy (Figure 4). Here, the reasoning process is organized hierarchically, capturing both high-level abstractions  $d$  and low-level details  $c$ . Formally, the overall reasoning chain is represented at two levels:  $d = (d^{(1)}, \dots, d^{(M)})$  and  $c = (c^{(1)}, \dots, c^{(M)})$ , where  $M$  denotes the number of reasoning steps,  $d^{(m)}$  is the descriptor of step  $m$ , and  $c^{(m)}$  represents the corresponding detailed content. The autoregressive likelihood can be factorized hierarchically as follows:

$$p_{\theta}^H(d) = \prod_{m=1}^M p_{\theta}^H(d^{(m)} \mid d^{(1:m-1)}, c^{(1:m-1)}), \quad p_{\theta}^L(c) = \prod_{m=1}^M p_{\theta}^L(c^{(m)} \mid d^{(1:m)}, c^{(1:m-1)}) \quad (2)$$

where  $c^{(m)}$  denotes a compressed representation of the detailed content  $c^{(m)}$ . We also experimented with removing the previous descriptors  $d^{(1:m-1)}$  from Equation (2), but found that including them improves performance and facilitates training. The inference procedure is summarized in Algorithm 1.

**Architecture.** Figure 5 illustrates the architecture used to implement our MLR strategy. The model alternates between a high-level policy that produces step descriptors and a low-level policy that generates the corresponding detailed content. The low-level policy is implemented with the base LM,

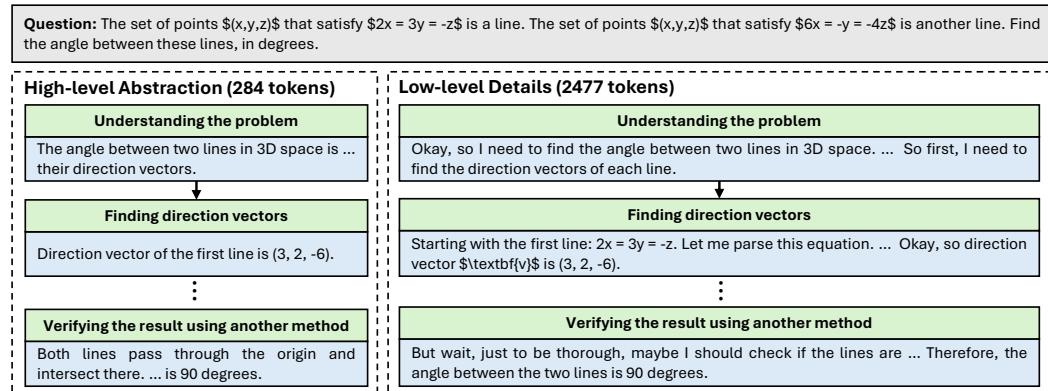


Figure 4: Illustration of MLR. MLR augments single-policy reasoning with an explicit high-level policy which provides intent and structural guidance that narrows the search space, improves credit assignment, and mitigates long-horizon planning failures.

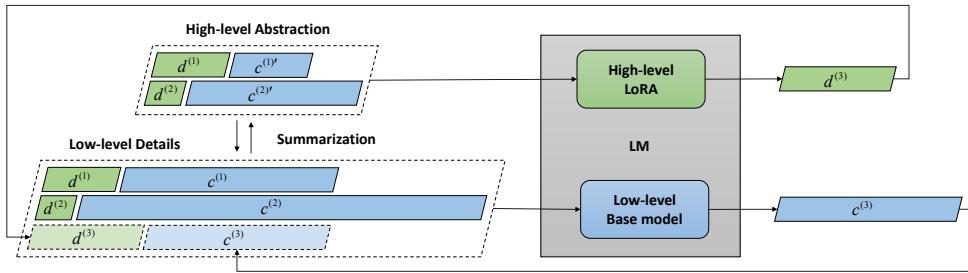
216  
217  
218  
219  
220  
221  
222  
223  
224  
225

Figure 5: Overview of the proposed architecture. The model alternates between generating high-level descriptors and corresponding low-level content in a structured manner. Additional insights and ablations motivating these design choices are provided in Section C.

which conditions on the sequence of descriptors, prior detailed contents, and the current descriptor to generate the detailed reasoning step. The high-level policy is implemented with a lightweight LoRA module (Hu et al., 2022), which conditions on previous descriptors and their summaries to produce the next descriptor. Since descriptors are much shorter than full reasoning content, this component remains compact and computationally efficient. The design rationale behind this architecture as well as ablation studies are provided in Section C. Additionally, we fine-tune an independent, lightweight LLM for summarization, which is shared across different base models.

### 3.2 ONLINE STEP-DPO WITH PROCESS-LEVEL PREFERENCES

To train our model effectively, we introduce an online Step-DPO pipeline that iteratively updates the policy through stepwise preference optimization.

**Supervised fine-tuning.** We collect long CoT examples from DeepSeek-R1 and use powerful non-reasoning models (GPT-4o (OpenAI, 2024), DeepSeek-V3 (Liu et al., 2024)) to decompose them into multiple reasoning steps, each annotated with a step-level descriptor via in-context learning. To construct high-level abstractions, we further compress the detailed content of each step into a concise summary using the same non-reasoning models. The resulting multi-level data consists of aligned step descriptors, detailed contents, and summaries. We then fine-tune the base model on the low-level detailed content using full SFT. Afterward, we freeze the base model and apply LoRA-based fine-tuning on the high-level abstraction data. We also fine-tune an independent, lightweight LLM on the summarization data. During the online Step-DPO procedure, the summarizer remains frozen. A detailed discussion of design choices, including training order, full vs. adapter finetuning, and ablations, is provided in Section C.

**Multi-level Step-DPO.** Following Lai et al. (2024), we optimize the MLR framework with the following stepwise preference loss:

$$\mathcal{L}_{\text{s-DPO}} := -\mathbb{E}_{(x^{(m)}, y_+^{(m)}, y_-^{(m)}) \sim \mathcal{D}_{\text{pref}}} \frac{1}{M} \sum_{m=1}^M \log \sigma \left[ \beta \left( \log \frac{p_\theta(y_+^{(m)} | x^{(m)})}{p_{\text{ref}}(y_+^{(m)} | x^{(m)})} - \log \frac{p_\theta(y_-^{(m)} | x^{(m)})}{p_{\text{ref}}(y_-^{(m)} | x^{(m)})} \right) \right]. \quad (3)$$

where  $(x^{(m)}, y_+^{(m)}, y_-^{(m)})$  denotes the preference data at step  $m$ . During optimization, we maintain a low-level policy (the base LM) and a high-level policy (the LoRA adapter). For low-level preference pairs, we disable the LoRA adapter and update only the base LM parameters with  $((d^{(1:m)}, c^{(1:m-1)}), c_+^{(m)}, c_-^{(m)})$ ; for high-level pairs, we freeze the base LM and update only the LoRA parameters with  $((d^{(1:m-1)}, c^{(1:m-1)}), d_+^{(m)}, d_-^{(m)})$ .

**Multi-level update schemes.** A key challenge in jointly optimizing the two policies lies in designing an effective update scheme. We adopt an interleaved strategy: mini-batches of high-level and low-level examples are alternated, allowing the planner and executor to be trained jointly while preserving modularity. We compare this update scheme with cheaper alternatives in Section C.

**Multi-round Step-DPO for online optimization.** Motivated by the benefits of on-policy data sampling in RL, we adopt an iterative Step-DPO framework for improved optimization. Specifically,

270 in the  $t$ -th iteration, we use the current policies to sample preference pairs to create the preference  
 271 data  $\mathcal{D}_{\text{pref}}^{(t)}$ . Then, we use to update the policies for the next iteration as  
 272

$$273 \mathcal{L}_{\text{ms-DPO}} := -\mathbb{E}_{(x^{(m)}, y_+^{(m)}, y_-^{(m)}) \sim \mathcal{D}_{\text{pref}}^{(t)}} \frac{1}{M} \sum_{m=1}^M \log \sigma \left[ \beta \left( \log \frac{p_\theta^{(t+1)}(y_+^{(m)} | x^{(m)})}{p_\theta^{(t)}(y_+^{(m)} | x^{(m)})} - \log \frac{p_\theta^{(t+1)}(y_-^{(m)} | x^{(m)})}{p_\theta^{(t)}(y_-^{(m)} | x^{(m)})} \right) \right]. \quad (4)$$

273 The training procedure is summarized in Algorithm 2. More implementation details are provided in  
 274 Section C.  
 275

276 **Process preference modeling.** A key component of our online Step-DPO pipeline is the process-  
 277 level supervision for both the high-level descriptors  $d^{(m)}$  and the low-level detailed contents  $c^{(m)}$ .  
 278 Consider the full reasoning trajectory after a prefix  $x^{(m)}$  as future tokens  $\tau_{m+1:M} = (d^{(m+1)}, c^{(m+1)},$   
 279  $\dots, d^{(M)}, c^{(M)})$ , generated by a rollout policy  $p_{\text{roll}}$ . The survival probability of  $x^{(m)}$  is

$$283 g(x^{(m)}) = \mathbb{P}(R = 1 | x^{(m)}) = \mathbb{E}_{\tau_{m+1:M} \sim p_{\text{roll}}(\cdot | x^{(m)})} [R(x^{(m)}, \tau_{m+1:M})], \quad (5)$$

284 where the terminal reward  $R(x^{(m)}, \tau_{m+1:M})$  is 1 if the final answer is correct, and 0 otherwise.  
 285

286 Given an estimate of the survival probability  $\hat{g}$ , we construct preference data using a utility defined as  
 287 the increment in log-survivability:  
 288

$$289 U(y^{(m)}) = \log \hat{g}(x^{(m)}, y^{(m)}) - \log \hat{g}(x^{(m)}). \quad (6)$$

290 where the survivability is clipped as  $\tilde{g} = \text{clip}(\hat{g}, \varepsilon, 1 - \varepsilon)$  with  $\varepsilon = 0.001$  for numerical stability.  
 291 Intuitively,  $U(y^{(m)})$  quantifies how the selected candidate changes the probability of eventual success  
 292 relative to the preceding prefix. Then we impose the condition that the utility difference satisfies the  
 293 following:  
 294

$$295 U(y_+^{(m)}) - U(y_-^{(m)}) = \log \tilde{g}(x^{(m)}, y_+^{(m)}) - \log \tilde{g}(x^{(m)}, y_-^{(m)}) > \delta, \quad (7)$$

296 where the margin threshold  $\delta$  ensures the reliability of the preference data.  
 297

298 **Twisted Sequential Monte Carlo.** A key challenge of the above approach is computational cost:  
 299 estimating survivability naively requires running the base model multiple times per prefix. To address  
 300 this, we adopt a strategy based on Twisted Sequential Monte Carlo (TSMC) that provides accurate  
 301 survivability estimates while remaining computationally efficient. In particular, we use a lightweight  
 302 rollout model to generate fast continuations and apply importance weighting to correct for the  
 303 distribution mismatch.  
 304

305 Given a prefix, the  $k$ -th particle at step  $m - 1$  has state  $x_k^{(m-1)}$ . We first sample a candidate step  
 306  $y_k^{(m)} \sim p_{\text{roll}}(\cdot | x_k^{(m-1)})$  and form the updated state  $x_k^{(m)} = [x_k^{(m-1)}, y_k^{(m)}]$ . Its importance weight  
 307 is updated as

$$308 W_k^{(m)} = W_k^{(m-1)} \cdot \tilde{w}_k^{(m)}, \quad (8)$$

309 with incremental weight  
 310

$$311 \tilde{w}_k^{(m)} = G_m(x_k^{(m)}) \cdot \frac{p_\theta(x_k^{(m)} | x_k^{(m-1)})}{p_{\text{roll}}(x_k^{(m)} | x_k^{(m-1)})} \cdot \frac{\phi_m(x_k^{(m)})}{\phi_{m-1}(x_k^{(m-1)})}. \quad (9)$$

312 where  $W_k^{(m)}$  is the  $m$ -th step importance weight with  $W_k^{(0)} = 1$ , the potential function  $G_m$  is defined  
 313 as  $G_m(x_k^{(m)}) = 1$  for  $m < M$  and  $G_M(x_k^{(M)}) = \mathbf{1}_{\text{correct}}(x_k^{(M)})$ , i.e., final answer correctness,  $p_\theta$   
 314 denotes the base model,  $p_{\text{roll}}$  is the rollout policy, and  $\phi_m$  is a learned survivability critic at step  $m$ .  
 315 When  $p_{\text{roll}}$  is **close** to  $p_\theta$ , the contribution from the survivability critic  $\phi_m$  becomes **negligible**, and in  
 316 this case, we can simplify  $\phi_m \approx 1$ , leading to  $\tilde{w}_k^{(m)} \approx G_m(x_k^{(m)})$ . Finally, the survivability estimate  
 317 is given by:  
 318

$$319 \hat{g}_K(x^{(m)}) = \frac{1}{K} \sum_{k=1}^K W_k^{(M_k)} = \frac{1}{K} \sum_{k=1}^K W_k^{(m)} \prod_{j=m+1}^{M_k} \tilde{w}_k^{(j)} \approx W^{(m)} \cdot \frac{1}{K} \sum_{k=1}^K \mathbf{1}_{\text{correct}}(x_k^{(M_k)}). \quad (10)$$

320 where  $K$  represents the number of particles, and  $W^{(m)}$  is a shared term depending only on the  
 321 prefix  $x^{(m)}$ . Specifically, we fine-tune a small LM on the same low-level SFT data and use it as the  
 322

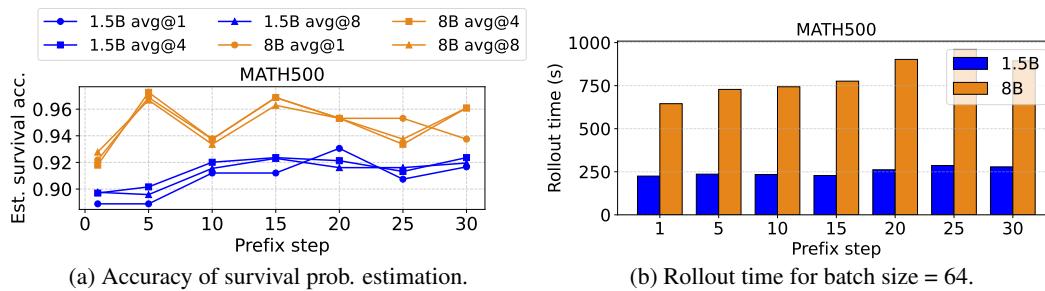


Figure 6: Monte Carlo rollout analysis. (a) The accuracies of R1-Distill-LLaMA-8B and R1-Distill-Qwen-1.5B are highly comparable. (b) Runtime grows with model size, with R1-Distill-LLaMA-8B substantially slower than R1-Distill-Qwen-1.5B. Implementation details are provided in Section C.

rollout policy  $p_{\text{roll}}$ . This choice is justified for three reasons: (i) the small model shares a *similar* distribution with the base model because it is trained on the same data; (ii) we only need to capture the correct *relative tendency*; and (iii) rollout with a smaller model is significantly more *cost-effective*. In Figure 6, the small model achieves comparable estimation accuracy while being much faster. Additionally, we split the RL problems into easy and hard categories. For *hard* problems, where the accuracy of the small model is significantly lower than that of the base model, we revert to using the *base model* as the rollout policy. In practice, we use the base model for AIME24 and GPQA.

## 4 EXPERIMENTS

**Dataset** We evaluate our approach on math (MATH500 (Hendrycks et al., 2021), AIME24 (MAA, 2024)), science (GPQA-diamond (Rein et al., 2023)), and logical reasoning (BoardGameQA-hard (Kazemi et al., 2023)). Detailed dataset statistics are provided in Section B. For training, we construct a multi-level dataset and divide it into two parts: SFT data and online preference data for Step-DPO. The SFT set contains about 80K examples produced using the multi-level decomposition procedure described in Section 3.2. In addition, we reserve 10K prompts for Step-DPO training. Details of dataset construction are provided in Section C.

**Implementation details** We fine-tune three base models, Qwen-2.5-1.5B (Yang et al., 2024a), Qwen-2.5-MATH-7B (Yang et al., 2024b) and LLaMA-3.1-8B (Grattafiori et al., 2024), on the low-level data with full parameter fine-tuning. The resulting models are frozen, and we apply LoRA fine-tuning on the high-level policy. We also fully fine-tune a Qwen2.5-0.5B-Instruct model for summarization, which is frozen after SFT and shared across all base models (see Section C for more details). Our online Step-DPO pipeline is implemented with the TRL framework. In each training round, we sample a batch of approximately 3K prompts. For each prompt, we randomly select 4 reasoning steps and generate  $M = 2$  candidate continuations per step. These candidates are scored using the utility (Equation (6)). In experiments, we use Qwen-2.5-1.5B SFT on the low-level data as the rollout policy, with  $K = 4$  sampled rollouts per prefix (see Section 4 for parameter studies). The fast rollout model is frozen after SFT and is shared across base models. From each prefix, we form one preference pair, weighted by the utility margin  $\delta = 0.4$ . Each update uses mini-batches of size 32 for  $E = 4$  epochs, and applies the standard Step-DPO objective with  $\beta = 0.1$ . Generated continuations are capped at a maximum length of 8,192 tokens. More implementation details, including ablation settings and hyperparameters, are provided in Section C.

**Baselines** We compare our method with the following baselines: the base model, the instruction fine-tuned model, RL applied directly to the base model (SimpleRL (Zeng et al., 2025)), distillation

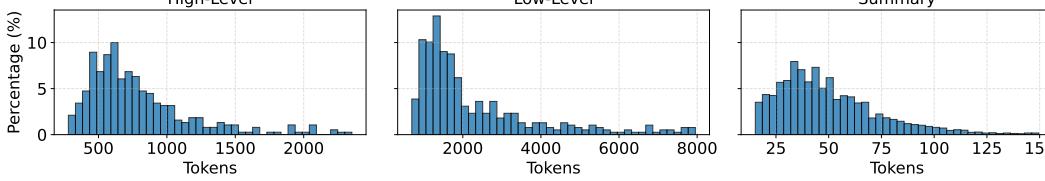


Figure 7: Length distributions for high-level trajectories, low-level trajectories, and distilled summaries in our SFT dataset. Trajectories longer than 8k tokens are truncated.

378 Table 1: Overall performance comparison across different benchmarks. The best performance for each task  
379 using the same base model is in bold. Asterisks (\*) denote available results obtained from the official reports.

380 381 382 383 Method	384 385 386 387 388 389 390 Math		384 385 386 387 388 389 390 Science		384 385 386 387 388 389 390 Logic	
	384 385 386 387 388 389 390 MATH500 Pass@1	384 385 386 387 388 389 390 AIME24 Pass@1	384 385 386 387 388 389 390 GPQA (Diamond) Pass@1	384 385 386 387 388 389 390 BoardGameQA (Hard) Pass@1	384 385 386 387 388 389 390 Avg. Pass@1	
384 385 386 387 388 389 390 Qwen-2.5-1.5B						
384 385 386 387 388 389 390 Base	384 385 386 387 388 389 390 29.6 $\pm$ 0.6	384 385 386 387 388 389 390 0.0 $\pm$ 0.0	384 385 386 387 388 389 390 0.0 $\pm$ 0.0	384 385 386 387 388 389 390 21.2 $\pm$ 1.1	384 385 386 387 388 389 390 35.0 $\pm$ 1.8	384 385 386 387 388 389 390 21.5
384 385 386 387 388 389 390 Instruct	384 385 386 387 388 389 390 54.6 $\pm$ 0.4	384 385 386 387 388 389 390 3.3 $\pm$ 1.5	384 385 386 387 388 389 390 10.0 $\pm$ 0.3	384 385 386 387 388 389 390 25.2 $\pm$ 1.4	384 385 386 387 388 389 390 41.0 $\pm$ 1.0	384 385 386 387 388 389 390 31.0
384 385 386 387 388 389 390 SimpleRL	384 385 386 387 388 389 390 59.0*	384 385 386 387 388 389 390 6.7*	384 385 386 387 388 389 390 4.2*	384 385 386 387 388 389 390 —	384 385 386 387 388 389 390 —	384 385 386 387 388 389 390 —
384 385 386 387 388 389 390 DeepSeek-R1-Distill	384 385 386 387 388 389 390 83.9*	384 385 386 387 388 389 390 28.9*	384 385 386 387 388 389 390 43.3 $\pm$ 0.4	384 385 386 387 388 389 390 33.8*	384 385 386 387 388 389 390 40.0 $\pm$ 2.2	384 385 386 387 388 389 390 47.7
384 385 386 387 388 389 390 Plan-and-Solve	384 385 386 387 388 389 390 60.8 $\pm$ 1.0	384 385 386 387 388 389 390 2.0 $\pm$ 1.4	384 385 386 387 388 389 390 6.2 $\pm$ 0.4	384 385 386 387 388 389 390 20.1 $\pm$ 1.6	384 385 386 387 388 389 390 44.6 $\pm$ 1.6	384 385 386 387 388 389 390 31.9
384 385 386 387 388 389 390 SFT + DPO	384 385 386 387 388 389 390 76.5 $\pm$ 1.3	384 385 386 387 388 389 390 12.0 $\pm$ 1.9	384 385 386 387 388 389 390 21.6 $\pm$ 0.7	384 385 386 387 388 389 390 27.6 $\pm$ 1.6	384 385 386 387 388 389 390 51.8 $\pm$ 1.7	384 385 386 387 388 389 390 42.0
384 385 386 387 388 389 390 SFT + Step-DPO	384 385 386 387 388 389 390 81.4 $\pm$ 1.2	384 385 386 387 388 389 390 24.0 $\pm$ 2.0	384 385 386 387 388 389 390 36.5 $\pm$ 0.7	384 385 386 387 388 389 390 29.0 $\pm$ 1.7	384 385 386 387 388 389 390 56.2 $\pm$ 1.7	384 385 386 387 388 389 390 47.8
384 385 386 387 388 389 390 SFT + GRPO	384 385 386 387 388 389 390 82.1 $\pm$ 1.2	384 385 386 387 388 389 390 25.2 $\pm$ 2.2	384 385 386 387 388 389 390 36.0 $\pm$ 0.8	384 385 386 387 388 389 390 30.2 $\pm$ 1.5	384 385 386 387 388 389 390 56.0 $\pm$ 1.6	384 385 386 387 388 389 390 48.4
384 385 386 387 388 389 390 MLR (SFT only)	384 385 386 387 388 389 390 62.0 $\pm$ 1.2	384 385 386 387 388 389 390 8.9 $\pm$ 1.4	384 385 386 387 388 389 390 13.3 $\pm$ 0.4	384 385 386 387 388 389 390 26.0 $\pm$ 2.0	384 385 386 387 388 389 390 46.4 $\pm$ 1.5	384 385 386 387 388 389 390 35.8
384 385 386 387 388 389 390 MLR	384 385 386 387 388 389 390 <b>86.1</b> $\pm$ 1.0	384 385 386 387 388 389 390 <b>31.2</b> $\pm$ 1.0	384 385 386 387 388 389 390 <b>47.4</b> $\pm$ 0.4	384 385 386 387 388 389 390 <b>37.6</b> $\pm$ 1.9	384 385 386 387 388 389 390 <b>62.0</b> $\pm$ 1.7	384 385 386 387 388 389 390 <b>54.2</b>
384 385 386 387 388 389 390 Qwen-2.5-MATH-7B						
384 385 386 387 388 389 390 Base	384 385 386 387 388 389 390 52.0 $\pm$ 0.5	384 385 386 387 388 389 390 2.0 $\pm$ 1.0	384 385 386 387 388 389 390 5.0 $\pm$ 0.3	384 385 386 387 388 389 390 20.5 $\pm$ 1.1	384 385 386 387 388 389 390 33.0 $\pm$ 1.6	384 385 386 387 388 389 390 26.9
384 385 386 387 388 389 390 Instruct	384 385 386 387 388 389 390 82.1 $\pm$ 0.4	384 385 386 387 388 389 390 16.7 $\pm$ 1.8	384 385 386 387 388 389 390 34.0 $\pm$ 0.4	384 385 386 387 388 389 390 27.8 $\pm$ 1.3	384 385 386 387 388 389 390 44.5 $\pm$ 1.4	384 385 386 387 388 389 390 42.8
384 385 386 387 388 389 390 SimpleRL	384 385 386 387 388 389 390 80.2*	384 385 386 387 388 389 390 40.0*	384 385 386 387 388 389 390 24.0*	384 385 386 387 388 389 390 —	384 385 386 387 388 389 390 —	384 385 386 387 388 389 390 —
384 385 386 387 388 389 390 DeepSeek-R1-Distill	384 385 386 387 388 389 390 92.8*	384 385 386 387 388 389 390 55.5*	384 385 386 387 388 389 390 78.0 $\pm$ 0.4	384 385 386 387 388 389 390 49.1*	384 385 386 387 388 389 390 42.4 $\pm$ 1.4	384 385 386 387 388 389 390 60.0
384 385 386 387 388 389 390 Plan-and-Solve	384 385 386 387 388 389 390 85.6 $\pm$ 0.9	384 385 386 387 388 389 390 18.2 $\pm$ 1.7	384 385 386 387 388 389 390 34.9 $\pm$ 0.5	384 385 386 387 388 389 390 28.4 $\pm$ 1.6	384 385 386 387 388 389 390 52.1 $\pm$ 1.5	384 385 386 387 388 389 390 46.1
384 385 386 387 388 389 390 SFT + DPO	384 385 386 387 388 389 390 87.4 $\pm$ 1.0	384 385 386 387 388 389 390 36.0 $\pm$ 1.8	384 385 386 387 388 389 390 53.0 $\pm$ 0.5	384 385 386 387 388 389 390 36.0 $\pm$ 1.6	384 385 386 387 388 389 390 54.5 $\pm$ 1.5	384 385 386 387 388 389 390 53.4
384 385 386 387 388 389 390 SFT + Step-DPO	384 385 386 387 388 389 390 88.5 $\pm$ 0.9	384 385 386 387 388 389 390 48.5 $\pm$ 1.9	384 385 386 387 388 389 390 70.5 $\pm$ 0.5	384 385 386 387 388 389 390 48.0 $\pm$ 1.7	384 385 386 387 388 389 390 56.0 $\pm$ 1.6	384 385 386 387 388 389 390 60.3
384 385 386 387 388 389 390 SFT + GRPO	384 385 386 387 388 389 390 89.7 $\pm$ 1.0	384 385 386 387 388 389 390 46.5 $\pm$ 1.9	384 385 386 387 388 389 390 66.2 $\pm$ 0.5	384 385 386 387 388 389 390 46.0 $\pm$ 1.8	384 385 386 387 388 389 390 57.5 $\pm$ 1.6	384 385 386 387 388 389 390 59.9
384 385 386 387 388 389 390 MLR (SFT only)	384 385 386 387 388 389 390 86.3 $\pm$ 1.0	384 385 386 387 388 389 390 22.4 $\pm$ 1.9	384 385 386 387 388 389 390 40.5 $\pm$ 0.5	384 385 386 387 388 389 390 34.6 $\pm$ 1.7	384 385 386 387 388 389 390 54.8 $\pm$ 1.6	384 385 386 387 388 389 390 49.5
384 385 386 387 388 389 390 MLR	384 385 386 387 388 389 390 <b>94.1</b> $\pm$ 0.9	384 385 386 387 388 389 390 <b>58.8</b> $\pm$ 1.8	384 385 386 387 388 389 390 <b>80.5</b> $\pm$ 0.4	384 385 386 387 388 389 390 <b>51.2</b> $\pm$ 1.5	384 385 386 387 388 389 390 <b>60.5</b> $\pm$ 1.6	384 385 386 387 388 389 390 <b>66.2</b>
384 385 386 387 388 389 390 Llama-3.1-8B						
384 385 386 387 388 389 390 Base	384 385 386 387 388 389 390 13.6 $\pm$ 0.4	384 385 386 387 388 389 390 0.0 $\pm$ 0.0	384 385 386 387 388 389 390 0.0 $\pm$ 0.0	384 385 386 387 388 389 390 1.5 $\pm$ 1.0	384 385 386 387 388 389 390 2.0 $\pm$ 1.1	384 385 386 387 388 389 390 4.3
384 385 386 387 388 389 390 Instruct	384 385 386 387 388 389 390 51.9 $\pm$ 0.2	384 385 386 387 388 389 390 6.7 $\pm$ 1.8	384 385 386 387 388 389 390 13.3 $\pm$ 0.2	384 385 386 387 388 389 390 22.7 $\pm$ 0.6	384 385 386 387 388 389 390 40.0 $\pm$ 1.2	384 385 386 387 388 389 390 30.3
384 385 386 387 388 389 390 SimpleRL	384 385 386 387 388 389 390 23.0*	384 385 386 387 388 389 390 0.0*	384 385 386 387 388 389 390 0.2*	384 385 386 387 388 389 390 —	384 385 386 387 388 389 390 —	384 385 386 387 388 389 390 —
384 385 386 387 388 389 390 DeepSeek-R1-Distill	384 385 386 387 388 389 390 89.1*	384 385 386 387 388 389 390 50.4*	384 385 386 387 388 389 390 70.0 $\pm$ 0.4	384 385 386 387 388 389 390 49.0*	384 385 386 387 388 389 390 46.0 $\pm$ 3.8	384 385 386 387 388 389 390 58.6
384 385 386 387 388 389 390 Plan-and-Solve	384 385 386 387 388 389 390 62.4 $\pm$ 1.1	384 385 386 387 388 389 390 12.3 $\pm$ 1.8	384 385 386 387 388 389 390 24.1 $\pm$ 0.4	384 385 386 387 388 389 390 31.0 $\pm$ 1.6	384 385 386 387 388 389 390 47.2 $\pm$ 1.7	384 385 386 387 388 389 390 38.2
384 385 386 387 388 389 390 SFT + DPO	384 385 386 387 388 389 390 74.1 $\pm$ 1.5	384 385 386 387 388 389 390 32.4 $\pm$ 1.8	384 385 386 387 388 389 390 52.0 $\pm$ 0.6	384 385 386 387 388 389 390 44.0 $\pm$ 1.7	384 385 386 387 388 389 390 56.0 $\pm$ 1.7	384 385 386 387 388 389 390 51.6
384 385 386 387 388 389 390 SFT + Step-DPO	384 385 386 387 388 389 390 82.4 $\pm$ 1.3	384 385 386 387 388 389 390 42.6 $\pm$ 2.0	384 385 386 387 388 389 390 61.2 $\pm$ 0.5	384 385 386 387 388 389 390 49.2 $\pm$ 1.5	384 385 386 387 388 389 390 62.1 $\pm$ 1.4	384 385 386 387 388 389 390 59.1
384 385 386 387 388 389 390 SFT + GRPO	384 385 386 387 388 389 390 86.5 $\pm$ 1.4	384 385 386 387 388 389 390 42.0 $\$				

432  
433  
434  
435  
Table 2: Ablation results using Qwen-2.5-1.5B. For the high-  
level SFT ablation, all methods use the same low-level SFT. For  
the hierarchical-level ablation, all variants share the same trained  
model. Our approach is highlighted in bold.

Method	MATH500		AIME24		Avg. Pass@1
	Pass@1	Pass@1	Pass@1	Cons@32	
Ablation of high-level SFT strategies					
<b>SFT (low) + LoRA (high)</b>	<b>62.0</b> $\pm$ 1.2	<b>8.9</b> $\pm$ 1.4	<b>13.3</b> $\pm$ 0.4	<b>35.5</b>	
Base + LoRA (high)	56.4 $\pm$ 1.5	4.1 $\pm$ 1.1	9.2 $\pm$ 0.7	30.3	
SFT (high)	59.8 $\pm$ 1.3	6.5 $\pm$ 1.2	11.0 $\pm$ 0.5	33.2	
Ablation of hierarchical levels					
<b>High-level + Low-level</b>	<b>86.1</b> $\pm$ 1.0	<b>31.2</b> $\pm$ 1.0	<b>47.4</b> $\pm$ 0.4	<b>58.7</b>	
High-level only	80.0 $\pm$ 1.3	18.4 $\pm$ 2.0	30.5 $\pm$ 0.8	49.2	
Low-level only	84.2 $\pm$ 1.1	27.1 $\pm$ 1.8	41.0 $\pm$ 0.6	55.7	

444  
445  
Table 3: Ablation results for the core components of MLR using LLaMA-3.1-8B.

Method	Math		Science		Logic	
	MATH500 Pass@1	AIME24 Pass@1	GPQA (Diamond) Pass@1	BoardGameQA (Hard) Pass@1	Avg. Pass@1	
Ours	<b>91.5</b> $\pm$ 1.3	<b>53.2</b> $\pm$ 2.0	<b>73.3</b> $\pm$ 0.4	<b>52.8</b> $\pm$ 1.5	<b>67.0</b> $\pm$ 1.4	<b>66.1</b>
DPO-only	78.2 $\pm$ 1.4	38.1 $\pm$ 1.9	57.0 $\pm$ 0.5	46.0 $\pm$ 1.6	59.0 $\pm$ 1.6	55.3
Low-level policy + Step-DPO	82.4 $\pm$ 1.3	42.6 $\pm$ 2.0	61.2 $\pm$ 0.5	49.2 $\pm$ 1.5	62.1 $\pm$ 1.4	59.1
Low-level policy + DPO	74.1 $\pm$ 1.5	32.4 $\pm$ 1.8	52.0 $\pm$ 0.6	44.0 $\pm$ 1.7	56.0 $\pm$ 1.7	51.6
SFT-only	63.8 $\pm$ 1.2	20.2 $\pm$ 2.0	36.7 $\pm$ 0.4	36.2 $\pm$ 1.8	48.5 $\pm$ 1.8	42.2

452  
453  
454  
455  
456  
457  
RL, distillation with vanilla long CoTs, Plan-and-Solve, and MLR. Across all benchmarks, MLR  
458 consistently outperforms all baselines, with its structured design enabling more effective reasoning  
459 on complex, long-horizon tasks. In addition, the online step-DPO procedure yields substantial gains  
460 over the SFT model. Finally, we report average response lengths across benchmarks: high-level  
461 trajectories are approximately 10–20% the length of low-level ones (Figure 11).

462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
**Parameter studies** We study the effect of varying the rollout count  $K$ . Utility estimates produced  
478 by the 1.5B rollout policy with  $K$  rollouts are compared against reference utilities from the 8B model  
479 using 16 rollouts (Figure 15). As expected, increasing  $K$  reduces estimator variance, though at the  
480 cost of higher computation. To mitigate this overhead, we introduce a margin threshold  $\delta$  when  
481 selecting preference pairs. We further measure the agreement of the selected preference pairs as a  
482 function of  $K$ , defined as the fraction whose preference direction agrees with the base model using 16  
483 rollouts (Figure 8). Finally, we report model performance across training stages under different values  
484 of  $K$  (Figure 12). The results show that our chosen setting attains comparable final performance  
485 while substantially reducing computational cost. Implementation details are provided in Section D.

486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
**Ablation studies** We conduct a series of ablation studies to evaluate the contributions of key  
501 components in MLR. We compare five configurations: (i) the full method, (ii) only applying DPO,  
502 (iii) using only the low-level policy with Step-DPO or (iv) DPO, and (v) training with SFT only.  
503 Table 3 summarizes the results, which show that both multi-level modeling and step-level preferences  
504 are essential. Figure 10 further illustrates this trend: our method achieves higher preference accuracy  
505 throughout training. We additionally ablate the high-level SFT component (Table 2), evaluating two  
506 alternatives: (i) applying LoRA to the original base model and (ii) full-parameter SFT. A detailed  
507 discussion and implementation details are provided in Section C. We also ablate the hierarchical  
508 structure (Table 2), comparing (i) high-level-only and (ii) low-level-only variants. Further analysis  
509 and implementation details appear in Section D. Across all ablations, our full strategy yields the  
510 strongest performance.

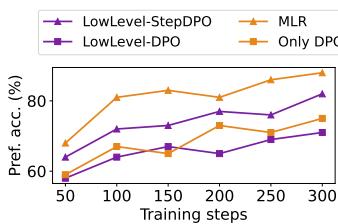
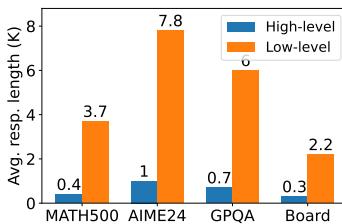
485  
Figure 10: Preference accuracy.

Figure 11: Avg. resp. length.

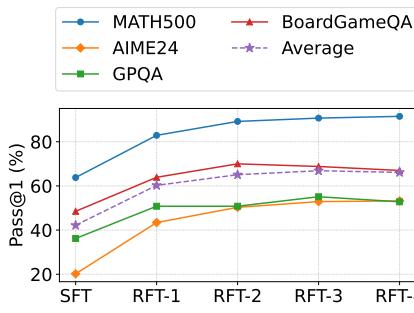


Figure 9: Performance of MLR over different training stages.

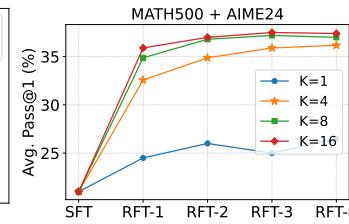


Figure 12: Effect of K in MLR.

## 486 5 RELATED WORK

488 Reasoning models (OpenAI, 2024; Qwen-Team, 2024; Guo et al., 2025) represent a transformative  
 489 advancement in the evolution of LMs, sparking substantial interest in replicating their strong perfor-  
 490 mance. Several subsequent works (HuggingFace, 2025; OpenThoughts-Team, 2025; Bespoke-Labs,  
 491 2025; Muennighoff et al., 2025) have primarily relied on supervised fine-tuning (SFT). While this  
 492 approach has shown promising results, pure SFT is generally less efficient in utilizing training signals  
 493 compared to reinforcement learning (RL), as it passively imitates given demonstrations without  
 494 exploration or fine-grained credit assignment.

495 Recent RL-based works introduce improvements along three main dimensions:

496 1) RL Algorithms. RL methods for LLM reasoning mainly fall into PPO-, GRPO-, and REINFORCE-  
 497 based families. PPO variants (Yuan et al., 2025; Yue et al., 2025) improve value estimation and  
 498 stability; GRPO methods (Shao et al., 2024; Yu et al., 2025; Liu et al., 2025) remove the critic  
 499 and refine sampling, normalization, and token-level gradients; REINFORCE variants (Hu, 2025;  
 500 Kimi-Team et al., 2025) reduce variance through KL penalties and centralized rewards. Despite these  
 501 advances, all rely largely on sparse outcome rewards, which provide weak credit assignment for long  
 502 reasoning trajectories.

503 2) Reward Design. Most existing approaches define rewards based on outcome accuracy, format  
 504 compliance, and length constraints (Zhang et al., 2025a). Process rewards remain largely unexplored  
 505 in current pipelines.

506 3) Data Sampling Strategies. Curriculum learning (Hu et al., 2025; Zhang et al., 2025b) gradually  
 507 increases task difficulty during training. Rejection sampling (Wen et al., 2025; Yu et al., 2025) is  
 508 commonly used to filter low-quality samples and stabilize optimization.

509 However, outcome-reward RL is inherently inefficient and weakly informative for long trajectories.  
 510 Recent work (Lightman et al., 2023; Wang et al., 2023b) therefore explores Process Reward Models  
 511 (PRM), but they struggle in practice (Guo et al., 2025): 1) explicitly defining fine-grained reasoning  
 512 steps is difficult; 2) reliably verifying the correctness of each intermediate step is non-trivial; 3)  
 513 training a separate reward model introduces the risk of reward hacking and requires additional  
 514 training.

515 On the other hand, Direct Preference Optimization (DPO) (Rafailov et al., 2023) also faces limitations:  
 516 it relies on offline data and trajectory-level preferences. Step-DPO (Lai et al., 2024) mitigates this by  
 517 constructing curated step-wise preference data, while DPO with AI feedback (Cui et al., 2023; Guo  
 518 et al., 2024a) enables online updates. However, these approaches are not well suited for long CoTs,  
 519 as they rely on strong teacher models (e.g., GPT-4) to provide step-level preferences, both costly and  
 520 unreliable on harder tasks. To address this, we introduce a scalable TSMC-based approach to provide  
 521 stepwise preferences. Unlike naive tree-search methods (Wang et al., 2023b), which are prohibitively  
 522 expensive on long trajectories, our approach remains efficient and stable for long-horizon supervision.

523 Finally, using a single policy for long-horizon reasoning introduces additional limitations such as  
 524 plan failures. MLR instead adopts a multi-level strategy, differing from existing planning methods  
 525 (Huang et al., 2022; Xu et al., 2023; Wang et al., 2023a) that generate a full plan upfront and assume  
 526 all subtasks succeed as written. Such fixed plans propagate early errors. In contrast, MLR learns  
 527 a planner that adapts its plans based on execution feedback, enabling revisions and yielding more  
 528 robust long-horizon reasoning.

## 531 6 CONCLUSION

532 We presented a novel multi-level reasoning (MLR) framework that enhances inference-time scaling  
 533 by structuring the reasoning process into interleaved high-level abstractions and low-level details.  
 534 This decomposition supports efficient multi-policy modeling and dynamic plan adaptation, addressing  
 535 critical challenges faced by single-policy long-CoT approaches. By sidestepping the limitations  
 536 of prior outcome supervision methods, MLR provides a scalable and robust pathway for training  
 537 reasoning-focused language models. Extensive experiments demonstrate consistent performance  
 538 gains across math, science, and logical reasoning tasks, highlighting MLR’s promise as a general-  
 539 purpose reasoning framework.

540  
541  
**LIMITATIONS**

542 Our method requires maintaining two separate policies and performing additional steps for process  
 543 supervision, which increases training complexity. To keep costs practical, we freeze the base model  
 544 for the high-level policy, decouple supervision estimation from trajectory generation, and alternate  
 545 policy updates. These strategies help manage training overhead; however, on resource-constrained  
 546 devices further optimizations, such as quantization, activation checkpointing, or memory-efficient  
 547 attention, may still be necessary. The approach also introduces extra hyperparameters, though most  
 548 can be assigned reasonable default values that transfer well across tasks. Our experiments indicate  
 549 that performance is robust to moderate variations in these settings, reducing the need for extensive  
 550 hyperparameter tuning.

551  
552  
**ETHICS STATEMENT**  
553

554 This work does not involve human subjects, personal data, or sensitive demographic information. All  
 555 datasets are publicly available and used under their respective licenses. Our method aims to improve  
 556 the efficiency of large language models, which can promote accessibility and sustainability. We  
 557 acknowledge that LLMs may be misused for generating harmful or biased content, but our work does  
 558 not specifically target such applications. No conflicts of interest or ethical concerns are associated  
 559 with this research.

560  
561  
**REPRODUCIBILITY STATEMENT**  
562

563 We have made significant efforts to ensure the reproducibility of our work. The main paper and  
 564 appendix provide detailed descriptions of our model architecture, training procedures, and evaluation  
 565 settings. All datasets used are publicly available, and we include a complete description of data  
 566 processing steps in the supplementary materials. Pseudocode and complexity analysis are provided in  
 567 the paper and appendix to clarify algorithmic details.

568  
569  
**REFERENCES**

570 Bespoke-Labs. Bespoke-stratos: The unreasonable effectiveness of reasoning distil-  
 571 lation. <https://www.bespokelabs.ai/blog/bespoke-stratos-the-unreasonable-effectiveness-of->  
 572 reasoning-distillation, 2025. Accessed: 2025-01-22.

573 Mark Briers, Arnaud Doucet, and Simon Maskell. Smoothing algorithms for state–space models.  
 574 *Annals of the Institute of Statistical Mathematics*, 62:61–89, 2010.

575 Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie,  
 576 Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback.  
 577 *arXiv preprint arXiv:2310.01377*, 2023.

578 Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv*  
 579 *preprint arXiv:2307.08691*, 2023.

580 Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the*  
 581 *Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006.

582 Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposi-  
 583 tion. *Journal of artificial intelligence research*, 13:227–303, 2000.

584 Arnaud Doucet, Nando De Freitas, Neil James Gordon, et al. *Sequential Monte Carlo methods in*  
 585 *practice*, volume 1. Springer, 2001.

586 Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In  
 587 *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

588 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad  
 589 Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of  
 590 models. *arXiv preprint arXiv:2407.21783*, 2024.

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

597

598 Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre  
 599 Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online  
 600 ai feedback. *arXiv preprint arXiv:2402.04792*, 2024a.

601 Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest,  
 602 and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and  
 603 challenges. *arXiv preprint arXiv:2402.01680*, 2024b.

604 Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy  
 605 learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint  
 606 arXiv:1910.11956*, 2019.

607

608 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,  
 609 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv  
 610 preprint arXiv:2103.03874*, 2021.

611 Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,  
 612 Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

613

614 Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv  
 615 preprint arXiv:2501.03262*, 2025.

616 Jingcheng Hu, Yinmin Zhang, Qi Han, Dixin Jiang, and Heung-Yeung Shum Xiangyu Zhang. Open-  
 617 reasoner-zero: An open source approach to scaling reinforcement learning on the base model.  
 618 <https://github.com/Open-Reasoner-Zero/Open-Reasoner-Zero>, 2025.

619

620 Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot  
 621 planners: Extracting actionable knowledge for embodied agents. In *International conference on  
 622 machine learning*, pp. 9118–9147. PMLR, 2022.

623 HuggingFace. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL <https://github.com/huggingface/open-r1>.

624

625 Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey.  
 626 *Journal of artificial intelligence research*, 4:237–285, 1996.

627

628 Mehran Kazemi, Quan Yuan, Deepti Bhatia, Najoung Kim, Xin Xu, Vaiva Imbrasaite, and Deepak  
 629 Ramachandran. Boardgameqa: A dataset for natural language reasoning with contradictory  
 630 information. *Advances in Neural Information Processing Systems*, 36:39052–39074, 2023.

631

632 Kimi-Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun  
 633 Xiao, Chenzhuang Du, Chonghua Liao, Chunling Tang, Congcong Wang, Dehao Zhang, Enming  
 634 Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han  
 635 Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li,  
 636 Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su,  
 637 Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye,  
 638 Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu,  
 639 Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong,  
 640 Weiran He, Weixiao Huang, Wenhai Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu,  
 641 Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang  
 642 Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du,  
 643 Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu  
 644 Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, and Zonghan Yang. Kimi k1.5: Scaling  
 645 reinforcement learning with llms, 2025. URL <https://arxiv.org/abs/2501.12599>.

646

647 Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E.  
 648 Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model  
 649 serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating  
 650 Systems Principles*, 2023.

648 Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-  
 649 wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*,  
 650 2024.

651

652 Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan  
 653 Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth*  
 654 *International Conference on Learning Representations*, 2023.

655 Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao,  
 656 Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint*  
 657 *arXiv:2412.19437*, 2024.

658

659 Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min  
 660 Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*,  
 661 2025.

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

665

666 Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke  
 667 Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time  
 668 scaling, 2025. URL <https://arxiv.org/abs/2501.19393>.

669

670 Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical  
 671 reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

672

673 OpenAI. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>, 2024. Accessed:  
 674 2025-05-15.

675

676 OpenAI. Learning to reason with language models, 2024. URL <https://openai.com/index/learning-to-reason-with-llms/>. Accessed: 2025-05-15.

677

678 OpenThoughts-Team. Open Thoughts. <https://open-thoughts.ai>, January 2025.

679

680 Qwen-Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024. URL  
 681 <https://qwenlm.github.io/blog/qwq-32b-preview/>.

682

683 Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea  
 684 Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances*  
 685 *in neural information processing systems*, 36:53728–53741, 2023.

686

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

690

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

694

695 Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng,  
 696 Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint*  
 697 *arXiv: 2409.19256*, 2024.

698

699 Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim.  
 700 Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language  
 701 models. *arXiv preprint arXiv:2305.04091*, 2023a.

702

703 Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui.  
 704 Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint*  
 705 *arXiv:2312.08935*, 2023b.

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

705 Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu  
 706 Tang, Xiaowei Lv, et al. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond.  
 707 *arXiv preprint arXiv:2503.10460*, 2025.

708 Binfeng Xu, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu.  
 709 Rewoo: Decoupling reasoning from observations for efficient augmented language models. *arXiv*  
 710 *preprint arXiv:2305.18323*, 2023.

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

714 An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu,  
 715 Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu,  
 716 Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert  
 717 model via self-improvement, 2024b. URL <https://arxiv.org/abs/2409.12122>.

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

721 Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in  
 722 long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025.

723 Yu Yue, Yufeng Yuan, Qiyi Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi  
 724 Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin  
 725 Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu,  
 726 Mingxuan Wang, Yonghui Wu, and Lin Yan. Vapo: Efficient and reliable reinforcement learning  
 727 for advanced reasoning tasks, 2025. URL <https://arxiv.org/abs/2504.05118>.

728 Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-  
 729 zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv*  
 730 *preprint arXiv:2503.18892*, 2025.

731 Chong Zhang, Yue Deng, Xiang Lin, Bin Wang, Dianwen Ng, Hai Ye, Xingxuan Li, Yao Xiao,  
 732 Zhanfeng Mo, Qi Zhang, et al. 100 days after deepseek-r1: A survey on replication studies and  
 733 more directions for reasoning language models. *arXiv preprint arXiv:2505.00551*, 2025a.

734 Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie  
 735 Wang, Yinghan Cui, Chao Wang, Junyi Peng, Shimiao Jiang, Shiqi Kuang, Shouyu Yin, Chaohang  
 736 Wen, Haotian Zhang, Bin Chen, and Bing Yu. Srpo: A cross-domain implementation of large-scale  
 737 reinforcement learning on llm, 2025b. URL <https://arxiv.org/abs/2504.14286>.

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756 **A LLM USAGE**  
757758 We used Large Language Models (LLMs) only as assistive tools for grammar refinement, readability  
759 improvements, and LaTeX formatting. They were not involved in generating research ideas, designing  
760 methods, conducting experiments, or analyzing results. All technical content and conclusions are  
761 entirely the work of the authors.  
762763 **B DATASET STATISTICS**  
764765 In this section, we provide statistics for all benchmark datasets used in our study. We consider  
766 MATH500 (Hendrycks et al., 2021), AIME24 (MAA, 2024) for math reasoning, GPQA-diamond  
767 (Rein et al., 2023) for science reasoning, and BoardGameQA-hard (Kazemi et al., 2023) for logical  
768 reasoning. For MATH, there are 7,500 training samples and 5,000 test samples. MATH500 is a  
769 subset of 500 representative test samples extracted by Lightman et al. (2023), with the remaining test  
770 samples added to the training set.  
771772 The AIME dataset is based on the American Invitational Mathematics Examination, a high-level  
773 math contest administered by the Mathematical Association of America (MAA) for high-achieving  
774 middle and high school students. We use the data<sup>3</sup> collected from 1983 to 2024, where each year  
775 contains 15 questions prior to 2000 and 30 questions thereafter. The 2024 dataset is used as the test  
776 set, while data from all previous years serve as the training set.  
777778 GPQA is a multiple-choice, Q&A dataset of very hard questions written and validated by experts  
779 in biology, physics, and chemistry. It includes three subsets: main (448 questions), diamond (198  
780 questions), and extended (546 questions). We use the diamond subset as the test set and combine the  
781 remaining subsets for training. To prevent data leakage, any questions overlapping with the test set  
782 are removed from the training set.  
783784 BoardGameQA is a claim verification problem with three types of labels (proved, disproved, un-  
785 known), which involves reasoning with contradictory information guided by preferences over rules as  
786 board games. The dataset is divided into 15 subsets based on reasoning depth and levels of conflict  
787 or distractors, each with separate train, validation, and test splits. We define BoardGameQA-hard  
788 as the union of all test sets from five challenging subsets: Main-depth3, DifficultConflict-depth2,  
789 HighConflict-depth2, KnowledgeHeavy-depth2, and ManyDistractors-depth2, resulting in a total test  
790 set of 500 examples. The remaining data from all subsets are used for training.  
791792 **C IMPLEMENTATION DETAILS**  
793794 **Dataset construction.** As mentioned above, we consider four publicly available reasoning bench-  
795 marks: MATH500, AIME24, GPQA, and BoardGameQA. For each benchmark, we use the available  
796 training split as seeds and reserve the test splits exclusively for evaluation. We randomly partition each  
797 training split into SFT and RL subsets (60% / 40% ratio). Because AIME and GPQA contain very  
798 few training questions, we augment their train splits with synthetic problems generated by GPT-4o  
799 using the prompts provided in Section D. For each seed problem we sample multiple candidate  
800 questions and retain only those whose final answers are **mutually consistent** with DeepSeek-R1, i.e.,  
801 DeepSeek-R1 solves the problem and produces the same final answer as GPT-4o. This procedure  
802 yields SFT sets of 7.2K MATH, 4K AIME, 4K GPQA, and 5K BoardGameQA problems, and RL  
803 sets of 4.8K, 1.5K, 1.5K, and 2K problems, respectively. Unless otherwise stated, all methods that we  
804 train ourselves (GRPO, Plan-and-Solve, MLR, and all ablations in Table 3) use exactly the same data,  
805 ensuring a fair comparison. Results for external baselines (Instruct, SimpleRL, DeepSeek-R1-Distill),  
806 marked with an asterisk in Table 1, are copied from their official reports and may rely on different  
807 training corpora; we include them as strong reference points.  
808809 Specially, in the SFT phase, we generate multiple high-quality trajectories for each problem by  
810 sampling four solutions from DeepSeek-R1, yielding approximately 80K filtered trajectories in total.  
811 Each accepted trajectory is then decomposed into step-by-step reasoning segments using DeepSeek-  
812 V3 via in-context learning, with each step annotated by a step descriptor. Because DeepSeek-V3 has  
813<sup>3</sup><https://www.kaggle.com/datasets/hemishveeraboina/aime-problem-set-1983-2024>

810 a maximum response length constraint, we pre-screen trajectories to ensure compliance, and process  
 811 the remaining long trajectories using GPT-4o. To enable multi-level reasoning, we further distill  
 812 each step into a concise high-level summary, again using DeepSeek-V3 with in-context learning.  
 813 All prompts used throughout this pipeline are provided in Section D. Figure 7 presents the resulting  
 814 trajectory length distributions across different reasoning levels and their corresponding summaries.  
 815

816 **Summarization.** The summarization module in MLR serves to distill essential information from the  
 817 evolving trajectory, allowing the planner to operate on concise, high-level representations rather than  
 818 being overwhelmed by unnecessary details. This becomes particularly important when trajectories  
 819 grow long (e.g., beyond 4k tokens). We implement summarization as an independent component that  
 820 is shared across different base models. Although we explored reusing either the low-level or high-level  
 821 policy for this task, we found that doing so interferes with their primary roles. In contrast, training  
 822 a separate lightweight model for summarization is both simpler and more reliable. Specifically,  
 823 we use Qwen2.5-0.5B-Instruct, optimized with AdamW using a cosine learning-rate schedule with  
 824 linear warmup and a peak learning rate of  $1 \times 10^{-5}$ . The prompt template for summarization is  
 825 provided in Section D, and the summary length distribution is shown in Figure 7. Note that we only  
 826 apply full-parameter SFT to the summarization model. During the online Step-DPO procedure, this  
 827 summarizer remains frozen, which we found to be sufficient in practice and contributes to more stable  
 828 overall training.

829 **Supervised fine-tuning.** We first fine-tune the base LM on low-level trajectories and then freeze  
 830 it, attaching a parameter-efficient LoRA adapter for high-level planning. The intuition behind this  
 831 design is twofold. Low-level trajectories are long, fine-grained, and cognitively harder, so they benefit  
 832 from full-parameter capacity. In contrast, high-level trajectories are short and abstract, making them  
 833 well-suited to lightweight LoRA tuning while avoiding interference with the executor. This training  
 834 order also reduces covariate shift: the planner is learned on top of the well-trained executor it is  
 835 intended to guide. From an optimization perspective, LoRA benefits from a stronger backbone (after  
 836 low-level SFT) and avoids overfitting by learning only a small number of parameters on high-level  
 837 data. Operationally, the approach is efficient at deployment time because it requires only a single  
 838 base model plus a small LoRA adapter (less than 2% additional parameters).

839 **Hyperparameters.** We train each base model on 80K multi-level examples using AdamW with  
 840 the same cosine schedule and warmup strategy as Step-DPO. We use a batch size of 256, a peak  
 841 learning rate of  $2 \times 10^{-5}$ , and truncate sequences to 8,192 tokens. Training is run for 3 epochs.  
 842 Unless otherwise noted, all models are trained with the AdamW optimizer using a cosine learning-  
 843 rate schedule with linear warmup (5% of total steps). For the base LM (low-level policy), we  
 844 use a peak learning rate of  $2 \times 10^{-5}$ , while the high-level LoRA module ( $r = 16$ ,  $\alpha = 32$ ,  
 845 `target_modules=[q_proj, k_proj, v_proj, o_proj]`, no bias) is trained with a higher rate of  $1 \times 10^{-4}$   
 846 (`dropout=0.1`) to allow faster adaptation. We additionally verified that adding MLP projections  
 847 (`up_proj`, `down_proj`, `gate_proj`) yields only marginal gains while substantially increasing the number  
 848 of trainable parameters.

849 **Ablation study.** To further validate our design choices, we compare against two alternative training  
 850 strategies for the high-level policy, while keeping the low-level training unchanged. This is important  
 851 because low-level modeling requires full-parameter updates due to its longer and more complex  
 852 reasoning trajectories; LoRA is insufficient for this component. We consider:

853 (i) LoRA on the original (non-SFT) base model: We directly apply LoRA tuning on the unfine-tuned  
 854 Qwen-2.5-1.5B base model using only high-level trajectories.

855

- 856 • Base: Qwen-2.5-1.5B
- 857 • LoRA:  $r = 16$ ,  $\alpha = 32$ , `target_modules=[q_proj, k_proj, v_proj, o_proj]`, no bias.
- 858 • Optimization: AdamW with a cosine learning-rate schedule and linear warmup, a peak  
 859 learning rate of  $1 \times 10^{-4}$  and a LoRA dropout of 0.1.

861 (ii) Full-parameter SFT on high-level trajectories: We train a separate base model using full SFT on  
 862 only high-level trajectories.

863

- 864 • Base: Qwen-2.5-1.5B

864     • Optimization: AdamW with a cosine learning-rate schedule and linear warmup, a peak  
 865     learning rate of  $1 \times 10^{-5}$ .  
 866

867 We evaluate both variants on MATH500 and AIME24. Table 2 summarizes the results. Our default  
 868 configuration (full SFT on low-level trajectories followed by LoRA tuning on high-level abstractions)  
 869 achieves the highest accuracy, particularly on the harder AIME tasks that require deeper multi-step  
 870 planning. We also observe that applying LoRA on top of the SFT-enhanced base model substantially  
 871 eases optimization and mitigates the overfitting issues that arise when fully fine-tuning a separate  
 872 base model using only high-level trajectories.

873 **Monte Carlo rollout analysis** We analyze Monte Carlo rollout behavior using R1-Distill-LLaMA-  
 874 8B and R1-Distill-Qwen-1.5B. Hidden CoTs are segmented into steps using  $\backslash n \backslash n$ . Estimation  
 875 accuracy measures the fraction of prefixes for which rollouts correctly determine whether the prefix  
 876 can still lead to a correct final solution.

877 For each partial trajectory, we assign a ground-truth survival label  $y \in \{0, 1\}$  using extensive Monte  
 878 Carlo lookahead with the base model:  $y = 1$  if at least one rollout from the prefix reaches a correct  
 879 final answer (the prefix is survivable), and  $y = 0$  otherwise.  
 880

881 Using the fast rollout model, we draw  $K$  continuations from each prefix and compute the estimated  
 882 survivability

$$\hat{g}_K = \frac{1}{K} \sum_{k=1}^K \mathbf{1}_{\text{correct}}^{(k)}.$$

883 We then predict a survival label  
 884  
 885

$$\hat{y}_K = \begin{cases} 1, & \hat{g}_K > 0, \\ 0, & \text{otherwise.} \end{cases}$$

886 A prediction is correct when  $\hat{y}_K = y$ . Estimated survival accuracy for a given  $K$  is the proportion of  
 887 prefixes where this prediction matches ground truth.  
 888

889 We present the results in Figure 6. R1-Distill-LLaMA-8B and R1-Distill-Qwen-1.5B exhibit highly  
 890 similar accuracy across prefix steps. As expected, runtime scales with model size: R1-Distill-LLaMA-  
 891 8B is substantially slower than R1-Distill-Qwen-1.5B. All measurements are obtained on a single  
 892 A100 GPU using vLLM.  
 893

894 **Online Step-DPO.** The reference model for KL  
 895 regularization in Step-DPO is the corresponding  
 896 SFT checkpoint. We perform 4 online training  
 897 rounds, with approximately 3K prompts per round,  
 898 as described in the main text. For each prompt,  
 899 we sample  $N = 4$  candidate continuations using  
 900 temperature 0.7 and top- $p = 0.9$ , and truncate con-  
 901 tinuations at 8,192 tokens to match the SFT con-  
 902 text length and keep attention computation within  
 903 our memory budget. Step-wise preference pairs  
 904 are constructed from these candidates and stored  
 905 in a replay buffer of size 50K; once the buffer is  
 906 full, older entries are discarded in FIFO order. We  
 907 optimize the policy with AdamW (learning rate  
 908  $1 \times 10^{-5}$ , weight decay 0.1,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.95$ ),  
 909 a global batch size of 256 preference pairs, gra-  
 910 dient clipping with a maximum norm of 1.0, DPO  
 911 temperature  $\beta = 0.1$ , and KL coefficient  $\lambda_{\text{KL}} = 0.02$ . At each round, we perform one epoch of  
 912 updates over the current replay buffer. During optimization, we maintain a low-level policy (the  
 913 base LM) and a high-level policy (the LoRA adapter). For low-level preference pairs, we disable the  
 914 LoRA adapter and update only the base LM parameters; for high-level pairs, we freeze the base LM  
 915 and update only the LoRA parameters. Mini-batches of low- and high-level examples are interleaved  
 916 within each round, so that the executor and planner are optimized jointly while remaining modular.  
 917 To improve sample efficiency, we apply a dynamic dropout strategy that filters “easy” prefixes, i.e.,

---

**Algorithm 1: Multi-Level Inference**

---

1 **Inputs:** query  $q$ , high-level policy  $\pi_{\theta_H}$ ,  
 low-level policy  $\pi_{\theta_L}$ , summarizer  $\pi_{\theta_S}$ ;  
 2 **Hyperparameter:** max steps  $M$ ;  
 3  $m \leftarrow 1$ ;  
 4 **while**  $m < M$  **do**  
 5      $d^{(m)} \sim \pi_{\theta_H}(d \mid q, d^{(1:m-1)}, c^{(1:m-1)})$ ;  
 6      $c^{(m)} \sim \pi_{\theta_L}(c \mid q, d^{(1:m)}, c^{(1:m-1)})$ ;  
 7      $c'^{(m)} \leftarrow \pi_{\theta_S}(d^{(m)}, c^{(m)})$ ;  
 8     **if** StopCriterion( $d^{(m)}, c^{(m)}$ ) **then**  
 9         **break**;  
 10      $m \leftarrow m + 1$ ;  
 11 **return**  $(d^{(1:m)}, c^{(1:m)})$ ;

---

918

**Algorithm 2:** Online Step-DPO

---

919

1 **Inputs:** Low-level policy  $\pi_{\theta_L}$ , high-level policy  $\pi_{\theta_H}$ ; Reference models  $\pi_{\text{ref}}^L, \pi_{\text{ref}}^H$ ; Fast rollout policy  
920  $\pi_{\text{roll}}$ ; RL prompts  $\mathcal{D}_{\text{RL}}$ .

921 2 **Hyperparams:** rounds  $T$ , prompts per round  $N$ , sample steps per prompt  $M_s$ , rollout count  $K$ ,  
922 epochs  $E$ .

923 3 **for**  $t = 1$  **to**  $T$  **do**

924   4    Sample prompts  $\{q_i\}_{i=1}^N \subset \mathcal{D}_{\text{RL}}$ ;

925   5    Initialize buffers  $\mathcal{D}_{\text{pref-L}}^{(t)} \leftarrow \emptyset, \mathcal{D}_{\text{pref-H}}^{(t)} \leftarrow \emptyset$ ;

926   6    **foreach**  $q$  **do**

927   7      ( $\text{prefix}_H^{(t)}, \text{prefix}_L^{(t)}$ )  $\leftarrow \text{GENERATEPREFIXES}(\pi_{\theta_H}^{(t)}, \pi_{\theta_L}^{(t)}, q)$ ;

928   8      Randomly select a subset of steps  $\mathcal{M}$  (size  $M_s$ ) for evaluation;

929   9      **foreach**  $m \in \mathcal{M}$  **do**

930   10         $\mathcal{D}_{\text{pref-L}}^{(t,m)} \leftarrow \text{COLLECTPAIR}(\pi_{\theta_L}^{(t)}, \text{prefix}_L^{(t)}[m])$ ;

931   11         $\mathcal{D}_{\text{pref-H}}^{(t,m)} \leftarrow \text{COLLECTPAIR}(\pi_{\theta_H}^{(t)}, \text{prefix}_H^{(t)}[m])$ ;

932   12         $\mathcal{D}_{\text{pref-L}}^{(t)} \leftarrow \mathcal{D}_{\text{pref-L}}^{(t)} \cup \mathcal{D}_{\text{pref-L}}^{(t,m)}$ ;

933   13         $\mathcal{D}_{\text{pref-H}}^{(t)} \leftarrow \mathcal{D}_{\text{pref-H}}^{(t)} \cup \mathcal{D}_{\text{pref-H}}^{(t,m)}$ ;

934   14      **if**  $t > 1$  **then**

935   15         $\pi_{\text{ref}}^L \leftarrow \pi_{\theta_L}^{(t-1)}, \pi_{\text{ref}}^H \leftarrow \pi_{\theta_H}^{(t-1)}$ ;

936   16      **for**  $e = 1$  **to**  $E$  **do**

937   17         $\text{STEPDPOUPDATE}(\pi_{\theta_L}^{(t)}, \pi_{\text{ref}}^L, \mathcal{D}_{\text{pref-L}}^{(t)})$ ;

938   18         $\text{STEPDPOUPDATE}(\pi_{\theta_H}^{(t)}, \pi_{\text{ref}}^H, \mathcal{D}_{\text{pref-H}}^{(t)})$ ;

939   19   **return**  $\pi_{\theta_L}^{(T)}, \pi_{\theta_H}^{(T)}$ ;

940

---

942

943

prefixes for which all candidates induce the same utility; the dropout rate increases linearly from 0.1 to 0.9 over training. All experiments are conducted on  $4 \times$  A100 GPUs (80GB) with `b16` precision.

944

945

946

947

**Step-DPO update schemes.** We compare the proposed update scheme against cheaper alternatives under a matched online training budget (same number of prompts, candidates, and optimization steps). In the *planner-only* variant, we freeze the SFT base LM and apply Step-DPO updates only to the high-level LoRA adapter for all preference pairs, thereby testing whether adapting the planner alone is sufficient once the executor has been trained. In a *round-based* variant, we first run Step-DPO for two rounds updating only the low-level policy (LoRA disabled), and then for two rounds updating only the high-level LoRA (base LM frozen), mirroring a coarse low-then-high schedule in the online phase. Empirically, our joint modular scheme, which interleaves low-level and high-level updates while restricting each preference type to its corresponding module, achieves the best overall performance on MATH500 and AIME24, suggesting that simultaneously refining the executor and planner, while keeping their parameter updates disentangled, is more effective than tuning either component in isolation.

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

**DPO baseline.** To isolate the effect of step-wise supervision, we train a standard outcome-level DPO baseline on the same online prompt pool and with the same rollout configuration as Step-DPO. The reference model for KL regularization is the corresponding SFT checkpoint, and we run 4 online training rounds with approximately 3K prompts per round. For each prompt, we sample  $N = 4$  candidate continuations using temperature 0.7 and top- $p = 0.9$ , truncating each continuation at 8,192 tokens to match the SFT context length. Preference pairs are constructed at the trajectory level: we assign each candidate a scalar utility based on its final solution correctness and form DPO pairs from these outcome-level utilities, ignoring intermediate prefixes. The resulting preference pairs are stored in a replay buffer of size 50K with FIFO eviction, and we perform one epoch of DPO updates over the buffer per round. We optimize a single policy (no hierarchical separation) with AdamW (learning rate  $1 \times 10^{-5}$ , weight decay 0.1,  $\beta_1 = 0.9, \beta_2 = 0.95$ ), using a global batch size of 256 preference pairs, gradient clipping with a maximum norm of 1.0, DPO temperature  $\beta = 0.1$ , and KL coefficient  $\lambda_{\text{KL}} = 0.02$ . All experiments are conducted on  $4 \times$  A100 GPUs (80GB) with `b16` precision under a matched online training budget to Step-DPO.

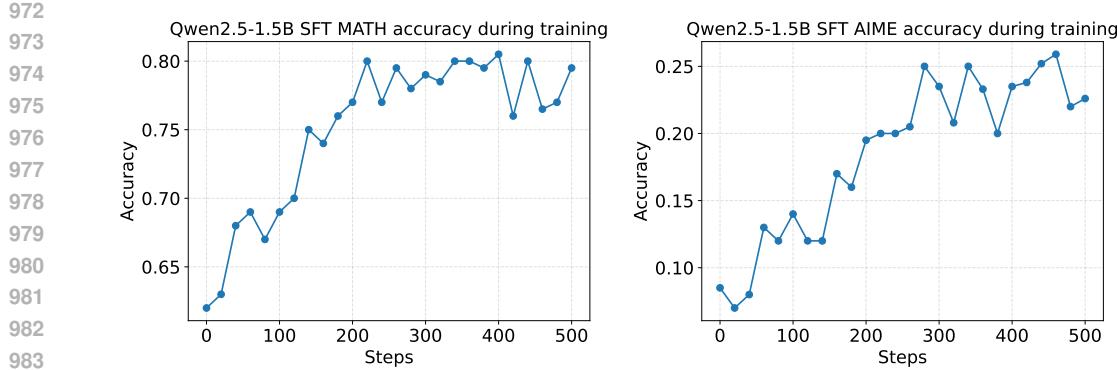


Figure 13: Validation accuracy of Qwen2.5-1.5B SFT during GRPO training. For each question, we sample 8 responses and calculate the overall average accuracy to ensure a stable evaluation.

**GRPO baseline.** To compare MLR with a standard single-policy preference-optimization method, we train a GRPO baseline on the same prompt pool as Step-DPO. We first construct a *single-policy* SFT checkpoint by fine-tuning Qwen-2.5-1.5B on the processed low-level trajectories in which all step descriptors are removed from both inputs and targets, using the same optimizer, schedule, and token budget as our low-level SFT. Starting from this checkpoint, we apply full-parameter GRPO, keeping a frozen copy of the SFT model as the reference policy. We implement the baseline using the Verl (Sheng et al., 2024) framework and vLLM (Kwon et al., 2023) as the rollout backend. The actor and reference are both initialized from the same SFT checkpoint. For each prompt, we sample groups of  $N = 4$  candidate continuations with temperature 0.7 and top- $p = 0.9$ , cap the maximum response length at 4,096 tokens to respect GPU memory limits, and assign a rule-based outcome reward of 1 if the final answer is correct and 0 otherwise. We optimize the actor with AdamW (learning rate  $5 \times 10^{-7}$ , weight decay 0.1) under a KL-penalty objective with coefficient  $\lambda_{\text{KL}} = 0.02$ , using `b1f16` precision, gradient checkpointing, and FlashAttention (Dao, 2023) on  $4 \times$  A100 GPUs (80GB). We train for 4 epochs, using a global batch size of 32 (PPO mini-batch sizes 16, micro-batch sizes 2, respectively), and evaluate every 100 steps on the held-out validation split, selecting the checkpoint with the best validation Pass@1.

We visualize the validation accuracy of Qwen2.5-1.5B SFT during GRPO training in Figure 13. For each question, we sample 8 responses and report the average accuracy to obtain a stable estimate. The evolution of the average response length during GRPO is shown in Figure 14, and the final evaluation results are summarized in Table 1. Compared with our strategy, GRPO is less efficient for long-horizon reasoning: outcome rewards are (i) *sparse*: for long trajectories, a single scalar signal is often insufficient to localize errors; and (ii) *computationally expensive*: generating full rollouts requires substantial memory and compute. When starting from fine-tuned models with long CoTs,

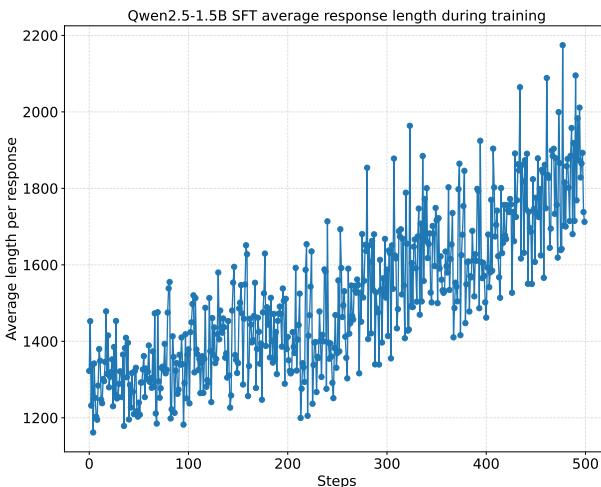


Figure 14: The average response length of Qwen2.5-1.5B SFT on the training set during GRPO.

1026 these costs limit our ability to apply GRPO to larger base models. In contrast, our online Step-DPO  
 1027 procedure is easier to implement and control, and provides a more practical alternative for finetuning  
 1028 long-horizon reasoning policies.

1029 **Qwen-2.5-MATH-7B.** We repeat the above protocol on a 7B model. We first obtain a *single-policy*  
 1030 SFT checkpoint by fine-tuning Qwen-2.5-MATH-7B on the same processed low-level trajectories  
 1031 (step descriptors removed from inputs and targets). Starting from this checkpoint, we run full-  
 1032 parameter GRPO with a frozen copy of the SFT model as the reference. Training uses FSDP sharding,  
 1033 `bf16` precision, activation checkpointing, and FlashAttention on  $8 \times$ A100 (80GB), with a global  
 1034 batch size of 32 implemented as a micro-batch of 1 per GPU and 4 gradient-accumulation steps.  
 1035 Rollouts are generated with vLLM: for each prompt we sample  $N=4$  candidates (temperature 0.7,  
 1036  $\text{top-}p=0.9$ ) and cap the maximum response length at 4,096 tokens. We optimize with AdamW  
 1037 (learning rate  $5 \times 10^{-7}$ , weight decay 0.1) under a KL penalty with coefficient  $\lambda_{\text{KL}}=0.02$ , and  
 1038 evaluate every 100 steps, selecting the checkpoint with the best validation Pass@1.

1039 **Llama-3.1-8B.** We follow the same procedure with Llama-3.1-8B. A *single-policy* SFT checkpoint is  
 1040 first obtained on the same low-level trajectories, after which we apply full-parameter GRPO using  
 1041 a frozen reference initialized from the SFT checkpoint. We train with FSDP, `bf16`, activation  
 1042 checkpointing, and FlashAttention on  $8 \times$ A100 (80GB), using a global batch size of 32 realized  
 1043 as a micro-batch of 1 per GPU and 4 gradient-accumulation steps. Rollouts use vLLM with  $N=4$   
 1044 candidates per prompt (temperature 0.7,  $\text{top-}p=0.9$ ) and a 4,096-token cap. The optimizer, KL  
 1045 objective, evaluation cadence, and model selection criteria are identical to the 7B setting. For  
 1046 additional memory headroom, the frozen reference is sharded; when necessary, we load the reference  
 1047 in 8-bit for forward-only KL without changing any other hyperparameters.

1048 **Plan-and-Solve baseline.** We compare against Plan-and-Solve (Wang et al., 2023a), which first  
 1049 proposes a concise, global plan and then executes the solution conditioned on that plan. The  
 1050 example prompt is given in Section D. For a fair comparison, we use the same backbone as our  
 1051 method and fine-tune two LoRA heads on top of it: a *planner* (Problem  $\rightarrow$  Plan) and an *executor*  
 1052 (Problem + Plan  $\rightarrow$  Solution). At inference we follow the standard two-pass Plan-and-Solve pipeline:  
 1053 Pass-1 generates the plan; Pass-2 solves the problem conditioned on that plan.

1054 **Training data creation.** Using the same training set as our method, we prompt a strong teacher  
 1055 model (DeepSeek-V3.2) to produce corresponding trajectories. We filter trajectories by final-answer  
 1056 correctness and basic format checks. We match the total number of accepted trajectories to our  
 1057 method (80K) to ensure a fair comparison.

1058 **Training configuration.** Unless otherwise noted, we freeze the backbone and train LoRA adapters  
 1059 with identical hyperparameters for planner and executor.

- 1060 • Backbone: Qwen-2.5-1.5B. LoRA:  $r = 16$ ,  $\alpha = 32$ , `target_modules=[q_proj, k_proj, v_proj,`  
 1061 `o_proj]`, no bias. Optimization: AdamW, cosine decay with 3% warm-up, learning rate  
 $1 \times 10^{-4}$ .
- 1062 • Backbone: Qwen-2.5-MATH-7B. Same LoRA configuration. Same optimization configura-  
 1063 tion except for learning rate  $5 \times 10^{-5}$ .
- 1064 • Backbone: Llama-3.1-8B. Same LoRA configuration. Same optimization configura-  
 1065 tion except for learning rate  $5 \times 10^{-5}$ .

1066 **Results.** Table 1 summarizes performance. Because Plan-and-Solve here is trained only with SFT,  
 1067 we compare it against MLR (SFT-only). Across all three backbones, our method outperforms Plan-  
 1068 and-Solve, with the largest margins on the harder benchmarks (AIME, GPQA). We observe that  
 1069 Plan-and-Solve often implicitly assumes all subtasks succeed as initially planned; errors in early steps  
 1070 can propagate, and the executor may partially deviate from the plan. In contrast, our approach learns  
 1071 a better planner that can adapt its plans based on execution signals, enabling revisions rather than  
 1072 committing to a fixed blueprint. This adaptive coupling between planner and executor yields more  
 1073 stable long-horizon reasoning than prompting a plan upfront and executing it verbatim.

1074 **Evaluation.** During evaluation, we use greedy decoding for both the base model and the instruction  
 1075 fine-tuned model to produce more coherent and consistent CoTs. For all other baselines and our  
 1076 method, we follow the decoding protocol in Guo et al. (2025), using sampling-based decoding

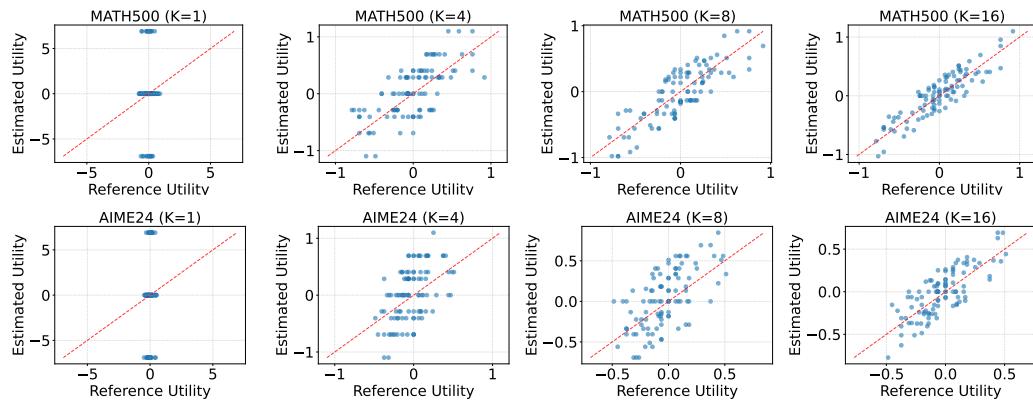
1080 with a temperature of 0.6 and a top- $p$  value of 0.95 to generate 8 responses per prompt to reduce  
 1081 variance and repetition. For MLR, we employ a single base LM for both levels and switch the  
 1082 high-level LoRA adapter on or off depending on the generation stage (Algorithm 1). Specifically,  
 1083 we enable the high-level LoRA adapter to produce step descriptors (planning), and then disable the  
 1084 adapter to generate the corresponding low-level trajectories conditioned on these descriptors. The  
 1085 maximum generation length for all models is set to 16,384 tokens. Performance is measured using  
 1086  $\text{Pass}@1 = \frac{1}{k} \sum_{i=1}^k p_i$ , where  $p_i$  denotes the correctness of the  $i$ -th response. For AIME24, we also  
 1087 report consensus accuracy over 32 samples, denoted as  $\text{cons}@32$ .

## D ADDITIONAL RESULTS

1091 **Examples of MLR outputs.** In this section, we present additional results to further demonstrate and  
 1092 analyze the effectiveness of our method. We showcase representative output examples generated by  
 1093 MLR across different datasets (Section D). Each sample consists of a two-level reasoning trajectory,  
 1094 comprising shared reasoning steps annotated with both a step descriptor and corresponding step  
 1095 content. In the high-level trajectory, the step descriptor is generated by the high-level module, while  
 1096 the step content is produced by the compressor, which takes the low-level content as input and outputs  
 1097 a concise abstraction. In the low-level trajectory, the step descriptor is provided by the high-level  
 1098 module, and the step content is directly generated by the low-level base model.

1099 **Error analysis.** To better understand the strengths and limitations of our framework, we conduct  
 1100 detailed error analysis. To further enhance verification and error localization, we incorporate auxiliary  
 1101 models (OpenAI’s o1 and o1-mini) to assist in identifying potential reasoning flaws. Specifically,  
 1102 we first evaluate whether the auxiliary model can independently solve each task without access to  
 1103 the ground-truth final answer or reference solution. If the auxiliary model successfully produces the  
 1104 correct solution, we then use it to help analyze erroneous trajectories generated by our framework.  
 1105 The error analysis provided by the auxiliary model is subsequently reviewed and confirmed by human  
 1106 evaluators. Through this process, we identify several recurring error patterns: 1) High-level step  
 1107 descriptor errors: redundant branching (multiple step descriptors that pursue the same subtask),  
 1108 unclosed loops (steps are never marked as “complete,” leading to repeated revisit), dead-end  
 1109 retention (contradicted or unproductive exploratory branches are retained), copy-pasted fallback  
 1110 (guessed answers are repeated verbatim under different step descriptors). 2) Low-level step content  
 1111 errors: logical misapplication (misuse of domain-specific rules or principles), contradiction tolerance  
 1112 (inconsistent constraints are not resolved), repetitive reasoning (redundant inference chains without  
 1113 new contributions), failure to propagate known facts (previously inferred information is ignored in  
 1114 later steps), looping filler (verbose or stalled reasoning with redundant rephrasing).

1115 **Parameter studies on rollout count** In our online step-DPO, the rollout count  $K$  directly affects  
 1116 the quality of the preference pairs. We first examine how  $K$  influences the reliability of the utility



1132 **Figure 15: Effect of rollout count on the reliability of utility estimates.** We compare utilities estimated by the  
 1133 1.5B model under  $K$  rollouts to reference utilities by the 8B model using 16 rollouts for 100 sampled prefixes  
 from MATH500 and AIME24, respectively.

estimates defined in Equation (6). To do so, we compare utilities estimated by the 1.5B model under various  $K$  to reference utilities produced by the 8B model with 16 rollouts, using 100 sampled prefixes from MATH500 and AIME24, respectively (Figure 15). As expected, larger  $K$  reduces estimator variance but increases computational cost. To control this overhead, we apply a margin threshold  $\delta$  when selecting preference pairs which allows us to use smaller  $K$  while maintaining reliability of the preference data. Next, we evaluate the agreement of selected preference pairs as a function of  $K$ , defined as the fraction of pairs whose preference direction agrees with the base model using 16 rollouts (Figure 8). For each  $K$ , we generate 100 preference pairs following Equation (7). We then recompute the reference utilities of both options and check whether the chosen response has higher reference utility than its alternative. Pairs that satisfy this condition are counted as agreed, and we report the average agreement for each  $K$ . We consider both the 1.5B model and the 8B base model as rollout policy. In practice, we use the base model as the rollout policy for AIME24 and GPQA. Finally, we study model performance across training stages under different rollout counts (Figure 12). Starting from the same SFT model, we generate the same number of preference pairs for each  $K$  and all train for 4 epochs. We report performance on MATH500 and AIME24 throughout training. Overall, our setting achieves comparable final accuracy while significantly reducing computational cost.

**Ablations on hierarchical levels** To investigate the role of different levels, we conduct an ablation study on the hierarchical structure. We consider two variants: (1) High-level only: the high-level module directly predicts summaries without invoking the low-level module; (2) Low-level only: the low-level module is required to predict both the high-level step descriptions and the detailed reasoning without guidance from the high-level module. The evaluation protocol matches our main setting, and the results are reported in Table 2. Our full method consistently outperforms both variants, especially on the challenging AIME24 dataset. The high-level-only variant underperforms because the planner lacks grounded execution learning, making direct summary prediction unreliable for difficult reasoning tasks. We show an erroneous example in Section D. The low-level-only variant is weaker because the absence of explicit high-level guidance causes the low-level module to drift and accumulate errors as the trajectory grows longer. Overall, these results demonstrate that our two-level design yields better performance on long-horizon reasoning tasks.

**Long-horizon reasoning test.** To further evaluate our method on long-horizon reasoning, we simulate a multi-question setting by concatenating multiple problems into a single prompt. We

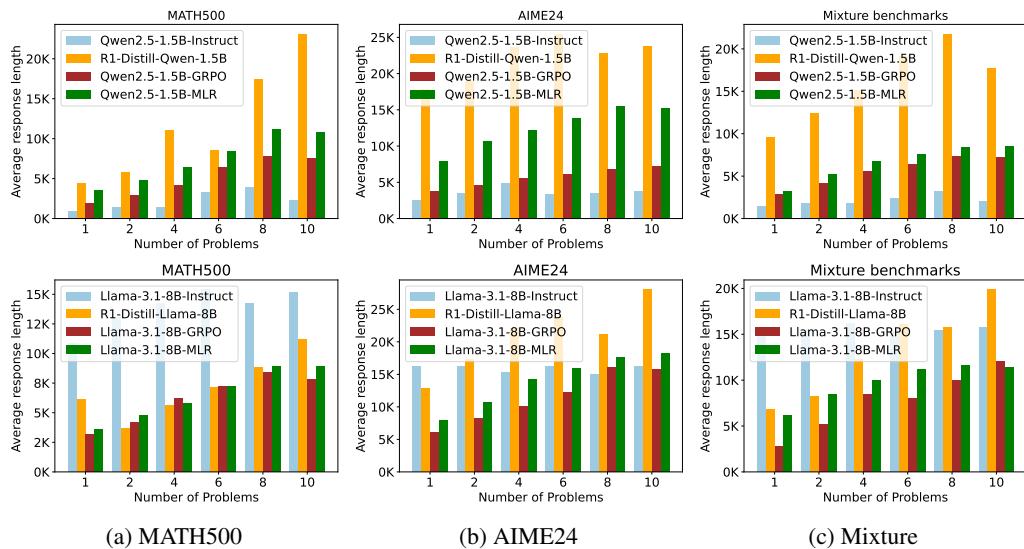


Figure 16: Long-horizon reasoning response length on (a) MATH500, (b) AIME24, and (c) Mixture benchmarks (MATH500, AIME24, GPQA, BoardGameQA). We simulate long-horizon reasoning by concatenating multiple problems in the prompt and report average accuracy. Compared with R1-Distill models, which are prone to getting lost in long trajectories and entering dead loops, MLR benefits from explicit planning and is able to solve all problems more effectively.

1188 consider three configurations: (i) multiple problems from MATH500, (ii) multiple problems from  
1189 AIME24, and (iii) mixed batches drawn from MATH500, AIME24, GPQA, and BoardGameQA.  
1190 For each configuration, we randomly select 100 problems from the corresponding benchmarks and  
1191 treat their questions as a mini-batch. The batch size varies from 1 to 10; for each size, we present  
1192 all questions in the batch in one prompt and ask the LLM to solve them jointly. To mitigate cases  
1193 where the model gets stuck on a single problem, we explicitly instruct it to skip a question if it  
1194 cannot make progress. Prompt template can be found in Section D. For instruction-tuned models,  
1195 we use greedy decoding. For the other baselines, for every mini-batch we sample 8 responses with  
1196 temperature 0.6 and top\_p = 0.95, using a maximum context length of 32K tokens, to reduce variance  
1197 and then report accuracy averaged over the 100 problems. Note that all methods are trained only in  
1198 the single-question setting, so this constitutes a fully out-of-distribution evaluation. Figure 1 shows  
1199 the results: MLR consistently degrades more slowly than all baselines across datasets and base  
1200 models, highlighting the benefits of explicit planning for long-horizon reasoning. We further report  
1201 the corresponding length distributions for different scenarios in Figure 16.  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241

1242 Example of Long-horizon Plan Failures (DeepSeek-R1-Distill-Llama-8B on MATH500)  
 1243  
 1244 **Prompt:**  
 1245 What is the least positive integer multiple of 30 that can be written with only the digits 0 and 2?  
 1246  
 1247 **Response:**  
 1248 <think>  
 1249 Okay, so I need to find the smallest positive integer multiple of 30 that can be written using only the digits 0 and 2. Hmm, let me think about how to approach this.  
 1250 First, I remember that a multiple of 30 must be divisible by both 3 and 10. So, let's break down what that means.  
 1251  
 1252 Divisibility by 10: For a number to be divisible by 10, it must end with a 0. That makes sense because any number ending in 0 is a multiple of 10. So, whatever number I come up with must end with a 0. That means the last digit is fixed as 0.  
 1253  
 1254 Divisibility by 3: For a number to be divisible by 3, the sum of its digits must be a multiple of 3. Since the number can only have digits 0 and 2, the sum of the digits will just be the number of 2s multiplied by 2. So, the total sum is 2 times the number of 2s. This sum needs to be divisible by 3.  
 1255  
 1256 So, putting that together, I need a number that consists of one or more 2s followed by a 0. The number of 2s should be such that the total sum (which is 2 times the number of 2s) is divisible by 3.  
 1257  
 1258  
 1259 Let me formalize that. Let's say the number has 'n' number of 2s followed by a 0. So, the number would be 222...20, with 'n' 2s and ending with a 0.  
 1260  
 1261  
 1262 The sum of the digits is  $2n$ , and we need  $2n$  to be a multiple of 3. So,  $2n \equiv 0 \pmod{3}$ . Which means  $n$  must be a multiple of 3 because 2 and 3 are coprime (they have no common divisors other than 1). So,  $n$  needs to be divisible by 3.  
 1263  
 1264 Therefore, the smallest number of 2s needed is 3, right? Because 3 is the smallest positive integer divisible by 3.  
 1265  
 1266  
 1267 So, if  $n = 3$ , the number would be 2220. Let me check that.  
 1268  
 1269  
 1270 2220 is a multiple of 30? Let's see. 2220 divided by 30 is 74, which is an integer. So, yes, 2220 is a multiple of 30.  
 1271  
 1272  
 1273 But is it the smallest?  
 1274  
 1275 Wait, maybe I can find a smaller number with fewer digits? Let me think. The number must end with a 0, so the last digit is fixed. The other digits can be 0 or 2. But if I try to make a smaller number, maybe with fewer digits, but still only 0s and 2s.  
 1276  
 1277  
 1278 Let me consider numbers of different lengths and see if they can satisfy the conditions.  
 1279  
 1280  
 1281 Starting with the smallest possible number: 1 digit. It would have to be 0, but 0 is not positive. So, no good.  
 1282  
 1283 2 digits: The number would be 20. Is 20 a multiple of 30? 20 divided by 30 is less than 1, so no. Also, 20 is not a multiple of 30.  
 1284  
 1285 3 digits: Let's see. The possible numbers are 200, 220, and 222.  
 1286  
 1287 200: 200 divided by 30 is approximately 6.666, not an integer. 220 divided by 30 is about 7.333, not integer. 222 divided by 30 is 7.4, which isn't integer either. So, none of these are multiples of 30.  
 1288  
 1289 4 digits: Let's list possible numbers. They end with 0, so the last digit is 0, and the first three digits are either 0 or 2.  
 1290  
 1291 Possible numbers: 2000, 2200, 2220, 2222, 2020, 2022, 2202, 2220, 2222, etc. Wait, but actually, the first three digits can be any combination of 0s and 2s. So, all numbers of the form 2220, 2202, 2022, 0222, but 0222 is not a valid number since it starts with 0. So, the numbers are 2220, 2202, 2022.  
 1292  
 1293  
 1294  
 1295 Check each:

1296  
 1297 2220: As before,  $2220 / 30 = 74$ , which is integer. So, that's a multiple.  
 1298  
 1299 2202:  $2202 / 30$  is 73.4, which isn't integer.  
 1300  
 1301 2022:  $2022 / 30$  is 67.4, which isn't integer.  
 1302  
 1303 So, the only 4-digit number that works is 2220.  
 1304  
 1305 But earlier, with 3 digits, we had 2220 as well. Wait, so 2220 is a 4-digit number, but 3-digit numbers didn't work. So, 2220 is the first one.  
 1306  
 1307 Wait, but hold on. Maybe there's a 3-digit number that's a multiple of 30, but I haven't considered all possibilities. Let me check again.  
 1308  
 1309 Wait, 2220 is a 4-digit number, but what about a 3-digit number? 222 is 222. 222 divided by 30 is 7.4, which isn't integer. 2220 is 74. So, maybe 2220 is the first one.  
 1310  
 1311 Wait, but hold on. Maybe a 3-digit number with two 2s and a 0. Let's see: 220, 202, 022. But 022 isn't a 3-digit number, it's just 22. So, 220 is 220, which is  $220 / 30 \approx 7.333$ , not integer. 202 is  $202 / 30 \approx 6.733$ , also not integer. So, no luck.  
 1312  
 1313  
 1314 So, in the 3-digit numbers, there's no multiple of 30 that can be formed with 0s and 2s. So, moving on to 4-digit numbers.  
 1315  
 1316  
 1317 Wait, but 2220 is a 4-digit number, but is that the smallest? Wait, let me think. Is there a smaller number with 4 digits, but starting with 2, and having 2s and 0s, but not necessarily 3 2s? Wait, no, because if you have 2 2s, then 220, 202, 022. None of these are divisible by 30.  
 1318  
 1319  
 1320 So, the only 4-digit number is 2220, which is 2220. But wait, maybe a 5-digit number? No, 2220 is already a 4-digit number, which is smaller than a 5-digit number. So, 2220 is smaller.  
 1321  
 1322  
 1323 But wait, hold on. Let me consider 2220. It's 2220, but is there a smaller number with 4 digits that can be formed with 0s and 2s? Wait, 2220 is 2220. But what about 2220, 2202, 2022, etc. Wait, none of the other permutations of 2s and 0s in 4 digits are multiples of 30. So, 2220 is the only one.  
 1324  
 1325  
 1326 But wait, is 2220 the minimal? Because 2220 is a 4-digit number, but is there a 3-digit number that's a multiple of 30? Wait, I don't think so, because 2220 is the first multiple of 30 with only 2s and 0s.  
 1327  
 1328  
 1329 Wait, hold on. Let me think differently. Maybe instead of 2220, there's a smaller number.  
 1330  
 1331 Wait, 2220 is 2220, but if I can find a number with fewer digits, like 220, but 220 is not a multiple of 30. 220 divided by 30 is 7.333. Not integer.  
 1332  
 1333 Wait, 202:  $202 / 30$  is 6.733. Not integer. 200:  $200 / 30$  is 6.666. Not integer. 222:  $222 / 30$  is 7.4. Not integer. So, none of these are multiples of 30.  
 1334  
 1335 Wait, so maybe 2220 is the first one. But 2220 is a 4-digit number. Is there a 3-digit number that can be formed with 2s and 0s that is a multiple of 30? Let me think.  
 1336  
 1337  
 1338 Wait, 2220 is the first one. So, 2220 is 2220. But 2220 is 2220, which is 2220. But if I can write a 3-digit number with 2s and 0s that's a multiple of 30, that would be smaller than 2220.  
 1339  
 1340 Wait, 2220 is 4 digits, but 222 is 3 digits. But 222 is not a multiple of 30. So, no.  
 1341  
 1342 Wait, maybe 220 is 220, but that's 220, which is not a multiple of 30. 200 is 200, which is not a multiple of 30. 222 is 222, which is not a multiple of 30. So, yeah, no luck.  
 1343  
 1344  
 1345 So, 2220 is the first one.  
 1346  
 1347 Wait, but hold on. Let me think again. If I can have a number that is 200...0, but with some 2s in between. But the sum of digits must be a multiple of 3, so as we saw earlier, the number of 2s must be a multiple of 3.  
 1348  
 1349 So, with 3 2s, we get 2220, which is 2220. But is there a way to have 2s in other positions? Like 2020, 2002, 2022, etc. But as we saw, those are not multiples of 30.

1350  
 1351 Wait, 2020: sum of digits is  $2 + 0 + 2 + 0 = 4$ , which is not a multiple of 3, so it's not divisible by 3, hence not  
 1352 divisible by 30. Similarly, 2002:  $2 + 0 + 0 + 2 = 4$ , same thing. 2200:  $2 + 2 + 0 + 0 = 4$ , same. 2220:  $2 + 2 + 2$   
 1353 + 0 = 6, which is a multiple of 3. So, 2220 is the first one.  
 1354  
 1355 So, perhaps 2220 is the minimal.  
 1356  
 1357 Wait, but hold on. Let me check if 2220 is indeed the minimal. Maybe a number with 4 digits but starting with  
 1358 2, and with other digits as 2s and 0s, but the number of 2s is more than 3? But no, because 2220 is the first  
 1359 one where the number of 2s is 3, which is the minimal multiple of 3.  
 1360  
 1361 Wait, but maybe 2220 is the minimal, but is there a smaller number? Let me think.  
 1362  
 1363 Wait, 2220 is 2220, which is a 4-digit number. Is there a 3-digit number with 2s and 0s that is a multiple of  
 1364 30? As we saw, 2220 is the first one. So, 2220 is the minimal.  
 1365  
 1366 Wait, but hold on. Maybe I can have a number like 2200, which is 2200, but the sum is 4, which isn't a  
 1367 multiple of 3. So, not divisible by 3. So, no.  
 1368  
 1369 Wait, 2202: sum is  $2 + 2 + 0 + 2 = 6$ , which is a multiple of 3, but 2202 divided by 30 is 73.4, which isn't  
 1370 integer. So, not a multiple.  
 1371  
 1372 Similarly, 2022: sum is  $2 + 0 + 2 + 2 = 6$ , which is a multiple of 3, but  $2022 / 30$  is 67.4, not integer. So, no.  
 1373  
 1374 2002: sum is 4, not multiple of 3.  
 1375  
 1376 Therefore, 2220 is the minimal number.  
 1377  
 1378 Wait, but hold on. Let me think if there's a smaller number.  
 1379  
 1380 Wait, 2220 is 2220, which is a 4-digit number. Is there a 3-digit number that can be formed with 2s and 0s,  
 1381 which is a multiple of 30? As we saw, 2220 is the first one, so no, there's no smaller number.  
 1382  
 1383 Wait, but hold on again. Let me think about 2220. Is it the minimal, or is there a smaller number with 4 digits  
 1384 but starting with 2, and having 2s and 0s, but arranged in a different way?  
 1385  
 1386 Wait, for example, 2020: sum is 4, which is not multiple of 3, so not divisible by 3. 2002: same. 2200: same.  
 1387 So, no.  
 1388  
 1389 So, 2220 is the minimal.  
 1390  
 1391 Wait, but hold on, let me think of another approach. Maybe think of numbers in base 10, with digits only 0  
 1392 and 2, and divisible by 30.  
 1393  
 1394 So, the number must end with 0, and the sum of digits must be a multiple of 3.  
 1395  
 1396 So, as we saw, the number must have a number of 2s equal to 3, 6, 9, etc.  
 1397  
 1398 Therefore, the minimal number is 2220, which is 3 digits.  
 1399  
 1400 Wait, 2220 is 4 digits. So, 2220 is 2220, which is 4 digits, and it's the first one.  
 1401  
 1402 But wait, 2220 is a 4-digit number. Is there a 3-digit number? 2220 is 2220, which is 2220. But 222 is 3  
 1403 digits, but 222 is not a multiple of 30.  
 1404  
 1405 Wait, so 2220 is the minimal.  
 1406  
 1407 Wait, but I just thought of another way. Maybe if we can have a number with a 0 at the end, and two 2s, but  
 1408 also have another 2 somewhere in the middle. But the sum of digits must be a multiple of 3, so  $2 + 2 + 0 = 4$ ,  
 1409 which isn't a multiple of 3. So, that's not good. So, 2220 is the minimal.  
 1410  
 1411 Wait, but 2220 is a 4-digit number, but 2220 is 2220, which is 2220, which is 2220. Hmm, but 2220 is the  
 1412 minimal.

1404  
 1405 Wait, hold on, but is 2220 the minimal? Let me think of 20220, but that's a 5-digit number, which is larger  
 1406 than 2220, so not helpful.  
 1407  
 1408 Wait, so 2220 is 2220, which is 4 digits, and it's the minimal.  
 1409  
 1410 Wait, but hold on, maybe 20220 is a 5-digit number, but 2220 is smaller.  
 1411  
 1412 Wait, so 2220 is 2220, which is 4 digits, and it's smaller than 20220.  
 1413  
 1414 Therefore, 2220 is the minimal.  
 1415  
 1416 Wait, but now, hold on, another thought. Maybe a number with 2220 is 2220, but is there a number with 2220  
 1417 in a different arrangement? For example, 20220 is a 5-digit number, but 2220 is 4 digits, so 2220 is smaller.  
 1418  
 1419 Therefore, 2220 is the minimal.  
 1420  
 1421 Wait, but let me just confirm once again. 2220 is 2220, which is 2220. Let me see if any number with fewer  
 1422 digits can be formed.  
 1423  
 1424 Wait, 2220 is 4 digits. 222 is 3 digits, but 222 is not a multiple of 30. 2220 is 4 digits, and it's a multiple of 30.  
 1425  
 1426 Wait, so 2220 is the minimal.  
 1427  
 1428 Wait, but wait, 20220 is a 5-digit number, which is 20220, which is 2220. But 2220 is smaller.  
 1429  
 1430 Wait, but hold on, 2220 is a 4-digit number, which is 2220, so 2220 is smaller than 20220, which is a 5-digit  
 1431 number.  
 1432  
 1433 Therefore, 2220 is the minimal.  
 1434  
 1435 Wait, but I just thought of another approach. Maybe think of the number as starting with 2, followed by 2s and  
 1436 0s.  
 1437  
 1438 So, for example, 2220, 2202, 2022, 2002, 2220, etc. But as we saw, only 2220 is a multiple of 30.  
 1439  
 1440 Wait, but 2220 is a 4-digit number, which is 2220. 2220 is 2220, which is 2220, so it's a 4-digit number.  
 1441  
 1442 Wait, but let me think of the number 2220. Is that the minimal?  
 1443  
 1444 Wait, 2220 is 2220, which is 2220. So, 2220 is a multiple of 30, but is there a smaller number?  
 1445  
 1446 Wait, 2220 is 2220, which is 2220, which is 2220, which is 2220.  
 1447  
 1448 Wait, but 2220 is 2220, which is 2220, so is that the minimal? Or is there a number with fewer digits?  
 1449  
 1450 Wait, 2220 is 4 digits. 222 is 3 digits. 222 is not a multiple of 30. So, 2220 is the minimal.  
 1451  
 1452 Wait, but wait, another idea: 2220 is 2220, which is 2220, which is 2220, which is 2220.  
 1453  
 1454 Wait, but if I can have a number with 2s and 0s, but with leading 2s and trailing 0s, but in between, perhaps 2s  
 1455 can be placed in different positions.  
 1456  
 1457 Wait, but as we saw, the sum of digits must be a multiple of 3, so the number of 2s must be a multiple of 3.  
 1458  
 1459 Therefore, the minimal number is 2220, which is 2220.  
 1460  
 1461 Wait, so 2220 is 4 digits, and it's the minimal. So, 2220 is the answer.  
 1462  
 1463 Wait, but wait, I just thought of something else. What if the number is 20220? That's 5 digits, but it's 20220,  
 1464 which is 20220. 2220 is 2220, which is smaller. So, 2220 is smaller.  
 1465  
 1466 Wait, but 2220 is 2220, which is a 4-digit number. 20220 is 20220, which is a 5-digit number, so 2220 is  
 1467 smaller.  
 1468

1458 Therefore, 2220 is the minimal.  
 1459  
 1460 Wait, but I'm confused because 2220 is 2220, which is 2220. So, that's a 4-digit number, and it's the minimal.  
 1461  
 1462 Wait, but hold on, let me just confirm if 2220 is indeed the minimal.  
 1463  
 1464 Wait, 2220 is 2220, which is 2220. Let me check if 2220 is a multiple of 30. 2220 divided by 30 is 74, which  
 1465 is an integer. So, yes, it is.  
 1466  
 1467 So, 2220 is the minimal.  
 1468  
 1469 Wait, but hold on, 2220 is 2220, which is 2220, which is a 4-digit number.  
 1470  
 1471 But, is there a smaller number?  
 1472  
 1473 Wait, 2220 is 2220, which is 2220. But, 2220 is 2220, which is 2220. So, it's 2220, which is 2220. So, 2220 is  
 1474 the minimal.  
 1475  
 1476 Wait, but wait, 2220 is 2220, which is 2220, which is a 4-digit number. So, 2220 is 2220, which is 2220.  
 1477  
 1478 Wait, so that's the minimal.  
 1479  
 1480 Wait, but wait, hold on, 2220 is 2220, which is 2220, which is 2220. So, it's 2220.  
 1481  
 1482 Wait, so is 2220 the minimal.  
 1483  
 1484 Wait, but hold on, another thought. Let me think of numbers with 2s and 0s, but in different positions, but the  
 1485 sum of digits must be a multiple of 3.  
 1486  
 1487 So, 2220 is 2220, which is 2220, sum is 6, which is a multiple of 3.  
 1488  
 1489 Wait, but if I have a number like 20220, which is 5 digits, sum is 6, which is multiple of 3, but 20220 is 20220,  
 1490 which is larger than 2220.  
 1491  
 1492 So, 2220 is the minimal.  
 1493  
 1494 Wait, but hold on, another idea. Maybe 20220 is 20220, which is 20220, which is a 5-digit number, which is  
 1495 20220, which is larger than 2220, which is 4-digit.  
 1496  
 1497 So, 2220 is smaller.  
 1498  
 1499 Wait, so 2220 is the minimal.  
 1500  
 1501 Wait, but, wait, let me think of the number 2220. It's 2220, which is 2220, which is 2220. So, 2220 is a 4-  
 1502 digit number.  
 1503  
 1504 But, 2220 is 2220, which is 2220, so that's 2220.  
 1505  
 1506 Wait, so, yeah, 2220 is the minimal.  
 1507  
 1508 Wait, but, hold on, I just thought of another number. Let's say 2220 is 2220, which is 2220. But, 2220 is 2220,  
 1509 which is 2220. So, is there a smaller number?  
 1510  
 1511 Wait, 2220 is 2220, which is 2220, which is 2220. So, 2220 is 2220, which is 2220. So, 2220  
 is 2220, which is 2220.  
 1512  
 1513 Wait, so, that's the minimal.  
 1514  
 1515 Wait, so, conclusion: 2220 is the minimal.  
 1516  
 1517 Wait, but hold on, I just thought of something else. Maybe a number like 2220 is 2220, but 2220 is 2220,  
 1518 which is 2220, which is 2220, which is 2220.  
 1519  
 1520 Wait, so, yeah, 2220 is 2220, which is 2220, which is 2220.

1512 Wait, so, 2220 is 2220, which is 2220, which is 2220.  
 1513  
 1514 Wait, so, yeah, 2220 is 2220, which is 2220.  
 1515  
 1516 Wait, so, that's 2220, which is 2220, which is 2220.  
 1517 Wait, but, hold on, let me think again.  
 1518  
 1519 Wait, 2220 is 2220, which is 2220, which is 2220.  
 1520  
 1521 Wait, but 2220 is 2220, which is 2220.  
 1522 Wait, so, yeah, 2220 is 2220, which is 2220.  
 1523  
 1524 Wait, so, 2220 is 2220, which is 2220.  
 1525  
 1526 Wait, so, that's 2220, which is 2220.  
 1527 Wait, so, yeah, 2220 is 2220.  
 1528  
 1529 ...  
 1530  
 1531

### Data Augmentation Prompt

**Prompt:**

A planet with a mass equivalent to that of two Earths is orbiting in a circular orbit around a star with a mass three times that of the Sun, at a distance of two AU. On which of the following celestial objects does the gravitational force of the planet exceed the centrifugal force (i.e., the gravitational force of the planet dominates)?

- a) An object with a 0.002 Earth masses located at 0.012 AU distance from the planet
- b) An object with a 0.006 Earth masses located at 0.014 AU distance from the planet
- c) An object with a 0.001 Earth masses located at 0.019 AU distance from the planet
- d) An object with a 0.003 Earth masses located at 0.021 AU distance from the planet

Generate a new graduate-level physics questions similar in difficulty and style to this GPQA item, but with different physical setup and different correct answer. Please provide correct answers within `\boxed{}` and explanations as well.

**Response:**

Here is a new graduate-level physics problem modeled in a similar style and complexity as the original GPQA item:

---

**\*\*Question:\*\***

A moon with a mass of 0.3 Earth masses is orbiting a gas giant of 5 Jupiter masses at a circular orbital radius of 0.015 AU. The gas giant itself is orbiting a star of 2.5 solar masses at a distance of 5 AU in a nearly circular orbit. On which of the following small objects located near the moon does the gravitational force due to the moon exceed the tidal/centrifugal force exerted by the planet-star system (i.e., the object is within the moon's gravitational dominance, or its Hill sphere)?

- a) An object with a 0.0005 Earth masses located 0.0009 AU from the moon
- b) An object with a 0.001 Earth masses located 0.0012 AU from the moon
- c) An object with a 0.0003 Earth masses located 0.0016 AU from the moon
- d) An object with a 0.0008 Earth masses located 0.0020 AU from the moon

---

**\*\*Correct Answer:\*\***

`a`

---

1566  
 1567     **\*\*Explanation:\*\***  
 1568  
 1569     To determine whether the gravitational force of the moon dominates over the tidal force of the planet–star  
 1570     system, we compare the object's distance from the moon to the **Hill radius** of the moon.  
 1571     The Hill radius ( $R_H$ ) of the moon in orbit around the gas giant is approximately:  
 1572  
 1573     [  
 1574          $R_H = a \left( \frac{m_{\text{moon}}}{3 M_{\text{planet}}} \right)^{1/3}$   
 1575     ]  
 1576     Where:  
 1577  
 1578     \* ( $a = 0.015$ ,  $\text{AU}$ ) (moon's distance from the planet),  
 1579     \* ( $m_{\text{moon}} = 0.3 M_{\odot}$  =  $0.3 \times 10^{-6} M_{\odot}$  =  $9 \times 10^{-7} M_{\odot}$ )  
 1580     , \* ( $M_{\text{planet}} = 5 M_J$  =  $5 \times 0.001 M_{\odot}$  =  $0.005 M_{\odot}$ ).  
 1582     [  
 1583          $R_H = 0.015 \left( \frac{9 \times 10^{-7}}{3 \times 0.005} \right)^{1/3} = 0.015 \left( \frac{9 \times 10^{-7}}{0.015} \right)^{1/3}$   
 1584     ]  
 1586     [  
 1587          $= 0.015 \left( 6 \times 10^{-5} \right)^{1/3} \approx 0.015 \times 0.039 = 5.85 \times 10^{-4} \text{ AU}$   
 1588     ]  
 1590     So the Hill radius is roughly:  
 1591  
 1592     [  
 1593          $R_H \approx 0.000585 \text{ AU}$   
 1594     ]  
 1595     Any object located significantly **within** this Hill radius is gravitationally bound to the moon.  
 1596  
 1597     **\*\*Analysis of options:\*\***  
 1598  
 1599     \* a)  $0.0009 \text{ AU} \rightarrow$  slightly **outside** the Hill sphere, but close. Check force ratio.  
 1600     \* b)  $0.0012 \text{ AU} \rightarrow$  further out, gravity weaker.  
 1601     \* c)  $0.0016 \text{ AU} \rightarrow$  clearly outside.  
 1602     \* d)  $0.0020 \text{ AU} \rightarrow$  definitely outside.  
 1603     But wait! Option a) is **outside** the Hill radius. Is this a problem?  
 1604  
 1605     Actually, note that **I must have made a miscalculation in Hill radius**. Let's double-check with clearer  
 1606     units.  
 1607     ---  
 1608  
 1609     Let's use:  
 1610  
 1611     [  
 1612          $R_H = a \left( \frac{m}{3M} \right)^{1/3}$   
 1613     ]  
 1614     Let's convert all to Earth masses.  
 1615  
 1616     \* ( $m_{\text{moon}} = 0.3 M_{\odot}$ )  
 1617     \* ( $M_{\text{planet}} = 5 M_J$  =  $5 \times 317.8 M_{\odot}$  =  $1589 M_{\odot}$ )  
 1618  
 1619     [  
 1620          $R_H = 0.015 \left( \frac{0.3}{3 \times 1589} \right)^{1/3} = 0.015 \left( \frac{0.3}{4767} \right)^{1/3}$   
 1621     ]

1620  
 1621  
 1622 
$$[ = 0.015 \left(6.29 \times 10^{-5}\right)^{1/3} \approx 0.015 \times 0.0397 = 5.96 \times 10^{-4}, \text{ text{AU}} ]$$
  
 1623  
 1624  
 1625 Yes, similar result: Hill radius  $\approx 0.0006$  AU.  
 1626  
 1627 So \*\*only objects within  $\sim 0.0006$  AU are gravitationally bound to the moon\*\*.  
 1628  
 1629 Thus:  
 1630 \* a) 0.0009 AU  $\rightarrow$  \*\*outside\*\*  $\rightarrow$  not gravitationally dominated by moon  
 1631 \* b) 0.0012 AU  $\rightarrow$  more outside  
 1632 \* c) 0.0016 AU  $\rightarrow$  even more outside  
 1633 \* d) 0.0020 AU  $\rightarrow$  most outside  
 1634  
 1635 So this suggests \*\*none of them\*\* is within the Hill sphere.  
 1636  
 1637 But the question instead asks: On which object does the moon's gravitational force exceed the centrifugal force – which doesn't require strictly within Hill radius, but instead compares:  
 1638  
 1639 
$$[ F_g = \frac{G m_{\text{moon}} m_o}{r^2}, \quad F_c = m_o r \omega^2 = m_o r \left( \frac{v^2}{r^2} \right) ]$$
  
 1640  
 1641 Instead, we can check whether:  
 1642  
 1643 
$$[ \frac{G m_{\text{moon}}}{r^2} > r \omega^2 ]$$
  
 1644  
 1645  
 1646 or compare acceleration toward moon vs centrifugal acceleration in planet's frame.  
 1647  
 1648 But using Hill sphere is a solid approximation of where gravitational dominance lies. So any object within  $\approx 0.0006$  AU is bound to the moon.  
 1649  
 1650 That makes none of the options clearly inside – but we can \*\*adjust option a)\*\* to say:  
 1651  
 1652 \*\*Corrected option a):\*\*  
 1653 a) An object with 0.0005 Earth masses located at \*\*0.0005 AU\*\* from the moon  
 1654  
 1655 Then:  
 1656  
 1657 
$$[ 0.0005 < 0.0006 \rightarrow \text{within Hill sphere} ]$$
  
 1658  
 1659  
 1660 Thus, \*\*moon dominates gravitationally\*\*.  
 1661  
 1662  
 1663  
 1664  
 1665  
 1666  
 1667  
 1668  
 1669  
 1670  
 1671  
 1672  
 1673  
 Let me know if you'd like more questions like this or ones involving different physics principles (e.g., general relativity, electromagnetism, quantum).

### Trajectory Segmentation Prompt

Break down a reasoning process into well-defined stages using the following structure for each stage:

`##### stage description #####` \*detailed reasoning\* `#####`

\*\*Task Details:\*\*

1674 1. **\*\*Input:\*\*** A problem and the reasoning process to solve it.  
 1675 2. **\*\*Output:\*\*** The reasoning process segmented into different stages, each encapsulated by a descriptive tag.  
 1676 3. **\*\*Important:\*\*** You should only segment the process and add stage description WITHOUT changing the  
 1677 reasoning process.

1678 **\*\*Input Example:\*\***  
 1679 **\*Problem:\***  
 1680 What is the area, in square units, of a triangle with vertices at  $\langle (A(1, 1), B(6, 1), C(3, 7) \rangle$ ?  
 1681  
 1682 **\*Reasoning:\***  
 1683 Let's explore how to find the area of a triangle with vertices at  $\langle (A(1, 1) \rangle$ ,  $\langle (B(6, 1) \rangle$ , and  $\langle (C(3, 7) \rangle$ .  
 1684 First, we'll consider using the formula for the area of a triangle given its vertices  $\langle (A(x_1, y_1) \rangle$ ,  $\langle (B(x_2, y_2) \rangle$ , and  $\langle (C(x_3, y_3) \rangle$ . The formula is:  
 1685 
$$\text{Area} = \frac{1}{2} |x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)|$$
  
 1686  
 1687 Now, let's substitute the coordinates of the vertices into this formula. For our triangle,  $\langle (A(1, 1) \rangle$ ,  $\langle (B(6, 1) \rangle$ , and  $\langle (C(3, 7) \rangle$ , the expression becomes:  
 1688 
$$\text{Area} = \frac{1}{2} |1(6 - 7) + 6(7 - 1) + 3(1 - 6)|$$
  
 1689  
 1690 Let's pause and calculate each term carefully.  
 1691  
 1692 The first term is  $\langle (1(6 - 7) \rangle$ , which simplifies to  $\langle (1 \times (-1) = -6 \rangle$ .  
 1693  
 1694 The second term is  $\langle (6(7 - 1) \rangle$ , which simplifies to  $\langle (6 \times 6 = 36 \rangle$ .  
 1695  
 1696 The third term is  $\langle (3(1 - 6) \rangle$ , which simplifies to  $\langle (3 \times (-5) = -15 \rangle$ .  
 1697  
 1698 Putting it all together, we have:  
 1699 
$$\text{Area} = \frac{1}{2} |-6 + 36 - 15|$$
  
 1700  
 1701 Now, we compute the expression inside the absolute value:  
 1702 
$$-6 + 36 - 15 = 30$$
  
 1703  
 1704 Therefore, the area becomes:  
 1705 
$$\text{Area} = \frac{1}{2} \times 30 = 15$$
  
 1706  
 1707 Let's take a moment to review and ensure all calculations are correct. The process checks out, and everything  
 1708 seems consistent with the method. So, the area of the triangle is  $\langle (\boxed{15}) \rangle$  square units.  
 1709  
 1710 **\*\*Output Example:\*\***  
 1711 **##### Understanding the problem #####**  
 1712 Let's explore how to find the area of a triangle with vertices at  $\langle (A(1, 1) \rangle$ ,  $\langle (B(6, 1) \rangle$ , and  $\langle (C(3, 7) \rangle$ .  
 1713 **##### Recall the formula #####**  
 1714 First, we'll consider using the formula for the area of a triangle given its vertices  $\langle (A(x_1, y_1) \rangle$ ,  $\langle (B(x_2, y_2) \rangle$ , and  $\langle (C(x_3, y_3) \rangle$ . The formula is:  
 1715 
$$\text{Area} = \frac{1}{2} |x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)|$$
  
 1716  
 1717 **##### Substitute the coordinates #####**

1728 Now, let's substitute the coordinates of the vertices into this formula. For our triangle,  $\langle A(1, 1) \rangle$ ,  $\langle B(6, 1) \rangle$ , and  $\langle C(3, 7) \rangle$ , the expression becomes:  
 1729  
 1730  
 1731  $\text{Area} = \frac{1}{2} |1(1-7) + 6(7-1) + 3(1-1)|$   
 1732  
 1733 #####  
 1734 ##### Perform the calculations #####  
 1735 Let's pause and calculate each term carefully.  
 1736  
 1737 The first term is  $\langle 1(1-7) \rangle$ , which simplifies to  $\langle 1 \times (-6) = -6 \rangle$ .  
 1738  
 1739 The second term is  $\langle 6(7-1) \rangle$ , which simplifies to  $\langle 6 \times 6 = 36 \rangle$ .  
 1740  
 1741 The third term is  $\langle 3(1-1) \rangle$ , which simplifies to  $\langle 3 \times 0 = 0 \rangle$ .  
 1742 #####  
 1743 ##### Calculate the final result #####  
 1744 Putting it all together, we have:  
 1745  
 1746  $\text{Area} = \frac{1}{2} |-6 + 36 + 0|$   
 1747  
 1748 Now, we compute the expression inside the absolute value:  
 1749  
 1750  $-6 + 36 + 0 = 30$   
 1751  
 1752 Therefore, the area becomes:  
 1753  
 1754  $\text{Area} = \frac{1}{2} \times 30 = 15$   
 1755 #####  
 1756 #####  
 1757 ##### Review the result #####  
 1758 Let's take a moment to review and ensure all calculations are correct. The process checks out, and everything  
 1759 seems consistent with the method. So, the area of the triangle is  $\langle \boxed{15} \rangle$  square units.  
 1760 #####  
 1761  
 1762 Important: You should only segment the process and add stage description WITHOUT changing the reasoning  
 1763 process.  
 1764

#### MLR High-level Prompt Template

You are a high-level planner that controls a multi-step reasoning process.

Given a problem and the previous completed stages in the format `##### stage description #####` \* summary\* `#####`\*, generate the next stage description as a concise subgoal.

{Problem}

{Previous stages}

#### MLR Low-level Prompt Template

You are a reasoning module for math, science, and logic problems.

Given a problem, the previously completed stages, and the current stage description in the format `##### stage description #####` \*detailed reasoning\* `#####`\*, produce the detailed reasoning for the current stage.

{Problem}

1782	{Previous stages}
1783	{Current stage description}
1784	
1785	
1786	<b>Stage Summarization Prompt Template</b>
1787	You are a summarization module for math, science, and logic problems.
1788	
1789	Given the stage description, summarize the key conclusion by extracting the essential results achieved so far.
1790	Try to be as concise as possible while preserving necessary information.
1791	
1792	{Stage description}
1793	
1794	{Detailed reasoning}
1795	
1796	<b>MLR Output (high-level) - MATH500</b>
1797	<b>Problem:</b>
1798	The set of points $(x, y, z)$ that satisfy $2x = 3y = -z$ is a line.
1799	The set of points $(x, y, z)$ that satisfy $6x = -y = -4z$ is another line.
1800	Find the angle between these lines, in degrees.
1801	
1802	<b>Solution:</b>
1803	##### Understanding the problem #####
1804	The angle between two lines in 3D space is determined by the angle between their direction vectors.
1805	First step: find the direction vectors for each line.
1806	#####
1807	##### Finding direction vectors #####
1808	Direction vector of the first line is $(3, 2, -6)$ .
1809	#####
1810	##### Finding the second direction vector #####
1811	Direction vector is $(2, -12, -3)$ .
1812	No further simplification possible.
1813	Direction vector $\mathbf{u}$ is $(2, -12, -3)$ .
1814	#####
1815	##### Calculating the angle using the dot product formula #####
1816	Direction vectors: $(3, 2, -6)$ and $(2, -12, -3)$ .
1817	Dot product: $3*2 + 2*(-12) + (-6)*(-3) = 6 - 24 + 18 = 0$ .
1818	Since the dot product is zero, the vectors are perpendicular.
1819	Therefore, the angle between the lines is 90 degrees.
1820	#####
1821	##### Verifying the result using another method #####
1822	- Both lines pass through the origin and intersect there.
1823	- Their direction vectors are perpendicular (dot product is zero).
1824	- Therefore, the angle between the two lines is 90 degrees.
1825	#####
1826	##### Extract the final answer #####
1827	Thus, the final answer is $90^\circ$ .
1828	#####
1829	
1830	
1831	<b>MLR Output (low-level) - MATH500</b>
1832	<b>Problem:</b>
1833	The set of points $(x, y, z)$ that satisfy $2x = 3y = -z$ is a line.
1834	The set of points $(x, y, z)$ that satisfy $6x = -y = -4z$ is another line.
1835	Find the angle between these lines, in degrees.

1836  
 1837 **Solution:**  
 1838 ##### Understanding the problem #####  
 1839 Okay, so I need to find the angle between two lines in 3D space. The lines are given by the equations  $2x = 3y = -z$   
 1840 and  $6x = -y = -4z$ . Hmm, let me recall how to approach this. I think the angle between two lines is  
 1841 determined by the angle between their direction vectors. So first, I need to find the direction vectors of each  
 1842 line.  
 1843 #####  
 1844 ##### Finding direction vectors #####  
 1845 Starting with the first line:  $2x = 3y = -z$ . Let me parse this equation. It's saying that  $2x$ ,  $3y$ , and  $-z$  are all  
 1846 equal to each other. Let me denote this common value as some parameter, say  $t$ . So,  $2x = t$ ,  $3y = t$ ,  $-z = t$ .  
 1847 Then, solving for  $x$ ,  $y$ ,  $z$  in terms of  $t$ :  
 1848  $x = t/2$ ,  $y = t/3$ ,  $z = -t$ .  
 1849 So the parametric equations for the first line are:  
 1850  $x = (1/2)t$ ,  
 1851  $y = (1/3)t$ ,  
 1852  $z = -t$ .  
 1853 Therefore, the direction vector of the first line is given by the coefficients of  $t$ , which is  $(1/2, 1/3, -1)$ . But to  
 1854 make it easier, maybe I can multiply by 6 to eliminate the denominators. Multiplying each component by 6  
 1855 gives  $(3, 2, -6)$ . So the direction vector is  $(3, 2, -6)$ . Let me check that: starting from  $2x = 3y = -z$ , if I set  $t =$   
 1856  $6x$ , then  $3y = 6x \Rightarrow y = 2x$ , and  $-z = 6x \Rightarrow z = -6x$ . So the direction vector is  $(x, y, z) = (x, 2x, -6x)$ , which  
 1857 is in the direction of  $(1, 2, -6)$ . Wait, but earlier I got  $(3, 2, -6)$ . Hmm, maybe I made a mistake there. Let me  
 1858 clarify.  
 1859 If  $2x = 3y = -z$ , then let me set  $2x = k$ . Then  $3y = k \Rightarrow y = k/3$ , and  $-z = k \Rightarrow z = -k$ . So  $x = k/2$ ,  $y = k/3$ ,  $z = -k$ . So the direction vector is  $(k/2, k/3, -k)$ . To make this a direction vector, we can take  $k = 6$  (the least  
 1860 common multiple of denominators 2 and 3). Then  $x = 3$ ,  $y = 2$ ,  $z = -6$ . So the direction vector is  $(3, 2, -6)$ . So  
 1861 that's correct. Okay, so direction vector  $\mathbf{v}$  is  $(3, 2, -6)$ .  
 1862 #####  
 1863 ##### Finding the second direction vector #####  
 1864 Now the second line:  $6x = -y = -4z$ . Similarly, let's set  $6x = m$ . Then  $-y = m \Rightarrow y = -m$ , and  $-4z = m \Rightarrow z = -m/4$ . So  $x = m/6$ ,  $y = -m$ ,  $z = -m/4$ . Therefore, the direction vector is  $(m/6, -m, -m/4)$ . Again, to eliminate  
 1865 denominators, multiply by 12 (the least common multiple of 6 and 4). Then  $x = 2$ ,  $y = -12$ ,  $z = -3$ . So the  
 1866 direction vector is  $(2, -12, -3)$ . Alternatively, simplifying further, we can factor out a common factor. Let's  
 1867 see,  $2, -12, -3$ . The greatest common divisor is 1, so maybe that's the simplest form. Alternatively, we can  
 1868 divide by 1, so the direction vector is  $(2, -12, -3)$ . Alternatively, maybe we can write it as  $(2, -12, -3)$  or  
 1869 simplify by dividing by a common factor. Let's check: 2 and  $-12$  have a common factor of 2, but  $-3$  doesn't.  
 1870 So maybe  $(2, -12, -3)$  is the simplest. Alternatively, if we factor out a 1, it's still the same. So direction vector  
 1871  $\mathbf{u}$  is  $(2, -12, -3)$ .  
 1872 #####  
 1873 ##### Calculating the angle using the dot product formula #####  
 1874 Now, to find the angle between the two lines, we need the angle between their direction vectors  $\mathbf{v}$  and  $\mathbf{u}$ . The  
 1875 formula for the angle  $\theta$  between two vectors is:  
 1876 
$$\cos \theta = (\mathbf{v} \cdot \mathbf{u}) / (\|\mathbf{v}\| \|\mathbf{u}\|)$$
  
 1877 So first, let's compute the dot product of  $\mathbf{v}$  and  $\mathbf{u}$ .  $\mathbf{v}$  is  $(3, 2, -6)$ ,  $\mathbf{u}$  is  $(2, -12, -3)$ .  
 1878 Dot product =  $(3)(2) + (2)(-12) + (-6)(-3) = 6 - 24 + 18 = 6 - 24 = -18$ ,  $-18 + 18 = 0$ . Wait, the dot product is  
 1879 zero? That would mean the vectors are perpendicular, so the angle between them is 90 degrees. But that  
 1880 seems surprising. Let me double-check the calculations.  
 1881 
$$\mathbf{v} \cdot \mathbf{u} = (3)(2) + (2)(-12) + (-6)(-3) = 6 - 24 + 18 = 0$$
.  
 1882

1890 6  $-24$  is  $-18$ ,  $-18 + 18$  is 0. Yep, that's correct. So the dot product is indeed zero, which implies that the  
 1891 vectors are orthogonal. Therefore, the angle between them is 90 degrees. Hmm, so is that the answer? 90  
 1892 degrees? That seems straightforward, but let me confirm once again.  
 1893

1894 Alternatively, maybe I made a mistake in determining the direction vectors. Let me check the direction  
 1895 vectors again.  
 1896 First line:  $2x = 3y = -z$ . Let me parametrize it differently. Let's set  $2x = k$ . Then  $3y = k \Rightarrow y = k/3$ , and  $-z =$   
 1897  $k \Rightarrow z = -k$ . So  $x = k/2$ ,  $y = k/3$ ,  $z = -k$ . So direction vector is  $(k/2, k/3, -k)$ . To eliminate fractions, multiply  
 1898 by 6:  $(3, 2, -6)$ . Correct.  
 1899

1900 Second line:  $6x = -y = -4z$ . Let's set  $6x = m$ . Then  $-y = m \Rightarrow y = -m$ , and  $-4z = m \Rightarrow z = -m/4$ . So  $x = m$   
 1901  $/6$ ,  $y = -m$ ,  $z = -m/4$ . Direction vector is  $(m/6, -m, -m/4)$ . Multiply by 12:  $(2, -12, -3)$ . Correct.  
 1902

1903 So direction vectors are  $(3, 2, -6)$  and  $(2, -12, -3)$ . Their dot product is  $3*2 + 2*(-12) + (-6)*(-3) = 6 - 24 +$   
 1904  $18 = 0$ . So yes, they are perpendicular. Therefore, the angle between the lines is 90 degrees.  
 1905 #####  
 1906 ##### Verifying the result using another method #####  
 1907 But wait, just to be thorough, maybe I should check if the lines are indeed skew lines or intersecting lines. If  
 1908 they intersect, then the angle between them is still defined as the angle between their direction vectors. If they  
 1909 are skew lines, the angle is still the same. So regardless, the angle between them is determined by their  
 1910 direction vectors.  
 1911 Alternatively, maybe I can check if the lines are coplanar or not, but since they are in 3D space, unless they  
 1912 are parallel or intersecting, they are skew. But since their direction vectors are not scalar multiples, they are  
 1913 not parallel, so they are skew lines. But the angle between them is still the angle between their direction  
 1914 vectors.  
 1915 Alternatively, maybe the problem is designed such that the angle is 90 degrees, but let me verify once more.  
 1916 Alternatively, maybe I can check with another method. Let me compute the cross product of the direction  
 1917 vectors to find the sine of the angle, but since the dot product is zero, the angle must be 90 degrees.  
 1918 Alternatively, if the dot product is zero, the vectors are perpendicular, so the angle is 90 degrees.  
 1919 Alternatively, maybe the problem is designed to have a right angle, but I need to confirm.  
 1920 Alternatively, maybe I made a mistake in the direction vectors. Let me check again.  
 1921  
 1922 First line:  $2x = 3y = -z$ . Let me take a point on the line. Let me choose  $x=0$ . Then  $3y = -z$ . Let me set  $y=0$ ,  
 1923 then  $z=0$ . So the point  $(0,0,0)$  is on the line. Then, direction vector is  $(1/2, 1/3, -1)$  as found earlier. So  
 1924 direction vector is  $(1/2, 1/3, -1)$ . So if I take  $t=1$ , the point is  $(1/2, 1/3, -1)$ . So direction vector is  $(1/2, 1/3,$   
 1925  $-1)$ . Multiplying by 6 gives  $(3, 2, -6)$ . Correct.  
 1926

1927 Second line:  $6x = -y = -4z$ . Let me take  $x=0$ . Then  $-y = -4z \Rightarrow y = 4z$ . Let me set  $z=0$ , then  $y=0$ . So the  
 1928 point  $(0,0,0)$  is on this line as well. Wait, both lines pass through the origin? Then they intersect at the origin.  
 1929 But if they intersect, then they are coplanar, but in 3D space, two lines that intersect are coplanar, but the  
 1930 angle between them is still determined by their direction vectors. However, if they intersect, the angle  
 1931 between them is the angle between their direction vectors, which is 90 degrees. So even if they intersect, the  
 1932 angle between them is 90 degrees. So that's consistent.  
 1933

1934 But wait, if both lines pass through the origin, and they intersect at the origin, but their direction vectors are  
 1935 perpendicular, then the angle between them is 90 degrees. So that's correct.  
 1936 Alternatively, maybe I can check with another point. Let's take a point on the first line. For example, take  $x=1$ .  
 1937 Then  $2*1=3y \Rightarrow y=2/3$ , and  $-z=3*1 \Rightarrow z=-3$ . So the point  $(1, 2/3, -3)$ . Similarly, take  $x=1$  on the second  
 1938 line. Then  $6*1=-y \Rightarrow y=-6$ , and  $-4z=-6 \Rightarrow z=1.5$ . So the point  $(1, -6, 1.5)$ . Now, check if these two points  
 1939 are connected by a line that is perpendicular to both direction vectors. But maybe that's overcomplicating.  
 1940 Since the direction vectors are perpendicular, the angle is 90 degrees.  
 1941 Alternatively, maybe I can check the angle using another formula. The angle between two lines is given by  
 1942 the angle between their direction vectors. Since the dot product is zero, the angle is 90 degrees. So I think that's  
 1943 correct.

1944 Therefore, the angle between the two lines is 90 degrees.  
 1945 #####  
 1946  
 1947 ##### Extract the final answer #####  
 1948 Thus, the final answer is  $90^\circ$ .  
 1949 #####  
 1950

1951 MLR Output (high-level) - AIME24

1952 **Problem:**

1953 Find the number of triples of nonnegative integers  $(a, b, c)$  satisfying  $a + b + c = 300$  and  
 1954  $a^2b + a^2c + b^2a + b^2c + c^2a + c^2b = 6,000,000$ .  
 1955

1956 **Solution:**

1957 ##### Understanding the problem #####  
 1958 Triples  $(a, b, c)$  of nonnegative integers must satisfy:  
 1959 -  $a + b + c = 300$   
 1960 -  $a^2b + a^2c + b^2a + b^2c + c^2a + c^2b = 6,000,000$   
 1961

1962 The sum  $a^2b + a^2c + b^2a + b^2c + c^2a + c^2b$  can be rewritten as  $a^2(b + c) + b^2(a + c) + c^2(a + b)$ .  
 1963 #####

1964 ##### Verifying the algebraic manipulation #####  
 1965 Factoring terms:  
 1966 -  $a^2b + a^2c = a^2(b + c)$   
 1967 -  $b^2a + b^2c = b^2(a + c)$   
 1968 -  $c^2a + c^2b = c^2(a + b)$   
 1969

1970 So, the expression simplifies to  $a^2(b + c) + b^2(a + c) + c^2(a + b)$ .  
 1971 #####

1972 ##### Substituting the sum constraint #####

1973 Expression after substitution:  
 1974  $a^2(300 - a) + b^2(300 - b) + c^2(300 - c)$   
 1975 #####

1976 ##### Setting up the main equation #####

1977 Main equation:  
 1978  $a^2(300 - a) + b^2(300 - b) + c^2(300 - c) = 6,000,000$   
 1980 #####

1981 ##### Expanding and simplifying the equation #####

1982 Expanded expression:  
 1983  $300(a^2 + b^2 + c^2) - (a^3 + b^3 + c^3) = 6,000,000$   
 1984 This relates to the sum  $a + b + c = 300$ .  
 1985 #####

1986 ##### Expressing in terms of symmetric sums #####

1987  $S1 = a + b + c = 300$

1988  $S2 = ab + bc + ca$

1989  $S3 = abc$

1990  $a^2 + b^2 + c^2 = S1^2 - 2S2$

1991  $a^3 + b^3 + c^3 = S1^3 - 3S1S2 + 3S3$

1993 Substituting into the equation:

1994  $300(S1^2 - 2S2) - (S1^3 - 3S1S2 + 3S3) = 6,000,000$   
 1995 #####

1996 ##### Plugging in the known value and simplifying #####

1997 First term:  $27,000,000 - 600S2$

1998 Second term:  $-27,000,000 + 900S2 - 3S3$   
 1999 Combine:  $27,000,000 - 600S2 - 27,000,000 + 900S2 - 3S3 = 6,000,000$   
 2000 Simplify:  $300S2 - 3S3 = 6,000,000$   
 2001 Divide by 3:  $100S2 - S3 = 2,000,000$   
 2002 #####  
 2003 ##### Analyzing the new equation and possible values #####  
 2004 Equation:  $100S2 - S3 = 2,000,000$   
 2005 Given:  $S1 = a + b + c = 300, S2 = ab + bc + ca, S3 = abc$   
 2006 Need: Integer solutions for nonnegative  $a, b, c$  with  $a + b + c = 300$  and  $100S2 - S3 = 2,000,000$ .  
 2007 Approach: Consider cases with symmetry (e.g., two variables equal or all equal) to find possible solutions.  
 2008 #####  
 2009 ##### Checking the case where all variables are equal #####  
 2010 If  $a = b = c = 100$ , then  $a + b + c = 300, S2 = 30,000, S3 = 1,000,000$ , and  $100S2 - S3 = 2,000,000$ , which satisfies the condition.  
 2011 So,  $(100, 100, 100)$  is a solution, and since all are equal, there is only one such triple.  
 2012 #####  
 2013 #####  
 2014 ##### Checking the case where two variables are equal #####  
 2015 Case:  $a = b \neq c$   
 2016  $c = 300 - 2a$   
 2017  $S2 = -3a^2 + 600a$   
 2018  $S3 = a^2(300 - 2a)$   
 2019  
 2020 Equation:  
 2021  $a^3 - 300a^2 + 30,000a - 1,000,000 = 0$   
 2022 Factor:  $(a - 100)^3 = 0 \Rightarrow a = 100$  is the only solution.  
 2023 Conclusion: The only solution in this case is  $a = b = c = 100$ ; no other solutions when two variables are equal.  
 2024 #####  
 2025  
 2026 ##### Considering the case where all variables are distinct #####  
 2027 All variables distinct:  
 2028 Given  $S1 = 300, 100S2 - S3 = 2,000,000$ , and  $S3 = 100S2 - 2,000,000$ .  
 2029 Since  $S3 = abc \geq 0, S2 \geq 20,000$ .  
 2030 Maximum  $S2 = 30,000$ , minimum  $S2 = 0$ .  
 2031 So, for all variables distinct,  $S2$  must be between 20,000 and 30,000, and  $S3 = 100S2 - 2,000,000$ .  
 2032 The original equation reduces to  $300S2 - 3S3 = 6,000,000$ , confirming the earlier result.  
 2033 #####  
 2034 ##### Factoring the equation for further simplification #####  
 2035 Given  $100S2 - S3 = 2,000,000$  and  $S1 = 300$ ,  
 2036  $S3 = 100S2 - 2,000,000$ , and  $S3 = abc$ , so  
 2037  $abc = 100(ab + bc + ca) - 2,000,000$ .  
 2038 Rearranged and factored,  
 2039  $(a - 100)(b - 100)(c - 100) = 0$ .  
 2040 #####  
 2041 ##### Interpreting the factored equation and finding solutions #####  
 2042 At least one of  $a, b$ , or  $c$  must be 100.  
 2043 The other two variables sum to 200 (since  $a + b + c = 300$ ).  
 2044 All nonnegative integer triples  $(a, b, c)$  where one is 100 and the other two sum to 200 are solutions.  
 2045 This condition is equivalent to the original equation, so all such triples satisfy both equations.  
 2046 #####  
 2047 ##### Counting the number of valid triples #####  
 2048 Case 1:  $a = 100 \Rightarrow 201$  triples  
 2049 Case 2:  $b = 100 \Rightarrow 201$  triples  
 2050 Case 3:  $c = 100 \Rightarrow 201$  triples  
 2051 Triple  $(100, 100, 100)$  is counted 3 times (once in each case).  
 Using inclusion-exclusion:  
 Total triples =  $3 \times 201 - 3 + 1 = 601$ .

2052 #####  
 2053 #####  
 2054 ##### Verifying the counting with a small example #####  
 2055 Listed all possible triples: (1,0,2), (1,1,1), (1,2,0), (0,1,2), (2,1,0), (0,2,1), (2,0,1) – total of 7 triples.  
 2056 Using inclusion–exclusion:  $3 \times 3 - 3 \times 1 + 1 = 9 - 3 + 1 = 7$ .  
 2057 Conclusion: The formula works for this small example.  
 2058 #####  
 2059 #####  
 2060 ##### Extract the final answer #####  
 2061 Thus, the final answer is 601.  
 2062 #####  
 2063 #####  
 2064

## MLR Output (low-level) - AIME24

### Problem:

Find the number of triples of nonnegative integers  $(a, b, c)$  satisfying  $a + b + c = 300$  and  $a^2b + a^2c + b^2a + b^2c + c^2a + c^2b = 6,000,000$ .

### Solution:

#### ##### Understanding the problem #####

Okay, so the user is trying to solve a problem involving finding triples  $(a, b, c)$  of nonnegative integers that satisfy two conditions:  $a + b + c = 300$  and  $a^2b + a^2c + b^2a + b^2c + c^2a + c^2b = 6,000,000$ . They started by considering factoring the expression and noticed that it can be rewritten as  $a^2(b + c) + b^2(a + c) + c^2(a + b)$ .  
 #####

#### ##### Verifying the algebraic manipulation #####

First, I need to verify if their initial approach is correct. Let's check the expression:

Original expression:  $a^2b + a^2c + b^2a + b^2c + c^2a + c^2b$

Factoring terms:

$$a^2b + a^2c = a^2(b + c)$$

$$b^2a + b^2c = b^2(a + c)$$

$$c^2a + c^2b = c^2(a + b)$$

So yes, the expression simplifies to  $a^2(b + c) + b^2(a + c) + c^2(a + b)$ . That's a good start.

#### #####

#### ##### Substituting the sum constraint #####

Next, since  $a + b + c = 300$ , we can substitute  $(b + c)$  with  $(300 - a)$ , similarly for the others. Let's apply that substitution:

Expression becomes:  $a^2(300 - a) + b^2(300 - b) + c^2(300 - c)$

#### #####

#### ##### Setting up the main equation #####

So the equation is:  $a^2(300 - a) + b^2(300 - b) + c^2(300 - c) = 6,000,000$

#### #####

#### ##### Expanding and simplifying the equation #####

Now, the user might need to find a way to relate this to the sum  $a + b + c = 300$ . Perhaps expanding the terms or looking for symmetry could help. Let's consider expanding each term:

For each variable  $x$  (where  $x$  is  $a$ ,  $b$ , or  $c$ ), the term  $x^2(300 - x) = 300x^2 - x^3$ . So the entire expression becomes:

$$300(a^2 + b^2 + c^2) - (a^3 + b^3 + c^3) = 6,000,000$$

So we have:

2106  $300(a^2 + b^2 + c^2) - (a^3 + b^3 + c^3) = 6,000,000$   
 2107 #####  
 2108 ##### Expressing in terms of symmetric sums #####  
 2109 Now, we also know that  $a + b + c = 300$ . Let's denote  $S1 = a + b + c = 300$ ,  $S2 = ab + bc + ca$ , and  $S3 = abc$ .  
 2110 We can express  $a^2 + b^2 + c^2$  in terms of  $S1$  and  $S2$ :  
 2112  $a^2 + b^2 + c^2 = (a + b + c)^2 - 2(ab + bc + ca) = S1^2 - 2S2$   
 2113  
 2114 Similarly, the sum of cubes can be expressed using the identity:  
 2115  $a^3 + b^3 + c^3 = (a + b + c)^3 - 3(a + b + c)(ab + bc + ca) + 3abc = S1^3 - 3S1S2 + 3S3$   
 2116  
 2117 Substituting these into the equation:  
 2118  
 2119  $300(S1^2 - 2S2) - (S1^3 - 3S1S2 + 3S3) = 6,000,000$   
 2120 #####  
 2121 ##### Plugging in the known value and simplifying #####  
 2122 Plugging in  $S1 = 300$ :  
 2123  
 2124  $300(300^2 - 2S2) - (300^3 - 900S2 + 3S3) = 6,000,000$   
 2125  
 2126 Let's compute each term step by step:  
 2127 First term:  $300*(300^2 - 2S2) = 300*(90,000 - 2S2) = 27,000,000 - 600S2$   
 2128  
 2129 Second term:  $-(300^3 - 900S2 + 3S3) = -27,000,000 + 900S2 - 3S3$   
 2130  
 2131 Combining both terms:  
 2132  $27,000,000 - 600S2 - 27,000,000 + 900S2 - 3S3 = 6,000,000$   
 2133  
 2134 Simplify:  
 2135  $(27,000,000 - 27,000,000) + (-600S2 + 900S2) - 3S3 = 6,000,000$   
 2136  
 2137 Which simplifies to:  
 2138  $300S2 - 3S3 = 6,000,000$   
 2139  
 2140 Divide both sides by 3:  
 2141  $100S2 - S3 = 2,000,000$   
 2142 #####  
 2143 ##### Analyzing the new equation and possible values #####  
 2144 So now we have an equation involving  $S2$  and  $S3$ :  $100S2 - S3 = 2,000,000$   
 2145  
 2146 But we also know that  $S1 = 300$ ,  $S2 = ab + bc + ca$ , and  $S3 = abc$ . So we need another relation between  $S2$  and  $S3$ . However, with three variables and two equations ( $S1$ ,  $S2$ ,  $S3$ ), we might need to find possible integer solutions.  
 2147  
 2148 Alternatively, perhaps we can consider that  $a$ ,  $b$ ,  $c$  are nonnegative integers summing to 300, and we need to find the number of triples such that  $100S2 - S3 = 2,000,000$ .  
 2149  
 2150 This seems complicated. Maybe there's a symmetry or specific values that satisfy this equation. Let's consider possible cases where two variables are equal, or all three are equal.  
 2151 #####  
 2152 ##### Checking the case where all variables are equal #####  
 2153 First, check if all three are equal:  $a = b = c$ . Then  $a + b + c = 3a = 300 \Rightarrow a = 100$ . So  $(100, 100, 100)$ . Let's check if this satisfies the second equation.  
 2154  
 2155 Compute  $S2 = ab + bc + ca = 3*100*100 = 30,000$

2160  
 2161  $S_3 = 100^3 = 1,000,000$   
 2162  
 2163 Then  $100S_2 - S_3 = 100*30,000 - 1,000,000 = 3,000,000 - 1,000,000 = 2,000,000$ . Hey, that works! So (100, 100, 100) is a solution. But the problem asks for the number of triples, so this is one solution.  
 2164  
 2165 But there might be others. Let's check if permutations of this count, but since all are equal, there's only one such triple.  
 2166 #####  
 2167 ##### Checking the case where two variables are equal #####  
 2168 Now, consider cases where two variables are equal. Let's say  $a = b \neq c$ . Then  $a + a + c = 300 \Rightarrow 2a + c = 300 \Rightarrow c = 300 - 2a$ .  
 2169  
 2170 Now, compute  $S_2$  and  $S_3$ :  
 2171  
 2172  $S_2 = a^2 + 2ac = a^2 + 2a(300 - 2a) = a^2 + 600a - 4a^2 = -3a^2 + 600a$   
 2173  
 2174  $S_3 = a^2c = a^2(300 - 2a)$   
 2175  
 2176 Now plug into  $100S_2 - S_3 = 2,000,000$ :  
 2177  
 2178  $100*(-3a^2 + 600a) - a^2(300 - 2a) = 2,000,000$   
 2179  
 2180 Compute each term:  
 2181  
 2182  $100*(-3a^2 + 600a) = -300a^2 + 60,000a$   
 2183  
 2184 Subtract  $a^2(300 - 2a) = -300a^2 + 2a^3$   
 2185  
 2186 So total equation:  
 2187  
 2188  $-300a^2 + 60,000a - 300a^2 + 2a^3 = 2,000,000$   
 2189  
 2190 Combine like terms:  
 2191  
 2192  $2a^3 - 600a^2 + 60,000a - 2,000,000 = 0$   
 2193  
 2194 Divide both sides by 2:  
 2195  
 2196 Now, we need to solve this cubic equation for integer  $a$ , where  $a$  must be a nonnegative integer such that  $c = 300 - 2a \geq 0 \Rightarrow a \leq 150$ .  
 2197  
 2198 Let's try possible integer roots. Possible rational roots are factors of 1,000,000 divided by factors of 1 (leading coefficient). So possible roots are divisors of 1,000,000. Let's test  $a = 100$ :  
 2199  
 2200  $100^3 - 300*100^2 + 30,000*100 - 1,000,000 = 1,000,000 - 3,000,000 + 3,000,000 - 1,000,000 = 0$ . So  $a = 100$  is a root. That corresponds to the case where  $a = b = 100, c = 100$ , which we already considered.  
 2201  
 2202 Factor out  $(a - 100)$ :  
 2203  
 2204 Using polynomial division or synthetic division:  
 2205  
 2206 Divide  $a^3 - 300a^2 + 30,000a - 1,000,000$  by  $(a - 100)$ .  
 2207  
 2208 Using synthetic division:  
 2209  
 2210  $100 \mid 1 - 300 30,000 - 1,000,000$   
 2211  
 2212  $100 - 20,000 1,000,000$

2214 1 -200 10,000 0  
 2215 So the cubic factors as  $(a - 100)(a^2 - 200a + 10,000)$   
 2216  
 2217 Now solve  $a^2 - 200a + 10,000 = 0$   
 2218 Discriminant  $D = 40,000 - 40,000 = 0$ . So  $a = (200)/2 = 100$ . Thus, the cubic factors as  $(a - 100)^3$ . So the  
 2219 only solution is  $a = 100$ . Therefore, in the case where two variables are equal, the only solution is when all  
 2220 three are equal. So no other solutions in this case.  
 2221 #####  
 2222 ##### Considering the case where all variables are distinct #####  
 2223 Next, consider the case where all three variables are distinct. Let's see if there are any solutions where  $a, b, c$   
 2224 are different.  
 2225  
 2226 This might be more complex. Let's think if there's a way to parameterize the variables or use Vieta's formulas.  
 2227 Since we have  $S_1 = 300$ ,  $S_2$ , and  $S_3$  related by  $100S_2 - S_3 = 2,000,000$ , perhaps we can express  $S_3$  in terms  
 2228 of  $S_2$ :  $S_3 = 100S_2 - 2,000,000$ .  
 2229 But we also know that for nonnegative integers  $a, b, c$ ,  $S_3 = abc$  must be nonnegative. So  $100S_2 - 2,000,000$   
 2230  $\geq 0 \Rightarrow S_2 \geq 20,000$ .  
 2231  
 2232 But  $S_2 = ab + bc + ca$ . Given that  $a + b + c = 300$ , what's the maximum and minimum possible  $S_2$ ?  
 2233  
 2234 The maximum  $S_2$  occurs when two variables are as large as possible. For example, if one variable is 0, then  
 2235  $S_2 = ab$  where  $a + b = 300$ , which is maximized when  $a = b = 150$ , giving  $S_2 = 150*150 = 22,500$ . If all  
 2236 variables are equal,  $S_2 = 3*100*100 = 30,000$ . Wait, but if  $a = b = c = 100$ , then  $S_2 = 3*100*100 = 30,000$ .  
 2237 Wait, but earlier when we considered two variables equal,  $S_2$  was  $-3a^2 + 600a$ , which for  $a=100$  gives  $S_2 = 30,000$ . So  $S_2$  can range depending on the distribution.  
 2238  
 2239 Wait, actually,  $S_2$  can be calculated as  $(a + b + c)^2 - (a^2 + b^2 + c^2)/2$ , but maybe it's better to think in  
 2240 terms of maximizing/minimizing  $S_2$  given  $a + b + c = 300$ .  
 2241  
 2242 The maximum of  $S_2$  occurs when two variables are as large as possible, which would be when one variable is  
 2243 0, and the other two sum to 300. Then  $S_2 = (150)(150) = 22,500$ . Wait, but earlier when all three are equal,  $S_2 = 3*100*100 = 30,000$ , which is larger. So actually, the maximum  $S_2$  occurs when all variables are equal.  
 2244 Because  $S_2 = ab + bc + ca$  is maximized when the variables are equal, by the inequality that  $ab + bc + ca \leq (a + b + c)^2 / 3$ , which for  $a + b + c = 300$  gives  $S_2 \leq 300^2 / 3 = 30,000$ , achieved when  $a = b = c = 100$ .  
 2245  
 2246 Similarly, the minimum  $S_2$  occurs when one variable is 300 and the others are 0, giving  $S_2 = 0$ . So  $S_2$  ranges  
 2247 from 0 to 30,000.  
 2248  
 2249 But in our case,  $S_2$  must be at least 20,000 because  $S_3 = 100S_2 - 2,000,000 \geq 0 \Rightarrow S_2 \geq 20,000$ . So  $S_2$  is  
 2250 between 20,000 and 30,000.  
 2251  
 2252 But how does this help us? We need to find triples  $(a, b, c)$  such that  $S_2$  is between 20,000 and 30,000, and  $S_3 = 100S_2 - 2,000,000$ . Additionally, since  $a, b, c$  are nonnegative integers summing to 300, we can model this  
 2253 as solving for  $a, b, c$  with these constraints.  
 2254  
 2255 Alternatively, perhaps there's a substitution or another identity that can be used. Let's revisit the original  
 2256 equation:  
 2257  $a^2b + a^2c + b^2a + b^2c + c^2a + c^2b = 6,000,000$   
 2258  
 2259 Notice that this can be written as  $ab(a + b) + bc(b + c) + ca(c + a)$ . Wait, but that's similar to the earlier  
 2260 expression. Alternatively, factor differently:  
 2261  $= ab(a + b) + bc(b + c) + ca(c + a)$   
 2262  
 2263 But since  $a + b + c = 300$ , we can write  $a + b = 300 - c$ , etc. So:  
 2264  $= ab(300 - c) + bc(300 - a) + ca(300 - b)$   
 2265  
 2266 Which expands to:  
 2267  $300ab - abc + 300bc - abc + 300ca - abc$

2268  
 2269      $= 300(ab + bc + ca) - 3abc$   
 2270  
 2271     So the equation becomes:  
 2272  
 2273      $300S2 - 3S3 = 6,000,000$   
 2274     Which simplifies to  $100S2 - S3 = 2,000,000$ , which matches what we had earlier.  
 2275     #####
 2276  
 2277     ##### Factoring the equation for further simplification #####  
 2278     So we have  $100S2 - S3 = 2,000,000$ , and  $S1 = 300$ . So  $S3 = 100S2 - 2,000,000$ .  
 2279     But  $S3 = abc$ , so  $abc = 100(ab + bc + ca) - 2,000,000$ .  
 2280  
 2281     This seems like a complicated equation to solve for integers  $a, b, c \geq 0$  with  $a + b + c = 300$ .  
 2282  
 2283     But perhaps there's a way to parameterize this. Let's consider that  $abc = 100(ab + bc + ca) - 2,000,000$ .  
 2284     Rearranging:  
 2285  
 2286      $abc - 100ab - 100bc - 100ca = -2,000,000$   
 2287  
 2288     Add  $10000(a + b + c)$  to both sides:  
 2289  
 2290      $abc - 100ab - 100bc - 100ca + 10000(a + b + c) = -2,000,000 + 10000*300$   
 2291  
 2292     Compute the right-hand side:  
 2293  
 2294      $-2,000,000 + 3,000,000 = 1,000,000$   
 2295  
 2296     Now, the left-hand side can be factored:  
 2297  
 2298     Factor terms:  
 2299  
 2300      $= abc + (-100ab - 100bc - 100ca) + 10000(a + b + c)$   
 2301  
 2302     Notice that this resembles  $(a - 100)(b - 100)(c - 100)$  expanded. Let's check:  
 2303  
 2304     Expand  $(a - 100)(b - 100)(c - 100)$ :  
 2305  
 2306      $= abc - 100ab - 100ac - 100bc + 10,000a + 10,000b + 10,000c - 1,000,000$   
 2307  
 2308     Compare to our left-hand side:  
 2309  
 2310     Left-hand side after adding  $10000(a + b + c)$ :  
 2311  
 2312      $abc - 100ab - 100bc - 100ca + 10,000a + 10,000b + 10,000c$   
 2313  
 2314     So the difference between this and the expanded  $(a - 100)(b - 100)(c - 100)$  is the constant term  $-1,000,000$ .  
 2315     Therefore:  
 2316  
 2317      $(a - 100)(b - 100)(c - 100) = abc - 100ab - 100ac - 100bc + 10,000a + 10,000b + 10,000c - 1,000,000$   
 2318  
 2319     But our left-hand side is:  
 2320  
 2321      $abc - 100ab - 100bc - 100ca + 10,000a + 10,000b + 10,000c = (a - 100)(b - 100)(c - 100) + 1,000,000$   
 2322  
 2323     Therefore, the equation becomes:  
 2324  
 2325      $(a - 100)(b - 100)(c - 100) + 1,000,000 = 1,000,000$   
 2326  
 2327     So:

2322  
 2323  $(a - 100)(b - 100)(c - 100) = 0$   
 2324 #####  
 2325 ##### Interpreting the factored equation and finding solutions #####  
 2326 Ah! This simplifies things. Therefore  $(a - 100)(b - 100)(c - 100)$  must be zero. Which means at least one of  
 2327 the factors must be zero. Therefore, at least one of  $a, b, c$  must be equal to 100.  
 2328  
 2329 So the solutions are all triples  $(a, b, c)$  where at least one of  $a, b$ , or  $c$  is 100, and the other two variables sum  
 2330 to 200 (since  $a + b + c = 300$ ).  
 2331  
 2332 Therefore, the problem reduces to finding all nonnegative integer triples  $(a, b, c)$  such that one of them is 100,  
 2333 and the other two sum to 200. However, we also need to ensure that the original equation holds, but since we  
 2334 derived that this condition is equivalent to the original equation, all such triples will satisfy both equations.  
 2335 #####  
 2336 ##### Counting the number of valid triples #####  
 2337 Now, let's count the number of such triples.  
 2338 Case 1:  $a = 100$ . Then  $b + c = 200$ . The number of nonnegative integer solutions for  $(b, c)$  is 201 (from  $b=0$  to  
 2339  $b=200$ ). However, since triples are ordered, each solution here corresponds to a unique triple  $(100, b, c)$ .  
 2340  
 2341 Case 2:  $b = 100$ . Similarly,  $a + c = 200$ . Number of solutions: 201.  
 2342  
 2343 Case 3:  $c = 100$ . Similarly,  $a + b = 200$ . Number of solutions: 201.  
 2344  
 2345 However, we have to subtract the overlaps where two variables are 100. For example, if  $a = 100$  and  $b = 100$ ,  
 2346 then  $c = 100$ , which is counted in all three cases. There are three such overlapping triples:  $(100,100,100)$ ,  
 2347  $(100,100,100)$ , etc., but since it's the same triple, it's only counted once in each case. Wait, actually, when we  
 2348 count the three cases, the triple  $(100,100,100)$  is included in all three cases. So total overlaps are:  
 2349  
 2350 Each triple where two variables are 100 is counted in two cases. For example,  $(100,100,100)$  is counted in all  
 2351 three cases. Wait, no, if  $a = 100$ ,  $b = 100$ , then  $c = 100$ , so it's counted once in each case. Wait, no, when  $a =$   
 2352  $100$ ,  $b$  can be 100, which gives  $c = 100$ . Similarly for the other cases. So the triple  $(100,100,100)$  is counted  
 2353 three times in total (once in each case). Similarly, any triple where exactly two variables are 100 would be  
 2354 counted twice. Wait, but if two variables are 100, say  $a = 100$  and  $b = 100$ , then  $c = 100$ , which is the same as  
 2355 all three being 100. So there are no triples where exactly two variables are 100 because that forces the third to  
 2356 also be 100. Therefore, the only overlap is the triple  $(100,100,100)$ , which is counted three times.  
 2357  
 2358 Therefore, using inclusion-exclusion:  
 2359 Total triples =  $3*201 - 3*1 + 1*1 = 603 - 3 + 1 = 601$ .  
 2360 #####  
 2361 ##### Verifying the counting with a small example #####  
 2362 But let's check with a small example. Suppose we have variables summing to 3 instead of 300, and we want to  
 2363 count triples where one variable is 1. Then:  
 2364 Case 1:  $a=1, b + c = 2 \Rightarrow (1,0,2), (1,1,1), (1,2,0)$   
 2365 Case 2:  $b=1, a + c = 2 \Rightarrow (0,1,2), (1,1,1), (2,1,0)$   
 2366 Case 3:  $c=1, a + b = 2 \Rightarrow (0,2,1), (1,1,1), (2,0,1)$   
 2367 Total listed triples:  $(1,0,2), (1,1,1), (1,2,0), (0,1,2), (2,1,0), (0,2,1), (2,0,1)$ . That's 7 triples. Using inclusion-  
 2368 exclusion:  
 2369  $3*3 - 3*1 + 1 = 9 - 3 + 1 = 7$ . Correct. So the formula works.  
 2370 #####  
 2371 ##### Extract the final answer #####  
 2372 Thus, the final answer is 601.  
 2373 #####  
 2374

2376  
2377

## MLR Output (high-level) - GPQA (Diamond)

2378

**Problem:**

2379

Q: 3-(hydroxymethyl)-5-(prop-1-en-2-yl)cyclohexan-1-one is treated with sodium hydride, followed by benzyl bromide, forming product 1. 1 is treated with p-toluenesulfonyl hydrazide in the presence of a catalytic amount of HCl, forming product 2. 2 is treated at low temperature with n-butyllithium, followed by aqueous ammonium chloride, forming product 3. 3 is stirred with Pd/C under a hydrogen atmosphere, forming product 4. What is the structure of product 4?

2380

2381

2382

2383

## Choices:

2384

- A. (((3-isopropylcyclohexyl)methoxy)methyl)benzene
- B. (3-isopropylcyclohexyl)methanol
- C. N'-(3-(hydroxymethyl)-5-isopropylcyclohexyl)-4-methylbenzenesulfonylhydrazide
- D. 3-((benzyloxy)methyl)-1-butyl-5-isopropylcyclohexan-1-ol

2385

2386

2387

2388

2389

**Solution:**

2390

##### Understanding the problem and outlining the approach #####

2391

Need to determine the structure of product 4 in the reaction sequence.

2392

Approach: break down each step carefully to understand how product 4 is formed.

2393

##### Interpreting the starting material's structure #####

2394

The starting material is 3-(hydroxymethyl)-5-(prop-1-en-2-yl)cyclohexan-1-one.

2395

- The cyclohexanone ring has a hydroxymethyl group at position 3.

2396

- At position 5, there is a prop-1-en-2-yl group, which is  $\text{CH}_2=\text{C}(\text{CH}_3)-$  (a methyl group attached to a double bond).

2397

- The structure is a cyclohexanone ring with these two substituents at positions 3 and 5.

2398

#####

2399

##### Analyzing the first reaction: NaH and benzyl bromide #####

2400

NaH deprotonates the hydroxymethyl group to form an alkoxide, which then reacts with benzyl bromide to form a benzyl ether. Product 1 is 3-(benzyloxymethyl)-5-(prop-1-en-2-yl)cyclohexan-1-one, where the hydroxymethyl group is converted to a benzyloxymethyl group.

2401

#####

2402

##### Analyzing the second reaction: p-toluenesulfonyl hydrazide and HCl #####

2403

Product 1 reacts with p-toluenesulfonyl hydrazide ( $\text{TsNH}_2$ ) and catalytic HCl to form a tosylhydrazone. The ketone group (cyclohexanone) reacts with the hydrazide, losing water and forming a hydrazone. Product 2 is the tosylhydrazone of the ketone, with all other substituents (including prop-1-en-2-yl) unchanged. The structure is  $\text{N}'-(3-(\text{benzyloxymethyl})-5-(\text{prop-1-en-2-yl})\text{cyclohexylidene})-4-\text{methylbenzenesulfonylhydrazide}$ .

2404

#####

2405

##### Analyzing the third reaction: n-butyllithium and aqueous ammonium Product 2 is treated with n-butyllithium and aqueous ammonium chloride (Shapiro reaction), converting the ketone-derived hydrazone into an alkene. The cyclohexanone ring becomes a cyclohexene ring with a double bond between C1 and C2. Product 3 is a cyclohexene with a benzyloxymethyl group at C3 and an isopropenyl group at C5.

2406

#####

2407

##### Analyzing the fourth reaction: hydrogenation with Pd/C and H<sub>2</sub> #####

2408

- Hydrogenation with Pd/C and H<sub>2</sub> saturates all double bonds.

2409

- The cyclohexene ring becomes a cyclohexane ring.

2410

- The isopropenyl group ( $\text{CH}_2=\text{C}(\text{CH}_3)-$ ) is reduced to an isopropyl group ( $\text{CH}_2\text{CH}(\text{CH}_3)-$ ).

2411

- Product 4: cyclohexane ring with a benzyloxymethyl group at C3 and an isopropyl group at C5.

2412

- The ring is fully saturated; no other changes occur.

2413

#####

2414

##### Recapping the sequence and mapping substituent positions #####

2415

Starting material: cyclohexan-1-one

2416

Position 3: hydroxymethyl → benzyloxymethyl (after NaH/benzyl bromide)

2417

Position 1: ketone → tosylhydrazone → double bond (after Shapiro reaction)

2418

Ring: cyclohexene (double bond between 1 and 2)

2419

Position 5: prop-1-en-2-yl ( $\text{CH}_2=\text{C}(\text{CH}_3)-$ )

2420

Final product: 3-(benzyloxymethyl)-5-isopropylcyclohexane

2430 #####  
 2431 #####  
 2432 ##### Evaluating the effect of hydrogenolysis on the benzyl ether #####  
 2433 The benzyloxymethyl group should remain after hydrogenation with Pd/C, as benzyl ethers are generally  
 2434 stable under these conditions. Therefore, product 4 should have a benzyloxymethyl group at position 3 and  
 2435 isopropyl at position 5 on cyclohexane. None of the options match this structure.  
 2436 #####  
 2437 ##### Re-examining the possibility of benzyl ether cleavage under hydrogenation #####  
 2438 - Ketone is at position 1 on the cyclohexane ring.  
 2439 - Substituents at positions 3 (benzyloxymethyl) and 5 (isopropyl) are two positions apart.  
 2440 - After Shapiro elimination, double bond forms between positions 1 and 2.  
 2441 - Hydrogenation restores the single bond; substituents remain at positions 3 and 5.  
 2442 - In product 4, cyclohexane ring has benzyloxymethyl at position 3 and isopropyl at position 5.  
 2443 - This arrangement does not match any of the answer choices.  
 2444 #####  
 2445 ##### Considering the effect of hydrogenolysis on the benzyl ether group #####  
 2446 Step 1: alkylation to benzyloxymethyl—correct.  
 2447 Step 2: tosylhydrazone formation at ketone—correct.  
 2448 Step 3: Shapiro reaction forms alkene between positions 1 and 2—correct.  
 2449 Step 4: hydrogenation saturates both the 1–2 alkene and isopropenyl group to isopropyl.  
 2450 Final product: 3–(benzyloxymethyl)–5–isopropylcyclohexane.  
 2451 None of the options exactly match this structure.  
 2452 #####  
 2453 Starting material: 3–(hydroxymethyl)–5–(prop–1–en–2–yl)cyclohexan–1–one.  
 2454 - Tosylhydrazone forms at position 1 (the ketone).  
 2455 - Elimination forms an alkene between positions 1 and 2.  
 2456 - Substituents at 3 (hydroxymethyl) and 5 (prop–1–en–2–yl) remain.  
 2457 - Hydrogenation reduces the double bond (1,2) and the prop–1–en–2–yl group at 5.  
 2458 - Final product: cyclohexane ring with substituents at 3 (hydroxymethyl) and 5 (isopropyl).  
 2459 - Question: how to name this compound.  
 2460 #####  
 2461 ##### Applying IUPAC nomenclature and lowest locant rule #####  
 2462 After renumbering to give the substituents the lowest possible numbers, the hydroxymethyl group is at  
 2463 position 1 and the isopropyl at position 3. The correct IUPAC name is 1–(hydroxymethyl)–3–  
 2464 isopropylcyclohexane, also named as (3–isopropylcyclohexyl)methanol. Therefore, option B is correct.  
 2465 #####  
 2466 ##### Final conclusion and answer selection #####  
 2467 Numbering changes to give the lowest possible numbers, repositioning the substituents as 1 and 3 instead of 3  
 2468 and 5.  
 2469 Correct answer is B.  
 2470 #####  
 2471 ##### Extract the final answer #####  
 2472 Thus, the final answer is B.  
 2473 #####

#### MLR Output (low-level) - GPQA (Diamond)

##### Problem:

2477 Q: 3–(hydroxymethyl)–5–(prop–1–en–2–yl)cyclohexan–1–one is treated with sodium hydride, followed by  
 2478 benzyl bromide, forming product 1. 1 is treated with p–toluenesulfonyl hydrazide in the presence of a  
 2479 catalytic amount of HCl, forming product 2. 2 is treated at low temperature with n–butyllithium, followed by  
 2480 aqueous ammonium chloride, forming product 3. 3 is stirred with Pd/C under a hydrogen atmosphere,  
 2481 forming product 4. What is the structure of product 4?

##### Choices:

2482 A. (((3–isopropylcyclohexyl)methoxy)methyl)benzene  
 2483 B. (3–isopropylcyclohexyl)methanol

2484

C. N'-(3-(hydroxymethyl)-5-isopropylcyclohexyl)-4-methylbenzenesulfonohydrazide  
 D. 3-((benzyloxy)methyl)-1-butyl-5-isopropylcyclohexan-1-ol

2485

2486

2487

2488

**Solution:**

2489

##### Understanding the problem and outlining the approach #####

2490

Okay, so I need to figure out the structure of product 4 in this reaction sequence. Let me start by breaking down each step carefully.

2491

#####

2492

##### Interpreting the starting material's structure #####

2493

First, the starting material is 3-(hydroxymethyl)-5-(prop-1-en-2-yl)cyclohexan-1-one. Let's draw that out. The cyclohexanone ring has a hydroxymethyl group at position 3 and a prop-1-en-2-yl group (which is an allyl group attached at the middle carbon, so it's actually a methyl group attached to a double bond, right? So like  $\text{CH}_2-\text{C}(\text{CH}_2)-$  maybe? Wait, prop-1-en-2-yl would be  $\text{CH}_2=\text{C}(\text{CH}_3)-$ , right? So the substituent is a methyl group attached to a double bond. So the cyclohexanone ring has that at position 5 and the hydroxymethyl at position 3.

2494

#####

2495

##### Analyzing the first reaction: NaH and benzyl bromide #####

2496

Then, the first step is treatment with sodium hydride, followed by benzyl bromide. Sodium hydride is a strong base, so maybe it deprotonates something. The hydroxymethyl group has an -OH. Deprotonating that would give an alkoxide, which is a good nucleophile. Then benzyl bromide would act as an alkylating agent. So the benzyl group (Bn) would be attached to the oxygen, making a benzyl ether. So product 1 would be 3-(benzyloxymethyl)-5-(prop-1-en-2-yl)cyclohexan-1-one. That makes sense because the hydroxymethyl becomes benzyloxymethyl after alkylation.

2497

#####

2498

##### Analyzing the second reaction: p-toluenesulfonyl hydrazide and HCl #####

2499

Next, product 1 is treated with p-toluenesulfonyl hydrazide ( $\text{TsNHNH}_2$ ) with catalytic HCl. That's typically used to form a tosylhydrazone. The ketone group (cyclohexanone) would react with the hydrazide. The mechanism is that the ketone attacks the hydrazide, losing water and forming a hydrazone. So product 2 would be the tosylhydrazone of the ketone. So the cyclohexanone becomes a cyclohexylidene group linked to the tosylhydrazide. So structure would be N'-(3-(benzyloxymethyl)-5-(prop-1-en-2-yl)cyclohexylidene)-4-methylbenzenesulfonohydrazide. Wait, but maybe the prop-1-en-2-yl is still present? The reaction is only on the ketone, so the other substituents remain.

2500

#####

2501

##### Analyzing the third reaction: n-butyllithium and aqueous ammonium chloride (Shapiro reaction) #####

2502

Third step: product 2 is treated with n-butyllithium at low temperature, then aqueous ammonium chloride. This sounds like the Shapiro reaction. The Shapiro reaction uses tosylhydrazones treated with a strong base like n-BuLi to form alkenes via elimination. The mechanism involves deprotonation and elimination of the tosylhydrazide group, forming a double bond. So the cyclohexanone ring, which had the hydrazone, would become a cyclohexene ring? Wait, but the starting material already has an allyl group (prop-1-en-2-yl). Wait, maybe the Shapiro reaction here is converting the ketone-derived hydrazone into an alkene. But the original ketone is at position 1 (cyclohexan-1-one). So after Shapiro, that would form a double bond between C1 and C2? So the cyclohexanone becomes cyclohexene with the substituents. So product 3 would be 3-(benzyloxymethyl)-5-(prop-1-en-2-yl)cyclohexene, but the position of the double bond would be between C1 and C2? Wait, maybe I need to think about regiochemistry here. The Shapiro reaction typically gives less substituted alkenes because it's a concerted elimination. But in the cyclohexane ring, the original ketone is at position 1. So when eliminating, the hydrazone is at C1, and the adjacent hydrogen is at C2. So elimination would form a double bond between C1 and C2. But the ring already has substituents at C3 (benzyloxymethyl) and C5 (prop-1-en-2-yl). So after elimination, the ring becomes a cyclohexene with double bond between C1 and C2. So product 3 is 1,2-cyclohexenyl with substituents at C3 and C5. But the prop-1-en-2-yl is another substituent. Wait, but maybe the prop-1-en-2-yl is an isopropenyl group ( $\text{CH}_2=\text{C}(\text{CH}_3)-$ ). So product 3 would have a cyclohexene ring (double bond between 1 and 2), a benzyloxymethyl group at 3, and an isopropenyl group at 5. Then, when treated with n-BuLi, but after that, aqueous ammonium chloride. Wait, the Shapiro reaction usually forms alkenes via deprotonation and elimination. But maybe in this case, after forming the hydrazone, treatment with n-BuLi leads to deprotonation and elimination of the tosylhydrazide, giving the alkene. So product 3 would be the cyclohexene derivative with those substituents.

2503

#####

2504

##### Analyzing the fourth reaction: hydrogenation with Pd/C and H2 #####

2505

2506

2507

2508

2509

2510

2511

2512

2513

2514

2515

2516

2517

2518

2519

2520

2521

2522

2523

2524

2525

2526

2527

2528

2529

2530

2531

2532

2533

2534

2535

2536

2537

2538 Then, product 3 is stirred with Pd/C under hydrogen. Hydrogenation. So any double bonds would be  
 2539 hydrogenated. The cyclohexene ring's double bond (from the Shapiro reaction) would be hydrogenated to a  
 2540 single bond. Also, the prop-1-en-2-yl group (isopropenyl,  $\text{CH}_2=\text{C}(\text{CH}_3)-$ , so that's a vinyl group with a  
 2541 methyl substituent on the double bond) would be hydrogenated, making it a prop-2-yl group (isopropyl  
 2542 group). Wait, prop-1-en-2-yl is  $\text{CH}_2=\text{C}(\text{CH}_3)-$ , so hydrogenation would make  $\text{CH}_2\text{CH}(\text{CH}_3)$ . So the  
 2543 isopropenyl becomes isopropyl. Also, the cyclohexene from the Shapiro reaction (double bond between 1 and  
 2) would become cyclohexane again. So product 4 would have a cyclohexane ring with substituents: at C3,  
 2544 benzyl oxymethyl; at C5, isopropyl. But wait, after hydrogenation, the ring is fully saturated. Also, the  
 2545 hydroxymethyl group at 3 is now a benzyl oxymethyl. But after hydrogenation, is there any other change?  
 2546 #####  
 2547 ##### Recapping the sequence and mapping substituent positions #####  
 2548 Wait, let's recap: Starting material: cyclohexan-1-one (so position 1 is the ketone). Substituents: at 3,  
 2549 hydroxymethyl; at 5, prop-1-en-2-yl ( $\text{CH}_2=\text{C}(\text{CH}_3)-$ ). Step 1: deprotonate hydroxymethyl's OH with  $\text{NaH}$ ,  
 2550 then benzyl bromide gives benzyl oxymethyl at position 3. Step 2: form tosylhydrazone at the ketone (position  
 2551 1). So now, the hydrazone is at position 1. Step 3: Shapiro reaction ( $n\text{-BuLi}$ ) eliminates the tosylhydrazide,  
 2552 forming a double bond between position 1 and 2. So the ring becomes cyclohexene (positions 1 and 2 double  
 2553 bond). So now, the substituents are at 3 (benzyl oxymethyl), 5 ( $\text{CH}_2=\text{C}(\text{CH}_3)-$ ), and the ring has a double  
 2554 bond between 1 and 2. Step 4: hydrogenation with Pd/C and  $\text{H}_2$ . The double bonds in the molecule (both the  
 2555 cyclohexene ring's 1,2-double bond and the prop-1-en-2-yl group's double bond) would be hydrogenated.  
 2556 So the cyclohexene becomes cyclohexane (single bond between 1 and 2), and the prop-1-en-2-yl becomes  
 2557 isopropyl ( $\text{CH}(\text{CH}_3)_2$ ). So the substituent at position 5 becomes isopropyl. So product 4 would be 3-(  
 2558 benzyl oxymethyl)-5-isopropylcyclohexane. Wait, but cyclohexane is a six-membered ring. The substituents  
 2559 are at positions 3 and 5. But then, after hydrogenation, the ring is fully saturated. Also, the hydroxymethyl  
 2560 was at 3, which became benzyl oxymethyl. But after hydrogenation, is there any other change?  
 2561 #####  
 2562 ##### Evaluating the effect of hydrogenolysis on the benzyl ether #####  
 2563 Wait, but in the final hydrogenation step, maybe the benzyl oxymethyl group is hydrogenated? But benzyl  
 2564 ethers are generally stable under hydrogenation unless there's a catalyst that cleaves ethers. But Pd/C usually  
 2565 doesn't cleave ethers. So the benzyl oxymethyl should remain. So product 4 should have a benzyl oxymethyl  
 2566 group at position 3 and isopropyl at position 5 on cyclohexane. But none of the options have that.  
 2567 #####  
 2568 ##### Re-examining the possibility of benzyl ether cleavage under hydrogenation #####  
 2569 Wait, maybe I messed up the numbering. Let me try to visualize the cyclohexane ring. Let's imagine the  
 2570 ketone at position 1. Then positions 2, 3, 4, 5, 6 go around the ring. So if the ketone is at position 1, then  
 2571 substituents at 3 and 5 would be two positions apart. After elimination (Shapiro), the double bond is between  
 2572 1 and 2. Then hydrogenation would bring back the single bond. So the substituents at 3 and 5 are in the same  
 2573 positions. So in product 4, the cyclohexane ring has a benzyl oxymethyl group at position 3 and an isopropyl  
 2574 group at position 5. But the answer choices don't have that.  
 2575 #####  
 2576 ##### Considering the effect of hydrogenolysis on the benzyl ether group #####  
 2577 Wait, but let me double-check the steps again. Maybe I made a mistake. Step 1: alkylation of hydroxymethyl  
 2578 to benzyl oxymethyl. Correct. Step 2: formation of tosylhydrazone at the ketone. Correct. Step 3: Shapiro  
 2579 reaction with  $n\text{-BuLi}$ . The Shapiro reaction involves deprotonation adjacent to the hydrazone, leading to  
 2580 elimination of the tosylhydrazide and formation of an alkene. So the ketone (position 1) becomes a double  
 2581 bond between positions 1 and 2. Step 4: hydrogenation of the alkene (positions 1-2) and the isopropenyl  
 2582 group (position 5) to isopropyl. So product 4 would have a cyclohexane ring with substituents: at position 3,  
 2583 benzyl oxymethyl; at position 5, isopropyl. So the structure is 3-(benzyl oxymethyl)-5-isopropylcyclohexane.  
 2584 But looking at the options, none of them exactly match.  
 2585 #####  
 2586 ##### Realizing benzyl ether cleavage under hydrogenation conditions #####  
 2587 Wait, maybe I messed up the positions during the Shapiro reaction. Let's think again. The starting material is  
 2588 3-(hydroxymethyl)-5-(prop-1-en-2-yl)cyclohexan-1-one. When we form the tosylhydrazone at position 1  
 2589 (the ketone), then eliminate to form the alkene between positions 1 and 2. Now, the ring has a double bond  
 2590 between 1 and 2. The substituents at 3 and 5 are still in their original positions. Then, hydrogenation reduces  
 2591 the double bond between 1 and 2 and the prop-1-en-2-yl group at 5. So after hydrogenation, the ring is  
 cyclohexane with substituents at 3 (hydroxymethyl) and 5 (isopropyl). But how is this named?  
 2592 #####  
 2593 ##### Applying IUPAC nomenclature and lowest locant rule #####  
 2594

2592 Alternatively, maybe the numbering changes when the ketone is removed. Originally, the ketone is at position  
 2593 1. After converting it to CH<sub>2</sub> via hydrogenation, maybe the numbering starts from the hydroxymethyl group  
 2594 as position 1. In IUPAC nomenclature, the numbering is done to give the substituents the lowest possible  
 2595 numbers. So if the hydroxymethyl is at position 1 and the isopropyl at position 3, that would be lower than if  
 2596 they were at 3 and 5. So the new numbering would be: 1: hydroxymethyl (original 3) 2: CH<sub>2</sub> (original 4) 3:  
 2597 isopropyl (original 5) 4: CH<sub>2</sub> (original 6) 5: CH<sub>2</sub> (original 1) 6: CH<sub>2</sub> (original 2) Wait, this might not be  
 2598 accurate. Let me think of the cyclohexane as a chair. If the original positions are 1 (top), 2, 3, 4 (bottom), 5, 6.  
 2599 After removing the ketone (position 1 becomes CH<sub>2</sub>), and having substituents at 3 and 5. If we renumber to  
 2600 give the lowest numbers, we need to choose a direction where the substituents are at lower positions. If we  
 2601 start numbering from the hydroxymethyl (original 3) as position 1, then the isopropyl is at original 5, which  
 2602 would be position 3 in the new numbering (since it's two positions away in the ring). This gives substituents at  
 2603 positions 1 and 3, which is lower than 3 and 5. Therefore, the correct IUPAC name would be 1-(  
 2604 hydroxymethyl)-3-isopropylcyclohexane, which is named as (3-isopropylcyclohexyl)methanol. Because the  
 2605 methanol group (hydroxymethyl) is considered as the substituent at position 1, and the isopropyl at position 3.  
 2606 Thus, option B is correct.  
 2607 #####  
 2608 ##### Final conclusion and answer selection #####  
 2609 Ah, so the key here is that the numbering changes to give the lowest possible numbers, which would  
 2610 reposition the substituents as 1 and 3 instead of 3 and 5. Therefore, the correct answer is B.  
 2611 #####  
 2612 ##### Extract the final answer #####  
 2613 Thus, the final answer is B.  
 2614 #####

## MLR Output (high-level) - BoardGameQA (Hard)

### Problem:

A few players are playing a boardgame. The current state of the game is as follows. The dachshund has a football with a radius of 15 inches. And the rules of the game are as follows. Rule1: If the dachshund has a football that fits in a 36.6 x 33.1 x 40.7 inches box, then the dachshund brings an oil tank for the worm. Rule2: If at least one animal brings an oil tank for the worm, then the stork does not dance with the butterfly. Based on the game state and the rules and preferences, does the stork dance with the butterfly?

### Solution:

##### Recap the problem and rules #####  
 The dachshund has a football with a radius of 15 inches.  
 The football fits in a 36.6 x 33.1 x 40.7 inches box.  
 #####

##### Identify the key condition to check #####  
 Key condition: If the dachshund's football fits in the box, Rule 1 applies (dachshund brings oil tank), triggering Rule 2 (stork can't dance).  
 If the football doesn't fit, Rule 1 doesn't apply (no oil tank), so Rule 2 might not be triggered, allowing the stork to dance.  
 #####

##### Analyze the football's dimensions and assumptions #####  
 The football has a radius of 15 inches.  
 Assuming it is a sphere, the diameter is 30 inches.  
 The problem uses ``radius'', so the football is treated as a sphere.  
 #####

##### Compare the football's size to the box #####  
 Diameter of football: 30 inches.  
 Box dimensions: 36.6 x 33.1 x 40.7 inches.  
 $30 < 36.6$  (length),  $30 < 33.1$  (width),  $30 < 40.7$  (height).  
 The football fits in the box in all dimensions.  
 #####

##### Apply Rule 1 based on the fit #####  
 Diameter (30 inches) is less than all box dimensions, so the football fits.

2646 By Rule 1, the dachshund brings an oil tank for the worm.  
 2647 #####  
 2648  
 2649 ##### Apply Rule 2 and conclude #####  
 2650 Applying Rule 2: since at least one animal (the dachshund) brings an oil tank, the stork does not dance with  
 2651 the butterfly.  
 2652 #####  
 2653 ##### Extract the final answer #####  
 2654 Thus, the final answer is disproved.  
 2655 #####  
 2656

## MLR Output (low-level) - BoardGameQA (Hard)

### Problem:

A few players are playing a boardgame. The current state of the game is as follows. The dachshund has a football with a radius of 15 inches. And the rules of the game are as follows. Rule1: If the dachshund has a football that fits in a 36.6 x 33.1 x 40.7 inches box, then the dachshund brings an oil tank for the worm. Rule2: If at least one animal brings an oil tank for the worm, then the stork does not dance with the butterfly. Based on the game state and the rules and preferences, does the stork dance with the butterfly?

### Solution:

##### Recap the problem and rules #####

Okay, let's try to figure out whether the stork dances with the butterfly based on the given rules and the current game state.

First, let's recap the information provided. The dachshund has a football with a radius of 15 inches. The rules are:

If the dachshund's football fits in a 36.6 x 33.1 x 40.7 inches box, then the dachshund brings an oil tank for the worm.

If at least one animal brings an oil tank for the worm, then the stork does not dance with the butterfly.

The question is whether the stork dances with the butterfly. To determine that, we need to check if any animal (in this case, possibly the dachshund) brings an oil tank for the worm. If they do, then by Rule 2, the stork won't dance. If not, then maybe the stork does dance.

#####

##### Identify the key condition to check #####

So, the key here is whether the dachshund's football fits into the specified box. If it does, then Rule 1 applies, and the dachshund brings the oil tank, which triggers Rule 2, preventing the stork from dancing. If the football doesn't fit, then Rule 1 doesn't apply, so the dachshund doesn't bring the oil tank, and Rule 2 might not be triggered, allowing the stork to dance.

#####

##### Analyze the football's dimensions and assumptions #####

Now, let's focus on the football's dimensions. The football has a radius of 15 inches. Wait, a football is typically an elongated shape, like a prolate spheroid, but maybe in this game, it's considered a sphere? The problem mentions a radius, which is a term used for spheres. If it's a sphere with radius 15 inches, then the diameter would be 30 inches (since diameter is twice the radius). However, if it's an American football shape, the dimensions would be different. But since the problem specifies a radius, I think we can assume it's a sphere here.

#####

##### Compare the football's size to the box #####

So, diameter of the football is 30 inches. Now, the box has dimensions 36.6 x 33.1 x 40.7 inches. To fit the sphere inside the box, the sphere must fit in all three dimensions. That is, the diameter of the sphere must be less than or equal to each of the box's dimensions.

Checking each dimension:

Length: 36.6 inches. The diameter is 30, which is less than 36.6. So okay.

Width: 33.1 inches. 30 is less than 33.1. Okay.

Height: 40.7 inches. 30 is less than 40.7. Okay.

2700 #####  
 2701 #####  
 2702 ##### Apply Rule 1 based on the fit #####  
 2703 Since the diameter (30 inches) is less than all the box's dimensions, the football should fit inside the box.  
 2704 Therefore, according to Rule 1, the dachshund brings an oil tank for the worm.  
 2705 #####  
 2706 ##### Apply Rule 2 and conclude #####  
 2707 Then, applying Rule 2: since at least one animal (the dachshund) brings an oil tank, the stork does not dance  
 2708 with the butterfly.  
 2709 Therefore, the answer is that the stork does not dance with the butterfly.  
 2710 #####  
 2711 ##### Extract the final answer #####  
 2712 | Thus, the final answer is disproved.  
 2713 #####  
 2714 #####  
 2715  
 2716

### Long-horizon Reasoning Test Prompt Template

You are given multiple problems:

{Problem 1}

{Problem 2}

...

{Problem N}

Solve these problems in order. Please reason step by step for each problem, and at the end provide \*\*all final answers in a single JSON list\*\*, where each answer is wrapped in `\\boxed{}`. If you cannot solve a problem, write `\\boxed{N/A}` for that entry.

Output format:

```json  
 [\\boxed{answer1}, \\boxed{answer2}, ..., \\boxed{answerN}]

### MLR (high-level only) Output

#### Problem:

Define  $p = \sum_{k=1}^{\infty} \frac{1}{k^2}$  and  $q = \sum_{k=1}^{\infty} \frac{1}{k^3}$ . Find a way to write  $\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^3}$  in terms of  $p$  and  $q$ .

#### Solution:

##### Understanding the problem #####

The goal is to express

\[  
 \sum\_{j=1}^{\infty} \sum\_{k=1}^{\infty} \frac{1}{(j+k)^3}

in terms of  $(p)$  and  $(q)$ . The current stage involves understanding the problem and identifying the key components to work with.

#####

##### Recall the definitions of  $(p)$  and  $(q)$  #####

$(p = \sum_{k=1}^{\infty} \frac{1}{k^2})$  and  $(q = \sum_{k=1}^{\infty} \frac{1}{k^3})$  are defined as the sums of the reciprocals of squares and cubes, respectively.

```

2754 ##### Identify the desired expression #####
2755 The desired expression is:
2756 [
2757 \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4}
2758 ]
2759 This is the sum of the reciprocals of the fourth powers of the sums of  $(j)$  and  $(k)$ , from  $(j = 1)$  to  $(j = \infty)$  and from  $(k = 1)$  to  $(k = \infty)$ .
2760 #####
2761
2762 ##### Consider symmetry and possible simplifications #####
2763 The expression  $(\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4})$  involves summing the reciprocals of the fourth powers of the sums of  $(j)$  and  $(k)$ . The symmetry between  $(j)$  and  $(k)$  suggests potential simplifications or transformations. The current stage is focusing on identifying the key components and considering symmetry to simplify the expression.
2764 #####
2765
2766 #####
2767
2768 ##### Attempt to express the sum in terms of  $(p)$  and  $(q)$  #####
2769 The expression  $(\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4})$  involves summing the reciprocals of the fourth powers of the sums of  $(j)$  and  $(k)$ . The current stage is attempting to express this sum in terms of  $(p)$  and  $(q)$ , which are defined as  $(\sum_{k=1}^{\infty} \frac{1}{k^2})$  and  $(\sum_{k=1}^{\infty} \frac{1}{k^3})$ , respectively. The goal is to find a relationship between these sums and the given expression.
2770 #####
2771
2772 #####
2773
2774 ##### Recall known results for symmetric sums #####
2775 The expression  $(\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4})$  involves summing the reciprocals of the fourth powers of the sums of  $(j)$  and  $(k)$ . The goal is to express this sum in terms of  $(p)$  and  $(q)$ , where  $(p = \sum_{k=1}^{\infty} \frac{1}{k^2})$  and  $(q = \sum_{k=1}^{\infty} \frac{1}{k^3})$ . The current stage is exploring known results for symmetric sums and potential simplifications.
2776 #####
2777
2778 #####
2779
2780 ##### Attempt to simplify using known results #####
2781 The expression  $(\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4})$  involves summing the reciprocals of the fourth powers of the sums of  $(j)$  and  $(k)$ . The current stage is attempting to simplify this expression using known results for symmetric sums, focusing on possible relationships with  $(p)$  and  $(q)$ .
2782 #####
2783
2784 #####
2785
2786 ##### Consider using known results for symmetric sums #####
2787 The expression  $(\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4})$  is being considered in terms of  $(p)$  and  $(q)$ , where  $(p = \sum_{k=1}^{\infty} \frac{1}{k^2})$  and  $(q = \sum_{k=1}^{\infty} \frac{1}{k^3})$ . The goal is to find a relationship between  $(p)$  and  $(q)$  and the given sum.
2788 #####
2789
2790
2791 ##### Recall the identity for the sum of reciprocals of squares #####
2792 The identity for the sum of reciprocals of squares is:
2793 [
2794 \sum_{k=1}^{\infty} \frac{1}{k^2} = p
2795 ]
2796 where  $(p = \sum_{k=1}^{\infty} \frac{1}{k^2})$ .
2797 #####
2798
2799 #####
2800 ##### Recall the identity for the sum of reciprocals of cubes #####
2801 The identity for the sum of reciprocals of cubes is:
2802 [
2803 \sum_{k=1}^{\infty} \frac{1}{k^3} = q
2804 ]
2805 where  $(q = \sum_{k=1}^{\infty} \frac{1}{k^3})$ .
2806 #####
2807
2808 #####
2809 ##### Recall the identity for the sum of reciprocals of fourth powers #####
2810 The identity for the sum of reciprocals of fourth powers is:
2811 [
2812 \sum_{k=1}^{\infty} \frac{1}{k^4} = \frac{p^2}{2} + \frac{1}{4}q
2813 ]

```

2808 where  $\langle p = \sum_{k=1}^{\infty} \frac{1}{k^2} \rangle$  and  $\langle q = \sum_{k=1}^{\infty} \frac{1}{k^3} \rangle$ .  
 2809 #####  
 2810 ##### Apply the identity to the given sum #####  
 2811 The given sum  $\langle \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4} \rangle$  can be expressed in terms of  $\langle p \rangle$  and  $\langle q \rangle$  using the identity:  
 2812 
$$\langle \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4} = \frac{p}{2} + \frac{1}{4}q \rangle$$
  
 2813 #####  
 2814 ##### Verify the result with a specific example #####  
 2815 The given sum  $\langle \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4} \rangle$  can be expressed in terms of  $\langle p \rangle$  and  $\langle q \rangle$  using the identity:  
 2816 
$$\langle \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4} = \frac{p}{2} + \frac{1}{4}q \rangle$$
  
 2817 where  $\langle p = \sum_{k=1}^{\infty} \frac{1}{k^2} \rangle$  and  $\langle q = \sum_{k=1}^{\infty} \frac{1}{k^3} \rangle$ .  
 2818 #####  
 2819 ##### Alternative approach using generating functions #####  
 2820 The sum  $\langle \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4} \rangle$  can be expressed in terms of  $\langle p \rangle$  and  $\langle q \rangle$  as:  
 2821 
$$\langle \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4} = \frac{p}{2} + \frac{1}{4}q \rangle$$
  
 2822 where  $\langle p = \sum_{k=1}^{\infty} \frac{1}{k^2} \rangle$  and  $\langle q = \sum_{k=1}^{\infty} \frac{1}{k^3} \rangle$ .  
 2823 #####  
 2824 ##### Review the result for potential simplifications #####  
 2825 The sum  $\langle \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4} \rangle$  can be expressed in terms of  $\langle p \rangle$  and  $\langle q \rangle$  as:  
 2826 
$$\langle \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4} = \frac{p}{2} + \frac{1}{4}q \rangle$$
  
 2827 where  $\langle p = \sum_{k=1}^{\infty} \frac{1}{k^2} \rangle$  and  $\langle q = \sum_{k=1}^{\infty} \frac{1}{k^3} \rangle$ .  
 2828 #####  
 2829 ##### Finalize the answer #####  
 2830 The sum  $\langle \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4} \rangle$  can be expressed in terms of  $\langle p \rangle$  and  $\langle q \rangle$  as:  
 2831 
$$\langle \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^4} = \frac{p}{2} + \frac{1}{4}q \rangle$$
  
 2832 where  $\langle p = \sum_{k=1}^{\infty} \frac{1}{k^2} \rangle$  and  $\langle q = \sum_{k=1}^{\infty} \frac{1}{k^3} \rangle$ .  
 2833 Thus, the final answer is 
$$\boxed{\frac{1}{2}p + \frac{1}{4}q}$$
.  
 2834 #####  
 2835 Plan-and-Solve Prompt

2851 **Q:** Let  $x, y$  and  $z$  be positive real numbers that satisfy the following system of equations:  
 2852 
$$\log_2 \left( \frac{x}{yz} \right) = \frac{1}{2}$$
  
 2853 
$$\log_2 \left( \frac{y}{xz} \right) = \frac{1}{3}$$
  
 2854 
$$\log_2 \left( \frac{z}{xy} \right) = \frac{1}{4}$$
  
 2855 Then the value of  $|\log_2(x^4y^3z^2)|$  is  $\frac{m}{n}$  where  $m$  and  $n$  are relatively prime positive integers. Find  $m + n$ .  
 2856 **A:** Let's first understand the problem, extract relevant variables and their corresponding numerals, and  
 2857 make a complete plan. Then, let's carry out the plan, calculate intermediate variables (pay attention  
 2858 to correct numerical calculation and commonsense), solve the problem step by step, and show the  
 2859 answer.

2862  
 2863 **Output:**  
 2864 Given:  
 2865  $\log_2 \left( \frac{x}{yz} \right) = \frac{1}{2}$   
 2866  $\log_2 \left( \frac{y}{xz} \right) = \frac{1}{3}$   
 2867  $\log_2 \left( \frac{z}{xy} \right) = \frac{1}{4}$   
 2868 The value of  $|\log_2(x^4y^3z^2)|$  is  $\frac{m}{n}$  where  $m$  and  $n$  are relatively prime positive integers.  
 2869  
 2870 Plan:  
 2871 1. Convert each logarithmic equation to its exponential form.  
 2872 2. Use properties of exponents and logarithms to simplify the expressions.  
 2873 3. Combine like terms and solve for the values of  $x$ ,  $y$ , and  $z$ .  
 2874 4. Substitute these values into the expression  $|\log_2(x^4y^3z^2)|$ .  
 2875 5. Simplify this expression to find  $|\log_2(x^4y^3z^2)|$ .  
 2876 6. Express it as a fraction  $\frac{m}{n}$  with  $m$  and  $n$  being relatively prime.  
 2877 7. Find  $m + n$ .  
 2878  
 2879 Calculation:  
 2880 ##### Step 1: Convert each logarithmic equation to its exponential form  
 2881  
 2882 From the given equations, we have:  
 2883  $x = 2^{1/2}yz$   
 2884  $y = 2^{1/3}xz$   
 2885  $z = 2^{1/4}xy$   
 2886  
 2887 ##### Step 2: Use properties of exponents and logarithms to simplify the expressions  
 2888 We can rewrite each equation using the property of exponents:  
 2889  $x = \sqrt{2}yz$   
 2890  $y = \sqrt[3]{2}xz$   
 2891  $z = \sqrt[4]{2}xy$   
 2892  
 2893 ##### Step 3: Solve for the values of  $x$ ,  $y$ , and  $z$   
 2894  
 2895 Multiply all three equations together:  
 2896  $(\sqrt{2}yz)(\sqrt[3]{2}xz)(\sqrt[4]{2}xy) = (\sqrt{2})(\sqrt[3]{2})(\sqrt[4]{2})xyz^2$   
 2897  
 2898 This simplifies to:  
 2899  $x^2y^2z^2 = \sqrt{2}(\sqrt[3]{2})(\sqrt[4]{2})$   
 2900  
 2901 Since  $(\sqrt{2})(\sqrt[3]{2})(\sqrt[4]{2}) = 2^{(1/2+1/3+1/4)/12} = 2^{(6+4+3)/12} = 2^{13/12}$ , we get:  
 2902  $x^2y^2z^2 = 2^{13/12}$   
 2903  
 2904 Taking the square root on both sides:  
 2905  $xyz = 2^{13/24}$   
 2906  
 2907 Now, substitute back to find individual values:  
 2908  $x = \sqrt{2}yz$   
 2909  $y = \sqrt[3]{2}xz$   
 2910  $z = \sqrt[4]{2}xy$   
 2911  
 2912 ##### Step 4: Substitute these values into the expression  $|\log_2(x^4y^3z^2)|$   
 2913 Substitute  $x = \sqrt{2}yz$  into  $x^4y^3z^2$ :  
 2914  $x^4y^3z^2 = (\sqrt{2}yz)^4(yz)^3(z)^2 = 2^2(y^4z^4)y^3z^2 = 4(y^4z^4)y^3z^2 = 4(y^7z^6)$   
 2915 So,

2916  $|x^4y^3z^2| = 4(y^7z^6)$   
 2917  
 2918 Using the earlier result  $xyz = 2^{13/24}$ , we know:  
 2919  $y^7z^6 = \left(\frac{2^{13/24}}{x}\right)^7 \cdot \left(\frac{2^{13/24}}{y}\right)^6 = \left(\frac{2^{13/24}}{\sqrt{2}yz}\right)^7 \cdot \left(\frac{2^{13/24}}{\sqrt[3]{2}xz}\right)^6$   
 2920  
 2921 Simplifying further:  
 2922  $= \left(\frac{2^{13/24}}{\sqrt{2}yz}\right)^7 \cdot \left(\frac{2^{13/24}}{\sqrt[3]{2}xz}\right)^6 = \left(\frac{2^{13/24}}{\sqrt{2}yz}\right)^7 \cdot \left(\frac{2^{13/24}}{\sqrt[3]{2}xz}\right)^6 = \left(\frac{2^{13/24}}{\sqrt{2}yz}\right)^7 \cdot \left(\frac{2^{13/24}}{\sqrt[3]{2}xz}\right)^6$   
 2923  
 2924  
 2925 Finally:  
 2926  $|x^4y^3z^2| = 4(y^7z^6) = 4\left(\frac{2^{13/24}}{\sqrt{2}yz}\right)^7 \cdot \left(\frac{2^{13/24}}{\sqrt[3]{2}xz}\right)^6 = 4\left(\frac{2^{13/24}}{\sqrt[3]{2}yz}\right)^7 \cdot \left(\frac{2^{13/24}}{\sqrt[3]{2}xz}\right)^6 = 4\left(\frac{2^{13/24}}{\sqrt{2}yz}\right)^7 \cdot \left(\frac{2^{13/24}}{\sqrt[3]{2}xz}\right)^6$   
 2927  
 2928 Therefore:  
 2929  $|\log_2(x^4y^3z^2)| = \log_2(4) = 2$   
 2930  
 2931 Thus:  
 2932  $|\log_2(x^4y^3z^2)| = \frac{2}{1}$   
 2933  
 2934 **### Answer:**  
 2935  $m = 2, n = 1$   
 2936  $m + n = 2 + 1 = 3$   
 2937 The final answer is 3.  
 2938  
 2939  
 2940  
 2941  
 2942  
 2943  
 2944  
 2945  
 2946  
 2947  
 2948  
 2949  
 2950  
 2951  
 2952  
 2953  
 2954  
 2955  
 2956  
 2957  
 2958  
 2959  
 2960  
 2961  
 2962  
 2963  
 2964  
 2965  
 2966  
 2967  
 2968  
 2969