Rethinking RL Evaluation: Can Benchmarks Truly Reveal Failures of RL Methods?

Zihan Chen[*1,2], **Yiming Zhang**[*1,2], **Hengguang Zhou**[3], **Zenghui Ding**[†1], **Yining Sun**[1], **Cho-Jui Hsieh**[†3,4]

[1]HFIPS, Chinese Academy of Sciences [2]University of Science and Technology of China
[3]University of California, Los Angeles [4]Arena

Project: RL-GAP.github.io

Abstract

Current benchmarks are inadequate for evaluating progress in reinforcement learning (RL) for large language models (LLMs). Despite recent benchmark gains reported for RL, we find that training on these benchmarks' training sets achieves nearly the same performance as training directly on the test sets, suggesting that the benchmarks cannot reliably separate further progress. To study this phenomenon, we introduce a diagnostic suite and the Oracle Performance Gap (OPG) metric that quantifies the performance difference between training on the train split versus the test split of a benchmark. We further analyze this phenomenon with stress tests and find that, despite strong benchmark scores, existing RL methods struggle to generalize across distribution shifts, varying levels of difficulty, and counterfactual scenarios: shortcomings that current benchmarks fail to reveal.We conclude that current benchmarks are insufficient for evaluating generalization and propose three core principles for designing more faithful benchmarks: sufficient difficulty, balanced evaluation, and distributional robustness.

1 INTRODUCTION

Reinforcement Learning (RL) has emerged as a powerful paradigm for post-training Large Language Models (LLMs), significantly enhancing their capabilities on complex, multi-step reasoning tasks (Ouyang et al., 2022). Methods based on Reinforcement Learning from Human Feedback (RLHF) and Direct Preference Optimization (DPO) (Rafailov et al., 2023) have become standard practice for aligning LLMs. These paradigms are often powered by foundational algorithms like Proximal Policy Optimization (PPO) (Schulman et al., 2017), with state-of-the-art variants such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024) pushing models to achieve remarkable performance on benchmarks like GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). These successes, marked by state-of-the-art results (Lewkowycz et al., 2022; Lightman et al., 2023), suggest that RL-based alignment is a key pathway toward robust reasoning systems. Yet, a critical question remains: do current benchmarks meaningfully assess generalization? We find that the traditional assumption—that performance on unseen data measures generalization—may be insufficient for RL, as models trained on training splits perform nearly identically to those trained directly on test splits. This suggests that "unseen-ness" may no longer be a sufficiently discriminative criterion, calling for evaluations that go beyond disjoint splits to reveal deeper weaknesses.

To systematically investigate this phenomenon, we introduce an empirical evaluation framework for assessing whether the conventional train–test split remains a meaningful indicator of generalization for RL-trained models (Yu et al., 2025a). At the core of this framework, we define the Oracle Performance Gap (OPG) as a primary diagnostic measuring the performance difference between train-split-optimized and test-split-optimized models on the same benchmark. We further complement our analysis with targeted stress tests that probe whether high benchmark scores continue to correlate with robust generalization. These tests examine performance under variations in difficulty, question type, and counterfactual settings, revealing discrepancies between saturated benchmark performance and generalization behavior. Importantly, the observed trends are consistent with the Oracle Performance Gap analysis, jointly suggesting that near-ceiling benchmark scores alone may be insufficient to reliably assess generalization. These findings are then used to motivate three principles for benchmark design. Overall, our contributions are:

[*]Equal contribution. [†] Corresponding authors. Correspondence to: Zenghui Ding <dingzenghui@iim.ac.cn> and Cho-Jui Hsieh <chohsieh@cs.ucla.edu>.

Findings of the Association for Computational Linguistics: ACL 2026, pages 15692–15709
July 2-7, 2026

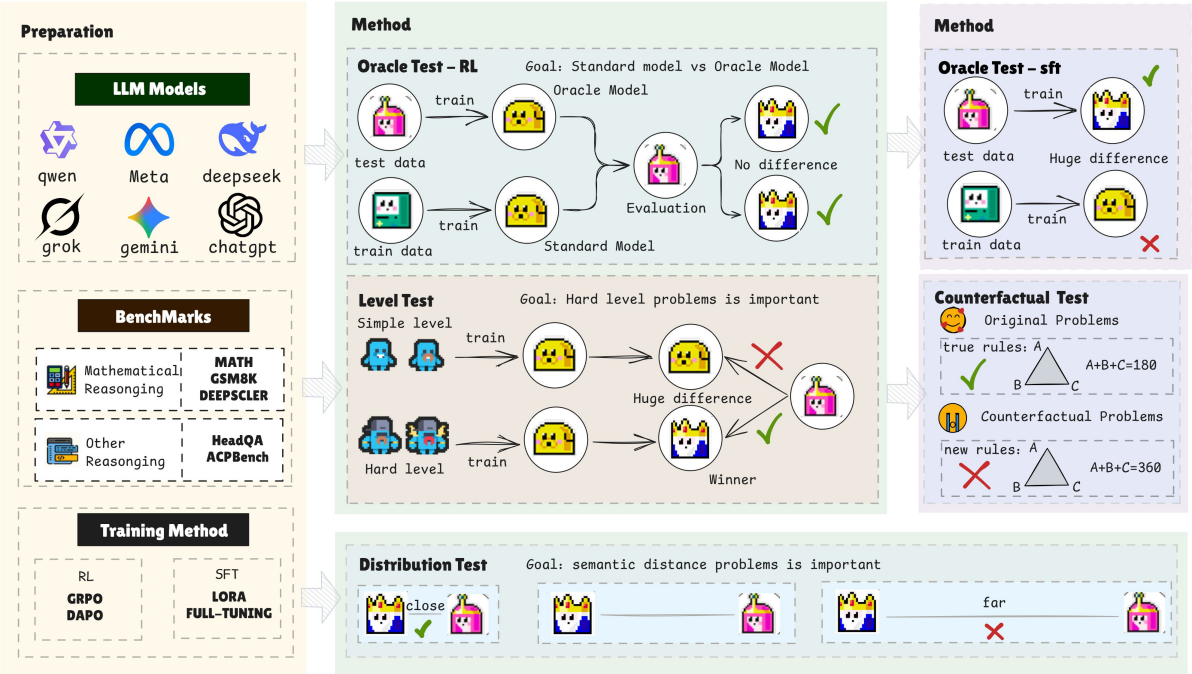


Figure 1: *Overview of our empirical framework.* The workflow begins by diagnosing benchmark flaws with novel metrics to uncover a core symptom: a vanishing generalization gap. It then proceeds through a suite of stress tests that reveal the brittle, shortcut-based nature of the learned skills, culminating in a new set of principles for more robust evaluation.

- **Illusion of Capability.** We present quantitative analyses suggesting that high performance on prevailing benchmarks does not necessarily correspond to robust generalization. Using the Oracle Performance Gap and aligned stress tests (including distributional and counterfactual evaluations), we identify structural limitations under which benchmarks assign strong scores despite observable discrepancies in generalization behavior. These findings highlight limitations in the reliability of current benchmarks scores as indicators of reasoning ability.
- **Novel Diagnostic Framework.** We introduce a new diagnostic framework, including the OPG and a set of evaluations (difficulty, distributional, and counterfactual), to systematically probe and quantify the extent to which benchmark scores remain informative about the generalization capability of RL models, rather than merely reflecting benchmark-specific fitting.
- **Actionable Design Principles.** Based on our findings, we propose a set of actionable principles for designing next-generation benchmarks that can more robustly evaluate an agent's true, transferable reasoning abilities under more challenging and realistic evaluation settings.

2 Diagnosing Generalization via OPG

The standard approach to evaluating LLM reasoning is to measure performance on a held-out test set, under the assumption that success on unseen data reflects generalization. To examine whether this assumption still holds for RL-based methods, we introduce a diagnostic framework that tests whether "unseen-ness", the common practice of relying on disjoint train/test splits, continues to provide a valid measure of generalization. Our framework compares RL models trained on training split with the Oracle model trained directly on the test split and finds that their performance is nearly identical, indicating that test-set "unseen-ness" alone has ceased to be a diagnostic signal of generalization.

2.1 Analysis Framework

2.1.1 Oracle Performance Gap (OPG)

We introduce the Oracle Performance Gap (OPG) as a diagnostic metric to audit the validity of a benchmark. Formally, let $P(M, \mathcal{D})$ denote the pass@1 accuracy of a model M on dataset $\mathcal{D}$ using algorithm $\mathcal{A} \in \{\text{SFT}, \text{RL}\}$. We define OPG as:

$$\text{OPG}_{\mathcal{A}} \triangleq \frac{P(M_{\mathcal{A},\text{test}}, \mathcal{D}_{\text{test}}) - P(M_{\mathcal{A},\text{train}}, \mathcal{D}_{\text{test}})}{P(M_{\mathcal{A},\text{test}}, \mathcal{D}_{\text{test}})}. \quad (1)$$

Table 1: *Benchmark Limitation Illustrated by Qwen2.5 Model Performance.* The table is reorganized by benchmark, comparing performance across 3B and 7B model scales.

Benchmark	Model Size	RL on Train Set Subsets (%)				RL Oracle ($M_{RL,test}$)	Baseline (M_{base})	OPG (%)
		10%	20%	50%	100%			
MATH	3B	$63.88_{\pm 1.04}$	$65.18_{\pm 0.84}$	$64.84_{\pm 1.25}$	$64.62_{\pm 0.98}$	$64.62_{\pm 1.11}$	62.20	0.00
	7B	$73.64_{\pm 0.48}$	$73.28_{\pm 0.68}$	$73.04_{\pm 0.91}$	$74.04_{\pm 0.39}$	$74.00_{\pm 0.68}$	68.80	-0.05
GSM8K	3B	$82.95_{\pm 0.38}$	$83.93_{\pm 0.47}$	$86.93_{\pm 0.34}$	$87.04_{\pm 0.49}$	$87.98_{\pm 0.35}$	83.02	1.07
	7B	$88.76_{\pm 0.47}$	$89.58_{\pm 0.44}$	$91.14_{\pm 0.30}$	$91.72_{\pm 0.31}$	$91.87_{\pm 0.31}$	88.40	0.16
DeepScaler	3B	$34.12_{\pm 0.94}$	$32.68_{\pm 1.02}$	$35.75_{\pm 0.96}$	$35.22_{\pm 0.89}$	$34.95_{\pm 0.91}$	33.38	-0.77
	7B	$42.05_{\pm 0.84}$	$42.09_{\pm 0.57}$	$42.84_{\pm 0.73}$	$42.36_{\pm 0.77}$	$42.64_{\pm 0.75}$	35.70	0.66
HeadQA	3B	$62.98_{\pm 0.77}$	$65.90_{\pm 1.19}$	$67.16_{\pm 0.79}$	$67.24_{\pm 0.59}$	$67.57_{\pm 0.70}$	54.96	0.49
	7B	$72.94_{\pm 0.90}$	$72.90_{\pm 0.79}$	$74.39_{\pm 0.60}$	$75.20_{\pm 0.66}$	$75.60_{\pm 0.50}$	52.24	0.53

Here, $M_{\mathcal{A},\text{train}}$ is the standard model and $M_{\mathcal{A},\text{test}}$ is the "Oracle" model fine-tuned explicitly on the test set. Unlike standard generalization gaps (which measure overfitting), OPG measures the *discriminative power* of the test set by quantifying the performance deficit against this Oracle baseline.

With OPG, we establish an upper bound for performance via memorization and assess whether the benchmark effectively challenges algorithm $\mathcal{A}$. We distinguish two outcomes:

1. **Effective Generalization (OPG$_{\mathcal{A}} \gg 0$):** A significant gap suggests that the benchmark poses a meaningful challenge, as the test set contains specific patterns or difficulties that cannot be trivially inferred from the training set.
2. **Weak Discriminative Signal (OPG$_{\mathcal{A}} \lesssim 0$):** A negligible gap indicates structural redundancy. It suggests the test set fails to differentiate between true generalization and simple pattern matching, as "seeing" the test data offers no performance advantage.

2.1.2 Experiment Setup

Our analysis spans four benchmarks: MATH, GSM8K, HeadQA, and DeepScaler. We use two base models from the `Qwen` family, `Qwen2.5-3B-Instruct` and `Qwen2.5-7B-Instruct`. To systematically isolate the effects of the fine-tuning paradigm and data distribution, we create and compare a suite of six model variants for each base model:

- **Baseline (M_{base}):** The original instruction-tuned model without any additional fine-tuning.
- **Standard SFT ($M_{SFT,train}$):** The base model fine-tuned on the official training set containing only standard question-answer pairs.
- **SFT with CoT ($M_{SFT,formatted}$):** The base model fine-tuned on a formatted training set that includes detailed, teacher-generated chain-of-thought (CoT) reasoning steps.
- **RL on Train Set ($M_{RL,train}$):** The base model fine-tuned on the official training set using GRPO, a state-of-the-art RL algorithm.
- **SFT Oracle ($M_{SFT,test}$):** The base model fine-tuned directly on the *test set* using SFT. This serves as a practical upper bound for SFT performance on the test distribution.
- **RL Oracle ($M_{RL,test}$):** The base model fine-tuned directly on the *test set* using the same GRPO setup. This provides an upper bound for the RL agent's ability to exploit the test set.

All models are evaluated on the official test sets using pass@1 accuracy. To ensure statistical rigor, we rigorously quantified performance stability by conducting 10 independent evaluation runs via sampling for each reported metric. Accordingly, we report the **Mean $\pm$ 95% Confidence Interval (CI)**, providing a robust measure of reliability. Full implementation details, hyperparameters, and evaluation protocols are provided in Appendix A.

2.2 Result

Finding 1: The Vanishing Generalization Gap Suggests Unseen-ness is an Insufficient Criterion. The OPG analysis reveals a stark contrast between SFT and RL paradigms, as detailed in Tables 1 and 2. While SFT models exhibit a large and expected OPG in a challenging generalization setting, this gap collapses to near-zero for RL-trained models. To rule out the concern that this is caused by data leakage in the base model, we verified that both our fine-tuned models significantly outperform the untrained baseline, confirming that this

Table 2: *SFT Performance Reveals the Expected Generalization Gap.* This table presents the SFT results organized by benchmark, with a direct comparison between the 3B and 7B model scales. All values are pass@1 accuracy.

Benchmark	**Model Size**	**SFT Performance Metrics (%)**			**OPG (%)**
		$M_{SFT,train}$	$M_{SFT,test}$	$M_{SFT,formatted}$	
MATH	3B	$17.20_{\pm 0.85}$	$40.00_{\pm 1.12}$	$31.02_{\pm 0.95}$	22.45
	7B	$23.60_{\pm 0.78}$	$64.20_{\pm 0.65}$	$42.00_{\pm 0.82}$	34.58
GSM8K	3B	$16.83_{\pm 0.45}$	$68.05_{\pm 0.41}$	$64.82_{\pm 0.39}$	4.75
	7B	$19.71_{\pm 0.42}$	$79.04_{\pm 0.33}$	$75.36_{\pm 0.35}$	4.66
DeepScaler	3B	$8.51_{\pm 0.92}$	$27.03_{\pm 0.98}$	$22.57_{\pm 0.94}$	16.50
	7B	$12.57_{\pm 0.85}$	$36.76_{\pm 0.76}$	$23.51_{\pm 0.80}$	36.04
HeadQA	3B	$10.45_{\pm 0.68}$	$54.96_{\pm 0.75}$	$41.22_{\pm 0.82}$	25.00
	7B	$12.80_{\pm 0.62}$	$52.24_{\pm 0.66}$	$35.52_{\pm 0.74}$	32.01

result reflects standard training behavior. To further assess if our conclusion is general, we also evaluated additional RL algorithms such as DAPO, as well as alternative architectures, domains, and inference settings; the corresponding results are presented in Appendix F.

Furthermore, to account for potential effect of the test set being much smaller than the training set, we trained RL models on various subsets of the training data. The OPG remained consistently low across all sizes, suggesting our conclusion is robust to the effects of data quantity. Together, these results suggest that the classical assumption—that performance on "unseen" test data is a sufficient measure of generalization—no longer holds for RL. This suggests that current benchmarks may no longer meaningfully assess future progress in RL generalization.

Takeaway ❶. Our OPG analysis reveals that RL models trained on the training split perform nearly identically to Oracle models trained directly on the test split. This performance convergence suggests that existing datasets likely suffer from structural redundancy, rendering the traditional criterion of test-set "unseen-ness" insufficient. Consequently, simple train-test splitting may no longer provide a sufficiently discriminative measure of true generalization for RL models.

Robustness beyond the primary setting. We additionally evaluated whether the near-zero OPG trend persists beyond the primary Qwen2.5-based mathematical setting. We found similarly small OPG values across an alternative RL algorithm (DAPO; Yu et al., 2025b), an additional open-weight model family (Llama-3-8B; Grattafiori et al., 2024), and non-mathematical reasoning domains including HotpotQA (Yang et al., 2018), MedQA (Jin et al., 2021), and LogiQA (Liu et al., 2020), as well as under varied KL coefficients and decoding parameters. These results provide further support that the observed vanishing-gap phenomenon is not specific to a single algorithm, architecture, domain, or inference setting. Detailed results are provided in Appendix F.

3 Benchmark Principles

Section 2 showed, via the Oracle Performance Gap, that standard benchmark evaluations may fail to reliably reflect generalization. In this section, we complement that analysis by evaluating models under more revealing evaluation settings that make specific dimensions of generalization explicit and provide insight into directions for improving benchmark design. By examining performance across difficulty levels, distributional shifts, and counterfactual conditions, we observe patterns consistent with the OPG findings: models achieving near-saturated benchmark performance can nonetheless exhibit pronounced performance degradation under these settings. Together, these analyses expose three structural limitations in existing benchmarks and motivate three corresponding principles for benchmark design: Cross-Difficulty Generalization, Distributional Robustness, and counterfactual Reasoning.

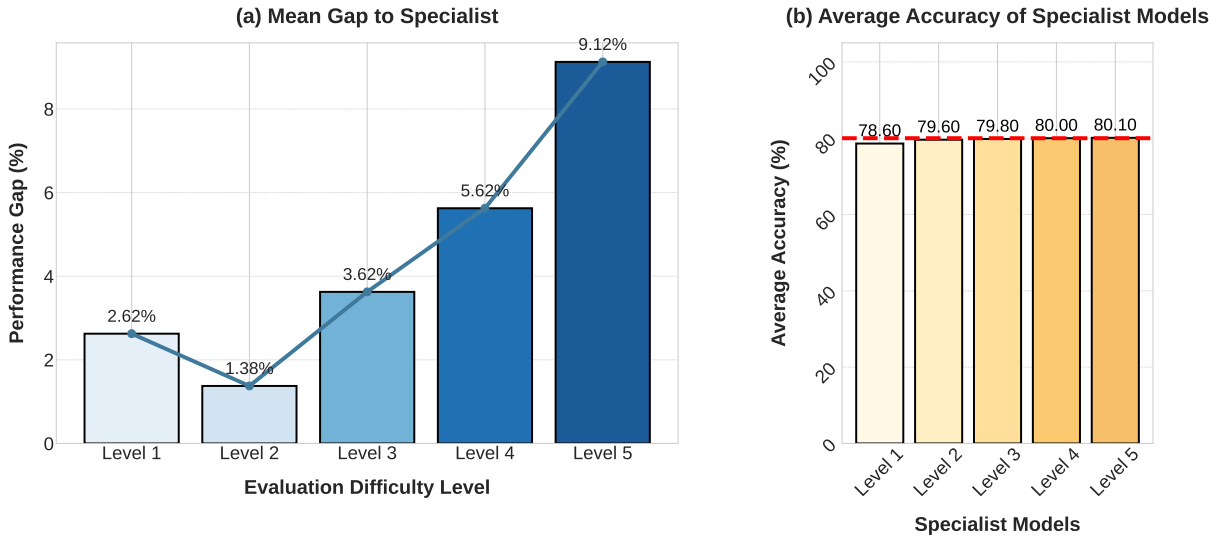


Figure 2: *The Illusion of Average Performance.* **(a)** The mean performance gap between the best (specialist) model and the average of all other models widens dramatically as task difficulty increases. **(b)** Surprisingly, the average scores of these specialists (calculated across all five difficulty partitions) are nearly identical. This contrast illustrates how a difficulty-agnostic evaluation can mask substantial differences in generalization capability. Full performance data is provided in Appendix B.2.

3.1 The Difficulty Test

One plausible factor contributing to the failure in evaluation discussed in Section 2 is that typical train–test splits do not account for variation in sample difficulty, instead summarizing performance by aggregating test instances across difficulty levels. This form of aggregation can obscure systematic differences in generalization behavior, particularly when failures are concentrated on more challenging cases and are diluted by easier ones. To make this dimension explicit, we stratify benchmarks by task difficulty and analyze performance beyond aggregate scores.

3.1.1 The Paradox of Average Scores

Setup. We conduct a cross-difficulty analysis by training five specialist models (M_{L_i}), each fine-tuned on a single difficulty partition of our constructed **MATH dataset** ($\mathcal{D}_{\text{train}}^{L_i}$, see Appendix B.1 for the partition protocol). Each specialist is then evaluated on all five training sets partitions, where a model's performance on its own training data serves as an oracle's performance benchmark against which its true generalization to unseen partitions is measured.

We observe that RL models exhibit asymmetric generalization: models trained on harder levels transfer well to easier ones, while those trained on easier levels struggle to generalize to harder tasks (Figure 3). Yet, when computing a single average score—mirroring the "vanishing generalization gap"—these models achieve nearly identical results (Figure 2(b)). This indicates that the OPG did not truly vanish; instead, it was concealed by difficulty-agnostic averaging. Thus, standard averaging masks meaningful differences, creating a misleading impression of equal capability that only stratified evaluation can reveal.

Finding 2: A difficulty-aware train–test split is an effective setting for evaluating generalization. Our cross-difficulty evaluation confirms failure modes concealed by average scores, suggesting that difficulty-aware partitioning is a superior paradigm for evaluation. More importantly, it reveals differences in transfer behavior that are invisible under standard aggregate metrics. This is supported by two key findings:

- **Masking Effect Confirmed:** Figure 2(b) reveals that significant capability variations are completely masked by final average scores. Although the specialist models differ substantially in their cross-difficulty behavior, the final average scores across them remain nearly identical.
- **Oracle Gap Re-emerges at the Micro-Level:** Contrary to the global near-zero OPG, micro-level analysis reveals a persistent gap. As shown in Figure 2(a), the gap against the specialist oracle (M_{L_j}) is non-zero; instead, it reappears and widens as task complexity increases. This shows that the Oracle Gap is not absent, but merely

hidden by aggregation.

Principle ❶. Our findings suggest that benchmarks should be explicitly stratified by difficulty. Standard aggregate metrics are insufficient, as they allow high success rates on trivial tasks to mask significant failures on complex ones. Instead of relying on a single average score, reporting performance across distinct difficulty levels enables a more granular assessment. This approach is essential for exposing hidden generalization weaknesses and ensuring that reported improvements reflect robust reasoning rather than superficial fitting to easy data.

3.1.2 The Impact of Training Difficulty

Setup and Phenomenon. To investigate the impact of training data difficulty on final generalization, we conduct a complexity test. We first train five generalist-optimized models, M_{L_i} for $i \in \{1, \ldots, 5\}$, on the previously defined difficulty-stratified training sets, $\mathcal{D}_{\text{train}}^{L_i}$. The key difference from our prior analysis lies in the evaluation protocol, which is centered around a novel, balanced test set.

- **`Test_Balanced`:** This is the unified and balanced evaluation suite, constructed by sampling an equal number of problems from each of the five difficulty levels. This results in a test set $\mathcal{D}_{bal}$ composed of five equal-sized partitions, $\{\mathcal{D}_{\text{test, bal}}^{L_j}\}_{j=1}^{5}$.

Unlike the models in the first experiment, these models are "generalist-optimized," meaning we select the checkpoint for each M_{L_i} with the highest overall accuracy on the `Test_Balanced` set. We then analyze the relationship between the difficulty of the training data and the model's final average performance on this balanced benchmark.Detailed performance data for this experiment is provided in Appendix C.

Finding 3: Training on Difficult Problems Boosts Transferable Generalization. Our analysis reveals that models trained on higher difficulty levels (L4–L5) exhibit "downward compatibility," effectively solving simple tasks while retaining complex reasoning capabilities. In contrast, models trained on easier data fail to generalize upward (Figure 3). Consequently, beyond evaluation considerations, these observations point to the potential benefit of including more challenging problems in training sets to encourage broader generalization.

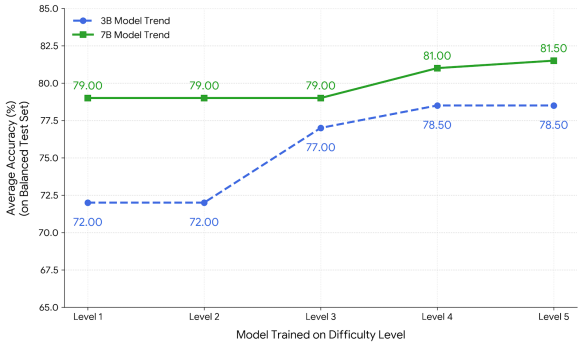


Figure 3: *Average performance on the balanced test set.* Consistent with Finding 3, we observe a positive correlation between training difficulty and generalization. Models trained on higher difficulty levels (L4–L5) consistently outperform those trained on easier data, yielding the strongest generalist performance.

3.2 Evidence for Improved Benchmarking

While the previous section highlighted the importance of problem complexity, difficulty is just one facet of generalization. To further probe our hypotheses regarding benchmark limitations and motivate our design principles, we introduce two stress tests for generalization forms: the Distribution Test (Section 3.2.1), quantifying brittleness to semantic shifts, and the Counterfactual Test (Section 3.2.2), distinguishing reasoning from memorization.

3.2.1 The Distribution Test

Setup. Standard train-test split typically evaluate models under an i.i.d. setting, in which generalization is assessed primarily within the training distribution. To evaluate generalization beyond this paradigm, we constructed a testbed using 44,785 problems from the benchmarks analyzed in Section 2. By clustering these via K-Means on embeddings, we modeled a spectrum from in-distribution to out-of-distribution scenarios based on semantic distance.

- **Core Training Set ($\mathcal{D}_{\text{core}}$):** We formed a concentrated training set by selecting the 2,000 problems closest to a cluster's centroid. This set was constructed primarily to serve as a distributional anchor to precisely identify OOD test sets based on semantic distance, while also simulating the strictly defined "seen" distribution of standard benchmarks(details in Appendix D).

- **Core-Trained Model (M_{core}):** We fine-tuned a specialist model exclusively on this dataset, for-

Table 3: *Validation of Performance Inversion across Model Scales.* This table validates the brittleness of RL-tuned models using Global Cosine Distance. Both 3B and 7B models exhibit a clear "Performance Inversion" trend: while they show gains on semantically close data (d1), these gains diminish and eventually turn into significant penalties on distant, out-of-distribution data (d5).

Metric / Bin	d1 (Closest)	d2	d3	d4	d5 (Farthest)	Trend
Qwen2.5-3B Gain	+2.25%	+1.25%	0.00%	-1.00%	-3.75%	↘ (Inverted)
Qwen2.5-7B Gain	+7.25%	+6.50%	+5.00%	+1.25%	-2.50%	↘ (Inverted)

mally $M_{\text{core}} \triangleq T(M_{base}, \text{RL}, \mathcal{D}_{\text{core}})$. This model is designed to be an expert solely on this specific distribution.

Finally, five test sets, $\{\mathcal{D}_{\text{test}}^{d_k}\}_{k=1}^{5}$, were constructed by sampling 80 problems each from the remaining data, which were binned according to increasing semantic distance d_k from the $\mathcal{D}_{\text{core}}$ centroid.

Hypothesis 1: Optimization for Specific Distributions Induces Brittle Generalization

We hypothesize that optimizing for a specific distribution yields brittle heuristics rather than robust skills. While we expect strong performance on data matching the training distribution (simulating benchmark conditions), we predict a **Performance Inversion** on out-of-distribution (OOD) data. We measure the specialist's gain over the baseline, $\text{Gain}(k) \triangleq P(M_{\text{core}}, \mathcal{D}_{\text{test}}^{d_k}) - P(M_{base}, \mathcal{D}_{\text{test}}^{d_k})$, testing if the advantage vanishes and eventually reverses as semantic distance d_k increases:

$$\exists k \in \{1, \ldots, 5\} \quad \text{s.t.} \quad \text{Gain}(k) < 0 \quad (2)$$

Confirming this would demonstrate that high i.i.d. scores can mask harmful, non-generalizable biases instilled during fine-tuning.

Finding 4: I.I.D. Test-Set Performance Is Not a Reliable Indicator of Generalization. Our distribution test (Table 3) confirms that excelling on a static distribution can be actively harmful to robustness. While the specialist model (M_{core}) dominates on in-distribution data (simulating high benchmark scores), this advantage is revealed to be brittle: it vanishes with semantic distance and culminates in a performance inversion on OOD sets. Here, the specialist's accuracy collapses below that of the untuned baseline, demonstrating that what appears to be "capability" on i.i.d. test sets is often merely a harmful, non-generalizable bias.

Principle ❷. Incorporating Distributional Robustness. Our findings suggest that a faithful benchmark should go beyond in-distribution evaluation to actively probe for robustness against distributional shifts. It should include a spectrum of out-of-distribution (OOD) challenges to penalize brittle, over-specialized models.

3.2.2 The Counterfactual Robustness Test

Setup. To rigorously test whether our models perform genuine deductive reasoning or merely recite pre-trained knowledge, we designed a counterfactual robustness test. The experiment is constructed around the following key components:

- **Test Sets ($\mathcal{D}_{bal}$ and $\mathcal{D}_{cf}$):** Our experiment uses two test sets: a standard balanced set, $\mathcal{D}_{bal}$, from the MATH benchmark, and our primary evaluation set, $\mathcal{D}_{cf}$, which is created by transforming a subset of problems from $\mathcal{D}_{bal}$ (see Appendix E.1 for details).
- **Counterfactual Transformation ($c_{\text{real}} \rightarrow c_{\text{fake}}$):** The transformation process involves identifying a problem's core, real-world mathematical rule, c_{real}, and explicitly replacing it with a novel, contrary-to-fact premise, c_{fake}.
- **Evaluation Criterion:** A model's response is marked as correct only if it correctly and exclusively applies the explicitly stated counterfactual premise, c_{fake}. This strict criterion ensures we are measuring on-the-fly reasoning rather than answer correctness based on memorized knowledge.

We then evaluated our main RL-tuned models, `Qwen2.5-3B-MATH` and `Qwen2.5-7B-MATH`, on the new counterfactual set $\mathcal{D}_{cf}$.To assess the quality

Table 4: Performance collapse under standard and counterfactual evaluation.

Model	$\mathcal{D}_{bal}$ (%)	$\mathcal{D}_{cf}$ (%)
3B-MATH	64.20	36.00
7B-MATH	74.80	41.20

of the generated counterfactual set, we additionally conducted a human audit on 50 randomly selected samples, evaluated by three PhD students specializing in LLMs. The audit found that 93.34% of samples were unambiguous and 94.67% were solvable under the stated counterfactual rule (Appendix E.2).

Hypothesis 2: Models Prioritize Recitation Over Reasoning

We test the hypothesis that models default to reciting memorized knowledge (c_{real}) instead of reasoning from a novel premise (c_{fake}). A confirmation is indicated by a significant performance collapse on the counterfactual set, formalized as:

$$P(M_{\text{RL},train}, \mathcal{D}_{cf}) \ll P(M_{\text{RL},train}, \mathcal{D}_{bal}) \quad (3)$$

Finding 5: Counterfactual Failures Reveal Recitation Over Reasoning. Our counterfactual robustness test (Table 4) reveals a critical failure in models' ability to reason from novel premises. This is quantitatively evident in the severe performance degradation on the counterfactual test, where accuracies for our 7B and 3B models drop from 74.80% and 64.20% to 41.20% and 36.00%, respectively. A qualitative analysis of the model's chain-of-thought process confirms the cause of this failure (see Appendix E.3 for a detailed example). When presented with a problem that redefines the order of operations to PESAMD, the model completely disregards the new rule and defaults to the standard PEMDAS operations it has memorized. This provides definitive evidence that it operates as a pattern-matching engine that recites knowledge, rather than as a flexible, deductive reasoner.

Principle ❸: Assessing Counterfactual Reasoning. A faithful benchmark requires distinguishing true deduction from mere recitation. Our counterfactual test highlights a critical failure mode: when faced with novel, contrary-to-fact rules, models consistently default to reciting memorized knowledge rather than applying the new premise. Consequently, to penalize this brittle behavior and reward flexible reasoning, effective evaluation entails including problems that create a direct conflict between memorized priors and on-the-fly deduction.

4 Related work

4.1 Reasoning in Large Language Models

Chain-of-Thought (CoT) prompting has become a cornerstone for eliciting complex reasoning in Large Language Models (LLMs) (Wei et al., 2022; Zelikman et al., 2022). Along with advanced strategies like Tree of Thoughts (Wang et al., 2022; Yao et al., 2023; Yu et al., 2025a), these approaches improve reasoning by guiding models to generate step-by-step rationales. Furthermore, fine-tuning on high-quality reasoning datasets remains a critical method for instilling these skills directly into model parameters (Lewkowycz et al., 2022). While these efforts have driven remarkable performance improvements on popular benchmarks, our work diverges from this trend of score optimization. Instead, we critically interrogate the benchmarks themselves, arguing that the resulting gains are often an illusion created by structural flaws rather than a sign of true reasoning acquisition.

4.2 Reinforcement Learning for LLM

To overcome the limitations of Supervised Fine-Tuning (SFT), Reinforcement Learning (RL) actively optimizes LLMs by directly rewarding correct outcomes (Ouyang et al., 2022). Foundational algorithms like PPO (Schulman et al., 2017) and stability-enhancing methods like GRPO (Shao et al., 2024)—which leverages group-based comparisons—have significantly boosted benchmark scores. Recent work has also explored reducing interference during alignment in multi-objective settings (Lin et al., 2025). However, the reliance on outcome rewards is contested by process-based approaches that scrutinize reasoning steps to ensure true understanding (Li et al., 2025). Our work adds to this discourse by demonstrating that, even with advanced RL, structural benchmark flaws often lead to the reinforcement of brittle, non-

generalizable behaviors.

4.3 Analysis and Critique of Benchmarks

While benchmarks like GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) are vital for driving progress, research increasingly shows that models often exploit dataset artifacts and "shortcuts" rather than learning robust skills (Geirhos et al., 2020). This has motivated rigorous evaluation methods, such as testing on out-of-distribution (OOD) or adversarially perturbed examples (Jia and Liang, 2017), to probe true generalization. However, these methods typically stress model capabilities without diagnosing the underlying benchmark properties that permit such brittle learning. We contribute to this critical analysis by introducing diagnostic tools—specifically the Oracle Performance Gap (OPG) and difficulty-stratified evaluations—that provide quantitative evidence revealing high scores as illusions of capability, thereby motivating our proposed principles.

5 Conclusion

In this work, we critically analyze RL-based reasoning benchmarks, arguing that high scores may often reflect brittle shortcut learning rather than robust generalization. Our empirical framework, anchored by the OPG diagnostic, reveals structural limitations such as data homogeneity and redundancy. Furthermore, stress tests suggest substantial model fragility, as evidenced by asymmetric generalization across difficulty levels and failures on counterfactual tasks. Finally, we distill our findings into three principles for next-generation benchmarks: *difficulty stratification*, *distributional robustness*, and *counterfactual reasoning*.

Limitations

Model architectures evaluated in our main experiments are still concentrated on the Qwen2.5 family (3B and 7B). Although we additionally validated the near-zero OPG trend on another open-weight model family (Llama-3-8B) and under an alternative RL algorithm (DAPO), we have not exhaustively tested a broader range of open-weight architectures or closed-source frontier models.

Task domains in our primary analysis remain centered on reasoning-intensive settings, especially mathematics and related benchmarks. Although we further extended the evaluation to additional reasoning domains such as HotpotQA, MedQA, and LogiQA, it remains unclear whether the same pattern persists in modalities with substantially different evaluation dynamics, such as code generation or creative writing. Future work is needed to determine whether our findings reflect a broader trend in RL fine-tuning or remain concentrated in reasoning-heavy tasks.

Semantic partitioning for our Distribution Test relies on specific tools—the `all-mpnet-base-v2` sentence encoder and K-Means clustering—to define data splits. While we employed rigorous metrics like Silhouette scores to validate these clusters, alternative embedding models or distance metrics could potentially yield different semantic boundaries.

Automated data generation for the Counterfactual Robustness Test relies on a pipeline driven by Gemini 2.5 Pro(Comanici et al., 2025). While this enabled large-scale evaluation, the diversity and complexity of the generated counterfactual rules remain bounded by the capabilities of the generator model. Although we supplemented this pipeline with a human audit of sample quality, broader validation of counterfactual diversity and quality remains an important direction for future work. As noted in our Ethics Statement, we also did not conduct a full audit of latent societal biases in the datasets or models used.

Ethical Considerations

This work is motivated by the need to improve the scientific rigor of reinforcement learning evaluation. By examining the "illusion of capability" in current benchmarks, we aim to support the development of more robust and trustworthy reasoning systems.

Broader Impact and Trustworthiness. The deployment of RL-tuned models that perform well on static benchmarks but fail to generalize may pose risks in real-world applications, particularly in reasoning-intensive domains. Our findings highlight that high benchmark scores can mask brittle behaviors induced by over-specialization. By proposing stricter evaluation principles—such as distributional robustness and counterfactual testing—we advocate for evaluation settings that better distinguish robust reasoning from superficial pattern matching.

Data Usage and Compliance. Our experiments use publicly available academic datasets, including MATH, GSM8K, HeadQA, HotpotQA, MedQA,

and LogiQA, together with a compiled in-house dataset (DeepScaler) derived from public sources. All data was used solely for academic research purposes. We acknowledge that these datasets may inadvertently contain sensitive information or reflect historical biases. While we did not conduct a full audit of latent societal biases in the external datasets used in this study, we recognize this as an important direction for future work on equitable model evaluation.

Computational Resources. The experiments were conducted on a single server with 4 NVIDIA A100 GPUs. We adhered to efficient training practices to minimize unnecessary computational costs and environmental impact.

AI Assistance Declaration. In accordance with conference policies, we state that Large Language Models (specifically Gemini 2.5 Pro) were used in this work. Their use was limited to two functions: (1) supporting automated data annotation and counterfactual generation within our experimental pipeline, and (2) assisting with grammatical refinement and language polishing of the manuscript. All scientific concepts, experimental designs, and core intellectual contributions originated from the human authors.

Acknowledgements

We sincerely thank all the anonymous reviewers and (S)ACs for their constructive comments and helpful suggestions. This work was supported by the National Key Research and Development Program of China (Grant No. 2024YFF0507603) and the Anhui Provincial Major Science and Technology Project (Nos. 202303a07020006 and 202304a05020071).

References

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

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint*. ArXiv:2507.06261 [cs].

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *arXiv preprint*. ArXiv:2407.21783 [cs].

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 3843–3857. Curran Associates, Inc.

Ming Li, Jike Zhong, Shitian Zhao, Yuxiang Lai, Haoquan Zhang, Wang Bill Zhu, and Kaipeng Zhang. 2025. Think or not think: A study of explicit thinking in rule-based visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.16188*.

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

Liang Lin, Zhihao Xu, Junhao Dong, Jian Zhao, Yuchen Yuan, Guibin Zhang, Miao Yu, Yiming Zhang, Zhengtao Yao, Huahui Yi, Dongrui Liu, Xinfeng Li, and Kun Wang. 2025. OrthAlign: Orthogonal Subspace Decomposition for Non-Interfering Multi-Objective Alignment. *arXiv preprint*. ArXiv:2509.24610 [cs].

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. 2026. Curriculum Reinforcement Learning from Easy to Hard Tasks Improves LLM Reasoning. *arXiv preprint*. ArXiv:2506.06632 [cs].

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint*. ArXiv:2402.03300 [cs].

David Vilares and Carlos Gómez-Rodríguez. 2019. Head-qa: A healthcare dataset for complex reasoning. *arXiv preprint arXiv:1906.04701*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Chengzhang Yu, Yiming Zhang, Zhixin Liu, Zenghui Ding, Yining Sun, and Zhanpeng Jin. 2025a. Frame: Feedback-refined agent methodology for enhancing medical research insights. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7690–7704.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025b. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *arXiv preprint*. ArXiv:2503.14476 [cs].

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

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

Yaowei Zheng, Junting Lu, Shenzhi Wang, Zhangchi Feng, Dongdong Kuang, and Yuwen Xiong. 2025. Easyr1: An efficient, scalable, multi-modality rl training framework. `https://github.com/hiyouga/EasyR1`.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

A Full Evaluation Setup

A.1 Post-training Methods

Reinforcement Learning Reinforcement Learning (RL) has recently proven effective at steering large language models toward complex, multi-step objectives by optimizing policies with scalar reward signals (Zeng et al., 2025). For our main experiments, we use the `easy-r1` framework, a fork of the original `veRL` project (Zheng et al., 2025). We adopt its implementation of Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to fine-tune `Qwen2.5-7B-Instruct`, using final-answer correctness as the reward signal. Our RL configuration uses a learning rate of 1×10^{-6} with the AdamW optimizer and a weight decay of 1.0×10^{-2}. We generate 5 responses per prompt with a maximum sequence length of 4096 tokens, using a temperature of 1.0 and a top-p of 0.99. The model is updated with a global batch size of 16. KL-divergence regularization is enabled with a coefficient of 1.0×10^{-2}. We train for 5 epochs and select the checkpoint with the best validation performance. Additional robustness experiments involving alternative algorithms, architectures, and inference settings are described in Appendix F.

Supervised Fine-Tuning Supervised Fine-Tuning (SFT) remains a fundamental technique for adapting large pre-trained models by directly minimizing cross-entropy on high-quality datasets (Parashar et al., 2026). We use the LLaMA-Factory framework (Zheng et al., 2024), which is an extensible and user-friendly framework supporting multiple architectures and advanced optimization algorithms, to fine-tune our model on teacher-generated chain-of-thought traces. We use 1×10^{-6} as learning rate, the batch size is 512 and we train for 5 epoch to align with our RL settings.

A.2 Datasets and Benchmarks

Our primary analysis was conducted on the following four benchmarks, chosen to cover a range of mathematical and general reasoning tasks. Additional reasoning domains used in our robustness analyses are described in Appendix F.

- **MATH** (Hendrycks et al., 2021): A challenging dataset of 12,500 competition mathematics problems designed to test mathematical problem solving.
- **GSM8K** (Cobbe et al., 2021): A dataset of 8,500 high-quality, linguistically diverse grade-school math word problems created to measure multi-step reasoning.
- **HeadQA** (Vilares and Gómez-Rodríguez, 2019): A multiple-choice question answering dataset sourced from Spanish medical board exams, covering a wide range of topics and requiring specialized knowledge.
- **DeepScaler**: A proprietary in-house dataset created to evaluate specific mathematical reasoning abilities. It contains approximately 40,000 unique math problem-answer pairs compiled from sources such as AIME, AMC, Omni-MATH, and Still.

A.3 Implementation Details

All experiments were conducted on a single server equipped with 4 NVIDIA A100 (80GB) GPUs. Our implementation relies on PyTorch and the Hugging Face Transformers library.

B Detailed Data for Difficulty-Stratified Analysis

B.1 Automated Difficulty Level Annotation

To ensure a systematic and reproducible partitioning of our datasets into difficulty levels (L1-L5), we employed an automated annotation pipeline. Instead of relying on subjective manual labeling, we developed a detailed rubric based on the cognitive complexity required for each problem and used a large language model (`Gemini 2.5 Pro`) to assign a difficulty score to each problem in our corpus.

The process was guided by the five-level standard defined below. For each problem, the full text of this rubric was provided to the LLM, which was then prompted to return the single most appropriate difficulty level.

Level 1: Direct Application of Basic Rules. Problems that can be solved in one or two steps, where each step is a direct application of a basic formula or operational rule. The solution path is linear and requires minimal strategic planning.

Level 2: Identification of Standard Models. Problems that require identifying the correct standard model or general formula from a set of known methods. This tests for "pattern recognition" of classic problem types.

Level 3: Multi-Step, Cross-Conceptual Planning. Problems that cannot be solved by a

Table 5: Cross-Difficulty Generalization Performance Matrix for the *Qwen2.5-3B-Instruct* model. All values are pass@1 accuracy.

Trained on	Evaluated on Training Set of Level					
	Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Average**
Level 1	94.50%	85.00%	71.00%	66.00%	41.00%	71.50%
Level 2	93.00%	87.50%	73.00%	65.00%	42.50%	72.20%
Level 3	92.50%	86.00%	75.00%	66.00%	40.00%	71.90%
Level 4	92.50%	86.50%	72.00%	68.00%	43.00%	72.40%
Level 5	94.00%	87.00%	73.00%	62.00%	46.50%	72.50%
Original	92.00%	83.50%	69.50%	62.50%	43.50%	70.20%

Table 6: Cross-Difficulty Generalization Performance Matrix for the *Qwen2.5-7B-Instruct* model. All values are pass@1 accuracy.

Trained on	Evaluated on Training Set of Level					
	Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Average**
Level 1	97.00%	90.00%	78.00%	76.00%	52.00%	78.60%
Level 2	94.00%	91.50%	82.50%	76.00%	54.00%	79.60%
Level 3	95.50%	91.00%	83.50%	72.50%	56.50%	79.80%
Level 4	93.50%	88.50%	81.00%	80.00%	57.00%	80.00%
Level 5	94.50%	91.00%	78.00%	73.00%	64.00%	80.10%
Original	95.50%	87.50%	76.50%	74.00%	52.00%	77.60%

single standard model and require a coherent plan that links multiple concepts or steps, often from different mathematical areas.

Level 4: Application of Abstract Concepts. Problems requiring a deep understanding and flexible application of a major, abstract mathematical theory. The solution process is often non-intuitive and relies on a foundational result within a branch of mathematics.

Level 5: Axiomatic Reasoning and Creation. Problems that require reasoning "from first principles" within an axiomatic framework. This involves performing logical deductions, constructing proofs, or finding counterexamples based on the foundational rules of a mathematical structure.

The entire dataset was processed using a parallelized script with a thread pool executor to efficiently query the LLM API. The script included robust error handling and checkpointing to ensure the complete and accurate annotation of the corpus.

B.2 Result

This section provides the full cross-difficulty generalization performance matrices that form the basis for the analysis in Section 3.1.1 and the visualizations in Figure 2. Table 5 and Table 6 present the results for the `Qwen2.5-3B-Instruct` and `Qwen2.5-7B-Instruct` models, respectively.

The data highlights two key phenomena discussed in the main text. First, asymmetric generalization is evident: in Table 6, the model trained on Level 5 achieves 94.50% on Level 1, while the model trained on Level 1 only achieves 52.00% on Level 5. This pattern is mirrored in the 3B model, confirming that training on complex tasks induces a “downward compatibility” that simple training lacks. Second, the average score proves deceptive. As shown in the ‘Average‘ column, scores across specialists are remarkably similar (e.g., 78.60%–80.10% for 7B). This similarity masks a critical distinction: low-difficulty specialists inflate their averages via simple tasks, whereas high-difficulty specialists achieve their scores through genuine robustness, a nuance the aggregate metric fails to capture.

Table 7: Performance of *Qwen2.5-7B* generalist-optimized models on the balanced test set. Each row represents a model trained on a specific difficulty level (L_i), evaluated across test questions of all five difficulty levels.

Trained on	Evaluated on Test Set Questions of Level					
	Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Average**
Level 1	97.50%	90.00%	82.50%	75.00%	50.00%	79.00%
Level 2	95.00%	90.00%	80.00%	77.50%	47.50%	79.00%
Level 3	97.50%	85.00%	85.00%	77.50%	50.00%	79.00%
Level 4	97.50%	87.50%	85.00%	80.00%	55.00%	81.00%
Level 5	97.50%	92.50%	82.50%	82.50%	52.50%	81.50%
Original	97.50%	87.50%	82.50%	77.50%	50.00%	79.00%

Table 8: Performance of *Qwen2.5-3B* generalist-optimized models on the balanced test set. The performance decay for models trained on easy levels (L1, L2) is particularly pronounced.

Trained on	Evaluated on Test Set Questions of Level					
	Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Average**
Level 1	97.50%	82.50%	75.00%	72.50%	32.50%	72.00%
Level 2	95.00%	87.50%	80.00%	65.00%	35.00%	72.00%
Level 3	97.50%	90.00%	80.00%	72.50%	45.00%	77.00%
Level 4	95.00%	87.50%	87.50%	75.00%	47.50%	78.50%
Level 5	95.00%	87.50%	87.50%	75.00%	47.50%	78.50%
Original	92.50%	87.50%	77.50%	65.00%	22.50%	69.00%

C A Supplementary Experiment to the Difficulty Test

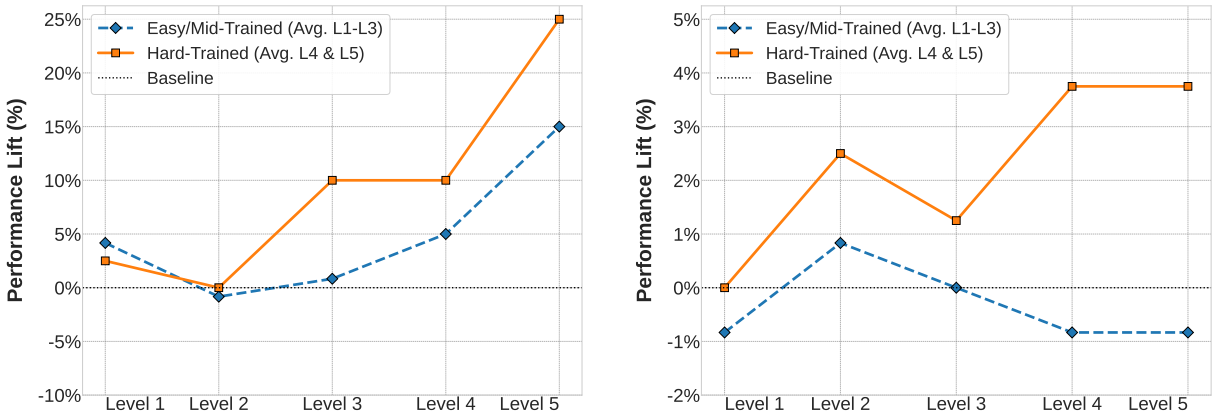
This appendix provides the full performance data for the "generalist-optimized" models described in our supplementary experiment on the difficulty test. The performance lift curves presented in Figure 4 in the main text are directly derived from the raw accuracy scores presented here. Table 7 details the results for the 7B model, while Table 8 shows the results for the 3B model.

Setup. To investigate the impact of training data difficulty on final generalization, we conduct a complexity test. We first train five generalist-optimized models, M_{L_i} for $i \in \{1, \ldots, 5\}$, on the previously defined difficulty-stratified training sets, $\mathcal{D}_{\text{train}}^{L_i}$. The key difference from our prior analysis lies in the evaluation protocol, which is centered around a novel, balanced test set.

- **`Test_Balanced`:** This is the unified and balanced evaluation suite, constructed by sampling an equal number of problems from each of the five difficulty levels. This results in a test set $\mathcal{D}_{bal}$ composed of five equal-sized partitions, $\{\mathcal{D}_{\text{test, bal}}^{L_j}\}_{j=1}^{5}$.

Unlike the models in the first experiment, these models are "generalist-optimized," meaning we select the checkpoint for each M_{L_i} with the highest overall accuracy on the `Test_Balanced` set.

Our complexity test reveals a stark pattern of asymmetric generalization, as illustrated in Figure 4. Models trained on high-difficulty problems (L4-L5) demonstrate a uniformly superior performance profile, outperforming their counterparts trained on easier data (L1-L3) across all evaluated task complexities. This suggests that complex reasoning skills naturally encompass the logic required for simpler tasks, whereas the reverse is not true. This finding has a critical implication for how we create datasets to train capable models: the training data must include a significant proportion of difficult problems. Therefore, for benchmark suites to drive meaningful progress, it is crucial that their provided training sets are sufficiently challenging to promote the development of truly robust models rather than merely encouraging the fitting of simple patterns.The data in these tables clearly illustrates this "asymmetric generalization" phenomenon. For example, in Table 8, the model trained on Level 1 (M_{L_1}) achieves high accu-



(a) Performance lift of the 3B model.

(b) Performance lift of the 7B model.

Figure 4: *Asymmetric Generalization is consistent across model scales.* Across both the 3B model (a) and the 7B model (b), training on high-difficulty problems (L4-L5, orange line) yields a uniformly superior performance lift over training on easier problems (L1-L3, blue line), proving that mastering complexity is essential for acquiring robust, transferable skills.Full performance data is provided in Table 8 and Table 7.

racy (97.50%) on Level 1 test problems but sees its performance drop to just 32.50% on Level 5 problems, indicating that the model relies on shallow heuristics that collapse under increased cognitive load. In contrast, the model trained on Level 5 (M_{L_5}) maintains robust performance across all levels, demonstrating a more generalizable capability that effectively transfers downwards to easier tasks.

D Data Construction Protocol for the Distribution Test

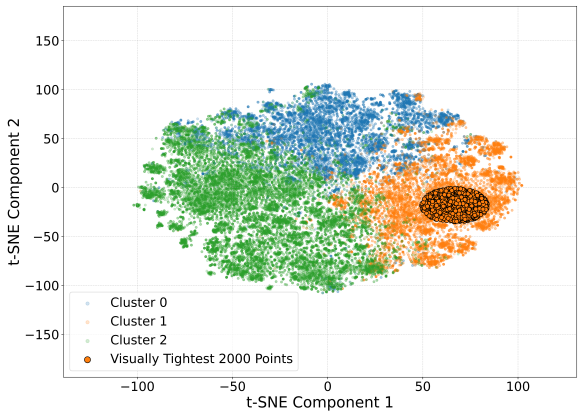
This section details the step-by-step procedure used to construct the specialized training and test sets for the Distribution Test, as described in Section 3.2.1. The entire process is designed to create a controlled environment for measuring generalization as a function of semantic distance. The process consists of three main stages:

Step 1: Semantic Embedding and Clustering. We began with our full corpus of approximately 44,785 mathematics problems. To understand their semantic relationships, we first encoded each problem into a high-dimensional vector representation using the `all-mpnet-base-v2` sentence encoder. We then applied K-Means clustering to this high-dimensional embedding space. Using a combination of the Elbow method and Silhouette score analysis, we determined the optimal number of clusters to be $k = 3$, effectively partitioning the entire dataset into three broad, semantically coherent groups.

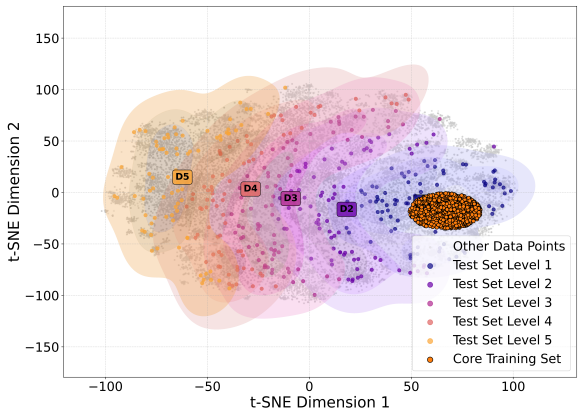
Step 2: Core Training Set (`Train_Core`) Selection. Our goal was to create a highly concentrated, semantically narrow training set. To achieve precise semantic selection, we performed the analysis using Global Cosine Distance directly on the original high-dimensional embeddings, avoiding potential information loss from dimensionality reduction. We first calculated the centroid of a single target cluster (e.g., Cluster 1) within the high-dimensional space. We then computed the cosine distance between this centroid and all data points in the cluster. The 2,000 problems with the smallest cosine distances—representing the points semantically closest to the cluster center—were selected to form our exclusive `Train_Core` training set.

Step 3: Distance-Stratified Test Set Construction. To systematically construct test sets representing a gradient of increasing semantic distance, we leveraged the remaining pool of 42,785 problems explicitly excluded from `Train_Core`. We first defined a stable reference origin by calculating the geometric centroid of the 2,000 `Train_Core` vectors within the high-dimensional embedding space. Subsequently, for every candidate point in the hold-out corpus, we computed its cosine distance relative to this core centroid to quantify its semantic divergence. All candidate points were then sorted by distance and stratified into five equal-sized bins (quintiles). Finally, to ensure a balanced evaluation, we randomly sampled 80 distinct problems from each bin to create our five final test sets, D1 (semantically closest) through D5 (semantically farthest).

The entire data construction pipeline is visually summarized in Figure 5. Panel (a) illustrates the



(a) Training set selection.



(b) Test set distribution.

Figure 5: *Visualization of the experimental data construction for the distribution test.* (a) The highly concentrated $\mathcal{D}_{\text{core}}$ set is selected from a semantic cluster. (b) The test sets are sampled and binned based on their increasing semantic distance from the $\mathcal{D}_{\text{core}}$ centroid.

outcome of the `Train_Core` selection process described in Step 2, while Panel (b) shows the resulting distribution of the five distance-stratified test sets as detailed in Step 3.

E The Counterfactual Robustness Test

This section provides detailed, qualitative examples of how fine-tuned models fail on counterfactual reasoning tasks, as discussed in Section 3.2.2. Each table analyzes a specific failure case, comparing the required reasoning path (based on the novel, counterfactual premise) with the model's actual thought process. These examples concretely illustrate the models' strong tendency to disregard explicit instructions and default to their pre-trained, memorized knowledge.

E.1 Methodology: Automated Dataset Generation

To ensure the diversity and systematic nature of our counterfactual examples, we developed and executed the following automated pipeline, moving beyond manual creation.

Step 1: Strategy — LLM as Data Creator. Our core strategy was to leverage a powerful Large Language Model to act as a creative research assistant. This approach allows for the large-scale and consistent application of complex transformation rules needed to create a high-quality counterfactual dataset.

Step 2:Task Definition — The Counterfactual Transformation. We provided the LLM (`Gemini 2.5 Pro`) with a detailed, multi-step prompt that precisely defined the transformation task. The instructions guided the model to first analyze a given standard problem to identify a core logical or mathematical rule. Subsequently, the model was tasked to invent a plausible but contrary-to-fact rule, rewrite the problem statement to include this new premise, and finally, generate a new step-by-step solution based exclusively on the novel rule.

Step 3: Execution — Parallelized Pipeline. This generation process was applied to our entire source dataset. To manage the scale, the pipeline was executed in parallel using a Python script with a `ThreadPoolExecutor` to handle concurrent API requests. The full, unabridged master prompt used in this process is available in our supplementary materials to ensure full reproducibility.

E.2 Human Audit of Counterfactual Data Quality

To quantitatively assess the quality of the generated counterfactual dataset, we conducted a human audit on 50 randomly selected samples. The audit was performed by three PhD students specializing in large language models. Each sample was evaluated according to the following criteria:

- **Unambiguous:** The counterfactual rule is explicitly stated and clearly overrides the model's pre-trained prior or default rule.
- **Solvable:** The problem contains sufficient information for a logically valid solution under the stated counterfactual rule.

The audit found that 93.34% of the sampled instances were judged to be unambiguous, and 94.67% were judged to be solvable. These results provide supporting evidence that the gener-

ated counterfactual test set is of sufficiently high quality for evaluating whether models follow explicitly stated novel rules rather than defaulting to memorized priors.

E.3 Case Study: Arithmetic Order of Operations

(Counterfactual Premise)

A novel order of operations, **PESAMD**, is defined: Parentheses, Exponents, **S/A**, then **M/D**. The model is asked to evaluate $f(x) = \frac{3x-2}{x-2}$.

Correct Reasoning (PESAMD)

1. **Numerator (S first):** $3 \times (0 - 2) = -6$
2. **Denominator:** $0 - 2 = -2$
3. **Division (last):** $\frac{-6}{-2} = 3$

The final correct answer is **3**.

Model's Actual Reasoning

1. **Numerator (M first):** $3 \times 0 = 0$, then $0 - 2 = -2$.
2. **Denominator:** $0 - 2 = -2$.
3. **Result:** $\frac{-2}{-2} = 1$.

The final incorrect answer is **1**.

E.4 Case Study: Number Theory Divisor Rule

(Counterfactual Premise)

A new system defines the number of divisors of $N = p_1^{a_1} \cdots$ as the **sum** of $(a_i + 1)$ values. Find the number of divisors for $N = 12$.

Correct Reasoning (Sum Rule)

1. Prime factorization of 12 is $2^2 \times 3^1$.
2. The exponents are $a_1 = 2, a_2 = 1$.
3. Apply the new **sum rule**: $(2 + 1) + (1 + 1) = 5$.

The final correct answer is **5**.

Model's Actual Reasoning

1. Correctly finds prime factorization: $12 = 2^2 \times 3^1$.
2. Ignores the "sum" rule and applies the memorized "product" rule: $(2 + 1) \times (1 + 1) = 6$.

The final incorrect answer is **6**.

E.5 Case Study: Physics Speed Formula

(Counterfactual Premise)

A car travels 120 km in 2 hours. In this reality, 'average speed' is calculated as: **speed = time / distance**. Find the speed.

Correct Reasoning (New Formula)

1. Identify Time = 2 hours, Distance = 120 km.
2. Apply the new formula time / distance: $2 \div 120 = \frac{1}{60}$.

The final correct answer is $\frac{1}{60}$ **km/h**.

Model's Actual Reasoning

1. Correctly identifies Time and Distance.
2. Ignores the new formula and applies the memorized, standard formula 'distance / time': $120 \div 2 = 60$.

The final incorrect answer is **60 km/h**.

F Additional Robustness Analyses for OPG

This section reports additional robustness analyses for the Oracle Performance Gap (OPG). We test whether the near-zero OPG trend persists across different RL algorithms, model families, task domains, and inference settings. Across all these settings, OPG remains small.

F.1 Across RL Algorithms: DAPO

We additionally evaluate DAPO to test whether the near-zero OPG trend extends beyond GRPO. Table 9 shows that the gap between Standard RL and Oracle RL remains minimal across GSM8K, DeepScaler, and HeadQA.

Table 9: Near-zero OPG under an alternative RL algorithm (DAPO).

Benchmark	Train	Oracle	Gap
GSM8K	86.40%	86.90%	0.50%
DeepScaler	43.85%	44.20%	0.35%
HeadQA	67.10%	67.78%	0.68%

F.2 Across Architectures and Domains

We further test whether the near-zero OPG phenomenon extends beyond the primary Qwen2.5-based mathematical setting. Table 10 shows similarly small OPG values on an additional open-weight model family, Llama-3-8B. Table 11 shows that the same trend also holds on non-mathematical reasoning domains, including HotpotQA, MedQA, and LogiQA.

Table 10: OPG on Llama-3-8B.

Benchmark	Train	Oracle	OPG (%)
GSM8K	87.45	88.40	1.07
MATH	43.75	43.82	0.16
DeepScaler	20.73	20.82	0.43

Table 11: OPG on additional reasoning domains.

Model	Domain	Train	Oracle	OPG (%)
Qwen2.5-7B	HotpotQA	77.85	78.05	0.26
Qwen2.5-7B	MedQA	78.60	79.00	0.51
Qwen2.5-7B	LogiQA	70.33	71.11	1.10
Llama-3-8B	HotpotQA	78.22	78.45	0.29
Llama-3-8B	MedQA	83.02	83.53	0.61
Llama-3-8B	LogiQA	76.48	77.19	0.92

F.3 Sensitivity to KL Coefficients

We vary the KL coefficient to examine whether the near-zero OPG trend depends on a particular regularization strength. Table 12 shows that OPG remains small across all tested KL settings.

Table 12: OPG under different KL settings.

Dataset	Setting	Train	Oracle	OPG (%)
GSM8K	1.0×10^{-3}	90.58	91.33	0.82
GSM8K	5.0×10^{-2}	91.24	91.47	0.25
MATH	1.0×10^{-3}	73.68	74.86	1.58
MATH	5.0×10^{-2}	73.65	73.86	0.28

F.4 Sensitivity to Decoding Parameters

We further vary decoding parameters, including temperature and top-p, to test whether the near-zero OPG trend is sensitive to the choice of inference configuration. Table 13 shows that OPG remains small across all tested settings, suggesting that the observed vanishing-gap phenomenon is robust to reasonable sampling variations at inference time.

Table 13: OPG under different decoding settings.

Dataset	Temperature	Train	Oracle	OPG (%)
GSM8K	0.7	91.02	91.33	0.34
GSM8K	0.9	91.74	92.07	0.36
MATH	0.7	74.05	75.06	1.35
MATH	0.9	74.08	75.10	1.36
Dataset	**Top-p**	**Train**	**Oracle**	**OPG (%)**
GSM8K	0.7	88.52	88.76	0.27
GSM8K	0.9	89.22	89.49	0.30
MATH	0.7	69.85	70.60	1.06
MATH	0.9	70.91	71.72	1.14