Reasoning Path Divergence: A New Metric and Curation Strategy to Unlock LLM Diverse Thinking

Feng Ju[1,2], Zeyu Qin[1], Rui Min[1], Zhitao He[1], Lingpeng Kong[3], Yi R. (May) Fung[1]

[1]**The Hong Kong University of Science and Technology**
[2]**University of Science and Technology of China**
[3]**The University of Hong Kong**

While Test-Time Scaling (TTS) has proven effective in improving the reasoning ability of large language models (LLMs), low diversity in model outputs often becomes a bottleneck; this is partly caused by the common *one problem, one solution* (1P1S) training practice, which provides a single canonical answer and can push models toward a narrow set of reasoning paths. To address this, we propose a *one problem, multiple solutions* (1PNS) training paradigm that exposes the model to a variety of valid reasoning trajectories and thus increases inference diversity. A core challenge for 1PNS is reliably measuring semantic differences between multi-step chains of thought, so we introduce Reasoning Path Divergence (RPD), a step-level metric that aligns and scores Long Chain-of-Thought solutions to capture differences in intermediate reasoning. Using RPD, we curate maximally diverse solution sets per problem and fine-tune Qwen3-4B-Base. Experiments show that RPD-selected training yields more varied outputs and higher pass@k, with an average +2.80% gain in pass@16 over a strong 1P1S baseline and a +4.99% gain on AIME24, demonstrating that 1PNS further amplifies the effectiveness of TTS.

1. Introduction

Large Language Models (LLMs) (Achiam et al., 2023; Chowdhery et al., 2023; Touvron et al., 2023) have made significant progress on tasks requiring complex reasoning that were previously challenging for automated systems, including competition-level mathematics and theoretical physics (Huang et al., 2025a; Wang et al., 2025a). Chain-of-Thought (CoT) prompting (Nye et al., 2021; Wei et al., 2022) plays an important role in that progress by eliciting step-by-step reasoning from language models. Built on CoT, Test-Time Scaling (TTS) methods, which have been widely adopted in recent research, further achieve substantial improvements on complex reasoning tasks, by generating multiple reasoning trajectories at inference time and selecting among them through techniques such as Best-of-N sampling (Brown et al., 2024; Song et al.,

2024) and self-consistency (Wang et al., 2022, 2025b). However, the effectiveness of TTS methods critically depends on the diversity of generated reasoning paths (Chen et al., 2025; Chow et al., 2025; Dang et al., 2025; Yao et al., 2025). If the model's reasoning paths show minimal variation, the gains from additional sampling remain limited.

This diversity bottleneck is partly a consequence of common training practices for reasoning: datasets typically pair each problem with a single solution, which, by habitually exposing the model to one pathway, effectively teaches it to converge on a single "correct" way of reasoning rather than to explore the space of valid alternatives. Consequently, models tend to adopt that canonical trajectory and seldom produce alternative reasoning paths. Several works have sought to counteract this by modifying objectives or introducing diversity-aware losses (Chen et al., 2025; Li et al., 2025b; Yao et al., 2025), yet it is still unclear how the diversity in the training examples maps onto the diversity of output at test time. That uncertainty motivates our core question:

Can a one problem, multiple solutions training paradigm effectively mitigate output homogenization and improve TTS performance?

In this work, we explore a pragmatic approach to address this diversity bottleneck: training models on datasets where each problem is paired with multiple distinct solutions. To build such datasets, we first need to solve a key challenge: how to reliably measure semantic diversity between complex reasoning paths. Common approaches, such as computing cosine similarity on embeddings (Reimers and Gurevych, 2019) of the entire solution text, fail for Long Chain-of-Thought solutions because they conflate high-level strategic differences with low-level computational details and narrative style. To tackle this issue, we introduce Reasoning Path Divergence (RPD), a new diversity metric that uses large language models to summarize solutions into their core logical steps and then applies an asymmetric matching process to measure semantic overlap. This approach allows RPD to tell true strategic novelty from superficial variations, forming a basis for systematically curating diverse data.

Using this metric, we chose the OpenThought3 dataset (Guha et al., 2025) for our experiments. It contains 53,125 math problems, each with 16 long chain-of-thought answers. These features make the dataset a well-suited testbed for our experiments on diversity-driven data selection.

Our main contributions in this work are:

- **A Novel Metric and Diversity Driven Curation Strategy.** We first propose and validate Reasoning Path Divergence (RPD), a novel metric for quantifying the semantic diversity between Long-CoT solutions. Building on this metric, we propose a novel data curation pipeline. This pipeline constructs a high-quality *one problem, multiple solutions* training set by selecting the most semantically distinct solutions for each problem.
- **Demonstrated Gains in Diversity and Performance.** Models fine-tuned on our multi-solution (1PNS) dataset achieve an average improvement of **2.80%** in *pass*@16 performance across challenging math benchmarks, highlighted by a peak gain of **4.99%** on the AIME24 benchmark, while simultaneously exhibiting higher output diversity as measured by our RPD metric. These gains confirm that our method alleviates the diversity bottleneck in Test-Time Scaling, boosting its overall performance.

Problem: Find the constant term of the polynomial $(x^2 + 2x + 1)(x^2 - 3x - 2) + (x^2 + 4x + 3)(x^2 - 2x - 1)$
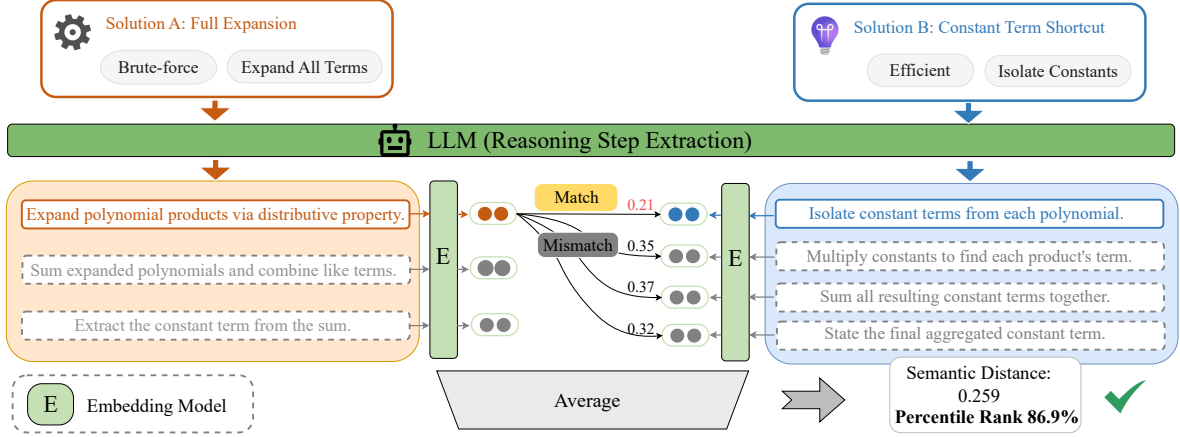
Figure 1 | The workflow of our Reasoning Path Divergence (RPD) metric. Given two solutions (A and B), an LLM first decomposes them into step-level summaries. An asymmetric matching is then performed: each step in the shorter summary (A) is matched to its semantically closest counterpart in the longer summary (B) based on embedding cosine distance. The final RPD score is the average of these minimum distances. Detailed examples with analysis is provided in Appendix C.

2. Related Work

Test-Time Scaling. A significant branch of Test-Time Scaling (TTS) focuses on improving performance by generating and aggregating multiple candidate solutions, which can be broadly divided into selection and fusion strategies. Selection-based methods choose the single best answer from a candidate pool. For example, some select the candidate with the highest verifier score, as in Best-of-N (Brown et al., 2024; Song et al., 2024), while others pick the most frequent answer via Majority Voting (Wang et al., 2022). To improve efficiency, certain studies filter candidates before final selection or voting (Chen et al., 2024; Munkhbat et al., 2025; Wu et al., 2025). Fusion-based methods, by contrast, merge multiple answers. This can be done by prompting an LLM to summarize the candidates (Jiang et al., 2023; Li et al., 2025a,c) or self-correct (He et al., 2025). A key challenge for these methods is low output diversity. Standard training often causes models to overfit to a single, canonical reasoning path, which limits the effectiveness of TTS.

LLM Generation Diversity. A large body of work confirms that standard supervised fine-tuning is detrimental to generation diversity (Chen et al., 2025; Li et al., 2025b; O'Mahony et al., 2024), prompting explorations into various training-phase optimizations to mitigate this issue, especially as recent studies establish a strong positive correlation between a model's solution diversity and its reasoning potential (Yao et al., 2025). These algorithm-centric approaches are varied, ranging from modifying the training objective with techniques like confidence regularization (Chen et al., 2025) or direct Best-of-N optimization (Chow et al., 2025), to altering the training process via sparse updates (Li et al., 2025b), checkpoint ensembling (Dang et al., 2025), and lightweight, diversity-aware parameter tuning (Chung et al., 2025). Complementing these effective, algorithm-centric strategies, our work explores a data-centric perspective aimed at directly enriching the reasoning diversity within the training data itself.

Data Curation. Curating high-quality, diverse datasets is essential for effective fine-tuning (Albalak et al., 2024). Most prior work targets *inter-problem* diversity, ensuring a broad mix of distinct problems, by synthesizing new questions at scale (Qin et al., 2025), using automated selection frameworks (Liu et al., 2024), removing semantic duplicates (Abbas et al., 2023), tuning domain mixtures (Xie et al., 2023), or scaling environment for agentic interactions (Huang et al., 2025b). By contrast, *intra-problem* diversity, teaching multiple ways to solve the same problem, remains under-explored. Closing this gap is the aim of our work.

3. Method

Enabling the *one problem, multiple solutions* training paradigm requires the ability to identify and select semantically distinct reasoning paths. To address this challenge, we introduce Reasoning Path Divergence (RPD), a new fine-grained metric for Long-CoT solutions. We then describe our 1PNS Curation Pipeline, which uses RPD to systematically build a high-diversity training set from the OpenThought3 dataset (Guha et al., 2025). This dataset contains 53,125 mathematical problems, each with 16 Long-CoT answers.

3.1. Reasoning Path Divergence (RPD): A Step-Level Diversity Metric

While prevailing approaches use embeddings over the entire solution to assess diversity, they struggle to adequately reflect solution diversity in Long-CoT reasoning, which is lengthy and structurally complex. Our RPD metric overcomes this limitation by analyzing the reasoning process at the step-summary level, computing embeddings over step summaries to capture semantic shifts in reasoning rather than superficial textual differences. For clarity, we illustrate the RPD computation framework in Figure 1, which comprises two core stages:

1. Reasoning Step Extraction. To reduce the complexity of Long-CoT reasoning paths and enable finer-grained diversity analysis, we first decompose the Long-CoT solutions into step-wise summaries. Formally, given two Long-CoT solutions S_A and S_B, we prompt an LLM (Qwen3-14B; Team, 2025) using the instructions provided in Appendix A.1 to split them into a short, ordered list of step summaries: $L_A = \{a_1, ..., a_m\}$ and $L_B = \{b_1, ..., b_n\}$.

2. Asymmetric Distance Computation. Next, we quantify the semantic distance between the two step lists using an asymmetric matching procedure. Each step summary is encoded into a high-dimensional vector using Qwen3-Embedding-8B (Zhang et al., 2025). Without loss of generality, we designate S_A as the shorter solution with $m \leq n$, and for each step a_i in S_A, we identify its closest semantic match in S_B by selecting the pair with the minimum cosine distance:

$$d_i = \min_{j=1,...,n} \left(1 - \frac{\vec{e}_{a_i} \cdot \vec{e}_{b_j}}{\|\vec{e}_{a_i}\| \|\vec{e}_{b_j}\|} \right) \tag{1}$$

The overall RPD score, $D(S_A, S_B)$, is calculated as the average of these minimum distances:

$$D(S_A, S_B) = \frac{1}{m} \sum_{i=1}^{m} d_i \tag{2}$$

Why is RPD superior to encoding the entire sentence? RPD's asymmetric design is robust because it accounts for semantic diversity in summarization granularity and mitigates the impact of step order on diversity analysis. In other words, it measures the extent to which the core logic of the shorter reasoning path is captured by the longer one. As a result, if one solution is merely

a more detailed restatement of another, the RPD score remains low, whereas fundamentally different reasoning strategies yield high scores. The formal definition and algorithmic details appear in Appendix A.2.

3.2. The 1PNS Curation Pipeline

Our pipeline processes the raw OpenThought3 dataset into a high-diversity 1PNS training set in two main phases.

Phase 1: Initial Quality Filtering. We started with 10,000 mathematical problems from OpenThought3. Since we lacked ground-truth labels, we applied a multi-stage filtering process to ensure data quality. This included length-based filtering to set a practical `max_new_tokens` limit for inference, and an LLM-based screening (using Qwen3-14B) to remove unclear problems and incomplete solutions missing final answers. This first phase gave us a high-quality candidate set of 1,600 problems, each with at least 10 candidate solutions that passed the screening. More details on the protocol are in Appendix B.1.

Before moving to the core selection step, we checked the natural diversity of this candidate set using a summary-based LLM Judge. As described in Appendix B.2, we used a Qwen3-14B model to evaluate the overall diversity of the solution summaries for each problem. The results showed a significant lack of diversity: for a majority of problems, **58%**, all solutions followed essentially the same single reasoning strategy, with only minor differences. This finding shows that having many solutions does not automatically mean they use diverse reasoning strategies, making a dedicated problem selection phase necessary.

Phase 2: Diversity-Driven Selection. This phase consists of a two-stage process guided by our RPD metric:

1. Problem Selection. We first rank problems by their intrinsic solution diversity. For each problem P with k solutions, we compute its overall diversity score, $\text{Score}_{\text{div}}(P)$, by averaging the pairwise RPD scores across all its unique solution pairs:

$$\text{Score}_{\text{div}}(P) = \frac{2}{k(k-1)} \sum_{1 \le i < j \le k} D(S_i, S_j)$$

We then select the top-N problems from this ranked list.

2. Solution Selection. For each of the top-N problems, we pick a set of M diverse solutions. We use a greedy selection that repeatedly adds the solution with the highest average RPD relative to the solutions already chosen.

This two-stage process results in a final training set rich in strategically diverse reasoning paths. The detailed algorithm is provided in Appendix A.3.

4. Experiments

To validate the core hypothesis of our work—that diversity-driven data curation can enhance a model's Test-Time Scaling (TTS) performance, we designed and conducted a series of experiments. Our evaluation had two parts. First, we measured how well the Reasoning Path Divergence (RPD) metric detects strategic diversity among solutions. Second, we measured how a training set selected using RPD affected a model's downstream `pass@k` performance.

4.1. RPD Metric Evaluation

Setup. To evaluate RPD's effectiveness in identifying semantically diverse reasoning paths, we randomly sample 100 problems and their solutions from the high-quality candidate set from our curation pipeline (Sec. 3.2). For each problem, each method selects the pair of solutions it considers most diverse. We use a separate LLM judge to assess whether the chosen pair employs different problem-solving strategies. The success rate is reported as the main evaluation metric. The reliability of the LLM judge is validated against human annotations (see Appendix D.1.2 for the full prompt and alignment study).

Methods Compared. We evaluate the following methods:

- **Random**: Randomly selects a pair of solutions, serving as a lower-bound baseline.
- **Raw Embedding (Raw Emb.)**: Selects the pair with the greatest cosine distance between the embeddings of the full solution texts.
- **Summary Embedding (Summary Emb.)**: Selects the pair with the greatest cosine distance between the embeddings of solution summaries.
- **LLM Selection**: A LLM (Qwen3-14B) selects the most diverse pair based on the summaries of all candidate solutions (see Appendix D.1.1 for details).
- **Ours (RPD)**: Our proposed asymmetric, step-level semantic distance metric.

Results and Analysis. As shown in Table 1, our RPD metric achieves a **53%** success rate, outperforming all baselines, including those based on raw embeddings (40%), summary embeddings (48%), and a powerful LLM selector (44%). These results offer two key insights. First, RPD's fine-grained, step-level analysis is crucial for overcoming the limitations of holistic embedding methods that conflate high-level strategy with superficial text. Second, RPD compares candidates in pairs systematically. This pairwise comparison is more reliable than heuristic LLM judgments for selecting the most diverse pair from many candidates. These results show that RPD works well as an automatic metric for our diversity-driven curation pipeline.

Table 1 | Effectiveness of various diversity metrics.

Method	Success Rate (%)
Random	27
Raw Emb.	40
LLM Selection	44
Summary Emb.	48
Ours (RPD)	**53**

4.2. Effectiveness of Multi-Solution Fine-Tuning

In this experimental section, we aim to answer the following research questions:

Q1: Does fine-tuning with the one problem, multiple solutions (1PNS) paradigm improve reasoning performance, as measured by `pass@k`, compared to the standard one problem, one solution (1P1S) setting?

Q2: Within the 1PNS paradigm, does selecting diverse solutions using our RPD metric yield higher `pass@k` than other selection methods?

4.2.1. Experimental Setup

Model. We use the Qwen3-4B-Base model (Team, 2025) for our primary experiments. To ensure the robustness of our findings, corresponding results for the Qwen2.5-3B model (Team, 2024) are provided in the Appendix E.2, and results for Llama-3.1-8B-Instruct (Dubey et al., 2024) are reported in Appendix E.3.

Benchmark. We evaluate the model's performance on three challenging mathematical reasoning benchmarks that align with our training data domain: AIME24[1], MATH500 Level 5 (Hendrycks et al., 2021), and Olympiad Bench[2] (He et al., 2024). Performance is measured using the `pass@k` metric.

Baselines. To comprehensively evaluate our diversity-driven data curation method, we conduct a comparison against several baselines. For our main experiments, we standardize the multi-solution format to **one problem, three solutions (1P3S)**. The impact of varying the number of solutions per problem is investigated in our ablation studies (Sec 4.2.3). For a fair comparison, all methods use the same total number of training instances—300.

Our proposed method, **Ours (RPD)**, constructs a training set of 100 problems and 3 solutions per problem, guided by our RPD metric's diversity scores. We compare it against the following baselines, which are grouped into two categories. The detailed construction methodology for each is provided in Appendix D.2.

1. Comparison of 1P1S vs. 1P3S paradigms.

- **Random 1P1S:** The standard SFT baseline, constructed by randomly selecting 300 unique problems and pairing each with one randomly chosen solution. This baseline is used to measure the fundamental performance gain of the 1P3S approach.

2. Comparison of diversity selection metrics (all using a 1P3S structure).

- **Random 1P3S:** A naive multi-solution approach. We randomly select 100 problems and use 3 randomly chosen solutions for each.
- **LLM Selection:** An LLM is prompted to select 100 problems and generate 3 diverse solutions for each.
- **Raw Embedding (Raw Emb.) :** We select the 100 problems and 3 corresponding solutions that maximize diversity based on the cosine distance between the embeddings of the full answer texts.
- **Summary Embedding (Summary Emb.):** We select data by maximizing the cosine distance between embeddings of AI-generated answer summaries for 100 problems and their 3 solutions.

Implementation Details. We fine-tune the Qwen3-4B-Base model using supervised fine-tuning with 4-bit QLoRA (rank=16, alpha=32). The model is trained for 12 epochs in BF16 precision on NVIDIA H20 GPUs. We use the AdamW optimizer with a batch size of 16 and a cosine learning rate scheduler, peaking at 5×10^{-5}. For inference, we use nucleus sampling (temperature=0.6, top_p=0.95) with maximum generation lengths tailored to each benchmark (14K for AIME24, 10K for MATH500, 8K for Olympiad). To ensure statistical robustness, we report average scores over multiple runs (4 for AIME24/MATH500, 2 for Olympiad).

4.2.2. Results and Analysis

We present the experiments in two parts to answer our research questions. First, we compare the *one problem multiple solutions* (1PNS) paradigm with the standard *one problem one solution* (1P1S) baseline. Second, we test our diversity metric against several alternative selection strategies.

---

[1] https://huggingface.co/datasets/Maxwell-Jia/AIME_2024

[2] For our evaluation, we selected an English, text-only, deterministic-answer mathematical subset of the Olympiad Bench to align with our training set.

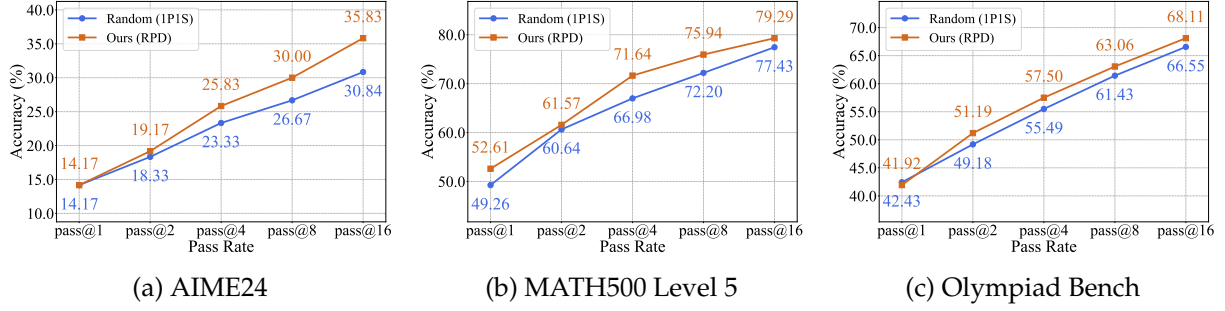(a) AIME24   (b) MATH500 Level 5   (c) Olympiad Bench

Figure 2 | Performance comparison of our 1P3S approach against the 1P1S baseline across three mathematical reasoning benchmarks. Each subplot corresponds to a different benchmark, showing the pass@k accuracy for k=1, 2, 4, 8, 16.

Table 2 | Comparison of different diversity selection methods on the MATH500 Level 5 benchmark. All methods except *Base* use a **1P3S** structure.

Method	pass@1 (%)	pass@2	pass@4	pass@8	pass@16
Base	46.08	56.90	64.37	71.27	75.00
Random (1P3S)	49.07	59.70	68.66	73.32	77.24
Raw Emb.	50.19	59.14	67.54	71.64	77.80
Summary Emb.	52.24	59.89	68.66	73.14	77.43
LLM Selection	49.81	58.96	66.23	73.51	77.61
Ours (RPD)	**52.61**	**61.57**	**71.64**	**75.94**	**79.29**

Q1: Superiority of the 1PNS Paradigm

We compare our 1P3S training method to the 1P1S baseline on three benchmarks (Figure 2). At `pass@1`, the two methods perform similarly. However, 1P3S outperforms the baseline for larger k. Across benchmarks, 1P3S achieves an average `pass@16` gain of **2.80%**, with the largest gain of **4.99%** on the AIME24 test. These results indicate that using multiple solutions per problem improves Test-Time Scaling on hard reasoning tasks.

Q2: Effectiveness of the RPD Metric

Next, we compare our diversity metric to other data selection strategies. All selection strategies use the 1P3S format (one problem, three solutions). Table 2 shows results on the MATH500 Level 5 benchmark. Results for AIME24 and the Olympiad benchmark are in Appendix E.1.

Table 2 shows that RPD-guided selection consistently outperforms the baseline methods on all `pass@k` metrics. Some methods, such as Summary Emb., are competitive at `pass@1`. However, our approach takes a clearer lead at larger k. For example, at `pass@4` our method leads by about **3.0%** over the next-best strategy. This gap shows that RPD better captures strategic diversity than whole-solution embedding distances or heuristic LLM selection.

4.2.3. Ablation Studies

We conduct a series of ablation studies to provide a comprehensive analysis of our method and its properties. We begin by evaluating our method's impact on the diversity of generated solutions. We then analyze the sensitivity to a key hyperparameter (the number of solutions curated for each problem) before disentangling the individual contributions of our problem and

Table 3 | Diversity scores for different methods on the MATH500 Level 5 test set, evaluated on the Qwen3 4B Base model. Scores are partitioned by the number of correct solutions (pass count) out of 16 attempts.

Method	Div-Self-BLEU		Our Metric	
	Pass Count 2-12	Pass Count 13-16	Pass Count 2-12	Pass Count 13-16
Random (1P1S)	35.27	15.26	15.17	13.30
Random (1P3S)	32.52	14.62	15.57	14.00
LLM Selection (1P3S)	36.36	14.19	15.11	13.31
Raw Emb. (1P3S)	33.94	14.23	15.39	13.08
Summary Emb. (1P3S)	37.42	14.46	15.69	12.98
RPD (1P3S)	38.20	14.31	15.80	12.62

answer selection components. Subsequently, we explore the interplay between our fine-tuning approach and inference-time temperature sampling. Furthermore, we investigate whether our paradigm provides a superior foundation for subsequent reinforcement learning. Finally, we validate the scalability of our paradigm on a larger training set.

Analysis of Solution Diversity. To verify that 1PNS training increases output diversity, we analyzed 16 generated solutions for each problem in MATH Level5 test set. We partitioned problems into a *moderately-solved group* (2-12 correct solutions) and a *well-solved group* (13-16 correct solutions) to analyze performance on problems of varying difficulty. Diversity was measured using our RPD metric, which is the average pairwise RPD among correct solutions within each problem, and Div-Self-BLEU (100 - Self-BLEU) (Kirk et al., 2023). For both metrics, a higher score indicates greater output diversity.

Table 3 shows that the fine-tuned model adjusts output diversity according to problem difficulty. For moderately-solved problems (2–12 correct solutions), our RPD method produces the most diverse outputs by both RPD and Div-Self-BLEU; for well-solved problems (13–16 correct solutions), diversity falls below the 1P1S baseline, suggesting convergence to a single high-confidence answer. We interpret this as an effective test-time scaling strategy: the model selectively increases exploration on challenging instances while exploiting confident solutions on simpler ones. This adaptability is key to optimizing overall `pass@k` performance.

Impact of the Number of Solutions per Problem. Keeping the training set fixed at 300 samples, we vary the number of solutions per problem and measure model performance. Table 4 compares configurations using 2, 3, 4, or 5 solutions per problem against the single-solution baseline.

As shown in Table 4, the `1PNS` paradigm consistently improves performance over the single-solution baseline, and this improvement generally grows as k increases. The gains peak at *RPD (1P3S)* and then decline when more solutions are added, indicating a trade-off between *diversity depth* and *problem breadth*. The best choice depends on the dataset; for OpenThought3, the 1P3S setting worked best in our tests.

Quantifying the Impact of Problem and Answer Selection Strategies. We isolate the effects of question selection (`RPD-Q`) and answer selection (`RPD-A`) by comparing the full method to ablations in which one component is replaced by random selection. Results are shown in Table 5. From these results, we draw three main conclusions. First, using diversity-driven selection for questions or for answers improves performance over the fully random baseline at higher `pass@k` values. Second, question selection (`RPD-Q`) has a larger effect on Test-Time Scaling than

Table 4 | Ablation study on the number of diverse solutions selected by **our RPD metric** per problem on the MATH500 Level 5 benchmark, compared against the 1P1S baseline. The total sample size is kept constant at 300.

Configuration	pass@1 (%)	pass@2	pass@4	pass@8	pass@16
Random (1P1S)	49.26	60.64	66.98	72.20	77.43
RPD (1P2S)	52.43	**61.57**	69.96	74.63	77.99
RPD (1P3S)	52.61	**61.57**	**71.64**	**75.94**	**79.29**
RPD (1P4S)	52.24	59.70	70.90	74.63	79.10
RPD (1P5S)	**53.92**	61.20	67.73	73.88	78.54

Table 5 | Ablation study on the contributions of the problem (Q) and answer (A) selection components on the MATH500 Level 5 benchmark. All configurations use a 100Q, 3A structure.

Method (Problem + Answer)	pass@1 (%)	pass@2	pass@4	pass@8	pass@16
Random-Q + Random-A	49.07	59.70	68.66	73.32	77.24
Random-Q + RPD-A	50.93	61.38	68.47	73.69	77.43
RPD-Q + Random-A	49.81	59.52	67.91	74.82	78.36
RPD-Q + RPD-A (Ours)	**52.61**	**61.57**	**71.64**	**75.94**	**79.29**

answer selection (`RPD-A`). At `pass@16`, RPD-Q yields a +1.12% gain over random, while RPD-A alone yields +0.19%. Finally, combining both strategies gives the best performance: at `pass@16` the full method improves by nearly one percentage point over the next-best configuration. This shows a clear synergy—Question Selection provides a strong foundation, but both components are needed to maximize reasoning performance.

Interaction with Inference-Time Sampling Temperature. Increasing the sampling temperature (T) is a common way to boost output diversity at inference. An important question is whether the diversity gains from our fine-tuning are redundant with, or complementary to, this technique. To test this, we compare our method (RPD 1P3S) to the baseline (Random 1P1S) across five temperatures from $T = 0.2$ to $T = 1.0$. We use `top_p=0.95` and `top_k=-1`.

Table 6 shows that the performance gap between our method and the baseline widens as k increases, regardless of temperature. For small k (e.g., `pass@1`, `pass@2`), our method and the random baseline perform similarly; neither dominates across all temperatures. At larger k (e.g., `pass@8`, `pass@16`), a clear pattern emerges: our method outperforms the baseline across all tested temperatures.

This observation supports our conclusion that RPD-guided training is orthogonal to, and complements, inference-time temperature sampling. By exposing the model to diverse reasoning pathways during training, our method expands the set of solutions the model can access. Temperature, in contrast, is a separate tool that controls the randomness of sampling from that solution space at inference. The stronger advantage at high k shows our approach gives a distinct, foundational benefit that tuning inference-time parameters alone cannot replace.

Effect of Subsequent Reinforcement Learning To evaluate whether the diverse reasoning trajectories encouraged by our 1PNS paradigm yield a better foundation for subsequent reinforcement learning (RL) fine-tuning, we applied an additional RL phase on both the RPD-curated (1P3S) model and the Random (1P1S) baseline using the SimpleRL-Zoo dataset (Zeng et al., 2025) (see Appendix E.4 for experimental details).

Table 6 | Performance comparison on the MATH500 Level 5 benchmark between our method and the random baseline across various sampling temperatures (*T*).

Method	Temp (*T*)	pass@1 (%)	pass@2	pass@4	pass@8	pass@16
Random	0.2	**51.12**	**58.96**	65.30	69.96	73.13
RPD	0.2	50.00	57.46	**66.79**	**71.46**	**74.82**
Random	0.4	**54.11**	**60.26**	68.10	72.39	76.31
RPD	0.4	47.95	60.08	**69.03**	**75.19**	**77.80**
Random	0.6	49.26	60.64	66.98	72.20	77.43
RPD	0.6	**52.61**	**61.57**	**71.64**	**75.94**	**79.29**
Random	0.8	**51.87**	**61.57**	69.22	74.44	78.36
RPD	0.8	50.56	60.82	**69.47**	**74.82**	**78.92**
Random	1.0	45.34	59.71	68.66	**73.88**	76.87
RPD	1.0	**48.51**	**59.89**	**69.22**	**73.88**	**77.80**

Table 7 | Performance comparison after an additional phase of Reinforcement Learning (RL) fine-tuning. The models were first fine-tuned with SFT (Random 1P1S vs. RPD 1P3S) and then further tuned with RL on the Simple-Zoo dataset.

Benchmark	Method	pass@1 (%)	pass@2	pass@4	pass@8	pass@16
AIME24	Random (1P1S) + RL	10.00	16.67	19.17	24.17	32.50
	RPD (1P3S) + RL	**12.50**	**17.50**	**24.17**	**30.84**	**37.50**
MATH500 Level 5	Random (1P1S) + RL	49.82	59.14	65.86	70.71	76.87
	RPD (1P3S) + RL	**51.31**	**59.89**	**69.22**	**73.88**	**77.80**
Olympiad Bench	Random (1P1S) + RL	41.77	50.23	57.27	62.84	67.66
	RPD (1P3S) + RL	**44.07**	**52.67**	**58.61**	**64.69**	**68.99**

Table 7 summarizes the results. RL fine-tuning improves performance for both models across the board. However, the model initialized with RPD-selected data consistently outperforms the 1P1S baseline on every benchmark and for every evaluation metric (pass@1/2/4/8/16). The advantage is most pronounced on the more difficult AIME24 benchmark: *RPD (1P3S) + RL* reaches pass@16 = 37.50%, which is 5.00 percentage points higher than the baseline.

These findings support the hypothesis that SFT with diverse, RPD-selected reasoning paths provides a stronger starting point for RL, enabling larger downstream gains.

Scalability to Larger Datasets. We also tested scalability by increasing the training set to 1,500 samples. As shown in Appendix E.5, our diversity-driven strategy retains a clear performance advantage over the 1P1S baseline at this larger scale.

5. Conclusion

To enable the *one problem, multiple solutions* (1PNS) paradigm, we introduce a novel metric for quantifying reasoning diversity, Reasoning Path Divergence (RPD), and use it to curate a dataset of maximally diverse solutions. Our experiments validate the superiority of the 1PNS paradigm over the standard 1P1S baseline: training on RPD-curated data reduces output homogenization and yields significant pass@k gains. These results show that our approach provides a direct way to improve test-time scaling.

Acknowledgements

References

A. K. M. Abbas, K. Tirumala, D. Simig, S. Ganguli, and A. S. Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

A. Albalak, Y. Elazar, S. M. Xie, S. Longpre, N. Lambert, X. Wang, N. Muennighoff, B. Hou, L. Pan, H. Jeong, et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

B. Brown, J. Juravsky, R. Ehrlich, R. Clark, Q. V. Le, C. Ré, and A. Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

F. Chen, A. Raventos, N. Cheng, S. Ganguli, and S. Druckmann. Rethinking fine-tuning when scaling test-time compute: Limiting confidence improves mathematical reasoning. *arXiv preprint arXiv:2502.07154*, 2025.

L. Chen, J. Davis, B. Hanin, P. Bailis, I. Stoica, M. Zaharia, and J. Zou. Are more lm calls all you need? towards the scaling properties of compound ai systems. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 45767–45790, 2024.

Y. Chow, G. Tennenholtz, I. Gur, V. Zhuang, B. Dai, A. Kumar, R. Agarwal, S. Thiagarajan, C. Boutilier, and A. Faust. Inference-aware fine-tuning for best-of-n sampling in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

H.-L. Chung, T.-Y. Hsiao, H.-Y. Huang, C. Cho, J.-R. Lin, Z. Ziwei, and Y.-N. Chen. Revisiting test-time scaling: A survey and a diversity-aware method for efficient reasoning. *arXiv preprint arXiv:2506.04611*, 2025.

X. Dang, C. Baek, K. Wen, Z. Kolter, and A. Raghunathan. Weight ensembling improves reasoning in language models. *arXiv preprint arXiv:2504.10478*, 2025.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

E. Guha, R. Marten, S. Keh, N. Raoof, G. Smyrnis, H. Bansal, M. Nezhurina, J. Mercat, T. Vu, Z. Sprague, A. Suvarna, B. Feuer, L. Chen, Z. Khan, E. Frankel, S. Grover, C. Choi, N. Muennighoff, S. Su, W. Zhao, J. Yang, S. Pimpalgaonkar, K. Sharma, C. C.-J. Ji, Y. Deng, S. Pratt, V. Ramanujan, J. Saad-Falcon, J. Li, A. Dave, A. Albalak, K. Arora, B. Wulfe, C. Hegde, G. Durrett, S. Oh, M. Bansal, S. Gabriel, A. Grover, K.-W. Chang, V. Shankar, A. Gokaslan, M. A. Merrill, T. Hashimoto, Y. Choi, J. Jitsev, R. Heckel, M. Sathiamoorthy, A. G. Dimakis, and L. Schmidt. Openthoughts: Data recipes for reasoning models, 2025. URL https://arxiv.org/abs/2506.04178.

C. He, R. Luo, Y. Bai, S. Hu, Z. L. Thai, J. Shen, J. Hu, X. Han, Y. Huang, Y. Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

J. He, H. Lin, Q. Wang, Y. R. Fung, and H. Ji. Self-correction is more than refinement: A learning framework for visual and language reasoning tasks. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6405–6421, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.331. URL https://aclanthology.org/2025.findings-acl.331/.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

S. Huang, H. Wang, W. Zhong, Z. Su, J. Feng, B. Cao, and Y. R. Fung. Adactrl: Towards adaptive and controllable reasoning via difficulty-aware budgeting, 2025a. URL https://arxiv.org/abs/2505.18822.

Y. Huang, S. Li, Z. Fan, M. LIU, W. Liu, and Y. R. Fung. Scaling environments for LLM agents: Fundamentals, approaches, and future directions. In *Workshop on Scaling Environments for Agents*, 2025b. URL https://openreview.net/forum?id=9axZcDTiJm.

D. Jiang, X. Ren, and B. Y. Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, 2023.

R. Kirk, I. Mediratta, C. Nalmpantis, J. Luketina, E. Hambro, E. Grefenstette, and R. Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*, 2023.

C. Li, T. Xu, and S. Y. Guo. Reasoning-as-logic-units: Scaling test-time reasoning in large language models through logic unit alignment. In *Forty-second International Conference on Machine Learning*, 2025a.

Z. Li, C. Chen, T. Xu, Z. Qin, J. Xiao, Z.-Q. Luo, and R. Sun. Preserving diversity in supervised fine-tuning of large language models. In *ICLR*, 2025b.

Z. Li, X. Feng, Y. Cai, Z. Zhang, T. Liu, C. Liang, W. Chen, H. Wang, and T. Zhao. Llms can generate a better answer by aggregating their own responses. *arXiv preprint arXiv:2503.04104*, 2025c.

W. Liu, W. Zeng, K. He, Y. Jiang, and J. He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

T. Munkhbat, N. Ho, S. H. Kim, Y. Yang, Y. Kim, and S.-Y. Yun. Self-training elicits concise reasoning in large language models. *CoRR*, 2025.

M. Nye, A. J. Andreassen, G. Gur-Ari, H. Michalewski, J. Austin, D. Bieber, D. Dohan, A. Lewkowycz, M. Bosma, D. Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.

L. O'Mahony, L. Grinsztajn, H. Schoelkopf, and S. Biderman. Attributing mode collapse in the fine-tuning of large language models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, volume 2, 2024.

Z. Qin, Q. Dong, X. Zhang, L. Dong, X. Huang, Z. Yang, M. KHADEMI, D. Zhang, H. H. Awadalla, Y. R. Fung, W. Chen, M. Cheng, and F. Wei. Scaling laws of synthetic data for language model. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=UmUXPXHtdl.

N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Y. Song, G. Wang, S. Li, and B. Y. Lin. The good, the bad, and the greedy: Evaluation of llms should not ignore non-determinism. *arXiv preprint arXiv:2407.10457*, 2024.

Q. Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Q. Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

R. Wang, Y. Li, Y. R. Fung, and T. Zhang. Let's reason formally: Natural-formal hybrid reasoning enhances llm's math capability, 2025a. URL https://arxiv.org/abs/2505.23703.

X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Y. Wang, Z. Fan, Q. Wang, Y. R. Fung, and H. Ji. CALM: Unleashing the cross-lingual self-aligning ability of language model question answering. In L. Chiruzzo, A. Ritter, and L. Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2809–2817, Albuquerque, New Mexico, Apr. 2025b. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.152. URL https://aclanthology.org/2025.findings-naacl.152/.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Y. Wu, Y. Wang, Z. Ye, T. Du, S. Jegelka, and Y. Wang. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*, 2025.

S. M. Xie, H. Pham, X. Dong, N. Du, H. Liu, Y. Lu, P. S. Liang, Q. V. Le, T. Ma, and A. W. Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023.

J. Yao, R. Cheng, X. Wu, J. Wu, and K. C. Tan. Diversity-aware policy optimization for large language model reasoning. *arXiv preprint arXiv:2505.23433*, 2025.

W. Zeng, Y. Huang, Q. Liu, W. Liu, K. He, Z. Ma, and J. He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *Second Conference on Language Modeling*, 2025.

Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, F. Huang, and J. Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

A. RPD Curation Method Implementation

A.1. Step-wise Solution Summarization via LLM

Our proposed diversity metric relies on a fine-grained, step-by-step summary of the reasoning path for each solution. To create these summaries, we use an LLM (Qwen3-14B) to break down each solution into its core logical steps. A key challenge is to ensure these summaries accurately reflect the original methodology while maintaining a consistent level of granularity. Overly concrete summaries might capture superficial numerical differences, while overly abstract summaries might fail to distinguish between genuinely different strategies.

To solve this, we design a detailed prompt that controls the LLM's output format and level of abstraction. This prompt instructs the model to produce a structured JSON object containing 3 to 5 method-focused steps. This strict format helps maintain uniformity across all summarized solutions. The complete prompt is provided below.

Prompt for Step-wise Solution Summarization

You are a specialized AI expert in analyzing mathematical solutions. Your task is to first provide a step-by-step analysis of a solution, and then, based on your analysis, generate a final JSON output that is concise, direct, and method-focused.

REQUIRED OUTPUT STRUCTURE
Your response **MUST** have two distinct parts in the following order:

Part 1: Analysis & Thinking Process

- Start this section with the heading `### Analysis`.
- Briefly explain your reasoning as you deconstruct the provided solution. This is your "scratchpad".

Part 2: Final JSON Output

- After your analysis, provide the final JSON output enclosed in `//boxed{{}}`.
- This part must contain *only* the `//boxed{{...}}` block and nothing else.

CONTENT RULES FOR THE FINAL JSON
1. **Step Count**: The JSON must contain **strictly 3 to 5 logical steps**.
2. **Output Style**:
    - **Use direct, active verb phrases.** Start each description with a verb (e.g., "Calculate", "Identify", "Apply").
    - **DO NOT use narrative phrasing** like "The author identifies..." or "The solution then calculates...".
3. **Abstraction Level**:
    - Be abstract about numbers and variables, but **be specific about the methodology**.
    - **BAD (Too Vague):** "Use a formula to get the result."
    - **BAD (Too Concrete):** "Calculate $1/3 + 1/6 = 1/2$."
    - **GOOD (Balanced):** "Combine the individual rates to find the total work rate."

JSON STRUCTURE SPECIFICATION
- The root object must have one key: `"logical_steps"`.
- The value of `"logical_steps"` must be a list (`[]`) of step objects.
- Each step object (`{{}}`) must contain two keys:
  - `"step_title"`: A short title for the step (e.g., "Step 1: Combine Rates"). Use `null` if not applicable.
  - `"step_description"`: A concise summary of the action, following all rules above.

EXAMPLE OF THE COMPLETE TWO-PART OUTPUT

Input Solution: "Pipe A fills a tank in 3 hours, so its rate is 1/3 tank/hr. Pipe B fills it in 6 hours, so its rate is 1/6 tank/hr. Together, their rate is 1/3 + 1/6 = 1/2 tank/hr. Therefore, the time to fill the tank together is the reciprocal of the rate, which is 1 / (1/2) = 2 hours."

Your Required Output:

```
### Analysis
The solution addresses a classic work-rate problem.
1.  First, it calculates the individual rate for each
    pipe.
2.  Second, it sums these rates to get a combined rate.
3.  Finally, it converts the combined rate back into
    total time.
The logic is broken down into three clear, abstract
    steps.

//boxed{{
  "logical_steps": [
    {{
      "step_title": "Step 1: Determine Individual Rates
          ",
      "step_description": "Determine the individual
          work rate of each component based on the time
          taken."
    }},
    {{
      "step_title": "Step 2: Combine Rates",
      "step_description": "Combine the individual rates
          to find the total system work rate."
    }},
    {{
      "step_title": "Step 3: Calculate Total Time",
      "step_description": "Calculate the total time by
          taking the reciprocal of the combined work rate
          ."
    }}
  ]
}}
```

> *YOUR TASK*
>
> **Math Problem**:
> {question_text}
>
> **Chain-of-Thought Solution to Analyze**:
> {answer_cot}

A.2. Reasoning Path Divergence (RPD) Calculation

After summarizing each solution into a series of core logical steps, the next phase is to compute the pairwise diversity using our **Reasoning Path Divergence (RPD)** metric. RPD is designed to quantify the semantic distance between the step-lists of two solutions, S_A and S_B.

The calculation begins by embedding each logical step using the **Qwen3-Embedding-8B** model. Subsequently, it computes an asymmetric score by finding the average minimum cosine distance from the steps of the shorter solution to all steps in the longer one. This asymmetric design is crucial: it ensures that a solution containing a genuinely novel step is considered distant, even if its other steps are subsumed by a more comprehensive solution. The formal algorithm is detailed below.

Algorithm 1 Reasoning Path Divergence (RPD) Calculation

Require: Two Long-CoT solutions, S_A and S_B.
Ensure: A scalar diversity score $D \in [0, 1]$.

1: $L_A \leftarrow \text{ExtractSteps}(S_A)$; $L_B \leftarrow \text{ExtractSteps}(S_B)$
2: **if** L_A is empty or L_B is empty **then**
3:     **return** 1.0
4: **end if**

5: $E_A \leftarrow \{\text{Embed}(a_i) \mid a_i \in L_A\}$; $E_B \leftarrow \{\text{Embed}(b_j) \mid b_j \in L_B\}$

6: $(E_{\text{shorter}}, E_{\text{longer}}) \leftarrow \begin{cases} (E_A, E_B) & \text{if } |E_A| \leq |E_B| \\ (E_B, E_A) & \text{otherwise} \end{cases}$

7: $\text{min\_distances} \leftarrow \emptyset$
8: **for all** $\vec{e}_s \in E_{\text{shorter}}$ **do**
9:     $d_{\min} \leftarrow \min_{\vec{e}_l \in E_{\text{longer}}} \left(1 - \frac{\vec{e}_s \cdot \vec{e}_l}{\|\vec{e}_s\| \|\vec{e}_l\|}\right)$
10:     $\text{min\_distances} \leftarrow \text{min\_distances} \cup \{d_{\min}\}$
11: **end for**

12: $D_{\text{final}} \leftarrow \text{Mean}(\text{min\_distances})$
13: **return** D_{final}

A.3. Diversity-Driven Data Curation

Our data curation process is a two-stage procedure designed to build a training set rich in strategic diversity. First, we perform **Problem Selection** to identify problems that naturally exhibit a wide range of solutions by scoring each problem based on its total intrinsic diversity. Second, for each of these top-ranked problems, we execute a greedy **Solution Selection** algorithm to curate a small but maximally diverse subset of M solutions. This two-stage approach ensures both inter-problem and intra-problem diversity. The algorithms for both stages are detailed below.

Algorithm 2 Stage 1: Problem selection by intrinsic diversity

---

Require: Candidate problem set $\mathcal{P}$, target count N, pairwise distance function $D(\cdot, \cdot)$
Ensure: Top-N problems $\mathcal{P}_{\text{top}}$ ranked by intrinsic diversity
 1: Initialize empty list of pairs $\mathcal{L} \leftarrow []$
 2: **for all** problem $P \in \mathcal{P}$ **do**
 3:      Let $\mathcal{S}_P = \{S_1, \ldots, S_{k_P}\}$ be its candidate solutions
 4:      **if** $k_P < 2$ **then**
 5:          append $(P, -\infty)$ to $\mathcal{L}$
 6:          **continue**
 7:      **end if**
 8:      Compute all pairwise distances $\{D(S_i, S_j) : 1 \le i < j \le k_P\}$
 9:      $\text{avgD} \leftarrow \dfrac{2}{k_P(k_P - 1)} \sum_{i<j} D(S_i, S_j)$
10:      append (P, avgD) to $\mathcal{L}$
11: **end for**
12: Sort $\mathcal{L}$ by score (second element) in descending order
13: $\mathcal{P}_{\text{top}} \leftarrow$ first $\min(N, |\mathcal{P}|)$ problems from sorted $\mathcal{L}$
14: **return** $\mathcal{P}_{\text{top}}$

Algorithm 3 Stage 2: Greedy Selection

Require: Candidate solutions $\mathcal{S}_{\text{cand}} = \{S_1, \ldots, S_k\}$, pairwise distance matrix $\mathbf{D} \in \mathbb{R}^{k \times k}$, target size M
Ensure: Selected index set $\mathcal{I}_{\text{select}}$ with $|\mathcal{I}_{\text{select}}| = \min(M, k)$
 1: **if** $M \le 0$ **or** $k = 0$ **then return** $\emptyset$
 2: **end if**
 3: **if** $M \ge k$ **then return** $\{1, \ldots, k\}$
 4: **end if**
 5: $i_{\text{first}} \leftarrow \arg\max_i \sum_{j \ne i} \mathbf{D}_{ij}$
 6: $\mathcal{I}_{\text{select}} \leftarrow \{i_{\text{first}}\}; \quad \mathcal{I}_{\text{remain}} \leftarrow \{1, \ldots, k\} \setminus \{i_{\text{first}}\}$
 7: **for each** $r \in \mathcal{I}_{\text{remain}}$ set $m[r] \leftarrow \mathbf{D}_{r, i_{\text{first}}}$
 8: **while** $|\mathcal{I}_{\text{select}}| < M$ and $\mathcal{I}_{\text{remain}} \ne \emptyset$ **do**
 9:      $r^\star \leftarrow \arg\max_{r \in \mathcal{I}_{\text{remain}}} m[r]$
10:      $\mathcal{I}_{\text{select}}.\text{append}(r^\star); \quad \mathcal{I}_{\text{remain}}.\text{remove}(r^\star)$
11:      **for each** $r \in \mathcal{I}_{\text{remain}}$: $m[r] \leftarrow \min\left(m[r], \mathbf{D}_{r, r^\star}\right)$
12: **end while**
13: **return** $\mathcal{I}_{\text{select}}$

B. Dataset Preprocessing and Analysis

B.1. Detailed Dataset Filtering Protocol

The OpenThought3 dataset is a valuable open-source resource, containing approximately 53,000 mathematical problems, each with 16 corresponding completions. However, the raw dataset presents several challenges for direct use in supervised fine-tuning. Key issues include the absence of ground truth labels, the possibility of encountering ambiguous or ill-posed problems, and the fact that some solutions may be unfinished or lack a definitive final answer. Furthermore, the length of the provided solutions varies dramatically.

To curate a high-quality training corpus and ensure computational efficiency during model inference, we implement a rigorous two-stage filtering protocol on a subset of 10,000 problems from OpenThought3. This protocol addresses both solution length and quality.

Stage 1: Length-Based Filtering. Our first step is to control for solution length. This measure is primarily motivated by the practical need to set a reasonable `max_new_tokens` parameter during inference. Accordingly, we filter out any problem whose average token count across all its solutions exceeds 14,000 tokens.

Stage 2: Quality and Completeness Filtering. Next, we address the issue of solution quality and completeness. We employ an LLM (Qwen3-14B) as a judge to verify whether each solution is valid. For every solution in the length-filtered set, we provide its final 500 tokens as input to the LLM. The model is instructed to determine if the solution concludes properly by presenting a clear and final answer. Solutions that the LLM judge flags as incomplete or inconclusive are discarded, and any problem subsequently left with fewer than 10 valid solutions is also removed.

This comprehensive filtering pipeline refines the initial pool of 10,000 problems into a high-quality, curated set of **approximately 1,600 problems**. Each problem in this final set has an average solution length of less than 14,000 tokens and is accompanied by at least 10 complete, validated solutions. This curated 1,600-problem dataset serves as the foundation for all subsequent experiments conducted in this work.

B.2. Dataset Diversity Analysis

To better inform our data curation, we first analyze the existing strategic diversity within our high-quality candidate set. We use a summary-based LLM Judge to classify whether the solutions for each problem are strategically uniform or diverse.

For each problem, we concatenate the step-wise summaries of all its candidate solutions (detailed in Appendix A.1) into a single string. This, along with the original problem statement, is provided to an LLM Judge (Qwen3-14B). The judge's task is to perform a binary classification on the entire set of solutions, identifying if at least two different solution strategies are present.

We specifically write the prompt to instruct the model to ignore superficial differences in wording or calculation, and instead focus on fundamental strategic choices, such as using direct casework versus complementary counting. We do this so that the classification reflects genuine methodological diversity, not just surface-level variations. The insights from this analysis, as reported in the main text, confirm the need for our subsequent diversity-driven problem selection phase. The complete prompt for this task is detailed below.

You are a master mathematician and an expert in pedagogical analysis. Your task is to classify a problem based on the methodological diversity of its proposed solutions.

Your goal is to perform a binary classification:

- **Class 2 (Diverse):** If there are at least two distinct core methodologies present across all the provided solution summaries.
- **Class 1 (Not Diverse):** If all solutions use the same core methodology, or if the differences are only superficial (e.g., a different order of calculation, or using standard procedural equivalents like substitution vs. elimination).

---

1. Your Analysis Framework & Core Criteria

Your primary task is to act as a discerning analyst. You must distinguish between minor procedural choices and significant differences in core steps. Assume that most solutions might share a high-level strategy; your goal is to find answers that execute core steps in a meaningfully different way.

Defining Methodological Difference (Your Core Criteria):

What IS NOT a Significant Difference (Methodologically Similar):

- **Order of Calculation:** Calculating value A then B, versus B then A, before combining them in the same way.
- **Algebraic Equivalence:** Using the form $(a + b)^2$ versus $a^2 + 2ab + b^2$.
- **Variable Naming or Notation:** Using n vs x.
- **Choice of Standard Procedural Equivalents:** One summary describes solving a system of equations using **substitution**, while the other uses **elimination**. These are considered standard, interchangeable procedures within the same overall algebraic approach.
- **Rigorous Proof vs. Heuristic Assumption:** If the overall strategy is the same, simply proving a result versus assuming it does not constitute a diverse approach. Both are still following the same high-level logical path.

What IS a Significant Difference (Methodologically Diverse):

- This difference represents a **completely distinct, independent, high-level strategic choice** that fundamentally alters the entire problem-solving path from beginning to end.
- **Example 1 (Different Overall Framework):** One solution to a geometry problem uses **coordinate geometry**, another uses **synthetic geometry**, and a third uses **vector analysis**.
- **Example 2 (Completely Different Logical Path):** To solve a counting problem, one answer uses **direct casework**, another uses **complementary counting**, and a third uses a **recurrence relation**.
- **Example 3 (Change in Analytical Tool):** A solution to an optimization problem uses **calculus**, a second uses **inequalities** (like AM-GM), and a third uses **linear programming**.

2. Content to Analyze

Problem:
`{question}`

Proposed Solutions (Summarized by Logical Steps):
`{summaries_text}`

3. Output Requirement

Based on the final criteria review, classify the diversity of the solutions.
Output Requirement:
Immediately after your classification, provide your final answer in a strict JSON format within a special block. The JSON should be a single integer, either 1 or 2. Do not provide any other text.

Example of Final Output Structure for a **Diverse** problem:

`//boxed{{2}}`

Example of Final Output Structure for a **Not Diverse** problem:

`//boxed{{1}}`

Begin Analysis and Provide Output:

This classification process is applied to the 1,600 high-quality problems in our candidate pool, yielding the diversity distribution statistics reported in Section 3.2.

C. Case Studies and Analysis of the RPD Metric

To provide a deeper insight into the effectiveness of our RPD metric, this section presents both a statistical overview and concrete, illustrative examples comparing it against a standard baseline.

C.1. Statistical Distribution of Diversity Scores

We first analyze the overall behavior of RPD compared to a common baseline. The baseline method calculates the cosine distance between the embeddings of the full, raw solution texts. We sampled 100 problems from our candidate pool and computed all pairwise diversity scores for their solutions using both methods, resulting in a total of 8,986 data points (i.e., solution pairs) for each distribution.

Figure 3 illustrates the resulting score distributions. The baseline scores are heavily concentrated in a very narrow range near zero (0.00–0.04). This indicates that full-text embeddings are largely insensitive to the underlying reasoning structure, assigning nearly identical low-diversity scores to most pairs and failing to distinguish between subtle and significant strategic differences. In contrast, our RPD metric produces a much wider and more uniform distribution. This indicates that RPD possesses significantly higher resolution and sensitivity, allowing it to capture a continuous spectrum of strategic differences, from the subtle to the substantial.
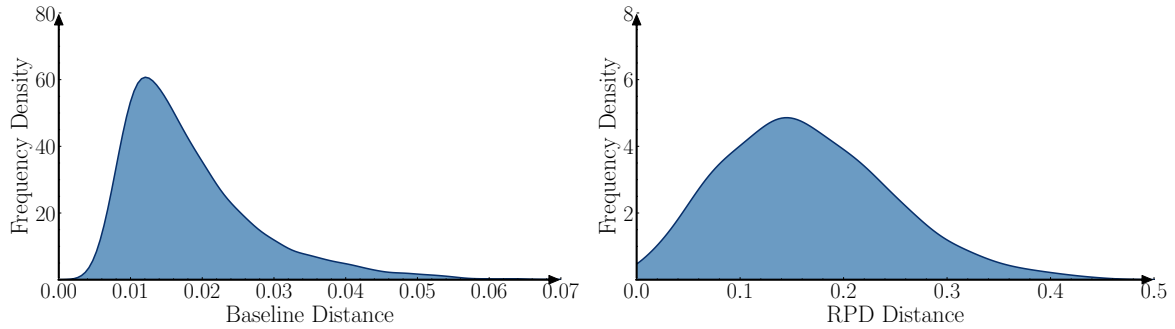

Figure 3 | Distribution of pairwise diversity scores on 100 problems for the baseline (left) and our RPD metric (right). RPD provides a significantly better-separated distribution.

C.2. Illustrative Examples

The following case studies provide concrete examples of this phenomenon.

(a) Case Study 1
Raw Emb. Distance: 0.015 (Percentile: 44.46%)
RPD Distance: 0.259 (Percentile: 86.92%)

(b) Case Study 2
Raw Emb. Distance: 0.016 (Percentile: 52.14%)
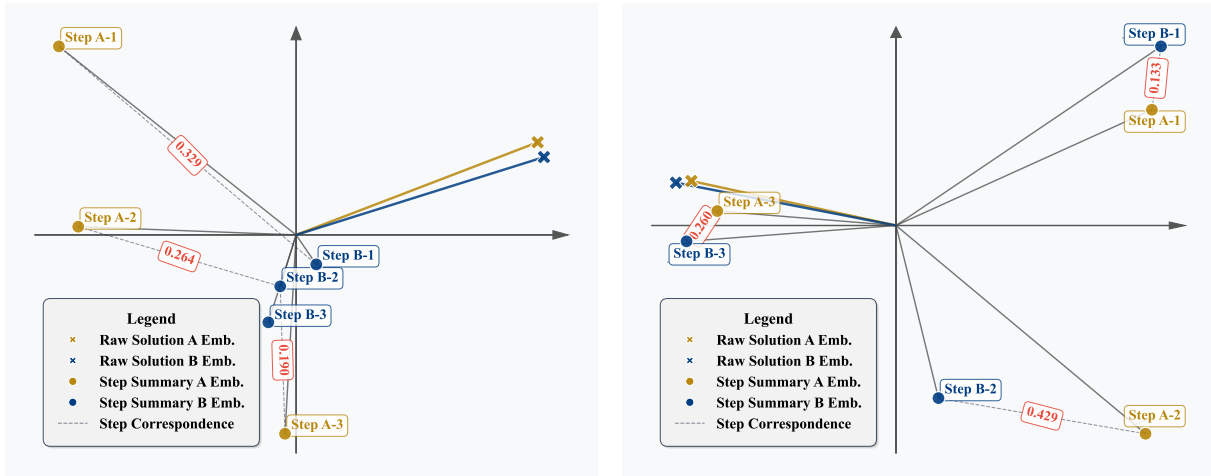RPD Distance: 0.274 (Percentile: 90.44%)

Figure 4 | PCA visualization of raw solution and step summary embeddings. The step embeddings for the two solutions occupy distinct regions of the space, reflecting a strategic diversity that our RPD metric correctly identifies. In contrast, the raw solution embeddings are nearly collinear, causing the baseline method to fail to distinguish them.

Case Study 1: Summaries for Figure 4a

- **Question:** Find the constant term in the polynomial $(x^2 + 2x + 1)(x^2 - 3x - 2) + (x^2 - 2x - 1)(x^2 + 4x + 3)$ after it is factored.
- **Solution A (Full Expansion):**
  - Step 1: Expand each trinomial product using the distributive property.
  - Step 2: Add the expanded polynomials together and combine like terms.
  - Step 3: Extract the constant term from the resulting polynomial.

- **Solution B (Constant Term Shortcut):**
  - Step 1: Determine the constant term of each product by multiplying the constant terms of the individual polynomials.
  - Step 2: Add the constant terms from each product to find the constant term of the entire expression.
  - Step 3: Verify that the constant term remains unchanged when the polynomial is factored.

Case Study 2: Summaries for Figure 4b

- **Question:** Determine the largest real value of a such that the equation

$$ax = x^3 + 1$$

  has a real solution.

- **Solution A (Calculus Approach):**
  - Step 1: Rewrite the equation to express a as a function of x, $a = \frac{x^3+1}{x}$.
  - Step 2: Find the critical points of the function $f(x) = \frac{x^3+1}{x}$ by taking its derivative and setting it to zero.
  - Step 3: Evaluate the function at the critical point to find the value of a where the equation has a double root, ensuring the largest a for which the equation has a real solution.

- **Solution B (Geometric Interpretation):**
  - Step 1: Interpret the equation as the intersection of a line $y = ax$ and a curve $y = x^3 + 1$.
  - Step 2: Set the derivative of the cubic function equal to the slope of the line to find the point of tangency.
  - Step 3: Solve the system of equations to find the largest real value of a corresponding to the tangency condition.

Analysis: The two case studies in Figure 4 illustrate a consistent pattern where our RPD metric succeeds and the baseline fails. In both examples, the solution pairs employ fundamentally different strategies. The baseline Raw Embedding Distance assigns very low scores (0.015 and 0.016) that correspond to mediocre percentiles (44-52%). This indicates the method is unable to reliably distinguish these solutions from the vast majority of superficially similar pairs. In stark contrast, our RPD metric assigns high scores (0.259 and 0.274) that fall into high percentiles (87-90%), correctly identifying the significant strategic divergence. The PCA visualizations visually corroborate this finding: the well-separated step embeddings in both (a) and (b) confirm that the solutions follow distinct reasoning paths, a fact that only RPD consistently captures.

D. Experiment Implementation Details

D.1. RPD Metric Evaluation (Details for Sec. 4.1)

In this section, we provide the implementation details for the RPD metric evaluation, including the prompts used for the LLM-based baseline and the evaluation judge.

D.1.1. *Prompt for the LLM-Selection Baseline*

To create the "LLM Selection" baseline, we prompt the Qwen3-14B model to identify the most diverse pair of solutions from all available candidates for a given problem. The prompt is designed to encourage a focus on strategic differences rather than superficial text variations.

Prompt for Selecting the Most Methodologically Diverse Solution Pair

You are a master mathematician and an expert in pedagogical analysis. Your task is to analyze multiple proposed solutions for a given problem and select a single pair of answers that represents the maximum possible methodological diversity. If no such pair exists, you must indicate this.

Your goal is to identify **one pair of answers** that represents a significant difference in a core step or sub-methodology. If all solutions follow a fundamentally similar strategy, your answer will be to select **"No"**.

---

1. Your Analysis Framework & Core Criteria

Your primary task is to act as a discerning analyst. You must distinguish between minor procedural choices and significant differences in core steps. Assume that most solutions might share a high-level strategy; your goal is to find answers that execute core steps in a meaningfully different way.

Defining Methodological Difference (Your Core Criteria):

What IS NOT a Significant Difference (Methodologically Similar):

- **Order of Calculation:** Calculating value A then B, versus B then A, before combining them in the same way.
- **Algebraic Equivalence:** Using the form $(a + b)^2$ versus $a^2 + 2ab + b^2$.
- **Variable Naming or Notation:** Using n vs x.
- **Choice of Standard Procedural Equivalents:** One summary describes solving a system of equations using **substitution**, while the other uses **elimination**. These are considered standard, interchangeable procedures within the same overall algebraic approach.
- **Rigorous Proof vs. Heuristic Assumption:** If the overall strategy is the same, simply proving a result versus assuming it does not constitute a diverse approach. Both are still following the same high-level logical path.

What IS a Significant Difference (Methodologically Diverse):

- This difference represents a **completely distinct, independent, high-level strategic**

choice that fundamentally alters the entire problem-solving path from beginning to end.

- **Example 1 (Different Overall Framework):** One solution to a geometry problem uses **coordinate geometry**, another uses **synthetic geometry**, and a third uses **vector analysis**.
- **Example 2 (Completely Different Logical Path):** To solve a counting problem, one answer uses **direct casework**, another uses **complementary counting**, and a third uses a **recurrence relation**.
- **Example 3 (Change in Analytical Tool):** A solution to an optimization problem uses **calculus**, a second uses **inequalities** (like AM-GM), and a third uses **linear programming**.

---

2. Content to Analyze

Problem:
{question}

Proposed Solutions (Summarized by Logical Steps):
{summaries_text}

---

3. Final Instructions & Output Requirement

Your Task:
Based on the final criteria review, analyze the solutions and make one of two possible determinations:

1. Identify the single pair of answers with the maximum methodological diversity.
2. Conclude that no pair meets the criteria for significant diversity, meaning all solutions follow a fundamentally similar approach.

Step 1: Brief Comparative Analysis

- **If you find a diverse pair:** Write a single, brief paragraph. Do not summarize each solution individually. Instead, group the solutions by common methodology and justify your selection of the most diverse pair. For example: "Solution A uses direct casework, while Solution B uses complementary counting. This represents the most significant methodological difference."
- **If you do NOT find a diverse pair:** Write a single, brief paragraph explaining why. State that all solutions follow a similar core strategy and briefly describe that common approach. For example: "All solutions utilize a system of linear equations to solve for the variables. While they use different methods like substitution or elimination, this does not represent a significant strategic divergence. Therefore, no pair is methodologically diverse."

Step 2: Final JSON Output   Immediately after your brief analysis paragraph, provide your final answer in a strict JSON format within a special block.

- **If a diverse pair is found:** The JSON should be a list containing the single selected answer ID pair.
- **If no diverse pair is found:** The JSON should contain the string "No" within the list structure to maintain format consistency.

Example of Final Output Structure (Diverse Pair):
[Your brief analysis justifying the choice...]

```
//boxed_json{{[[id_A, id_B]]}}
```

Example of Final Output Structure (No Diverse Pair):
[Your brief analysis explaining the lack of diversity...]

```
//boxed_json{{[["No"]]}}
```

Begin Analysis and Provide Output:

To automate the calculation of the "success rate," a LLM Judge (Qwen3-14B) is used to provide a final verdict on the diversity of a solution pair selected by a given method (e.g., RPD, Raw Emb., etc.). This section details the prompt used to guide the judge and the study conducted to validate its alignment with human judgment.

Judge Prompt. The judge is provided with the problem statement and a single pair of solutions. Its task is to assess whether the two solutions employed genuinely different problem-solving strategies. The prompt explicitly instructs the judge to ignore minor differences in wording or calculation and focus on the core reasoning approach.

Prompt for Methodological Similarity Rating

You are an expert Answer Analysis Assistant, specializing in understanding and comparing the logic and methodology behind problem-solving. Your task is to receive a question, two full answers with their summaries, and rate them strictly based on the similarity of their **methodology**.
Note: Based on your prior analysis, you should assume that all proposed solutions for this problem follow a similar high-level strategy. Your task is to find and rate the **methodological diversity within this shared high-level strategy**.

Rating Criteria

Your task is to determine if the two answers are **Methodologically Similar** or **Methodologically Diverse** based on the criteria below, and assign a corresponding rating.

- **Rating 1 (Methodologically Similar):** The two answers are considered similar if the differences are superficial. The following are **NOT** considered significant methodological differences:
  - **Order of Calculation:** Calculating value A then B, versus B then A, before combining them in the same way.
  - **Algebraic Equivalence:** Using the form `(a+b)^2` versus `a^2 + 2ab + b^2`.
  - **Variable Naming or Notation:** Using n vs x.
  - **Choice of Standard Procedural Equivalents:** One summary describes solving a system of equations using **substitution**, while the other uses **elimination**. These are considered standard, interchangeable procedures within the same overall algebraic approach.
  - **Rigorous Proof vs. Heuristic Assumption:** If the overall strategy is the same, simply proving a result versus assuming it does not constitute a diverse approach. Both are still following the same high-level logical path.

- **Rating 2 (Methodologically Diverse):** The two answers are considered diverse if the difference represents a **completely distinct, independent, high-level strategic choice** that fundamentally alters the entire problem-solving path from beginning to end.
  - **Example 1 (Different Overall Framework):** One solution to a geometry problem uses **coordinate geometry**, another uses **synthetic geometry**, and a third uses **vector analysis**.

Validation. To ensure the reliability of the LLM Judge used as our primary evaluation criterion in Sec. 4.1, we conduct an alignment study with human annotations.

To validate the judge, we first construct a dedicated test set. Human annotators select 100 pairs of solutions from our candidate pool, creating a balanced ground-truth dataset composed of 50 pairs with semantically *diverse* reasoning paths and 50 pairs with the *same* underlying reasoning path.

The LLM Judge is then tasked with making a binary diversity judgment on each of these 100 pairs. The results are presented in the confusion matrix in Table 8. Overall, the LLM Judge achieves an accuracy of 78%, demonstrating a strong alignment with human judgment and performing significantly better than a random baseline (50%). We observe that the judge is quite effective at identifying truly diverse

Table 8 | Confusion matrix of LLM Judge verdicts against human annotations on 100 solution pairs.

		LLM Judge Verdict	
		Diverse	Same
Human	**Diverse**	41 (TP)	9 (FN)
Label	**Same**	13 (FP)	37 (TN)

pairs (Recall 82%), though it is slightly prone to false positives (classifying similar paths as diverse). This level of agreement validates our use of the LLM Judge as a reliable automated proxy for evaluating reasoning diversity in our main experiment.

D.2. Details for Multi-Solution Fine-Tuning (Sec. 4.2)

This section provides detailed implementation procedures for the main fine-tuning experiment, focusing on how the baseline training sets were constructed. Each method aims to select 100 problems and 3 solutions per problem, but they differ in their core selection strategy.

D.2.1. Random Selection Baseline

The **Random 1P3S** baseline was constructed through a naive sampling process. We first randomly selected 100 problems from our 1,600-problem candidate pool without replacement. For each of these 100 problems, we then randomly selected 3 of its available solutions to form the training data. This method serves as a fundamental baseline to measure the benefits of any systematic diversity-driven selection.

D.2.2. LLM Selection Baseline

This baseline leverages the powerful Qwen3-14B model to simulate an expert's judgment in a two-stage curation process. First, the LLM performs a binary classification to identify whether a problem's solutions are methodologically diverse. We then selected 100 problems that were positively classified as containing diverse solution methods. Second, for these selected problems, the LLM is prompted again to choose the set of 3 solutions that are maximally distinct from each other. The specific prompts for each stage are provided below.

Prompt for Problem Diversity Classification

You are a master mathematician and an expert in pedagogical analysis. Your task is to classify a problem based on the methodological diversity of its proposed solutions.
Your goal is to perform a binary classification:

- **Class 2 (Diverse):** If the provided solution summaries showcase more than one distinct core methodology.
- **Class 1 (Not Diverse):** If all solutions use the same core methodology, or if the differences are only superficial (e.g., a different order of calculation, or using standard procedural equivalents like substitution vs. elimination).

---

1. Your Analysis Framework & Core Criteria
Your primary task is to act as a discerning analyst. You must distinguish between minor procedural choices and significant differences in core steps. Assume that most solutions might share a high-level strategy; your goal is to find answers that execute core steps in a meaningfully different way.

Defining Methodological Difference (Your Core Criteria):

What IS NOT a Significant Difference (Methodologically Similar):

- **Order of Calculation:** Calculating value A then B, versus B then A, before combining them in the same way.
- **Algebraic Equivalence:** Using the form $(a + b)^2$ versus $a^2 + 2ab + b^2$.
- **Variable Naming or Notation:** Using n vs x.
- **Choice of Standard Procedural Equivalents:** One summary describes solving a system of equations using **substitution**, while the other uses **elimination**. These are considered standard, interchangeable procedures within the same overall algebraic approach.
- **Rigorous Proof vs. Heuristic Assumption:** If the overall strategy is the same, simply proving a result versus assuming it does not constitute a diverse approach. Both are still following the same high-level logical path.

What IS a Significant Difference (Methodologically Diverse):

- This difference represents a **completely distinct, independent, high-level strategic choice** that fundamentally alters the entire problem-solving path from beginning to end.
- **Example 1 (Different Overall Framework):** One solution to a geometry problem uses **coordinate geometry**, another uses **synthetic geometry**, and a third uses **vector analysis**.
- **Example 2 (Completely Different Logical Path):** To solve a counting problem, one answer uses **direct casework**, another uses **complementary counting**, and a third uses a **recurrence relation**.
- **Example 3 (Change in Analytical Tool):** A solution to an optimization problem uses **calculus**, a second uses **inequalities** (like AM-GM), and a third uses **linear programming**.

---

2. Content to Analyze

Problem:
{question}

Proposed Solutions (Summarized by Logical Steps):
{summaries_text}

---

3. Output Requirement

Based on the final criteria review, classify the diversity of the solutions.

Output Requirement: Immediately after your classification, provide your final answer in a strict JSON format within a special block. The JSON should be a single integer, either 1 or 2. Do not provide any other text.

Example of Final Output Structure for a Diverse problem:
`//boxed{{2}}`

Example of Final Output Structure for a Not Diverse problem:
`//boxed{{1}}`

Begin Analysis and Provide Output:

Prompt for Diverse Solution Selection

You are a master mathematician and an expert in pedagogical analysis. Your task is to analyze multiple proposed solutions for a given problem and select a set of {num_to_select} answers that, as a set, represents the maximum possible methodological diversity.

Your goal is to identify a single set of {num_to_select} answers where each chosen answer has a significant methodological difference from every other answer in the set. Think of it as finding a set of three solutions that are all mutually distinct in their core approach.

1. Your Analysis Framework & Core Criteria

Your primary task is to act as a discerning analyst. You must distinguish between minor procedural choices and significant differences in core steps. Assume that most solutions might share a high-level strategy; your goal is to find answers that execute core steps in a meaningfully different way.

Defining Methodological Difference (Your Core Criteria):

What IS NOT a Significant Difference (Methodologically Similar):

- **Order of Calculation:** Calculating value A then B, versus B then A, before combining them in the same way.
- **Algebraic Equivalence:** Using the form $(a + b)^2$ versus $a^2 + 2ab + b^2$.
- **Variable Naming or Notation:** Using n vs x.
- **Choice of Standard Procedural Equivalents:** One summary describes solving a system of equations using **substitution**, while the other uses **elimination**. These are considered standard, interchangeable procedures within the same overall algebraic approach.
- **Rigorous Proof vs. Heuristic Assumption:** If the overall strategy is the same, simply proving a result versus assuming it does not constitute a diverse approach. Both are still following the same high-level logical path.

What IS a Significant Difference (Methodologically Diverse):

- This difference represents a **completely distinct, independent, high-level strategic choice** that fundamentally alters the entire problem-solving path from beginning to end.
- **Example 1 (Different Overall Framework):** One solution to a geometry problem uses **coordinate geometry**, another uses **synthetic geometry**, and a third uses **vector analysis**.
- **Example 2 (Completely Different Logical Path):** To solve a counting problem, one answer uses **direct casework**, another uses **complementary counting**, and a third uses a **recurrence relation**.
- **Example 3 (Change in Analytical Tool):** A solution to an optimization problem uses **calculus**, a second uses **inequalities** (like AM-GM), and a third uses **linear programming**.

2. Content to Analyze

Problem:
{question}

Proposed Solutions (Summarized by Logical Steps):
{summaries_text}

3. Final Instructions & Output Requirement

Your Task: Based on the final criteria review, analyze the solutions.

Step 1: Brief Comparative Analysis First, write a single, brief paragraph for your analysis. Do not summarize each solution individually. Instead, group the solutions by common methodology and justify your selection of the set of {num_to_select} most diverse answers. For example, Solutions A and C use direct casework, while Solution B uses complementary counting, and Solution D uses a geometric approach. The most diverse set is [A, B, D] as it captures these three distinct methods.

Step 2: Final JSON Output Immediately after your brief analysis paragraph, provide your final answer in a strict JSON format within a special block. The JSON should be a list containing the {num_to_select} selected answer IDs.

Example of Final Output Structure:
[Your brief analysis...]
//boxed_json{{[id_A, id_B, id_C]}}

Begin Analysis and Provide Output:

D.2.3. Embedding-Based Baseline

To rigorously evaluate the effectiveness of our RPD metric, we compare it against two baseline distance metrics. For a fair comparison, all training datasets, both for our method and the baselines, are constructed using the identical **two-stage data curation framework** detailed previously. This framework consists of **Stage 1: Problem Selection** (Algorithm 2) and **Stage 2: Greedy Solution Selection** (Algorithm 3).

The sole difference between our method and the baselines is the specific pairwise distance function, $\mathcal{D}(S_i, S_j)$, that is plugged into this framework. The baseline metrics are defined below.

Raw Solution Cosine Distance ($D_{\mathbf{raw}}$) This baseline metric computes the cosine distance between the embedding vectors of the complete solution texts. For all embedding tasks, we use the Qwen3-Embedding-8B model. Let $\mathcal{M}_{\text{embed}}$ be this model.

$$D_{\text{raw}}(S_i, S_j) = 1 - \frac{\mathcal{M}_{\text{embed}}(S_i) \cdot \mathcal{M}_{\text{embed}}(S_j)}{\|\mathcal{M}_{\text{embed}}(S_i)\| \|\mathcal{M}_{\text{embed}}(S_j)\|}$$

Summary Cosine Distance ($D_{\mathbf{summary}}$) This baseline first concatenates the step-level summaries for a solution to form a single composite summary text. The diversity is then computed as the

cosine distance between the embeddings of these composite summaries.

$$D_{\text{summary}}(S_i, S_j) = 1 - \frac{\mathcal{M}_{\text{embed}}(\text{Summary}_{\text{comp}}(S_i)) \cdot \mathcal{M}_{\text{embed}}(\text{Summary}_{\text{comp}}(S_j))}{\|\mathcal{M}_{\text{embed}}(\text{Summary}_{\text{comp}}(S_i))\| \|\mathcal{M}_{\text{embed}}(\text{Summary}_{\text{comp}}(S_j))\|}$$

Based on the framework detailed previously, we generate three distinct training datasets:

- **Ours (RPD)**: Constructed by applying the two-stage framework with our proposed RPD metric (D_{RPD}).
- **Raw Emb.**: Constructed using the same framework but with the D_{raw} metric.
- **Summary Emb.**: Constructed using the same framework but with the D_{summary} metric.

E. Experiment Results

This appendix presents the complete experimental results for both models. The tables are structured to clearly distinguish between the pre-trained baseline model, fine-tuning with a one-problem-one-solution (1P1S) paradigm, and fine-tuning with a one-problem-three-solution (1P3S) paradigm.

E.1. Complete Results for Qwen3-4B-Base Model

The following tables present the comprehensive performance of the Qwen3-4B-Base model on the AIME24 and Olympiad Benchmarks, which complements the MATH500 Level 5 results from the main paper. As shown in Table 9, our **RPD** method demonstrates a significant performance improvement by adopting the *one problem, multiple solutions* paradigm. It elevates the `pass@16` score to **35.83%** on AIME24, surpassing the standard 1P1S baseline (Random 1P1S) by an impressive **4.99** percentage points. Furthermore, our RPD-guided curation strategy also proves its superiority over other 1P3S methods, with its `pass@16` score outperforming the next-best baseline (Random 1P3S) by **2.50** percentage points on the same benchmark. This pattern holds for the Olympiad Bench (Table 10), where our method achieves a leading `pass@16` score of **68.11%**, which is **1.56** percentage points higher than the 1P1S baseline and **0.75** percentage points higher than the best alternative 1P3S method. These results provide strong evidence for the effectiveness of our approach in both paradigm and data curation strategy.

Table 9 | Full comparison on the **AIME24** benchmark using the **Qwen3-4B-Base** model.

Paradigm	Method	pass@1 (%)	pass@2 (%)	pass@4 (%)	pass@8 (%)	pass@16 (%)
Pre-trained	Base	8.34	13.33	16.67	21.67	27.50
1P1S	Random 1P1S	**14.17**	18.33	23.33	26.67	30.84
1P3S	Random 1P3S	9.17	12.50	19.17	28.34	33.33
	Raw Emb.	12.50	16.67	20.00	25.84	33.33
	Summary Emb.	10.00	12.50	17.50	25.00	29.17
	LLM Selection	10.83	15.84	20.83	25.83	30.83
	Ours (RPD)	**14.17**	**19.17**	**25.83**	**30.00**	**35.83**

Table 10 | Full comparison on the **Olympiad Bench** using the **Qwen3-4B-Base** model.

Paradigm	Method	pass@1 (%)	pass@2 (%)	pass@4 (%)	pass@8 (%)	pass@16 (%)
Pre-trained	Base	39.54	47.11	53.56	61.13	65.95
1P1S	Random 1P1S	**42.43**	49.18	55.49	61.43	66.55
1P3S	Random 1P3S	40.13	50.15	56.75	62.61	67.36
	Raw Emb.	39.91	47.48	56.38	61.42	66.62
	Summary Emb.	40.88	49.78	57.05	62.69	66.92
	LLM Selection	39.62	48.30	56.60	62.83	67.06
	Ours (RPD)	41.92	**51.19**	**57.50**	**63.06**	**68.11**

E.2. Complete Results for Qwen2.5-3B Model

To demonstrate the robustness and generalizability of our findings, we also fine-tuned the Qwen2.5-3B model. Specifically, we employed supervised fine-tuning using 4-bit QLoRA

(rank=16, alpha=32), training the model for **15 epochs** in BF16 precision. We utilized the AdamW optimizer with a cosine learning rate scheduler, setting the peak learning rate to 4×10^{-5}. We then evaluated its performance across the same three benchmarks (Tables 11, 12, and 13).

The results consistently reaffirm our core hypothesis. For instance, on the AIME24 benchmark (Table 11), our **RPD** method's advantage is particularly pronounced when evaluating with a larger sample set. Focusing on the key `pass@16` metric, our approach achieves a score of **22.50%**. This represents a substantial 5.00 percentage point improvement over the 1P1S baseline and demonstrates a clear advantage over other multi-solution strategies, outperforming the next-best 1P3S methods by 0.83 percentage points. The outperformance on AIME24 exemplifies a consistent trend also observed on the MATH500 and Olympiad benchmarks, which solidifies the conclusion that our RPD-guided data curation is a general and effective technique for enhancing Test-Time Scaling.

Table 11 | Full comparison on the **AIME24** benchmark using the **Qwen2.5-3B** model.

Paradigm	Method	pass@1 (%)	pass@2 (%)	pass@4 (%)	pass@8 (%)	pass@16 (%)
Pre-trained	Base	4.17	4.17	10.00	16.67	16.67
1P1S	Random 1P1S	4.17	8.34	10.00	13.33	17.50
1P3S	Random 1P3S	6.67	8.33	14.17	18.33	20.00
	Raw Emb.	5.84	8.34	14.17	18.33	20.83
	Summary Emb.	3.33	6.67	13.33	18.33	21.67
	LLM Selection	2.50	5.00	13.33	16.67	21.67
	Ours (RPD)	**7.50**	**10.00**	**15.00**	**20.00**	**22.50**

Table 12 | Full comparison on the **MATH500 Level 5** benchmark using the **Qwen2.5-3B** model.

Paradigm	Method	pass@1 (%)	pass@2 (%)	pass@4 (%)	pass@8 (%)	pass@16 (%)
Pre-trained	Base	23.70	32.65	43.84	55.60	63.62
1P1S	Random 1P1S	29.11	41.05	51.31	60.45	67.73
1P3S	Random 1P3S	**31.72**	**42.91**	50.94	60.82	68.28
	Raw Emb.	28.92	40.86	51.31	60.08	69.22
	Summary Emb.	27.05	38.06	51.12	60.26	67.35
	LLM Selection	27.61	37.87	49.82	60.26	67.91
	Ours (RPD)	28.55	40.30	**51.49**	**61.20**	**69.97**

Table 13 | Full comparison on the **Olympiad Bench** using the **Qwen2.5-3B** model.

Paradigm	Method	pass@1 (%)	pass@2 (%)	pass@4 (%)	pass@8 (%)	pass@16 (%)
Pre-trained	Base	21.81	30.27	37.54	45.55	51.93
1P1S	Random 1P1S	19.14	27.45	35.68	45.48	52.89
1P3S	Random 1P3S	22.33	30.79	39.10	47.11	53.93
	Raw Emb.	22.03	30.05	39.10	46.52	52.90
	Summary Emb.	**22.85**	**31.34**	38.95	46.63	53.82
	LLM Selection	21.96	30.79	**39.25**	47.11	53.94
	Ours (RPD)	20.40	30.19	39.10	**47.18**	**54.16**

E.3. Complete Results for Llama-3.1-8B-Instruct Model

To further validate our approach on a different architecture, we extended our experiments to the **Llama-3.1-8B-Instruct** model. We maintained a similar supervised fine-tuning setup using 4-bit QLoRA (rank=16, alpha=32) and BF16 precision. For this model, we adjusted the training duration to **24 epochs** and set the peak learning rate to 5×10^{-5} with the AdamW optimizer and a cosine scheduler. The model was then evaluated on the same three mathematical reasoning benchmarks (Tables 14).

The results from Llama-3.1-8B-Instruct further corroborate the effectiveness of our RPD-guided data selection. The benefits of our approach are especially pronounced on the **MATH500 Level 5** benchmark. Our method consistently outperforms the 1P1S baseline across all sampling levels, achieving the most significant gain at `pass@8` with a score of **47.76%** — a **3.35 percentage point** improvement over the baseline. This strong performance trend extends to the other challenging benchmarks, with our model also demonstrating clear advantages over the 1P1S baseline on AIME24 and Olympiad Bench. These findings strongly indicate that our RPD-based data curation is a generalizable and effective strategy for enhancing Test-Time Scaling, independent of the base model architecture.

Table 14 | Performance comparison of our 1PNS approach (RPD) against the 1P1S baseline (Random) using Llama-3.1-8B-Instruct across three mathematical reasoning benchmarks.

Benchmark	Method	pass@1 (%)	pass@2	pass@4	pass@8	pass@16
AIME24	Random (1P1S)	3.75	4.58	8.75	12.50	16.67
	RPD (1P3S)	**6.67**	**8.75**	**11.25**	**15.00**	**18.75**
MATH500 Level 5	Random (1P1S)	17.35	26.31	35.45	44.41	52.62
	RPD (1P3S)	**20.52**	**28.36**	**37.69**	**47.76**	**55.23**
Olympiad Bench	Random (1P1S)	13.99	21.29	26.86	33.46	39.62
	RPD (1P3S)	**14.54**	**21.59**	**27.45**	**33.83**	**40.58**

E.4. Experimental Details for RL Fine-Tuning

We performed an additional phase of Reinforcement Learning (RL) fine-tuning on our SFT checkpoints. The training was conducted for one epoch using the Group Relative Policy Optimization (GRPO; DeepSeek-AI, 2025) algorithm implemented in the veRL framework. Key hyperparameters and configuration details for this stage are summarized in Table 15.

Table 15 | Hyperparameters for Reinforcement Learning Fine-Tuning.

Hyperparameter	Value
Algorithm & Framework	
Framework	veRL
Algorithm	Group Relative Policy Optimization
Dataset	SimpleRL-Zoo, Medium difficulty level
Training Epochs	1
KL Coefficient (λ_{KL})	0.001
Actor Model & Optimization	
Actor Learning Rate	2×10^{-5}
LoRA Rank	32
LoRA Alpha	16
LoRA Target Modules	All linear layers
Data & Batching	
Global Batch Size	8
PPO Mini-batch Size	8
Micro-batch Size (per GPU)	4
Max Prompt Length	1800 tokens
Max Response Length	8000 tokens
Rollout & Generation	
Rollout Engine	vLLM
Rollout Samples (n)	4
Rollout Temperature	0.7
Hardware Configuration	
GPUs	2
Parallelism Strategy	Tensor Parallel

E.5. Ablation Study: Performance at a Larger Scale (1500 Samples)

To assess the scalability of our 1PNS paradigm, we conducted an additional experiment by increasing the total training data size to 1,500 samples. This study compares our diversity-driven (500Q, 3A) configuration against a traditional (1500Q, 1A) baseline. The results, presented in Table 16, show that our approach maintains a significant advantage, particularly on the `pass@k` metrics. This confirms that the benefits of multi-solution fine-tuning are robust and effective even at a larger data scale.

Table 16 | Performance comparison of our 1PNS approach against the 1P1S baseline across three mathematical reasoning benchmarks.

Benchmark	Method	pass@1 (%)	pass@2	pass@4	pass@8	pass@16
AIME24	Random (1P1S)	**13.33**	16.67	18.34	25.00	30.00
	RPD (1P3S)	12.50	**19.17**	**22.50**	**25.84**	**35.00**
MATH500 Level 5	Random (1P1S)	**52.80**	**62.32**	**68.66**	72.58	75.94
	RPD (1P3S)	51.49	58.96	66.61	**72.95**	**78.18**
Olympiad Bench	Random (1P1S)	39.77	**49.48**	55.86	62.24	66.62
	RPD (1P3S)	**39.99**	49.33	**57.20**	**63.21**	**67.51**