WHAT MATTERS IN ON-POLICY DISTILLATION? A PERSPECTIVE ON DATA EFFICIENCY AND DATA SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-Policy Distillation (OPD) has emerged as a widely adopted post-training paradigm for enhancing large language models in reasoning domains. However, the data-centric mechanisms in OPD remain relatively underexplored. This paper presents an empirical study of data efficiency and data selection in OPD. We begin by investigating an extreme setting: training OPD on only one example, namely **1-shot OPD**. Surprisingly, we find that 1-shot OPD is consistently effective across all sampled training examples and harder examples often yield superior performance gain. We next investigate what actually drives the student model's improvement in the training data. Our analysis reveals that the improvement is not driven by high token entropy, but the longer CoT paths which hard problem naturally generate. Training on longer CoT can help maintain closer alignment with the teacher over a long reasoning horizon, and learn critical thinking patterns usually missing in short CoTs, such as reflection (e.g., `"Alternatively"`). Based on these insights, we propose a simple data selection method that selects only hard examples for training, where even "unsolvable" examples that completely exceed the teacher's capability can be successfully used. Our experiments conducted on five models ranging from 1.5B to 8B show that training the student model on only 8 selected hard examples matches the performance of the 17K dataset baseline. Code and models will be publicly available.

1 INTRODUCTION

On-policy distillation (OPD) has rapidly emerged as a core technique for large language model (LLM) post-training (Lu & Lab, 2025). OPD allows a smaller student model to align its policy with a stronger teacher model by learning directly from on-policy rollouts, enabling the student to effectively inherit complex reasoning capabilities. Consequently, recent pioneering industry efforts, including Qwen3 (Yang et al., 2025), MiMo (Xiao et al., 2026), and GLM-5 (Zeng et al., 2026), have integrated OPD into their post-training pipelines, establishing it as an indispensable complement to supervised fine-tuning (SFT) and reinforcement learning with verified reward (RLVR).

However, while researchers have focused heavily on designing better training algorithms (e.g. EOPD (Jin et al., 2026) and REOPOLD (Ko et al., 2026)), the data side of OPD remains relatively underexplored. Specifically, How much data is truly necessary? What data is most effective? And what actually drives the student model's improvement in the training data? Answering these questions is of significant practical value because in many real-world scenarios, high-quality data are extremely scarce or expensive to collect, particularly in highly specialized domains such as medicine and engineering. This bottleneck makes a systematic study of data necessity highly important.

To answer these questions, we first investigate a special setting where we train OPD on only one example, namely **1-shot OPD**. Our empirical evaluations demonstrate that 1-shot OPD is consistently effective across all sampled training examples, which is also observed in a concurrent work (Fu et al., 2026). Furthermore, we show that its training is exceptionally stable and the validation performance remains remarkably stable up to even 2,000 steps without overfitting or catastrophic policy collapse. We further analyze why 1-shot OPD works so well and find that the student policy aligns closest with the teacher on structural reasoning tokens (e.g. "Alternative", "Wait") after dis-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tillation. This indicates that the student can successfully learn and generalize the teacher's reasoning patterns by distilling on only a single training example.

Then, we investigate what kind of data makes OPD most effective. We discover a clear trend: training on harder examples often yields better performance than training on easy or medium ones. Interestingly, even when these examples are so difficult that they completely exceed the capabilities of both the teacher and student models, keeping the training accuracy strictly at 0%, training on these harder examples still allows the validation accuracy to continuously improve. We further analyze why the example difficulty can influence the OPD and what actually drives the student model's improvement in the training data. We find that this improvement is not driven by high-entropy tokens, but is instead determined by the longer CoT. Specifically, we observe that training with longer CoTs helps the student maintain a closer alignment with the teacher across long horizons. Moreover, these longer paths help the model acquire thinking patterns that are rarely learned from short CoTs, such as self-reflection (e.g., "Alternatively").

Based on these findings, we propose a simple, difficulty-driven data selection method that selects only hard examples for training. We then study how many of these examples are sufficient under this selection. Surprisingly, we find that training a 1.5B student model on only 8 selected hard examples matches the validation accuracy of the full 17K dataset baseline (53.6% vs 53.7%). Furthermore, we show that this extreme data efficiency generalizes robustly across diverse model architectures and scales, ranging from 1.5B to 8B parameters.

2 PRELIMINARIES

On-Policy Distillation. OPD computes supervision on trajectories sampled from the current student policy π_θ. Given a prompt $x \sim \mathcal{D}_x$, the student samples a response sequence $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_N) \sim \pi_\theta(\cdot \mid x)$. Both the student model π_θ and the teacher model π_T are evaluated on the student-generated prefixes $\hat{y}_{<t}$, yielding two next-token probability distributions at each step t: $p_t(v) \triangleq \pi_\theta(v \mid x, \hat{y}_{<t})$ and $q_t(v) \triangleq \pi_T(v \mid x, \hat{y}_{<t})$ over the vocabulary $v \in \mathcal{V}$.

A standard formulation of OPD minimizes the sequence-level reverse Kullback-Leibler (KL) divergence over the trajectories generated by the student:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}_x}\Big[D_{\text{KL}}(\pi_\theta(\cdot \mid x) \,\|\, \pi_T(\cdot \mid x))\Big]. \tag{1}$$

Using autoregressive factorization, this sequence-level objective can be decomposed into an exact token-level formulation:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}_x,\, \hat{y} \sim \pi_\theta(\cdot|x)}\left[\sum_{t=1}^{N} D_{\text{KL}}(p_t \,\|\, q_t)\right]. \tag{2}$$

In practice, existing OPD formulations vary in their supervision granularity for computing this token-level KL, generally categorizing into sampled-token, full-vocabulary, and top-k OPD. Among these, top-k OPD restricts the divergence to a subset of high-probability tokens, significantly reducing memory overhead while preserving dense, multi-token supervision in the student's active generation region. Our experiments are primarily based on top-k OPD.

Top-k OPD. Top-k OPD restricts the divergence computation to a subset of tokens $S_t \subseteq \mathcal{V}$ at each step. Specifically, we adopt the student top-k variant, which selects the k tokens assigned the highest probabilities under the student distribution, defined as $S_t = \text{TopK}(p_t, k)$. To compute the loss on this subset, we renormalize the student and teacher distributions over S_t:

$$\bar{p}_t^{(S_t)}(v) = \frac{p_t(v)\mathbf{1}[v \in S_t]}{\sum_{u \in S_t} p_t(u)}, \quad \bar{q}_t^{(S_t)}(v) = \frac{q_t(v)\mathbf{1}[v \in S_t]}{\sum_{u \in S_t} q_t(u)}. \tag{3}$$

Distillation is then performed by minimizing the subset KL divergence $D_{\text{KL}}(\bar{p}_t^{(S_t)} \,\|\, \bar{q}_t^{(S_t)})$ over the trajectory, yielding the final objective:

$$\mathcal{L}_{\text{OPD}}^{\text{top-}k}(\theta) = \mathbb{E}_{x \sim \mathcal{D}_x,\, \hat{y} \sim \pi_\theta(\cdot|x)}\left[\sum_{t=1}^{N} D_{\text{KL}}(\bar{p}_t^{(S_t)} \,\|\, \bar{q}_t^{(S_t)})\right]. \tag{4}$$

3 PILOT EXPERIMENTS: 1-SHOT OPD

We first study an extreme setting: training a student model on only a single example, termed 1-shot OPD. This special setting serves as a clean baseline to test whether OPD can achieve policy alignment without the influence of data variety. The observation of 1-shot OPD also provides the guidance for our subsequent analyses on what kind of data is suitable and how much data is sufficient.

3.1 EXPERIMENTAL SETUP

Models. We employ DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025) as our student model, and JustRL-DeepSeek-1.5B (He et al., 2025a) as the teacher model. By default, these two models are used for our primary experiments. We also conduct experiments across other model architectures and scales in Section 5.2 to validate the generalizability of our findings.

Dataset. Due to computational resource limits, we randomly sample a subset of 1,000 examples from the DAPO-Math-17K dataset (Yu et al., 2026) to construct an example pool. To ensure reproducibility and consistent identification across all experiments. We simply rank the training examples by difficulty. Specifically, for each example in our pool, we compute the accuracy of both the student and teacher models across 16 rollouts per problem. Let S_i and T_i denote the rollout accuracies of the student and teacher models for the i-th problem, respectively. The average accuracy is computed as $A_i = (S_i + T_i)/2$. The training examples are sorted primarily in descending order of A_i, and secondarily by their original DAPO-Math-17K index to guarantee a deterministic sequence. The resulting ordered sequence is denoted as $\{\pi_i\}_{i=1}^{1000}$. Based on A_i, we further categorize the training examples into three categories: (1) **Easy** ($A_i > 0.9$), corresponding to indices π_1 to π_{199} (199 examples); (2) **Medium** ($0.1 \leq A_i \leq 0.9$), corresponding to indices π_{200} to π_{825} (626 examples); and (3) **Hard** ($A_i < 0.1$), corresponding to indices π_{826} to π_{1000} (175 examples).

Training. OPD is implemented by using the open-source `verl` (Sheng et al., 2024) framework. Following Li et al. (2026a), the training batch size and mini-batch size are 64, and we sample 8 responses for each prompt. For response generation, we utilize the `vLLM` engine with a rollout temperature of 1.0 and maximum response length of 7,168 tokens. The per-token reverse KL divergence is computed against only the top-16 tokens. We train the OPD in the DAPO-Math-17K for 1 epoch as full dataset baseline (Full-Set OPD), yields 279 training steps. For fair comparisons, we train our 1-shot OPD under 279 training steps by default. To test the domain generalizability of our findings, we also extend our investigation to the code generation domain by distilling on training prompts sampled from the LeetCodeDataset v3.0.1 (Xia et al., 2025). The complete experimental results for the coding domain are presented in Appendix C.1. All experiments are executed on an 8 $\times$ NVIDIA H800 80GB GPU cluster. More training details are included in Appendix A.1.

Evaluation. We evaluate the models across six standard mathematical reasoning benchmarks: AIME 2024 (Art of Problem Solving, a), AIME 2025 (Art of Problem Solving, a), AMC 2023 (Art of Problem Solving, b), MATH500 (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). Additionally, we extend our evaluation to non-mathematical domains including LiveCodeBench (Jain et al., 2025) and GPQA (Rein et al., 2023). The evaluation results for these non-mathematical datasets are provided in Appendix B. For all benchmarks, we sample with a temperature of 0.7 and a maximum output length of 16,384 tokens. Due to the limited sample sizes of AIME 2024, AIME 2025, and AMC 2023, we perform 16 rollouts per problem and report the average accuracy (*mean@16*) to ensure evaluation stability. For the remaining datasets, we report the average accuracy over 4 rollouts (*mean@4*).

3.2 1-SHOT OPD IS EFFECTIVE FOR MANY EXAMPLES

First, we randomly sample 8 training examples from each of the Easy, Medium, and Hard categories and train on them individually. The specific indices of sampled examples are listed in Table 10. During training, we utilize AMC 2023, AIME 2024, and AIME 2025 as the validation set to monitor performance changes over steps. Consistent with our evaluation, we sample 16 solutions per problem with a temperature of 0.7 and a maximum response length of 16,384 tokens, reporting the average accuracy over the 16 samples as our validation metric. Additionally, we track the overlap ratio (Li et al., 2026a) to measure policy alignment between the student and the teacher.

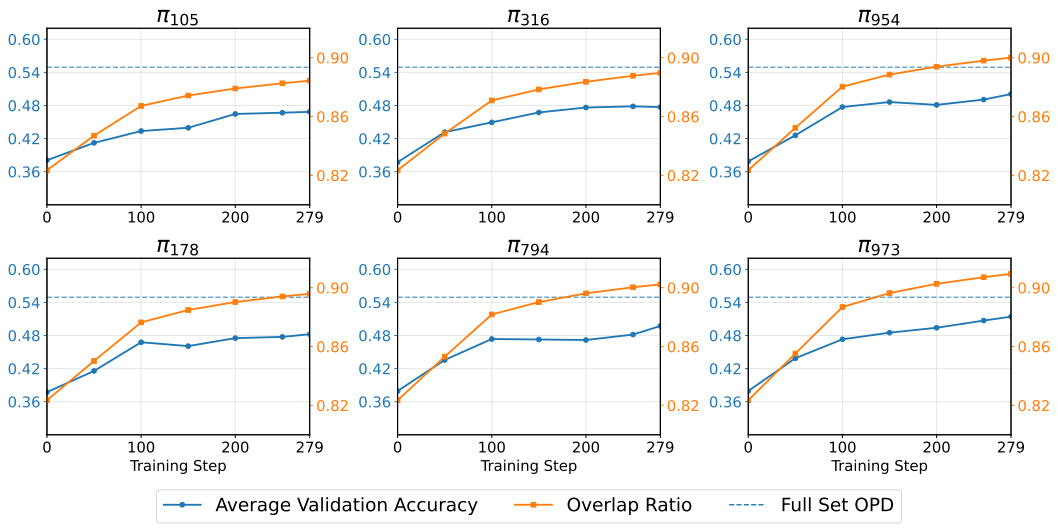

Figure 1: **Learning Dynamics of 1-shot OPD.** We visualize the training trajectories of six representative examples across three difficulty levels: two easy problems ($\{\pi_{105}\}$ and $\{\pi_{178}\}$, left), two medium problems ($\{\pi_{316}\}$ and $\{\pi_{794}\}$, middle), and two hard problems ($\{\pi_{954}\}$ and $\{\pi_{973}\}$, right). The solid blue curves denote the Average Validation Accuracy (computed across AMC 2023, AIME 2024, and AIME 2025). The solid orange curves with markers track the Overlap Ratio, indicating progressive token-level policy alignment with the teacher model. The horizontal dashed lines represent the full-set baseline (Full-Set OPD) for comparison.

Specifically, we randomly select 100 examples from the DAPO-Math-17K as a held-out set. At each training checkpoint, we perform 16 independent rollouts for each example. Then we report the average proportion of tokens that appear simultaneously in the top-16 vocabulary distributions of both the student and the teacher as the overlap ratio.

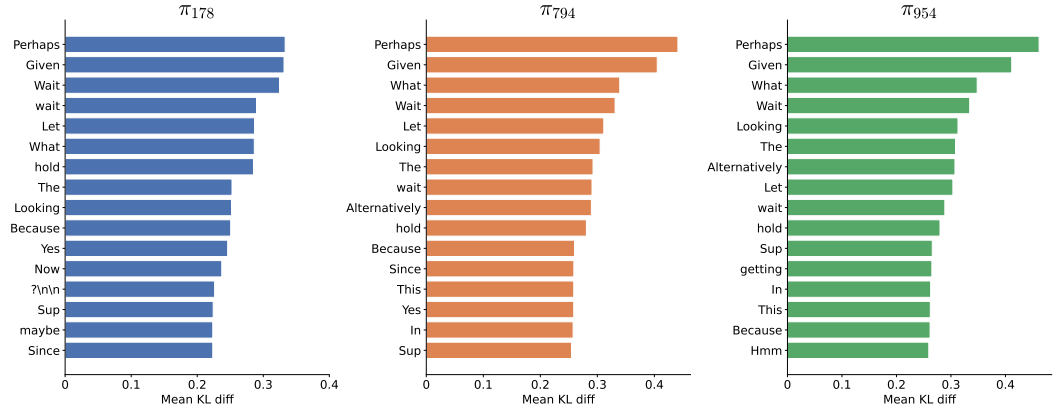

Figure 2: **Per-Token KL Divergence Reduction post 1-Shot OPD.** We report the top 16 tokens with the highest expected KL reduction after distillation for models trained respectively on $\{\pi_{178}\}$ (left), $\{\pi_{794}\}$ (middle), and $\{\pi_{954}\}$ (right).

We present the evaluation results across benchmarks in Table 1. Moreover, we select some examples to visualize their training trajectories in Figure 1. As illustrated in Figure 1, the overlap ratio continuously increases, indicating that the student model is progressively aligning with the teacher model. Moreover, the performance improvement demonstrates the effectiveness of 1-shot OPD. We further perform a prolonged training run of 2,000 steps to observe whether 1-shot OPD suffers from

overfitting. As shown in Figure 7, the validation performance plateaus after 300 steps and remains stable up to 2,000 steps without any performance decay or policy collapse.

To understand why 1-shot OPD is so effective, we analyze which tokens shift closest to the teacher after distillation. Specifically, we use the held-out set (100 problems) to generate 16 rollouts per problem. Then we compute the reduction in token-level KL divergence between the student and teacher distributions post-distillation and average this KL reduction over all token IDs. For statistical significance, we discard the rare tokens with an overall occurrence frequency of less than 0.01%. In Figure 2, we report the top 16 tokens with the highest KL reduction for models trained respectively on $\{\pi_{178}\}$, $\{\pi_{794}\}$, and $\{\pi_{954}\}$. The results reveal that the tokens aligning most significantly with the teacher after OPD are predominantly structural reasoning, including reflection (e.g., "Alternatively"), transitions (e.g., "Wait", "Perhaps"), and deductive reasoning (e.g., "Because", "Since"). This indicates that the student can successfully learn the teacher's reasoning patterns by distilling on only a single training example.

Table 1: **1-Shot OPD Performance Across Mathematical Reasoning Benchmarks.** We compare the student model DeepSeek-R1-Distill-Qwen-1.5B (Base), the teacher model JustRL-DeepSeek-1.5B (Teacher), and the full-dataset distillation baseline (Full-Set) against our 1-shot OPD models. Notably, 1-shot models trained on hard problems consistently outperform those trained on easy problems, with the best-performing example ($\{\pi_{973}\}$) achieving 51.7% average accuracy, nearly matching the Full-Set baseline (53.7%).

Model	AIME24	AIME25	AMC23	MATH500	Olympiad	Minerva	Avg
Base	31.7	23.0	60.7	82.6	34.1	22.4	42.4
Teacher	55.8	35.8	83.4	87.4	43.4	29.8	55.9
Full-Set	51.9	34.0	78.9	86.7	44.3	26.6	53.7
Easy Problems							
$\{\pi_5\}$	39.2	28.8	69.9	85.3	39.3	25.3	48.0
$\{\pi_{20}\}$	40.3	30.6	69.1	86.2	40.8	25.7	48.8
$\{\pi_{21}\}$	41.8	29.0	71.1	85.9	39.7	26.2	48.9
$\{\pi_{63}\}$	42.2	30.8	71.0	86.2	39.7	25.0	49.1
$\{\pi_{84}\}$	41.0	30.1	72.6	85.6	40.6	25.8	49.3
$\{\pi_{105}\}$	42.5	27.7	70.3	86.0	41.1	25.0	48.8
$\{\pi_{178}\}$	41.1	30.4	71.2	86.7	40.4	25.9	49.3
$\{\pi_{192}\}$	39.8	30.2	72.0	85.7	40.1	25.7	48.9
Medium Problems							
$\{\pi_{316}\}$	39.8	30.0	73.3	86.4	40.6	25.7	49.3
$\{\pi_{487}\}$	41.2	29.8	72.7	85.7	40.4	25.9	49.3
$\{\pi_{530}\}$	42.1	30.7	73.1	86.5	40.2	26.8	49.9
$\{\pi_{543}\}$	40.4	29.4	73.0	86.4	41.3	26.9	49.6
$\{\pi_{661}\}$	43.1	31.2	73.4	86.9	41.3	26.4	50.4
$\{\pi_{763}\}$	41.5	28.1	75.6	86.7	41.2	26.8	50.0
$\{\pi_{794}\}$	44.2	30.8	74.2	86.5	41.6	25.7	50.5
$\{\pi_{820}\}$	41.5	29.1	73.6	84.9	40.7	25.9	49.3
Hard Problems							
$\{\pi_{874}\}$	43.3	33.1	74.3	86.6	41.0	27.1	50.9
$\{\pi_{890}\}$	45.2	29.6	76.0	85.6	42.4	26.5	50.9
$\{\pi_{948}\}$	45.0	29.8	75.3	86.0	41.8	27.0	50.8
$\{\pi_{954}\}$	44.8	28.5	76.9	86.9	42.6	26.7	51.1
$\{\pi_{961}\}$	44.4	30.6	73.2	87.2	41.4	26.7	50.6
$\{\pi_{973}\}$	47.5	31.7	75.1	87.7	41.5	26.9	51.7
$\{\pi_{987}\}$	44.8	27.7	73.6	87.5	41.2	26.4	50.2
$\{\pi_{997}\}$	44.0	30.2	75.8	85.6	42.1	26.8	50.8

4 Data Selection for OPD

4.1 Harder Problems Often Lead to Better Performance

We further investigate whether different data behave differently in 1-shot OPD. For each category, we record both the average training accuracy and validation accuracy every 50 training steps, and present these trajectories in Figure 3. Surprisingly, we find that harder training examples often yield better performance. Moreover, we observe that for easy and hard problems, even after their

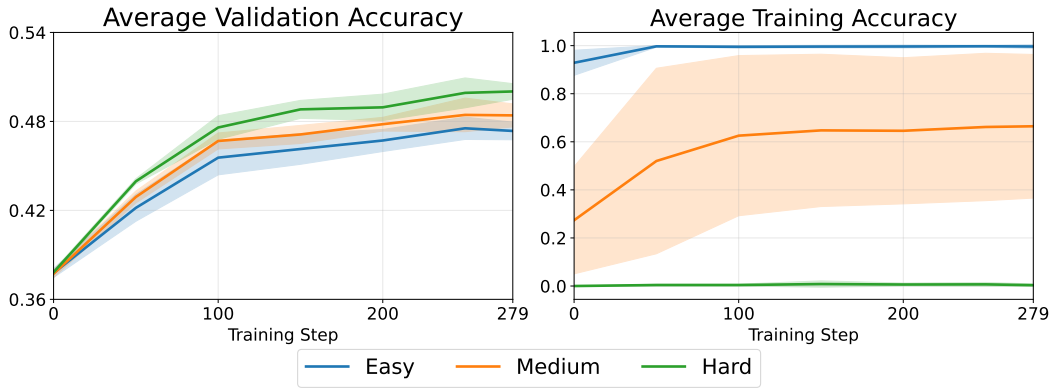

Figure 3: **Learning Trajectories across Difficulty Levels.** Left: Average Validation Accuracy over training steps. Right: Average Training Accuracy on the selected training example over training steps. Solid lines indicate the mean and shaded areas represent the standard deviation across the examples in each category.

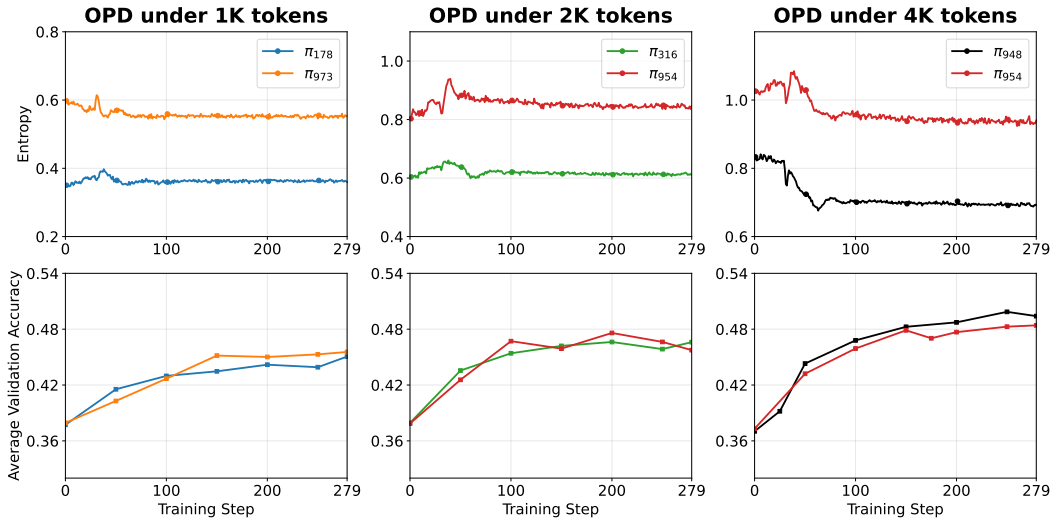

Figure 4: **Training Dynamics in Length-Constrained Setting** Training dynamics are presented for problem pairs grouped by difficulty: Easy vs. Hard (1K token limit, left), Medium vs. Hard (2K token limit, middle), and Hard vs. Hard (4K token limit, right). The top plots monitor token entropy, while the bottom plots track validation accuracy.

training accuracies saturate at 100% (for easy tasks) or remain flat at 0% (for hard tasks), their validation accuracies continuously improve. This behavior contrasts sharply with RL (e.g., GRPO-style algorithms) where training on extreme easy and hard samples is highly ineffective because the computed advantage or gradient updates tend to be zero.

4.2 Deeper Analysis

We next explore what underlying properties drive the efficacy of difficult prompts, examining potential factors such as the higher token entropy and the longer CoT reasoning sequences they naturally generate.

Setup. To isolate the role of token entropy from length related factors, we set up a length-controlled experiment. By adjusting the maximum rollout length, we expect that almost all rollouts generated for some examples will exceed this limit and get cut off. Consequently, training the 1-shot OPD uses almost the same number of tokens for those examples because the length of every response is forced

to be the maximum rollout length. Specifically, we conduct three sets of experiments. In each set, we select two problems with different difficulty or entropy, and set an appropriate maximum rollout length such that empirically over 99.5% of the rollouts for both problems exceed this threshold. The three problem pairs are: 1) Easy problem π_{178} and Hard problem π_{973} limited to 1024 tokens, 2) Medium problem π_{316} and Hard problem π_{954} limited to 2048 tokens, and 3) two Hard problems with highly different token entropy, π_{948} and π_{954} limited to 4096 tokens.

Higher Token Entropy Alone Does Not Improve OPD As illustrated in Figure 4, under length-controlled settings, we observe no significant difference in validation performance between the compared pairs, despite their highly divergent token entropy. Specifically, once rollout lengths are restricted to match, the performance gap between the pairs collapses to 0.5% (for $\{\pi_{973}\}$ vs. $\{\pi_{178}\}$) and -0.9% (for $\{\pi_{954}\}$ vs. $\{\pi_{316}\}$). However, in the default setting (7,168 maximum rollout length) where the student model can leverage the longer CoT sequences generated by harder examples, the harder problems demonstrate a notable advantage, outperforming $\{\pi_{178}\}$ and $\{\pi_{316}\}$ by 3.9% and 2.4% in validation accuracy, respectively. This collapse under length limits indicates that higher token entropy alone does not translate to better distillation performance. Additionally, we perform an auxiliary experiment where we sample responses for the Easy problem π_{178} under higher temperatures to artificially increase token entropy. As shown in Figure 9, the resulting model shows no accuracy gains, indicating that the performance improvement of difficult tasks is not driven by their higher token entropy.

Distillation Benefits from Longer CoTs We further compare student models distilled on the hard problem $\{\pi_{954}\}$ under different maximum response length of 1024, 2048, 4096, and 7168 tokens (referred to as 1K, 2K, 4K, and 7K models, respectively). Figure 5 illustrates the validation accuracy dynamics across training steps under these length limitation. The empirical results show a clear trend within the 1K to 7K token range: increasing the maximum response length enables the student model to engage in longer reasoning processes, leading to progressively higher validation accuracy. To explore the underlying mechanics, we track the token level KL loss over generation position in Figure 8. As depicted in Figure 8, the 7K model maintains a consistently lower KL divergence across extended generation horizons, whereas shorter variants exhibit a sharp rise in KL loss as the generation sequence lengthens. This indicates that training on longer CoT sequences directly enhances the model's longer-horizon reasoning capabilities, enabling it to sustain policy alignment with the teacher over extended context lengths.

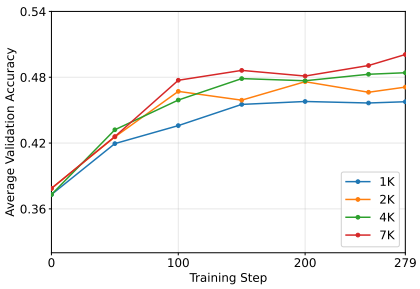

Figure 5: **Validation performance across different maximum response length.** We compare distillation validation accuracy across training steps under 1K, 2K, 4K, and 7K maximum token limits.

Harder Problems can Trigger More Reasoning Patterns. Furthermore, we observe that training on harder problems does not simply scale token volume, but instead learns more reasoning patterns, such as self-reflection and backtracking. As shown in Figure 2, for the model training on easy problem ($\{\pi_{178}\}$) where trajectories are naturally short, the self-reflection token "`Alternatively`" is absent from the top-16 KL reduction list. However, it emerges in the medium model ($\{\pi_{794}\}$), and ranks higher in the hard model ($\{\pi_{954}\}$). This finding suggests that longer CoT trajectories allows the student model to acquire some sophisticated reasoning patterns that are inherently absent in shorter, simpler reasoning paths.

5 How Much Data is Sufficient for OPD?

5.1 Few-Shot OPD Can Match Full-set Performance

Based on Section 4, we simply select hard examples as training set and investigate how scaling the training set size affects performance. Specifically, we scale the number of Hard training examples $N \in \{1, 4, 8, 16, 64\}$. To construct our training sets, for $N = 1, 4$, and 8, we directly use the hard examples evaluated in Table 10 (i.e., $\{\pi_{973}\}$ for 1-shot, $\{\pi_{874}, \pi_{890}, \pi_{954}, \pi_{973}\}$ for 4-shot, and $\{\pi_{874}, \ldots, \pi_{997}\}$ for 8-shot). For the larger sizes of $N = 16$ and 64, in addition to 8 hard

examples in Table 10, we randomly sample the remaining examples from our hard example pool. For comparison, we evaluate Easy and Medium baselines of size $N = 8$ using the eight corresponding examples (i.e., $\{\pi_5, \pi_{20}, \ldots, \pi_{192}\}$ for Easy, and $\{\pi_{316}, \pi_{487}, \ldots, \pi_{820}\}$ for Medium). Additionally, we also construct a Random baseline by randomly sampling 8 examples from the hold out training pool. Table 2 reports these results. To ensure that our data selection strategy is robust against sampling randomness, we further evaluate its stability across three independent random seeds in Appendix D. To evaluate whether our methodology generalizes beyond mathematical reasoning, we also evaluate data selection in the code generation domain, with complete experimental setups and results presented in Appendix C.1.

From the results in Table 2, we find that: 1) increasing the training set of Hard examples from $N = 1$ to $N = 8$ steadily improves performance, with 8 examples already matching the full-dataset (17K) distillation baseline, while further increasing the size to 16 or 64 examples yields no additional gains; and 2) consistent with our 1-shot findings, under a fixed size of 8 examples, training on Hard examples yields the best performance over the Easy, Medium, and Random baselines.

Table 2: **Few-Shot OPD Performance across Different Dataset Sizes.** The baselines include the full-dataset distillation (DAPO-Math-17K) and a randomly sampled 1K subset (DAPO-Subset). Distilling on just 8 hard examples achieves 53.6% average accuracy, nearly matching the Full-Set 17K baseline (53.7%).

Dataset	Size	AIME24	AIME25	AMC23	MATH500	Olympiad	Minerva	Avg
DAPO-Math-17K	17K	51.9	34.0	78.9	86.7	44.3	26.6	53.7
DAPO-Subset	1K	50.2	34.8	79.3	87.1	42.8	27.3	53.6
$\{\pi_5, \pi_{20}, \ldots, \pi_{192}\}$	8	45.8	30.6	76.1	87.1	41.6	26.2	51.2
$\{\pi_{316}, \pi_{487}, \ldots, \pi_{820}\}$	8	49.0	32.7	76.8	86.8	42.1	26.0	52.2
Random	8	48.1	33.8	76.2	86.3	41.9	26.4	52.1
$\{\pi_{973}\}$	1	47.5	31.7	75.1	87.7	41.5	26.9	51.7
$\{\pi_{874}, \pi_{890}, \pi_{954}, \pi_{973}\}$	4	50.8	31.5	78.5	86.7	41.5	28.2	52.9
$\{\pi_{874}, \ldots, \pi_{997}\}$	8	51.7	34.2	79.2	87.2	42.7	26.4	53.6
$\{\pi_{874}, \ldots, \pi_{997}, \ldots\}$	16	49.4	33.3	78.5	87.7	43.0	27.7	53.3
$\{\pi_{874}, \ldots, \pi_{997}, \ldots\}$	64	50.2	36.7	79.1	87.8	42.7	27.1	53.9

5.2 Few-Shot OPD on Other Models

To investigate whether this extreme data efficiency generalizes across other models, we additionally evaluate four teacher–student pairs: 1) **Skywork-OR1-Math-7B** $\rightarrow$ **DeepSeek-R1-Distill-Qwen-7B**; 2) **DeepSeek-R1-Distill-Llama-8B-GRPO** $\rightarrow$ **DeepSeek-R1-Distill-Llama-8B**, where the former is obtained by applying GRPO to the latter (detailed training settings are provided in Appendix A.2); 3) **Qwen3-4B** $\rightarrow$ **Qwen3-1.7B-Base**; 4) **Qwen3-4B** $\rightarrow$ **Qwen3-4B-Base**. The setting vary scale, capability, and backbone architecture. For Qwen3-4B, we disable thinking mode. Similar to Section 5.1, we train the student models on either 8 selected hard examples ($\{\pi_{874}, \ldots, \pi_{997}\}$) or 8 randomly sampled examples to compare their distillation performance. As shown in Table 3, few shot OPD on selected hard examples consistently triggers substantial performance gains, outperforming the Random baseline and closely approaching full dataset distillation.

6 Related Work

On-Policy Distillation. Distillation for LLMs is broadly categorized into off-policy and on-policy settings. While off-policy distillation (e.g., SFT) suffers from exposure bias due to the mismatch between static teacher-generated data and student-generated rollouts at inference (Agarwal et al., 2024), OPD mitigates this issue by training on trajectories sampled directly from the student's current policy under a reverse KL objective (Gu et al., 2024). Recently, OPD has since appeared in some reasoning and post-training recipes, such as Qwen3 (Yang et al., 2025), Deepseek-V4 (Xu et al., 2026) and GLM-5 (Zeng et al., 2026). Building on these successes, a growing body of work has investigated various algorithmic variants of OPD (Oh et al., 2026; Yang et al., 2026), optimized its practical training recipes (Li et al., 2026b; Ko et al., 2026), and integrated it into large-scale post-training pipelines (Ma et al., 2026). Moreover, several studies have further analyzed the underlying mechanisms of OPD (Wang et al., 2026b; Li et al., 2026a).

Table 3: **Selecting few hard examples for OPD is effective across different models.** We report benchmark accuracies across diverse model pairings and training sizes.

Model	Size	AIME24	AIME25	AMC23	MATH500	Olympiad	Minerva	Avg
Skywork-OR1-Math-7B → DeepSeek-R1-Distill-Qwen-7B								
DeepSeek-R1-Distill-Qwen-7B	–	51.9	38.8	81.0	89.6	43.6	32.6	56.3
Skywork-OR1-Math-7B	–	62.1	45.6	84.0	91.9	47.7	34.1	60.9
Full-Set	17K	58.3	44.6	83.4	91.1	46.0	34.2	59.6
Random	8	57.3	42.1	81.1	91.0	45.2	33.2	58.3
$\{\pi_{874}, \ldots, \pi_{997}\}$	8	58.3	43.9	83.1	91.3	46.3	34.0	59.5
DeepSeek-R1-Distill-Llama-8B-GRPO → DeepSeek-R1-Distill-Llama-8B								
DeepSeek-R1-Distill-Llama-8B	–	46.0	28.4	78.5	86.0	41.9	27.1	51.3
DeepSeek-R1-Distill-Llama-8B-GRPO	–	50.8	32.4	85.3	87.7	44.2	28.7	54.9
Full-Set	17K	50.1	31.2	84.7	87.4	43.0	28.4	54.1
Random	8	48.9	29.7	82.2	86.7	42.3	27.7	52.9
$\{\pi_{874}, \ldots, \pi_{997}\}$	8	50.3	30.9	84.3	87.1	42.8	28.1	53.9
Qwen3-4B → Qwen3-1.7B-Base								
Qwen3-1.7B-Base	–	1.4	2.5	12.3	13.3	4.8	3.7	6.3
Qwen3-4B	–	22.9	17.1	62.0	81.1	45.4	26.1	42.4
Full-Set	17K	10.2	4.0	28.6	53.9	24.4	14.1	22.5
Random	8	6.5	3.6	24.6	46.0	22.3	11.1	19.0
$\{\pi_{874}, \ldots, \pi_{997}\}$	8	9.4	4.4	26.8	51.0	23.7	12.7	21.3
Qwen3-4B → Qwen3-4B-Base								
Qwen3-4B-Base	–	7.3	5.0	24.6	22.3	11.9	4.8	12.7
Qwen3-4B	–	22.9	17.1	62.0	81.1	45.4	26.1	42.4
Full-Set	17K	14.4	14.4	40.5	65.6	33.3	16.5	30.8
Random	8	11.9	10.6	36.1	59.3	29.1	15.0	27.0
$\{\pi_{874}, \ldots, \pi_{997}\}$	8	13.1	12.5	38.6	64.1	31.3	15.6	29.2

Data Selection for LLM Post-Training. Selecting a small yet effective subset of training data has become an important problem for improving the efficiency of LLM post-training, with most efforts focusing on data selection for SFT (Ivison et al., 2025) and RLVR (Wu et al., 2026). Existing approaches include LLM-based quality assessment (Chen et al., 2024), gradient-based selection (Xia et al., 2024) and more. However, data selection for OPD remains underexplored. Our work is inspired by recent advances in RLVR, which show that RLVR can achieve significant reasoning gains with as few as a single training example (Wang et al., 2026a). We extend this exploration to the OPD paradigm, discovering different and interesting phenomena. A concurrent work by Fu et al. (2026) also investigates 1-shot OPD and analyzes its efficacy from a state-space coverage perspective. Differently, we analyze 1-shot OPD from the perspective of CoT trajectory and reasoning patterns. Then we propose a simple and effective data selection method that selects only hard examples for training, where even "unsolvable" examples completely exceeding the teacher's capability can be successfully used.

7 Conclusions

In this work, we show that 1-shot OPD is sufficient to trigger substantial improvements in mathematical reasoning tasks, with as few as 8 selected examples nearly matching the performance of distilling on the full DAPO-Math-17K dataset. Moreover, we observe that distilling on harder tasks consistently outperforms easy ones, and even "unsolvable" tasks where both the teacher and student models fail to produce correct answers still yield strong performance gains. Our analysis reveals that this improvement is fundamentally driven by the longer CoT trajectories rather than high token-level entropy. Future work includes: 1) designing effective dynamic batching strategies algorithms of OPD based on our findings; and 2) developing novel OPD algorithms that can learn from longer CoT reasoning trajectories, where the teacher's dense reward may lose local exploitability.

AI USE STATEMENT

In this work, we used generative AI tools for assisting writing, including language polishing and grammatical corrections. We have not used generative AI tools for generating ideas and experimental coding. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our empirical findings, we have made extensive efforts across datasets, models, and experimental configurations. All benchmarks, datasets, the student and teacher base models, are publicly available. The sampled training prompts along with their corresponding ground truths are explicitly provided in Appendix G. Furthermore, training details, hyperparameter configurations, and evaluation details are provided in Section 3.1 and Appendix A. To facilitate further research, our complete codebase, training scripts, and distilled model weights will be publicly released upon paper acceptance.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Art of Problem Solving. Aime problems and solutions. `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`, a. Accessed: 2025-04-20.

Art of Problem Solving. Amc problems and solutions. `https://artofproblemsolving.com/wiki/index.php?title=AMC_Problems_and_Solutions`, b. Accessed: 2025-04-20.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpagasus: Training a better alpaca with fewer data. In *International Conference on Learning Representations*, volume 2024, pp. 34767–34797, 2024.

Zixuan Fu, Bingxiang He, Yuxin Zuo, Haohuan Huang, Jinqian Zhang, Ruhang Xiao, Cheng Qian, Qinyu Luo, Huan-ang Gao, Yudong Wang, et al. Rethinking on-policy distillation of large language models ii: One training example. *arXiv preprint arXiv:2609.04172*, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

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

Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, et al. Justrl: Scaling a 1.5 b llm with a simple rl recipe. *arXiv preprint arXiv:2512.16649*, 2025a.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. 2025b. URL `https://arxiv.org/abs/2504.11456`.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. Large-scale data selection for instruction tuning. *arXiv preprint arXiv:2503.01807*, 2025.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026a.

Yuying Li, Leqi Zheng, Yongzi Yu, Wenrui Zhou, Xuchang Zhong, Xing Hu, Jing Jin, Hangjie Yuan, and Tao Feng. Filter, then reweight: Rethinking optimization granularity in on-policy distillation. *arXiv preprint arXiv:2606.02684*, 2026b.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, et al. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training. *arXiv preprint arXiv:2606.30406*, 2026.

Minjae Oh, Sangjun Song, Gyubin Choi, Yunho Choi, and Yohan Jo. Kl for a kl: On-policy distillation with control variate baseline. *arXiv preprint arXiv:2605.07865*, 2026.

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

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

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *Advances in Neural Information Processing Systems*, 38: 122721–122764, 2026a.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation. *arXiv preprint arXiv:2605.26844*, 2026b.

Jianghao Wu, Jianfei Cai, Weiqiang Wang, Jin Ye, Daniel F Schmidt, and Yasmeen George. Single-rollout hidden-state dynamics for training-free rlvr data selection. *arXiv preprint arXiv:2605.28631*, 2026.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Yunhui Xia, Wei Shen, Yan Wang, Jason Klein Liu, Huifeng Sun, Siyue Wu, Jian Hu, and Xiaolong Xu. Leetcodedataset: A temporal dataset for robust evaluation and efficient training of code llms, 2025. URL https://arxiv.org/abs/2504.14655.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

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

Zhicheng Yang, Zhijiang Guo, Yifan Song, Minrui Xu, Yongxin Wang, Yiwei Wang, Xiaodan Liang, and Jing Tang. Prune-opd: Efficient and reliable on-policy distillation for long-horizon reasoning. *arXiv preprint arXiv:2605.07804*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

APPENDIX TABLE OF CONTENTS

A More Training Details **14**

A.1 OPD Training Details 14

A.2 GRPO Training Details 14

B More Evaluations **14**

C More Experiments **15**

C.1 1-shot OPD in Coding Domain 15

C.2 1-shot OPD over Extended Training 16

C.3 1-shot OPD under Different CoT Lengths 16

C.4 1-shot OPD under Different Temperature 17

D Robustness of Data Selection to Random Seeds **17**

E Limitations **18**

F Discussions **18**

G Example Details **19**

G.1 Accuracy of Sampled Examples 19

G.2 Content of Sampled Examples 20

A More Training Details

A.1 OPD Training Details

For every input prompt during the OPD training, we generate $n = 8$ responses. We constrain the maximum prompt length to 1,024 tokens, while the maximum response length is limited to 7,168 tokens. The optimization process spans a single epoch across 8 H800 80G GPUs, utilizing a learning rate of 1×10^{-6}. We set both the student sampling temperature and the teacher temperature to 1.0, disable KL regularization, and adopt token-mean loss aggregation. Unless otherwise noted, all experiments use the default OPD hyperparameters listed in Table 4.

Table 4: Default hyperparameters for OPD.

Item	Value
Training temperature	1.0
Global batch size	64
Mini batch size	64
Rollout number	8
Temperature	1.0
LogProb top-K	16
Top-K strategy	Student Top-K
Top-p	1.0
Max prompt length	1024
Max response length	7168
Learning rate	1e-6
Epoch	1
KL Coefficient	0.0

A.2 GRPO Training Details

We train DeepSeek-R1-Distill-Llama-8B for one epoch on the dataset DeepMath-103K (He et al., 2025b). The main hyperparameters are summarized in Table 5.

Table 5: Default hyperparameters for GRPO.

Item	Value
Training temperature	1.0
Global batch size	64
Mini batch size	64
Rollout number	8
Temperature	1.0
Max prompt length	1024
Max response length	7168
Learning rate	1e-6
Epoch	1
KL Coefficient	0.0

B More Evaluations

To verify whether the performance gains acquired through 1-shot OPD generalize beyond the mathematical domain, we evaluate our models on out-of-distribution benchmarks, including LiveCodeBench for code reasoning and GPQA for scientific reasoning. Specifically, we report the LiveCodeBench and GPQA accuracy for 1-shot OPD models from Table 1 in Table 6, and the performance of few-shot OPD models from Table 2 in Table 7. The empirical results further validate that our observations remain highly consistent across out-of-distribution tasks. First, 1-shot OPD trained

on a single mathematical example successfully transfers reasoning abilities to coding and scientific domain, yielding notable performance gains over the base student model. As shown in Table 6, the 1-shot OPD training on single hard problems achieves an average accuracy of 40.5% on LiveCodeBench and 24.2% on GPQA, significantly outperforming the base student model baselines of 32.4% and 19.4%, respectively. Second, training on harder problems consistently achieves superior cross domain performance compared to easy problems. According to Table 6, the average accuracy of 1-shot OPD on hard problems (40.5% on LiveCodeBench and 24.2% on GPQA) markedly surpasses that training on easy problems (37.6% on LiveCodeBench and 22.1% on GPQA). Third, distilling on 8 selected hard examples is sufficient to closely approach the full dataset baseline. As reported in Table 7, it reaches 42.0% on LiveCodeBench and 27.2% on GPQA, nearly matching the full 17K dataset performance of 42.2% on LiveCodeBench and 27.8% on GPQA.

Table 6: **Out of Distribution Generalization of 1-Shot OPD Models from Table 1.** We report the accuracy on LiveCodeBench (LCB) and GPQA across individual models. The **Avg** rows denote the average accuracy across the 8 models trained on Easy, Medium, and Hard samples, respectively.

Easy Problems			**Medium Problems**			**Hard Problems**		
Model	**LCB**	**GPQA**	**Model**	**LCB**	**GPQA**	**Model**	**LCB**	**GPQA**
$\{\pi_5\}$	37.3	21.4	$\{\pi_{316}\}$	39.3	22.2	$\{\pi_{874}\}$	40.9	24.6
$\{\pi_{20}\}$	38.3	21.1	$\{\pi_{487}\}$	39.6	23.7	$\{\pi_{890}\}$	40.1	23.4
$\{\pi_{21}\}$	37.1	23.5	$\{\pi_{530}\}$	39.4	23.0	$\{\pi_{948}\}$	41.8	25.9
$\{\pi_{63}\}$	38.8	21.6	$\{\pi_{543}\}$	38.9	22.3	$\{\pi_{954}\}$	40.8	23.7
$\{\pi_{84}\}$	36.9	22.7	$\{\pi_{661}\}$	39.1	23.6	$\{\pi_{961}\}$	39.7	23.8
$\{\pi_{105}\}$	36.8	21.5	$\{\pi_{763}\}$	40.0	24.8	$\{\pi_{973}\}$	39.3	23.1
$\{\pi_{178}\}$	38.7	23.1	$\{\pi_{794}\}$	39.4	22.7	$\{\pi_{987}\}$	40.4	24.3
$\{\pi_{192}\}$	37.0	21.9	$\{\pi_{820}\}$	38.4	22.3	$\{\pi_{997}\}$	40.8	24.5
Avg	**37.6**	**22.1**	**Avg**	**39.3**	**23.1**	**Avg**	**40.5**	**24.2**

Table 7: **Out of Distribution Generalization of Few-Shot OPD Models from Table 2.** We report the average accuracy in LiveCodeBench and GPQA.

Model	**Size**	**LiveCodeBench**	**GPQA**
Student	N/A	32.4	19.4
Teacher	N/A	42.8	28.0
DAPO-Math-17K	17K	42.2	27.8
$\{\pi_5, \pi_{20}, \ldots, \pi_{192}\}$	8	40.6	23.7
$\{\pi_{316}, \pi_{487}, \ldots, \pi_{820}\}$	8	40.1	24.1
Random	8	40.3	24.3
$\{\pi_{874}, \pi_{890}, \pi_{954}, \pi_{973}\}$	4	41.4	24.8
$\{\pi_{874}, \ldots, \pi_{997}\}$	8	42.0	27.2
$\{\pi_{874}, \ldots, \pi_{997}, \ldots\}$	16	40.6	26.7
$\{\pi_{874}, \ldots, \pi_{997}, \ldots\}$	64	41.7	27.5

C More Experiments

C.1 1-shot OPD in Coding Domain

To evaluate whether our findings generalize beyond mathematical reasoning, we extend our empirical investigation to the code generation domain using the LeetCodeDataset v3.0.1 (Xia et al., 2025), comprising 2,641 problems in total. The dataset provides pre-existing difficulty classifications, containing 638 Easy, 1,397 Medium, and 606 Hard problems. Directly utilizing these native difficulty tags, we sample single training prompts from the Easy, Medium, and Hard categories, respectively. We employ DeepSeek-R1-Distill-Qwen-1.5B as the student model and JustRL-DeepSeek-1.5B as the teacher model. For evaluation, we track LiveCodeBench as the in distribution validation set. To monitor out of distribution performance, we average the accuracy across AMC23, AIME24, and AIME25 as Math Accuracy metric. As illustrated in Figure 6, 1-shot OPD trained on code generation tasks yields substantial validation gains across training steps, while the steady convergence of

distillation loss shows effective policy alignment between the student and teacher models. Furthermore, consistent with our mathematical domain observations, distilling on hard coding problems achieves superior performance on both in-distribution and out-of-distribution metrics compared to easy or medium problems.

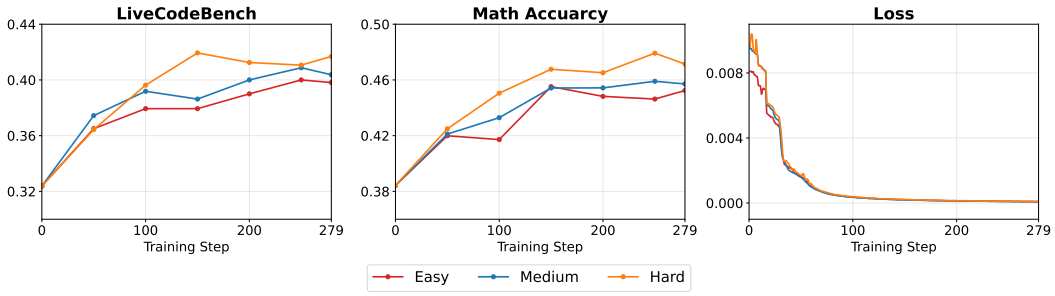

Figure 6: **Training dynamics and validation trajectories of 1-shot OPD in the coding domain.** We report the in-distribution validation accuracy on LiveCodeBench, the out of distribution Math Accuracy (computed as the average accuracy across AMC23, AIME24, and AIME25), and the distillation loss over training steps.

To further evaluate whether our difficulty-driven data selection strategy remains effective in the coding domain, we conduct experiments scaling training prompts from the LeetCodeDataset. Specifically, we randomly sample 8 hard problems to construct our Hard dataset. For comparison, we establish a Full-Set baseline using the entire LeetCodeDataset of 2,641 problems, as well as a Random baseline trained on 8 randomly sampled problems across all difficulty levels. As reported in Table 8, our difficulty-driven data selection demonstrates efficiency in the coding domain. Training on only 8 selected hard coding problems yields an average accuracy of 47.1%, significantly outperforming the Random baseline (45.1%) and nearly recovering the entire performance ceiling of the Full-Set baseline (47.3%). These results indicate that our data selection strategy generalizes robustly to code generation.

Table 8: **Data Selection Efficacy in the Coding Domain.** We report benchmark performance across six mathematical reasoning benchmarks alongside LiveCodeBench and GPQA. We evaluate the student model (DeepSeek-R1-Distill-Qwen-1.5B) distilled on LeetCodeDataset training prompts under different selection strategies.

Model	Size	AIME24	AIME25	AMC23	MATH500	Olympiad	Minerva	LiveCodeBench	GPQA	Avg
Student	N/A	31.7	23.0	60.7	82.6	34.1	22.4	32.4	19.4	38.3
Teacher	N/A	55.8	35.8	83.4	87.4	43.4	29.8	42.8	28.0	50.8
Full-Set	2641	44.4	32.1	78.7	86.3	40.8	27.0	42.7	26.4	47.3
Random	8	41.2	29.0	72.0	86.1	39.3	26.1	41.6	25.6	45.1
Hard	8	**43.7**	**31.8**	**77.2**	**87.7**	**39.9**	**27.0**	**42.8**	**26.3**	**47.1**

C.2 1-SHOT OPD OVER EXTENDED TRAINING

We further investigate whether 1-shot OPD suffers from overfitting. To do so, we perform a prolonged training experiment by extending the optimization duration to 2,000 steps for 1-shot OPD training on $\{\pi_{105}\}$, $\{\pi_{794}\}$, and $\{\pi_{973}\}$, respectively. As shown in Figure 7, we observe that the validation accuracy across all three models rises steadily during the first 300 steps and subsequently remains remarkably stable up to 2,000 steps, showing no signs of performance decay or catastrophic policy collapse. This behavior is fundamentally different from SFT and RL in single-example settings (Wang et al., 2026a), which often suffer from severe overfitting or policy degradation under extended optimization. One potential reason is that the sequence-level KL divergence between the teacher and student models naturally decreases as training progresses, leading to loss convergence and preventing policy collapse even under prolonged training.

C.3 1-SHOT OPD UNDER DIFFERENT COT LENGTHS

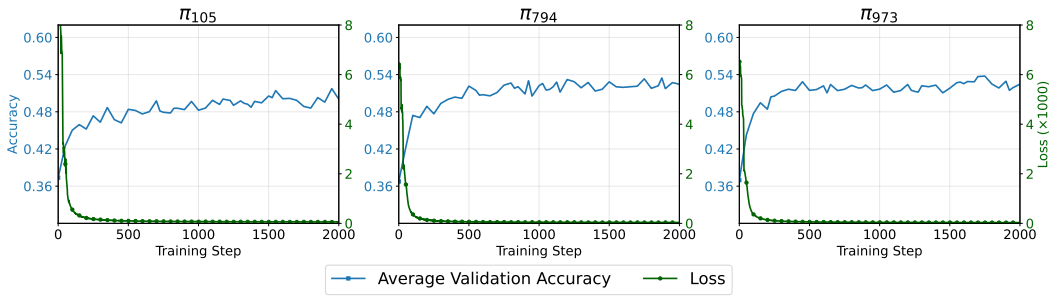

Figure 7: **Validation Accuracy over Extended Training in 1-Shot OPD.** We plot the validation trajectories over 2,000 steps for 1-shot OPD models trained on three representative examples of varying difficulty levels: $\{\pi_{105}\}$ (Easy, left), $\{\pi_{794}\}$ (Medium, middle), and $\{\pi_{973}\}$ (Hard, right). Despite extended training, the validation performance across all models remains exceptionally stable without policy collapse.

We further compare student models distilled on the hard problem $\{\pi_{954}\}$ under maximum rollout length constraints of 1024, 2048, 4096, and 7168 tokens (referred to as the 1K, 2K, 4K, and 7K models, respectively). For evaluation, we generate 16 rollout completions per problem across the 100 held-out problems. At each generation step, we compute the per-token KL divergence between the distilled student and the teacher policies and plot the average KL loss over positions in Figure 8. The empirical results in Figure 8 reveal that the 7K model consistently maintains a lower KL divergence across the entire generation horizon compared to the 1K, 2K and 4K models. Crucially, while the alignment of the 4K model is nearly identical to that of the 7K model within the 0 to $4,000$ token range, the KL gap between them widens significantly as the generation horizon expands beyond $4,000$ tokens. This divergence demonstrates that training on longer CoT sequences directly enhances the model's longer-horizon reasoning capabilities, enabling it to sustain policy alignment with the teacher over extended context lengths.

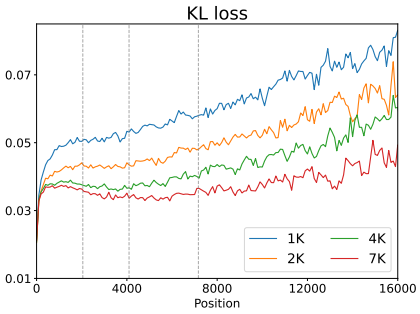

Figure 8: **Token-level KL Loss over Generation Position.** For visual clarity, we partition the token positions into non-overlapping intervals of 100 tokens and plot the binned average KL loss over positions.

C.4 1-SHOT OPD UNDER DIFFERENT TEMPERATURE

We conduct an auxiliary experiment by varying the student rollout temperature T in $\{0.6, 0.8, 1.0, 1.1\}$ during 1-shot OPD. This experiment is specifically designed around one easy problem π_{178}. By adjusting the sampling temperature of the rollout, we artificially inflate the token-level entropy of the student model's rollouts without altering the inherent difficulty of the training example itself. As illustrated in Figure 9, while higher temperatures markedly elevate the average token entropy throughout training, the corresponding validation accuracy almost keep unchanged across all temperature settings. This empirical result demonstrates that increasing token entropy does not translate to validation gains.

D ROBUSTNESS OF DATA SELECTION TO RANDOM SEEDS

To evaluate the stability and robustness of our data selection method against sampling randomness, we extend the setup in Section 5.1 by introducing two additional independent random seeds, totaling three independent seeds. Specifically, for each seed, we sample a distinct subset of 8 Hard problems (Hard 8) as well as 8 Random problems (Random 8) from the training pool, where the first seed corresponds to that reported in the main text. All models are evaluated using the student model (DeepSeek-R1-Distill-Qwen-1.5B) and the teacher model (JustRL-DeepSeek-1.5B) across eight

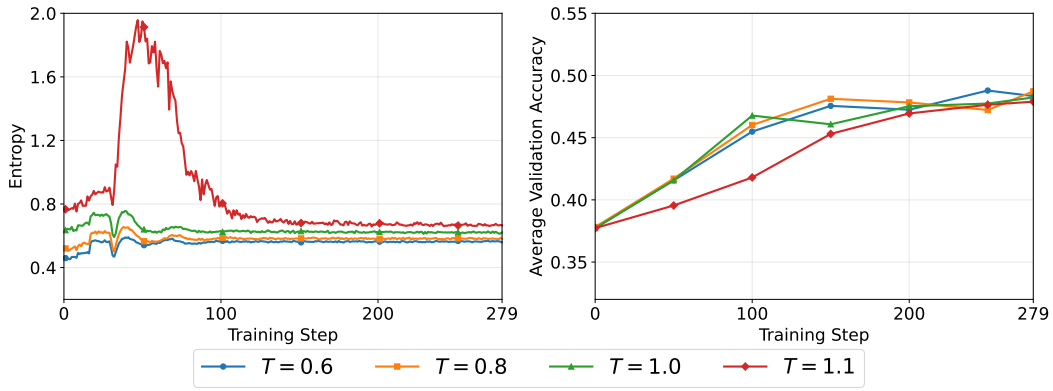

Figure 9: **Study on student rollout temperature T for the easy example π_{178} during 1-shot OPD.** The left subfigure monitors the average per-token entropy of the student's rollouts across training steps under different sampling temperatures $T \in \{0.6, 0.8, 1.0, 1.1\}$. The right subfigure tracks the corresponding Average Validation Accuracy (computed over AMC 2023, AIME 2024, and AIME 2025).

benchmarks, comprising six mathematical reasoning tasks alongside LiveCodeBench and GPQA. As reported in Table 9, the performance of our data selection consistently outperforms the random selection across all seeds.

Table 9: **Robustness of Data Selection Across Different Random Seeds.** We report benchmark accuracies across eight evaluation benchmarks alongside their average (Avg) for different data selection under three independent random sampling seeds. Group 1 represents the default seed used in the main text, while Group 2 and Group 3 represent additional independent random seeds.

Model	AIME24	AIME25	AMC23	MATH500	Olympiad	Minerva	LiveCodeBench	GPQA	Avg
Random Data Selection									
Seed 1	48.1	33.8	76.2	86.3	41.9	26.4	40.3	24.3	47.2
Seed 2	46.7	35.0	76.1	87.0	41.6	26.1	39.9	21.9	46.8
Seed 3	46.2	33.8	77.0	86.4	42.4	26.2	40.1	24.9	47.1
Our Data Selection									
Seed 1	51.7	34.2	79.2	87.2	42.7	26.4	42.0	27.2	**48.8**
Seed 2	50.3	34.3	78.8	87.1	42.0	28.1	41.7	27.6	**48.7**
Seed 3	51.3	34.9	79.3	87.5	42.3	27.8	42.2	27.8	**49.1**

E LIMITATIONS

A key limitation of our work is that our difficulty-driven data selection is a simple empirical choice rather than a mathematically optimal one. Finding the theoretically optimal training sets for OPD may be computationally intractable. However, despite not being theoretically optimal, our method is extremely simple to use. It requires zero complex optimization, zero extra training parameters, and can be implemented simply by sorting accuracy of examples. We believe this extreme simplicity and ease of implementation make our findings highly valuable for practical applications.

F DISCUSSIONS

CoT Length as a Practical Proxy in Verifier-Free Domains. In mathematical and coding domains, rule-based verifiers are readily available to compute rollout accuracy for task selection. However, in many general reasoning domains, such verifiers do not exist, making difficulty-based selection impossible. For these domains, our findings suggest a simple and effective alternative: using the length of the CoT as a proxy for data selection. As demonstrated in Section 4, CoT length is the key driver of policy alignment. Distilling on longer CoT sequences not only enables the student

model to maintain alignment stability over a long reasoning horizon, but also successfully activates advanced reasoning patterns (e.g. reflection), that may miss in short trajectories.

Discussion on Hard Problems with Short Response. A natural question arises regarding potential edge cases in our difficulty-driven selection strategy: what if a problem is categorized as hard due to low model accuracy, yet yields relatively short CoT reasoning responses? We term such instances pseudo-hard problems. These cases often stem from subtle trick questions, common calculation traps, or counterintuitive prompts where models frequently make premature errors without engaging in multi-step exploration. Because these prompts fail to elicit essential reasoning behaviors such as self-reflection, error correction, or backtracking, distilling on them may not yield superior performance gains. Nevertheless, our empirical observations indicate that such pseudo-hard examples are exceptionally rare in practice, as problem difficulty and generated response length exhibit a strong positive correlation. Consequently, while difficulty serves as a robust proxy for selection, incorporating response length as a secondary filtering metric could further refine data selection in edge scenarios.

The Limits of Scaling Rollout Horizon and the Exposure Bias Trade-off. A natural question is whether continuously increasing the maximum response length will always yield better policy alignment. We argue that there exists a empirical trade-off: while longer trajectories contain richer reasoning patterns, excessively long on-policy rollouts suffer from severe exposure bias. Because OPD optimizes the student policy against its own dynamically sampled trajectories, any reasoning errors made by the student model will compound over a long horizon. This compounding error drifts the student into out-of-distribution states, which significantly degrades the quality of the teacher's policy supervision. The maximum rollout length in our experiment is chosen as 7,168 following the empirical choice in Li et al. (2026a). To train on even longer CoT sequences without policy decay, we think designing the appropriate training algorithm (e.g. Prune-OPD (Yang et al., 2026)) is necessary.

G Example Details

G.1 Accuracy of Sampled Examples

To classify the difficulty of tasks in our example pool and select representative problems, we evaluate the baseline performance of both the student and teacher models. Specifically, we run 16 independent rollouts per problem for the student model (DeepSeek-R1-Distill-Qwen-1.5B) and the teacher model (JustRL-DeepSeek-1.5B), respectively. For each problem i, let S_i and T_i denote the rollout accuracies of the student and teacher models. We then compute the average accuracy as $A_i = (S_i + T_i)/2$ to represent the task difficulty.

Based on this metric, we rank all training examples and group them into three categories: (1) Easy ($A_i > 0.9$), (2) Medium ($0.1 \leq A_i \leq 0.9$), and (3) Hard ($A_i < 0.1$). Table 10 reports the detailed rollout accuracies of both models on the representative training problems selected in our experiments.

Table 10: **Accuracies of Sampled Problems.** We report the rollout accuracies of both the student and teacher models across 16 rollouts per problem during our data preparation phase.

Easy Problems			Medium Problems			Hard Problems		
ID	**Student**	**Teacher**	**ID**	**Student**	**Teacher**	**ID**	**Student**	**Teacher**
π_{5}	100.0	100.0	π_{316}	68.8	100.0	π_{874}	0.0	6.3
π_{20}	100.0	100.0	π_{487}	37.5	100.0	π_{890}	0.0	0.0
π_{21}	100.0	100.0	π_{530}	25.0	100.0	π_{948}	0.0	0.0
π_{63}	93.8	100.0	π_{543}	50.0	75.0	π_{954}	0.0	0.0
π_{84}	93.8	100.0	π_{661}	25.0	68.8	π_{961}	0.0	0.0
π_{105}	87.5	100.0	π_{763}	6.3	37.5	π_{973}	0.0	0.0
π_{178}	87.5	100.0	π_{794}	0.0	37.5	π_{987}	0.0	0.0
π_{192}	87.5	100.0	π_{820}	6.3	18.8	π_{997}	0.0	0.0

G.2 Content of Sampled Examples

Table 11: **Details of example π_5.**

Prompt:

```
Let $a$ and $b$ be positive real numbers.  Find the minimum value of
\[a^2 + b^2 + \frac{1}{(a + b)^2}.\]The answer is in the form k\sqrt{
m}+n,. Please provide the value of k + m + n. Please reason step by
step, and put your final answer within \boxed{{}}.
```

Ground truth: 3.

Table 12: **Details of example π_{20}.**

Prompt:

```
In trapezoid $ABCD$ the lengths of the bases $AB$ and $CD$ are 8 and
17 respectively. The legs of the trapezoid are extended beyond $A$
and $B$ to meet at point $E$. What is the ratio of the area of
triangle $EAB$ to the area of trapezoid $ABCD$? Express your answer
as a common fraction.The answer is in the form rac{m}{n}, where gcd(m
, n) = 1. Please provide the value of m + n. Please reason step by
step, and put your final answer within \boxed{{}}.
```

Ground truth: 289.

Table 13: **Details of example π_{21}.**

Prompt:

```
Lines $l_1^{}$ and $l_2^{}$ both pass through the origin and make
first-quadrant angles of $\frac{\pi}{70}$ and $\frac{\pi}{54}$
radians, respectively, with the positive $x$-axis. For any line  $l$,
 the transformation $R(l)$ produces another line as follows: $l$ is
reflected in $l_1$, and the resulting line is reflected in $l_2$. Let
 $R^{(1)}(l)=R(l)$ and $R^{(n)}(l)=R\left(R^{(n-1)}(l)\right)$. Given
 that $l$ is the line $y=\frac{19}{92}x$, find the smallest positive
integer $m$ for which $R^{(m)}(l)=l$. Please reason step by step, and
 put your final answer within \boxed{{}}.
```

Ground truth: 945.

Table 14: **Details of example π_{63}.**

Prompt:

```
On the party, every boy gave $1$ candy to every girl, and every girl
gave $1$ candy to every boy. Then every boy ate $2$ candies, and
every girl ate $3$ candies. It is known that $\frac{1}{4}$ of all
candies were eaten. Find the greatest possible number of children at
the party. Please reason step by step, and put your final answer
within \boxed{{}}.
```

Ground truth: 35.

Table 15: **Details of example π_{84}.**

Prompt:

```
The prime numbers $a$, $b$, and $c$ satisfy the equation $a + bˆ2 = 4
cˆ2$. Determine the sum of all possible values of $a + b + c$. Please
 reason step by step, and put your final answer within \boxed{{}}.
```

Ground truth: 31.

Table 16: **Details of example π_{105}.**

Prompt:

```
Compute the smallest positive integer $N$ for which $N \cdot 2ˆ{2024}
$ is a multiple of $2024$. Please reason step by step, and put your
final answer within \boxed{{}}.
```

Ground truth: 253.

Table 17: **Details of example π_{178}.**

Prompt:

```
Determine the number of pairs $(a,b)$ of real numbers such that $10,
a, b, ab$ is an arithmetic progression. Please reason step by step,
and put your final answer within \boxed{{}}.
```

Ground truth: 2.

Table 18: **Details of example π_{192}.**

Prompt:

```
Compute the remainder when $2ˆ{3ˆ5}+ 3ˆ{5ˆ2}+ 5ˆ{2ˆ3}$ is divided by
$30$. Please reason step by step, and put your final answer within \
boxed{{}}.
```

Ground truth: 6.

Table 19: **Details of example π_{316}.**

Prompt:

```
A dartboard consists of three concentric circles with radii 4, 6, and
 8. Three darts are thrown at the board, sticking at random locations
. Determine the probability that each dart lands in a different
region of the dartboard. The probability can be expressed as \( \frac
{m}{n} \), where \( m \) and \( n \) are relatively prime positive
integers. Calculate \( m + n \). Please reason step by step, and put
your final answer within \boxed{{}}.
```

Ground truth: 617.

Table 20: **Details of example π_{487}.**

Prompt:

```
In how many ways can the integers from 1 to n be ordered subject to
the condition that, except for the first integer on the left, every
integer differs by 1 from some integer to the left of it? Please
provide the number of ways for $n = 6$. Please reason step by step,
and put your final answer within \boxed{{}}.
```

Ground truth: 32.

Table 21: **Details of example π_{530}.**

Prompt:

```
Determine the largest integer $N$ for which there exists a $6 \times
N$ table $T$ that has the following properties:\n\n- Every column
contains the numbers $1, 2, \ldots, 6$ in some ordering.\n- For any
two columns $i \ne j$, there exists a row $r$ such that $T(r,i) = T(r
,j)$.\n- For any two columns $i \ne j$, there exists a row $s$ such
that $T(s,i) \ne T(s,j)$. Please reason step by step, and put your
final answer within \boxed{{}}.
```

Ground truth: 120.

Table 22: **Details of example π_{543}.**

Prompt:

```
Let f be a polynomial of degree $3$ with integer coefficients such
that $f(0) = 3$ and $f(1) = 11$.
If f has exactly $2$ integer roots, how many such polynomials $f$
exist? Please reason step by step, and put your final answer within \
boxed{{}}.
```

Ground truth: 0.

Table 23: **Details of example π_{661}.**

Prompt:

```
Find the area of the figure on the coordinate plane bounded by the
straight lines $x = 0$, $x = 2$, and the graphs of the functions $y =
 \\sqrt{x^3 + 1}$ and $y = -\\sqrt[3]{x^2 + 2x}$. Please reason step
by step, and put your final answer within \\boxed{{}}.
```

Ground truth: 6.

Table 24: **Details of example π_{763}.**

Prompt:

```
An organization has $30$ employees, $20$ of whom have a brand A
computer while the other $10$ have a brand B computer. For security,
the computers can only be connected to each other and only by cables.
 The cables can only connect a brand A computer to a brand B computer
. Employees can communicate with each other if their computers are
directly connected by a cable or by relaying messages through a
series of connected computers. Initially, no computer is connected to
 any other. A technician arbitrarily selects one computer of each
brand and installs a cable between them, provided there is not
already a cable between that pair. The technician stops once every
employee can communicate with each other. What is the maximum
possible number of cables used? Please reason step by step, and put
your final answer within \\boxed{{}}.
```

Ground truth: 191.

Table 25: **Details of example π_{794}.**

Prompt:

```
Tanya wrote numbers in the form $n^7 - 1$ for $n = 2, 3, \ldots$ and
noticed that for $n = 8$, she obtained a number divisible by $337$.
For what minimal $n$ did she get a number divisible by $2022$? Please
 reason step by step, and put your final answer within \boxed{{}}.
```

Ground truth: 79.

Table 26: **Details of example π_{820}.**

Prompt:

```
Vasya has $n$ candies of several types, where $n > 145$. It is known
that for any group of at least 145 candies, there is a type of candy
which appears exactly 10 times. Find the largest possible value of
$n$. Please reason step by step, and put your final answer within \
boxed{{}}.
```

Ground truth: 160.

Table 27: **Details of example π_{874}.**

Prompt:

```
Find the number of subsets of $\{1,3,5,7,9,11,13,15,17,19\}$ where
the elements in the subset add to $49$. Please reason step by step,
and put your final answer within \boxed{{}}.
```

Ground truth: 22.

Table 28: **Details of example π_{890}.**

Prompt:

```
Let $x_1, x_2, \dots, x_{100}$ be real numbers such that $|x_1| = 63$
 and $|x_{n+1}| = |x_n + 1|$ for $n = 1, 2, \dots, 99$. Find the
largest possible value of $(-x_1 - x_2 - \cdots - x_{100})$. Please
reason step by step, and put your final answer within \boxed{{}}.
```

Ground truth: 2034.

Table 29: **Details of example π_{948}.**

Prompt:

```
Two real numbers $x$ and $y$ are chosen at random in the interval
(0,1) with respect to the uniform distribution. What is the
probability that the closest integer to $x/y$ is even? The original
answer is in the form $r + s\pi$, please give the value of $r + s$.
Please reason step by step, and put your final answer within \boxed
{{}}.
```

Ground truth: 4.

Table 30: **Details of example π_{954}.**

Prompt:

```
Let $f:\{1,2,\dots,2019\}\to\{-1,1\}$ be a function, such that for
every $k\in\{1,2,\dots,2019\}$, there exists an $\ell\in\{1,2,\dots
,2019\}$ such that $$ \sum_{i\in\mathbb{Z}:(\ell-i)(i-k)\geqslant 0}
f(i)\leqslant 0. $$ Determine the maximum possible value of $$ \sum_{
i\in\mathbb{Z}:1\leqslant i\leqslant 2019} f(i). $$ Please reason
step by step, and put your final answer within \boxed{{}}.
```

Ground truth: 673.

Table 31: **Details of example π_{961}.**

Prompt:

```
A herder has forgotten the number of cows she has and does not want
to count all of them. She remembers these four facts about the number
 of cows:\n\n- It has $3$ digits.\n- It is a palindrome.\n- The
middle digit is a multiple of $4$.\n- It is divisible by $11$.\n\
nWhat is the sum of all possible numbers of cows that the herder has
?\n\n$\\textbf{(A) }343 \\ \\textbf{(B) }494 \\ \\textbf{(C) }615 \\
\\textbf{(D) }635 \\ \\textbf{(E) }726$ Please reason step by step,
and put your final answer within \\boxed{{}}.
```

Ground truth: 726.

Table 32: **Details of example π_{973}.**

Prompt:

```
Find all ordered pairs $(a, b)$ of positive integers for which\n$$\n\
frac{1}{a}+\frac{1}{b}=\frac{3}{2018} .\n$$\nPlease provide the sum
of all integers in the ordered pairs. Please reason step by step, and
 put your final answer within \boxed{{}}.
```

Ground truth: 1438383.

Table 33: **Details of example π_{987}.**

Prompt:

```
Find the smallest positive integer $n$ with the following property:
for every sequence of positive integers $a_1,a_2,\ldots , a_n$ with
$a_1+a_2+\ldots +a_n=2013$, there exist some (possibly one)
consecutive term(s) in the sequence that add up to $70$. Please
reason step by step, and put your final answer within \boxed{{}}.
```

Ground truth: 1033.

Table 34: **Details of example π_{997}.**

Prompt:

```
A quadratic polynomial $f(x)$ is called sparse if its degree is
exactly 2 , if it has integer coefficients, and if there exists a
nonzero polynomial $g(x)$ with integer coefficients such that $f(x) g
(x)$ has degree at most 3 and $f(x) g(x)$ has at most two nonzero
coefficients. Find the number of sparse quadratics whose coefficients
 lie between 0 and 10, inclusive. Please reason step by step, and put
 your final answer within \boxed{{}}.
```

Ground truth: 228.