Hard Examples Are All You Need: Maximizing GRPO Post-Training Under Annotation Budgets

Benjamin Pikus[*]
Luma AI
bpikus@gmail.com

Pratyush Ranjan Tiwari[*]
Eternis Labs
pratyush@eternis.ai

Burton Ye
Writer, Inc
burton@writer.com

Abstract

Collecting high-quality training examples for language model fine-tuning is expensive, with practical budgets limiting the amount of data that can be procured. We investigate whether example difficulty affects GRPO training effectiveness by comparing selection strategies (easy, medium, hard, random) across multiple models and reasoning tasks. Training on the hardest 10% of examples (those where the base model fails most often) yields dramatic performance gains up to 47%, while easy examples produce minimal improvements of 3-15%. This occurs because GRPO requires outcome variance to generate learning signals; hard examples maintain mixed success/failure outcomes throughout training while easy examples quickly converge to consistent success, eliminating learning opportunities. Moreover, models trained on hard examples show superior out-of-distribution generalization, with only hard-trained models achieving meaningful gains on the AIME2025 benchmark. Our findings provide clear guidance: when budget-constrained, prioritize collecting and annotating examples where your base model struggles, as these drive nearly all learning value in GRPO fine-tuning.

1 Introduction

Large language model (LLM) alignment and post-training are fundamentally constrained by the expense of acquiring high-quality supervision data, making the selection of training examples a critical factor in achieving optimal model performance [15].[2] While recent advances in reinforcement learning from human feedback (RLHF) and related techniques have demonstrated remarkable improvements in model capabilities, practitioners face a practical challenge: given limited resources for data annotation and curation, which examples should be prioritized to maximize post-training performance?

In this work, we focus on Group Relative Policy Optimization (GRPO), a PPO-style algorithm that replaces a learned value function with group-normalized advantages, reducing memory and relying on within-group reward variance for learning signals. We address a specific instantiation of this budget-aware selection problem: given a fixed budget to select and train on only a fraction of available prompts, how should we choose this subset to maximize the effectiveness of Group Relative Policy Optimization (GRPO) fine-tuning? Specifically, we investigate whether selecting examples based on their difficulty (as measured by the base model's success rate across multiple sampling attempts) leads to systematic differences in final model performance on held-out test sets.

Our key research questions are:

1. Should practitioners prioritize examples that are hard (where the base model frequently fails), easy (where it frequently succeeds), medium, or a selection of random difficulty examples?

[*]Equal contribution
[2]Code available here

2. Does the optimal selection strategy vary across model scales and families?
3. What mechanisms explain the differential effectiveness of difficulty-based selection strategies under GRPO's learning dynamics?
4. How do different difficulty-based selection strategies impact out-of-distribution generalization to substantially harder problem sets?
5. Can we achieve similar performance gains by training only on examples the base model gets wrong, avoiding the circular dependency of knowing which examples will improve?

Our findings reveal a striking pattern: **hard examples are all you need**. Training on the hardest 10% of examples yields performance gains up to 47%, while easy examples produce minimal improvements of 3-15%. This 30+ percentage point gap transforms marginally effective fine-tuning into highly successful model improvement.

To answer these questions, we make the following contributions:

Budget-aware evaluation protocol. We develop a systematic framework for comparing difficulty-conditioned training subsets under GRPO, using multi-sample base-model probing to robustly estimate example difficulty [24]. This protocol enables fair comparison across selection strategies while controlling for computational budget and other confounding factors.

Comprehensive experimental evaluation. We conduct extensive experiments on GSM8K grade-school math problems and BIG-Bench Hard's Tracking Shuffled Objects task across models (Qwen3-4B, Qwen3-14B, Phi-4, and Llama3.1-8B) [7, 25, 1]. Our results consistently show that training on the hardest 10% of examples yields superior test performance on reasoning tasks.

Mechanistic understanding of selection effects. We analyze why hard examples prove most effective under GRPO: the algorithm requires within-group outcome variance to generate learning signals. When all samples in a group produce identical rewards, the advantages become zero and learning stops. Our analysis shows that hard examples maintain mixed outcomes longer than easy examples, which quickly converge to deterministic success [18].

Out-of-distribution generalization analysis. We evaluate models trained on different difficulty-based subsets against the AIME2025-I benchmark, a substantially harder test set than the in-distribution GSM8K [14, 2]. We find that training on the hardest examples not only improves in-distribution performance, but is also the only strategy that yields meaningful gains on OOD data.

Base wrong vs base right analysis. We validate our "hard examples" principle from a complementary perspective by categorizing examples based on base model performance. This analysis reveals that nearly all learning value comes from examples the base model initially gets wrong, with training on "base wrong" examples achieving up to 23.5% relative improvements over "base right" examples. This convergent evidence that both low pass@k examples and base-wrong examples drive learning confirms that GRPO fundamentally benefits from training at the frontier of model capabilities.

Our findings have immediate practical implications: when using GRPO for reasoning task fine-tuning under budget constraints, practitioners should prioritize collecting and annotating examples where the base model struggles. We validate this principle from two complementary angles: examples with low pass@k success rates and examples the base model gets wrong both drive superior learning outcomes. This convergence demonstrates that "hard examples" identified through either metric capture the same underlying phenomenon: problems at the frontier of model capabilities that provide rich learning signals. This hard-example focus can yield performance improvements *exceeding 30 percentage points* over easy selection on some configurations, transforming marginally effective fine-tuning into highly successful model improvement.

Our study isolates *offline* difficulty-based selection under GRPO. It complements *online* difficulty-targeted selection with rollout replay that prioritizes moderate difficulty [19], advances in process-reward RL (PRIME) [5], and analyses of GRPO optimization biases such as Dr. GRPO [13]. We focus on outcome-reward GRPO; extending our protocol to dense process rewards or debiased GRPO is left to future work.

The remainder of this paper is organized as follows. Section 2 presents our experimental protocol for difficulty estimation and subset selection. Section 3 reports our main experimental results across tasks and model scales. Section 4 provides detailed analysis of learning dynamics and model behavior. Section 5 concludes with practical recommendations and future directions. Related work can be found in the appendix.

2 Experimental Protocol

This section describes our experimental framework for evaluating the impact of example difficulty on GRPO fine-tuning under budget constraints. We detail our difficulty estimation procedure, subset selection policies, training methodology, and evaluation metrics.

2.1 Goal

Our primary research goal is to determine which subset selection strategy maximizes post-GRPO test accuracy when constrained to using only $p = 10\%$ of the available training pool. Specifically, given a pool of unlabeled prompts $\mathcal{X}$, we select subsets $S \subset \mathcal{X}$ with $|S| = \lfloor p|\mathcal{X}| \rfloor$ according to different difficulty-based policies, train models using GRPO on each subset, and compare their performance on held-out test sets. This setup mirrors practical scenarios where annotation budgets limit the number of examples that can be used for fine-tuning.

2.2 Difficulty estimation via multi-sample probing

To robustly estimate example difficulty, we employ multi-sample evaluation using the base model before any fine-tuning. For each prompt x in the training pool, we sample K independent completions from the base model π_{base} using temperature $\tau = 1.0$ and chain-of-thought prompting where applicable [24]. For each prompt x, we compute:

$$\hat{p}(x) = \frac{1}{K} \sum_{i=1}^{K} \mathbf{1}[\text{completion}_i \text{ is correct}] \tag{1}$$

where $\hat{p}(x)$ represents the empirical success rate.

Task-specific correctness criteria:

- **GSM8K:** Exact match of the final numerical answer after extracting from the generated solution [4]
- **Tracking Shuffled Objects:** Correct identification of all object positions after the described swaps [20]

We use $K = 10$ samples for Tracking Shuffle Objects, and $K = 5$ for GSM8K due to its larger size and associated computational costs. This multi-sample approach provides stable difficulty estimates while remaining computationally tractable.

2.3 Subset selection policies

Given difficulty estimates for all prompts, we implement four selection policies, each choosing exactly 10% of the training pool:

1. **HARD(-est):** Select prompts with the lowest success rates:
$$S_{\text{hard}} = \arg \min_{S:|S|=\lfloor p|\mathcal{X}| \rfloor} \sum_{x \in S} \hat{p}(x)$$
2. **EASY(-est):** Select prompts with the highest success rates:
$$S_{\text{easy}} = \arg \max_{S:|S|=\lfloor p|\mathcal{X}| \rfloor} \sum_{x \in S} \hat{p}(x)$$
3. **MIDDLE:** Select prompts nearest to the median difficulty, specifically those in the interquartile range around the median $\hat{p}(x)$ value
4. **RANDOM:** Uniform random sample without replacement from the full pool

Figure 4 illustrates the complete pipeline from difficulty estimation through subset selection to GRPO training.

Budget-Aware GRPO Pipeline
1. **Difficulty Estimation:** Sample K completions per prompt from base model → Compute success rate $\hat{p}(x)$
2. **Subset Selection:** Apply policy (Hard, Easy, Middle, Random) → Select 10% subset
3. **GRPO Training:** Fine-tune model on selected subset using group advantages
4. **Evaluation:** Test on held-out benchmark

Figure 1: Schematic overview of our experimental protocol

2.4 Training procedure

We employ Group Relative Policy Optimization (GRPO) following the formulation introduced in DeepSeekMath [18]. More details on the specifics of the GRPO algorithm and hyperparameters used can be found in the appendix.

Our experiments span two benchmark datasets that probe different aspects of model capabilities. GSM8K provides 7,473 grade-school math problems requiring multi-step reasoning and arithmetic [4]. The Tracking 7 Shuffled Objects task from BIG-Bench Hard contains 250 problems testing the ability to maintain state through sequences of object swaps [20], which we randomly split 50%/50% into train/test. We hold the number of prompts, rollouts per prompt, group size, KL schedule, max length, and total RL update steps constant across policies. This avoids claims being confounded by extra tokens/updates.

We evaluate three models: Qwen3-4B with 4 billion parameters representing a smaller but efficient architecture, Qwen3-14B with 14 billion parameters as a medium-scale model, and Phi-4 with 14 billion parameters, representing a different model family. All models are initialized from instruction-tuned checkpoints, ensuring they have basic instruction-following capabilities before GRPO fine-tuning begins.

3 Experiments and Results

We present our main experimental findings comparing difficulty-based subset selection strategies for GRPO fine-tuning. Our results consistently demonstrate that training on the hardest examples yields superior performance on reasoning tasks.

3.1 Hard examples drive reasoning improvements

Figure 2 and Table 3.1, summarize the absolute accuracy change (in percentage points) from the baseline after GRPO training, across selection policies, datasets, and models.

Dataset	Model	Easy	Middle	Hard	Random
GSM8K	Qwen3-4B	3.49	26.69	**34.19**	29.49
	Qwen3-14B	8.26	24.34	**39.42**	34.87
	Phi-4	14.94	28.51	**37.30**	36.16
Shuffled Objects	Qwen3-4B	0.80	-3.20	**13.60**	-1.60
	Qwen3-14B	3.20	10.12	**22.40**	12.52
	Phi-4	29.60	28.80	**32.80**	28.80

Table 1: Absolute performance change (%) after GRPO training with different subset selection policies compared to baseline. Bold indicates best performance, underline indicates second-best.

The results reveal striking patterns in how difficulty-based selection impacts GRPO training effectiveness. On GSM8K, the superiority of hard example selection is unambiguous across all models. The Hard policy achieves remarkable performance improvements of 34.19% for Qwen3-4B, 39.42% for Qwen3-14B, and 37.3% for Phi-4. These gains dwarf those achieved by easy selection, which manages only 3.49%, 8.26%, and 14.94% respectively. The magnitude of these differences (exceeding 30 percentage points) suggests that training on challenging examples fundamentally alters the learning dynamics under GRPO. We see training on either medium or random difficulties yield significant

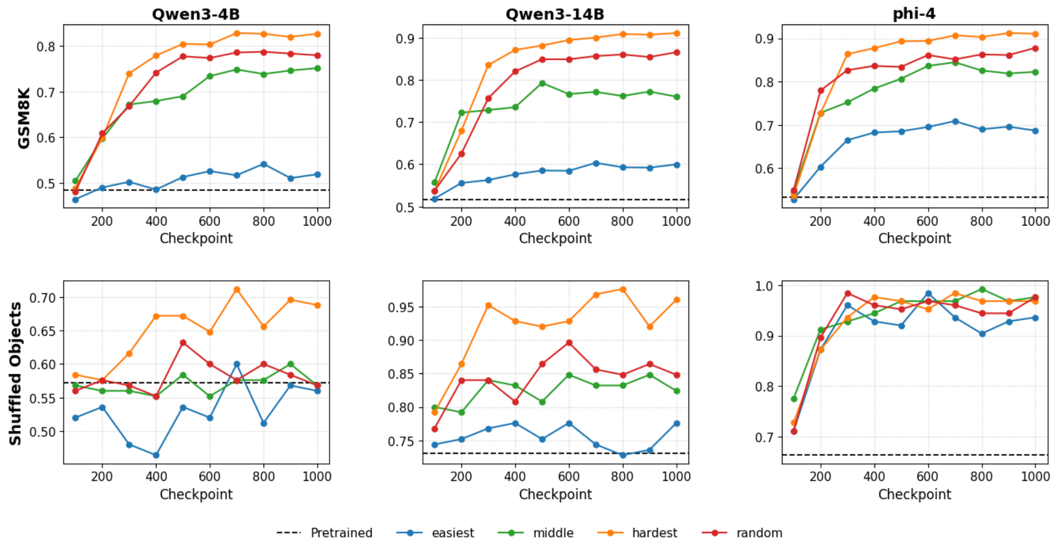

Figure 2: GRPO training dynamics reveal early and persistent advantages of hard example selection. Each subplot shows test accuracy over 1000 training steps for models trained on different difficulty-based subsets (10% of full data each). The hardest subset establishes superiority by step 300 and maintains this advantage.

gains compared to training on easy examples, but not near the gains of training on the hardest subset - affirming the hypothesis that the harder the training subset, the larger the gains we see[4].

The Tracking Shuffled Objects task amplifies these patterns. Hard selection yields positive gains of 13.60%, 22.40%, and 32.80% for Qwen3-4B, Qwen3-14B, and Phi-4 respectively. For Qwen3-4B, we see all other subsets produce either zero or negative gains. For Qwen3-14B, improvements show a stepwise pattern: training on the easy subset performs poorly, medium and random subsets provide moderate gains, and the hard subset delivers the largest gains. For Phi-4, training on the hardest subset gives the highest gains, but training on any subset still produces substantial improvements, effectively saturating the benchmark [20].

3.2 Performance dynamics during training

Figure 2 reveals how selection strategies shape learning trajectories from the earliest stages of training. The advantage of hard example selection manifests rapidly and decisively: by checkpoint 300, models trained on the hardest subset have already established clear superiority over all other selection strategies. This early divergence is particularly striking given that it represents less than a third of the total training steps, yet the performance ordering established at this point persists and often amplifies through the remainder of training. The consistency of this pattern across all three models suggests that hard examples provide fundamentally different learning signals that reshape the optimization landscape from the outset.

4 Analysis

In this section, we try to better understand why hard examples are most effective for GRPO training in three ways. First, we examine learning dynamics through the lens of new metrics - “learnable percentage". Second, we inspect performance on an out-of-distribution set to measure generalization capabilities when trained on different subsets. Third, we look at training just on examples where the base model is wrong vs where it is right.

4.1 Why hard examples help: maintaining learnable samples

A key insight into GRPO's preference for hard examples comes from tracking what fraction of training examples maintain non-zero outcome variance during training. Recall that GRPO learns by contrasting outputs within a group; when all outputs in a group produce identical rewards (aka standard deviation of reward is zero), the advantages become zero and no learning occurs.

Learnable percentage. Let T be the number of training updates. At update t, let $\mathbf{r}_t = (r_{t,1}, \ldots, r_{t,G}) \in \{0,1\}^G$ denote the vector of outcome rewards for the G rollouts in the GRPO group under π_t. We define the *learnable percentage* as

$$\text{Learnable \%} = 100 \cdot \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\{\, \mathsf{StdDev}(\mathbf{r}_t) > 0 \,\},$$

i.e., the fraction of training steps whose within-group reward standard deviation is nonzero (and therefore advantage is nonzero).

Table 2 shows the % learnable across strategies and datasets. We see that, unsurprisingly, the hardest strategies have the highest % learnable examples, meaning training on this data subset gives the most opportunity for the model to learn during training. Conversely, we see that the easy strategy has very few learnable examples. To test whether this effect depends on the model's baseline ability, we evaluate Llama-3.1-8B-Instruct [7], which has much lower initial performance on GSM8K. With a weaker baseline, we expect smaller performance gaps between subsets. Table 8 shows that while the hardest subset still achieves the largest gain, it is closer to the other subsets. Notably, all subsets have much higher % learnable values than the corresponding Qwen results (e.g., for the easiest subset, 21% vs. 4% for Qwen3-4B).

	Qwen3-4B		Qwen3-14B		Phi-4	
Strategy	shuffleobj	gsm8k	shuffleobj	gsm8k	shuffleobj	gsm8k
Easy	9.00	3.70	2.30	7.80	14.20	6.60
Medium	5.70	<u>24.50</u>	7.20	<u>18.90</u>	15.30	<u>23.30</u>
Hard	**14.40**	**34.10**	**13.50**	**29.50**	<u>19.10</u>	**40.20**
Random	<u>11.50</u>	19.00	<u>8.00</u>	18.40	**19.30**	21.50

Table 2: % Learnable across models, strategies and datasets

Strategy	Improvement over Base	% Learnable
Easy	36.54	20.90
Medium	39.42	42.00
Hard	**40.33**	**55.20**
Random	39.35	39.30

Table 3: Absolute accuracy improvement and % learnable for Llama-3.1-8B-Instruct on GSM8K dataset

Figure 3 plots the relationship between the % learnable and performance improvement, across all the models and strategies. We see a strong positive correlation, confirming that the underlying reason the harder subsets give better results is because they maintain more learnable examples throughout training.

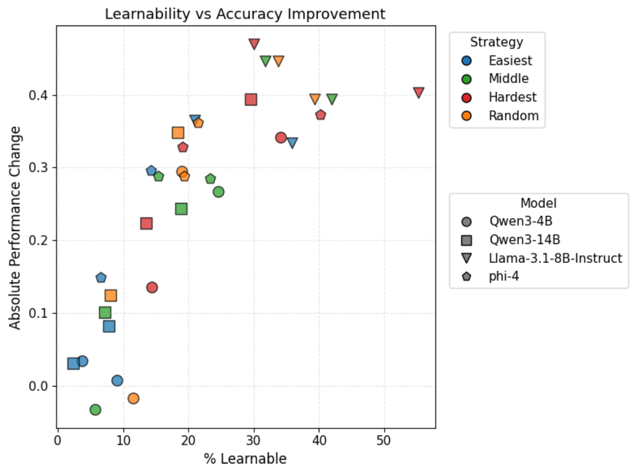

Figure 3: Scatter plot showing the relationship between the percentage of learnable training examples and the resulting absolute improvement in model performance, across strategies and models. Colors indicate the strategy and marker shapes indicate the model. We see a strong positive correlation (R^2 = 0.66), indicating that performance improves with more learnable examples.

Our analysis reveals why hard example selection proves dramatically more effective for GRPO training. Hard examples continue providing useful learning signal throughout training, while easy examples quickly become "solved" and stop contributing to improvement. This sustained learning opportunity explains the remarkable performance gaps we observe (often exceeding 30 percentage points between hard and easy selection strategies).

4.2 Out-Of-Distribution Performance

Having shown that harder training subsets improve test-set performance, we further evaluate on an out-of-distribution (OOD) test set. Specifically, we take the Qwen3-4B models trained on each subset and evaluate on AIME2025-I, a dataset of 15 questions significantly more difficulty than GSM8K [2, 14]. Table 4 presents the results.

Strategy	Pass@8
Base	33.3%
Easy	33.3%
Medium	26.7%
Hard	**40.0%**
Random	33.3%

Table 4: Pass@8 on AIME-2025 of Qwen3-4B trained on different subsets

We see that the training on the hardest subset is the only one to outperform the base Qwen3-4B model. Surprisingly, training on the medium subset actually does worse than training on easy or random subsets.

We see that training on the hardest subset is the only condition that meaningfully outperforms the base Qwen3-4B model, achieving a relative improvement of +20%. This reinforces the trend observed on the in-distribution test set: exposure to more challenging problems during training generalizes better to harder OOD problems. Interestingly, overall performance on the easy and random subsets remains identical to the base model, with closer inspection showing that these subsets improved performance on certain problems but regressed performance in other areas. In contrast, training on the medium difficulty subset yields a notable drop in performance, with no other gains.

Overall, these results show that the benefits of hard-example training extend beyond the original distribution, improving robustness to unseen, more challenging problems.

4.3 Base Wrong vs Base Right: Decomposing GRPO Value

Having established that hard examples (those with low pass@k success rates) drive the most value in GRPO training, we now validate this principle from a complementary perspective. We investigate whether the superiority of hard examples stems from a more fundamental property: that they represent problems the base model cannot reliably solve.

We categorize training examples into four types based on base model performance (here base means the model pre GRPO training):

1. **Base Wrong**: Examples the base model gets wrong (accuracy <25%)
2. **Base Right**: Examples the base model gets right (accuracy ≥ 25%)
3. **Base Right, Same Size As Base Wrong**: A subset of the full Base Right Set that is the same size as Base Wrong, to account for model size
4. **All**: All the available examples in the train set

We hypothesize that training just on the examples where the base model is wrong will significantly match the base right subsets, as it provides the most opportunity for genuine capability improvement and higher learnable percentage.

4.3.1 Results

Table 5 presents the absolute performance after training on each subset:

	ShuffleObj			GSM8K		
Training Set	Phi-4	Qwen3-4B	Qwen3-8B	Phi-4	Qwen3-4B	Qwen3-8B
Base Wrong	**0.98**	**0.73**	0.81	**0.92**	**0.84**	**0.86**
Base Right	**0.98**	0.64	0.83	0.80	0.70	0.72
Base Right (Same Size)	**0.98**	0.67	0.75	0.78	0.68	0.71
All	0.97	0.66	**0.84**	0.85	0.77	0.85

Table 5: Absolute performance on test sets after training on different subsets based on base model performance. Bold indicates best or tied-best performance.

Table 6 shows the relative improvements of Base Wrong training over both Base Right subsets. Table 7 shows the sizes of each subset - Base Wrong is, on average, less than half the size of Base Right.

	ShuffleObj			GSM8K			
Metric	Phi-4	Qwen3-4B	Qwen3-8B	Phi-4	Qwen3-4B	Qwen3-8B	Average
Improvement Over Base Right (Same Size)							
Absolute	0.0%	6.0%	5.8%	14.0%	16.0%	15.0%	10.2%
Relative	0.0%	9.0%	7.7%	17.9%	23.5%	21.1%	14.3%
Improvement Over Base Right (Full)							
Absolute	0.0%	9.0%	-2.0%	12.0%	14.0%	14.0%	7.8%
Relative	0.0%	14.1%	-2.4%	15.0%	20.0%	19.4%	11.0%

Table 6: Improvements from training on Base Wrong examples compared to Base Right examples. Relative improvement calculated as (Base Wrong - Base Right) / Base Right.

4.3.2 Analysis

The results strongly support our hypothesis that GRPO learning value concentrates in examples the base model gets wrong:

Base Wrong consistently outperforms Base Right: Across models and tasks, training exclusively on Base Wrong examples achieves equal or superior performance to training on Base Right, even when Base Right has significantly more examples. The average relative improvement of 14.3% over size-matched Base Right subsets demonstrates that these examples provide fundamentally different learning signals. Even when we train on the full Base Right set, we see that Base Wrong outperforms it by 11% on average.

Base Wrong matches / outperforms All: Training on all Base Wrong samples actually slightly outperforms training on All samples on average (85% vs 82%), further supporting that most, if not all, meaningful training signal is concentrated in the Base Wrong subset.

Task-specific patterns emerge: On GSM8K, the advantage is particularly pronounced, with Base Wrong training achieving 15-23.5% relative improvements. This aligns with our earlier findings that mathematical reasoning benefits most from challenging examples. ShuffleObj shows more modest gains, potentially due to the discrete nature of object tracking limiting the diversity of solution paths.

Model-specific sensitivity: Phi-4 shows less differentiation between strategies on ShuffleObj (all achieving 0.98), suggesting potential ceiling effects. However, on GSM8K, even Phi-4 shows substantial 17.9% relative improvement, confirming the generality of the phenomenon.

Practical implications: These findings provide a practical recipe for GRPO dataset curation: practitioners can identify valuable training examples by simply evaluating base model performance, without needing to know which examples will ultimately improve. This avoids the circular dependency inherent in selecting "learnable" examples and provides a straightforward criterion for data collection.

	ShuffleObj			GSM8K		
Training Set	Phi-4	Qwen3-4B	Qwen3-8B	Phi-4	Qwen3-4B	Qwen3-8B
Base Wrong	17	37	13	343	409	371
Base Right	108	88	112	657	591	629
All	125	125	125	1000	1000	1000

Table 7: Subset sizes for each dataset, model, and training strategy. On average, Base Wrong is only 42% the size of Base Right, and yet consistently significantly outperforms this subset. Note that GSM8K is clipped to 1000 since we only run for 1000 steps with 1 prompt per update.

The concentration of learning value in Base Wrong examples validates our central thesis from a complementary angle. Our two experimental approaches converge on the same insight: examples with low pass@k rates (our "hardest" selection) are predominantly those where the base model fails ("base wrong"), while examples with high pass@k rates (our "easy" selection) are those the base model already masters ("base right"). This convergence demonstrates that whether we measure difficulty through multi-sample success rates or binary base model performance, we identify the same high-value training examples: those at the frontier of model capabilities where learning can actually occur.

5 Conclusion

This paper investigated a critical question for resource-constrained language model fine-tuning: under a fixed budget for example selection, which difficulty-based strategy maximizes the effectiveness of GRPO? Through comprehensive experiments across multiple models and tasks, we established that training on the hardest examples consistently yields superior performance on reasoning benchmarks.

We validated this "hard examples are all you need" principle from two complementary perspectives. First, our difficulty-based selection experiments showed that examples with the lowest pass@k success rates drive performance gains up to 47%, while easy examples yield minimal improvements. Second, our analysis of training data categorized by base model performance revealed that nearly all GRPO learning value concentrates in examples the base model initially fails to solve. Training exclusively on examples where the base model achieves low success rates ($\hat{p}(x) < 0.25$) yields up to 23.5% relative improvements compared to training on examples the base model already solves correctly ($\hat{p}(x) \geq 0.25$).

This convergence of evidence from both perspectives confirms a fundamental insight: GRPO learning requires training at the frontier of model capabilities. Examples the base model already masters reliably (high pass@k or those it solves correctly) are effectively the easiest and contribute minimal learning value, while examples where the model struggles (low pass@k or those it fails to solve) provide the variance in outcomes necessary for GRPO's contrastive learning mechanism. This unified understanding provides practitioners with multiple practical approaches for identifying high-value training data while avoiding circular dependencies in dataset curation.

References

[1] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024.

[2] Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions, February 2025.

[3] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4299–4307, 2017.

[4] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021. Introduces GSM8K.

[5] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025. PRIME: online process rewards from outcome labels.

[6] Michael Han Daniel Han and Unsloth team. Unsloth, 2023.

[7] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Akhil Kadian, Ahmad Al-Dahle, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[8] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *ICML*, 2019.

[9] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *ICML*, 2019.

[10] Alex Havrilla, Edward Hughes, Mikayel Samvelyan, and Jacob Abernethy. Sparq: Synthetic problem generation for reasoning via quality-diversity algorithms, 2025.

[11] Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *ICML*, 2021.

[12] KrishnaTeja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh K. Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *AAAI*, 2021.

[13] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025. Introduces Dr. GRPO; analyzes GRPO length bias.

[14] OpenCompass. Aime2025 dataset. Hugging Face Hub, 2025. `https://huggingface.co/datasets/opencompass/AIME2025`, accessed 2025-08-14.

[15] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[16] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *NeurIPS*, 2021.

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

[18] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024.

[19] Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *arXiv preprint arXiv:2506.05316*, 2025.

[20] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of ACL*, 2023. BIG-Bench Hard; includes Tracking Shuffled Objects.

[21] Xiaoyu Tian, Sitong Zhao, Haotian Wang, Shuaiting Chen, Yiping Peng, Yunjie Ji, Han Zhao, and Xiangang Li. Deepdistill: Enhancing llm reasoning capabilities via large-scale difficulty-graded data training, 2025.

[22] Mariya Toneva, Alessandro Sordoni, Rémi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *ICLR*, 2019.

[23] Brandon Trabucco, Qasim Wani, Benjamin Pikus, and Vasu Sharma. Understanding trade offs when conditioning synthetic data, 2025.

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

[25] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.

A Related Work

Our work intersects several research areas: reinforcement learning from human feedback, example selection and curriculum learning, reasoning benchmarks, and knowledge evaluation tasks. We review each area and position our contributions within this broader context.

A.1 Reinforcement Learning from Human Feedback and GRPO

Reinforcement learning from human feedback (RLHF) has emerged as a dominant paradigm for aligning language models with human preferences [3, 15]. The standard RLHF pipeline involves training a reward model from human preferences, then optimizing the language model policy using algorithms like Proximal Policy Optimization (PPO) [17]. However, PPO-based RLHF suffers from high computational costs, training instability, and significant variance in gradient estimates.

Group Relative Policy Optimization (GRPO), introduced in DeepSeekMath [18], addresses these challenges by reformulating the optimization problem. Instead of training a separate reward model, GRPO directly optimizes using group-relative advantages computed from multiple samples per prompt. For each training prompt, GRPO samples a group of G outputs, computes their rewards (e.g., correctness on math problems), and uses the deviation from the group mean as the advantage signal. This approach reduces variance through the group baseline while eliminating the need for reward model training.

Our work builds directly on GRPO's formulation but introduces a critical new dimension: budget-aware example selection. While previous GRPO applications assume uniform sampling from the training distribution, we show that strategic selection based on example difficulty can substantially improve learning efficiency, particularly relevant when data acquisition costs constrain the training set size.

A.2 Example Difficulty and Data Selection

The importance of example selection in machine learning has long been recognized, with multiple research threads addressing different aspects of this problem.

Curriculum learning suggests that training on examples in order of increasing difficulty can improve convergence and final performance [9]. However, curriculum learning typically assumes access to the full dataset and focuses on presentation order rather than subset selection. Our work differs by operating under a strict budget constraint where only a fraction of examples can be used.

Data valuation and influence methods aim to identify the most valuable training examples. Data Shapley [8] uses cooperative game theory to assign value to each training example based on its marginal contribution. However, these methods are computationally expensive and designed for supervised learning rather than RLHF settings.

Coreset selection techniques identify representative subsets that approximate the full dataset's gradient [11, 12]. GradMatch selects examples whose gradients best match the full dataset gradient, while GLISTER focuses on generalization-based selection. These methods provide theoretical guarantees but require access to labels and gradient computation that may not be available in our budget-constrained setting.

Example forgetting and memorization studies have shown that neural networks exhibit consistent patterns in which examples are learned or forgotten during training [22].

Synthetic data is often an area of investigation for training, especially since it is often easier to control for difficulty. Our work doesn't explicitly investigate using synthetic data and instead selects from a real, static larger dataset [23, 21, 10].

"Unforgettable" examples are learned early and retained, while others are repeatedly learned and forgotten. The Data Diet work [16] leverages Error L2-Norm (EL2N) scores computed early in training to identify important examples, showing that networks can be trained on small subsets without performance loss.

Our approach differs from these prior works in several ways: (1) We operate in the RLHF/GRPO setting where rewards are binary and computed at inference time; (2) We use the base model's multi-sample success rate as a difficulty proxy, requiring no training to compute; (3) We explicitly compare different difficulty-based selection strategies under identical budgets.

A.3 Reasoning Benchmarks and Evaluation

Evaluating reasoning capabilities in language models requires carefully designed benchmarks that test different aspects of logical and mathematical thinking.

GSM8K [4] has become the standard benchmark for grade-school mathematical reasoning, containing 8,792 problems requiring multi-step arithmetic and logical reasoning. The dataset's problems are linguistically diverse but mathematically elementary, making it ideal for studying reasoning without confounding advanced mathematical knowledge.

BIG-Bench Hard (BBH) [20] curates 23 challenging tasks from the broader BIG-Bench suite where language models initially showed poor performance. The Tracking Shuffled Objects task, which we use in our experiments, requires models to track entity positions through a series of swapping operations: a pure reasoning task with minimal knowledge requirements.

Chain-of-thought prompting and self-consistency [24] have proven effective for improving reasoning performance. Self-consistency samples multiple reasoning paths and aggregates answers, often yielding substantial accuracy improvements. We leverage this insight in our difficulty estimation, using multi-sample success rates to robustly measure example hardness.

A.4 GRPO

The GRPO algorithm operates by sampling groups of outputs for each training prompt and using relative performance within each group to compute advantages. Specifically, for each prompt q in the selected subset, the algorithm samples G outputs $\{o_1, ..., o_G\}$ from the current policy π_θ, computes binary rewards $r_i \in \{0, 1\}$ based on correctness, and calculates a group baseline $\bar{r} = \frac{1}{G}\sum_{i=1}^{G} r_i$. The advantage for each output is then computed as $A_i = r_i - \bar{r}$, which naturally centers the learning signal around the group's average performance. These advantages drive policy updates while KL regularization to a reference policy π_{ref} prevents catastrophic distribution shifts.

The training objective maximizes the expected advantage-weighted log-likelihood while minimizing KL divergence from the reference policy:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{q\sim S}\mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\left[A(o,q)\cdot\log\pi_\theta(o|q) - \beta\cdot\text{KL}(\pi_\theta||\pi_{\text{ref}})\right]$$

where β controls the strength of KL regularization.

A.5 Closest GRPO Work

Recent work has explored various aspects of GRPO optimization that relate closely to our budget-aware selection study:

Difficulty-targeted selection and replay. Sun et al. propose a data-efficiency recipe for GRPO-style RL fine-tuning that (i) targets adaptive difficulty online, prioritizing questions of moderate difficulty expected to yield informative gradients, and (ii) introduces rollout replay to reuse recent samples and lower per-step cost. Across six LLM-dataset pairs they report 25-65% faster time-to-target performance than vanilla GRPO, suggesting that what you train on (and when) can matter as much as how much you train. Our paper shares the focus on difficulty-aware allocation under constraints, but differs in scope: we study budgeted, offline subset selection (hard vs. medium vs. easy vs. random) with a fixed labeling budget and analyze why "hardest" can dominate on reasoning tasks, whereas Sun et al. optimize online time-to-quality with an adaptive, typically moderate-difficulty target and replay [19].

Process rewards compatible with GRPO. Cui et al. (PRIME) tackle the sparsity and credit-assignment limits of outcome-only rewards by learning implicit process reward models online from outcome labels alone, then combining dense token-level signals with outcome rewards during RL. PRIME is compatible with standard advantage formulations (including GRPO) and removes a separate reward-modeling stage; empirically, they show sizable gains on competitive math and coding (e.g., 15% average over SFT with a 7B base). Our study focuses on which examples to buy under an outcome-reward GRPO setup; PRIME is complementary, and our framework could be extended to ask whether difficulty-aware selection interacts with dense process rewards in similar ways (e.g., whether "hardest" remains optimal when per-step feedback reduces variance) [5].

GRPO biases and token efficiency. Liu et al. present a critical analysis of R1-Zero-like training and identify an optimization bias in GRPO: the objective can inflate response length, particularly for incorrect outputs, harming token efficiency. They introduce Dr. GRPO, an unbiased variant that preserves reasoning quality while improving efficiency, and demonstrate strong AIME-24 results with a 7B base model. For our budget-aware study, their findings motivate length-controlled evaluations and guardrails when comparing difficulty-conditioned subsets (e.g., ensuring that harder-subset gains are not confounded by pathological length growth and considering Dr. GRPO as a robustness check) [13].

Positioning. Together, these works closest to GRPO underscore three levers we explicitly separate and test: (a) example selection by difficulty (Sun et al.), (b) reward shaping via dense process feedback (Cui et al.), and (c) objective design to avoid length bias (Liu et al.). Our contribution is orthogonal and complementary: a budget-aware, offline selection protocol and theory-informed analysis showing why, under outcome-level GRPO, training on the hardest decile can sustain a higher fraction of "learnable" examples (non-degenerate group variance) and thus larger gains on reasoning tasks, while we adopt length controls in line with Dr. GRPO insights [19, 5, 13].

A.6 Positioning Our Contributions

Our work makes several distinct contributions relative to this prior literature:

1. We are the first to systematically study budget-aware example selection specifically for GRPO fine-tuning, addressing a practical constraint faced by practitioners.
2. We provide a unified evaluation framework comparing difficulty-based selection strategies across multiple task types and model scales.
3. We offer theoretical insight connecting GRPO's group-advantage mechanism to the effectiveness of hard-example selection, grounded in variance reduction principles from reinforcement learning theory.
4. We demonstrate that selection strategy effectiveness depends critically on task type, with reasoning tasks benefiting from hard examples while knowledge tasks show no such preference.

These contributions provide both theoretical understanding and practical guidance for efficient RLHF fine-tuning under resource constraints.

B Extended Experimental Details

B.1 GRPO Algorithm Details

Group Relative Policy Optimization (GRPO) is a variant of PPO that replaces the learned value function with group-normalized advantages computed from within-batch reward contrasts. For each group of G rollouts from the same prompt, GRPO computes advantages as:

$$A_i = \frac{r_i - \bar{r}}{\sigma_r + \epsilon}$$

where r_i is the reward for rollout i, $\bar{r}$ is the mean reward within the group, σ_r is the standard deviation, and ϵ is a small constant for numerical stability. This formulation means that when all rollouts in a group receive identical rewards (i.e., $\sigma_r = 0$), all advantages become zero and no learning occurs for that group.

B.2 Detailed Training Pipeline

B.3 Multi-Sample Difficulty Estimation

Our difficulty estimation procedure uses the following specific parameters:

Budget-Aware GRPO Pipeline
1. **Difficulty Estimation:** Sample K completions per prompt from base model → Compute success rate $\hat{p}(x)$
2. **Subset Selection:** Apply policy (Hard, Easy, Middle, Random) → Select 10% subset
3. **GRPO Training:** Fine-tune model on selected subset using group advantages
4. **Evaluation:** Test on held-out benchmark

Figure 4: Schematic overview of our experimental protocol

- Temperature $\tau = 1.0$ for all sampling
- For GSM8K: 5-shot chain-of-thought prompting with examples from the training set
- For Shuffled Objects: 3-shot prompting with step-by-step tracking demonstrations
- Answer extraction using task-specific regex patterns to ensure consistent evaluation

The choice of $K = 5$ for GSM8K and $K = 10$ for Shuffled Objects balances computational cost with estimation stability. Preliminary experiments showed that these values provide sufficiently stable difficulty estimates (standard error < 0.1 for most examples).

B.4 Implementation Details

All experiments were conducted using the following infrastructure:

- 8x NVIDIA A100 80GB GPUs for model training
- PyTorch 2.0 with mixed precision training
- Gradient accumulation to achieve effective batch sizes
- Checkpoint saving every 100 training steps for trajectory analysis

C Extended Analysis

C.1 Performance Dynamics During Training

Figure 2 in the main paper reveals how selection strategies shape learning trajectories from the earliest stages of training. The advantage of hard example selection manifests rapidly and decisively: by checkpoint 300, models trained on the hardest subset have already established clear superiority over all other selection strategies. This early divergence is particularly striking given that it represents less than a third of the total training steps, yet the performance ordering established at this point persists and often amplifies through the remainder of training. The consistency of this pattern across all three models suggests that hard examples provide fundamentally different learning signals that reshape the optimization landscape from the outset.

C.2 Analysis with Weaker Baseline Models

To test whether the learnable percentage effect depends on the model's baseline ability, we evaluate Llama-3.1-8B-Instruct [7], which has much lower initial performance on GSM8K. With a weaker baseline, we expect smaller performance gaps between subsets. Table 8 shows that while the hardest subset still achieves the largest gain, it is closer to the other subsets. Notably, all subsets have much higher % learnable values than the corresponding Qwen results (e.g., for the easiest subset, 21% vs. 4% for Qwen3-4B).

C.3 Extended Out-of-Distribution Analysis

Our OOD evaluation on AIME2025-I reveals interesting patterns beyond the main finding. Training on the easiest and random subsets remains identical to the base model performance, but closer inspection shows that these subsets improved performance on certain problems while regressing performance in other areas, resulting in no net gain. In contrast, training on the middle-difficulty subset yields a notable drop in performance with no compensating gains. This suggests that middle-difficulty examples may create a problematic "uncanny valley" where the model learns patterns that

Strategy	Improvement over Base	% Learnable
easiest	36.54	20.90
middle	39.42	42.00
hardest	**40.33**	**55.20**
random	39.35	39.30

Table 8: Absolute accuracy improvement and % learnable for Llama-3.1-8B-Instruct on GSM8K dataset

are neither simple enough to generalize broadly nor complex enough to handle truly challenging problems.

The 20% relative improvement achieved by hard-example training on AIME problems is particularly noteworthy given the substantial difficulty gap between GSM8K (grade-school level) and AIME (competition mathematics). This suggests that hard examples don't just teach specific problem-solving patterns but rather develop more fundamental reasoning capabilities that transfer across difficulty levels.

C.4 Task-Specific Observations

GSM8K: The consistent superiority of hard selection across all models (34-40% improvement) suggests that mathematical reasoning particularly benefits from training on problems at the edge of the model's capabilities. The magnitude of these improvements—often exceeding 30 percentage points compared to easy selection—indicates that this is not a marginal optimization but a fundamental factor in training success.

Shuffled Objects: The even larger relative gaps for some models indicate that sequential tracking tasks may be especially sensitive to training difficulty. For Qwen3-4B, we observe that all non-hard subsets produce either zero or negative gains, suggesting that only sufficiently challenging examples can teach the complex state-tracking required for this task. Phi-4's strong performance across all subsets (but still best with hard) suggests some architectures may have better inductive biases for this task type.

D Hyperparameters

We set the group size $G = 8$ for our main experiments based on preliminary studies showing this balances variance reduction with computational efficiency. The KL coefficient $\beta = 0.1$ provides sufficient regularization without overly constraining learning. Training proceeds for 1000 steps with a learning rate of 3×10^{-5} using cosine decay, processing 1 prompt per gradient update due to memory constraints. We use 4-bit quantized models from Unsloth [6]. We evaluate model performance every 100 steps to track learning dynamics, using the initial SFT checkpoint as our reference policy throughout training.