

# SPaCe: Unlocking Sample-Efficient Large Language Models Training With Self-Pace Curriculum Learning

Dai Do<sup>1</sup>, Manh Nguyen<sup>1</sup>, Svetha Venkatesh<sup>1</sup>, Hung Le<sup>1</sup>  
<sup>1</sup>Deakin Applied AI Initiative, Deakin University, Australia  
v.do@deakin.edu.au

## Abstract

Large language models (LLMs) have shown strong reasoning capabilities when fine-tuned with reinforcement learning (RL). However, such methods require extensive data and compute, making them impractical under many realistic training budgets. Many existing pipelines sample training examples uniformly across steps or epochs, ignoring differences in difficulty, redundancy, and learning value, which slows learning and wastes computation. We propose **SPaCe**, a self-paced learning framework that enables efficient learning based on the capability of the model being trained through optimizing which data to use and when. First, we apply *cluster-based data reduction* to partition training data by semantics and difficulty, extracting a compact yet diverse subset that reduces redundancy. Then, a *multi-armed bandit* treats data clusters as arms, allocating training samples based on the model’s solve rates and learning progress. Experiments across multiple reasoning benchmarks show that SPaCe achieves comparable or better accuracy than state-of-the-art baselines while using up to 100× fewer samples. Ablation studies and analyses further highlight the importance of both data clustering and adaptive selection. Our results demonstrate that carefully curated, performance-driven training curricula can unlock strong reasoning abilities in LLMs with minimal resources.

## 1 Introduction

Large Language Models (LLMs) have achieved remarkable progress in tasks requiring reasoning, problem-solving, and generalization, driven largely by scaling trends in model size, data, and compute (Google, 2024; OpenAI, 2024). As the cost and complexity of pretraining continue to rise, research attention has increasingly shifted toward post-training techniques, which aim to improve LLM capabilities more efficiently. Among these, Reinforcement Fine-Tuning (RFT) has emerged

as a promising method that aligns model behavior with outcome-based reward signals, often relying on lightweight supervision without elaborate reward engineering or inference-time computation (Kumar et al., 2025; DeepSeek-AI, 2025; Lightman et al., 2023).

Standard RFT uniformly samples batches from the full dataset (DeepSeek-AI, 2025). While simple, this approach ignores each example’s difficulty, informativeness, and uncertainty, wasting limited reward feedback on trivial or noisy instances and slowing convergence (Ouyang et al., 2022; Dong et al., 2023). This raises two key underexplored dimensions: how to select which examples to train on, and how to present them to LLMs over time.

Data reduction methods prioritize informativeness by estimating example difficulty or uncertainty. For example, variance-based filtering based on multiple forward passes through a reference model (Wang et al., 2025b). Although effective at denoising, these approaches incur significant computational overhead, impractical for resource-constrained models. They are also sensitive to the selected training examples and the uncertainty estimator, which hinders generalization to new LLMs.

In parallel, curriculum design plays a central role in guiding the learning trajectory (Bengio et al., 2009). As the model improves during fine-tuning, the useful difficulty level shifts dynamically, yet static curricula or random orders often fail to reflect this progression. Recent attempts at adaptivity filter examples with heuristic thresholds (Shi et al., 2025), but such mechanisms are fragile, especially for small or weak models that rely on imperfect difficulty metrics, prematurely excluding challenging, informative examples, and stalling progress.

We introduce **SPaCe**, a self-paced RFT framework that improves training efficiency by selecting informative examples and adapting the training schedule online. We cast RFT as a Multi-Armed Bandit (MAB) problem (Sutton et al., 1998),

where each arm is a cluster of examples with similar semantics and per-example attribute, enabling data selection beyond fixed heuristics. SPaCe first performs a one-time clustering by jointly clustering latent representations and per-example attributes, then reduces redundancy by retaining a fixed number of diverse representatives per cluster, selected by iteratively maximizing embedding distance (Wang et al., 2025b). During training, SPaCe pulls an arm, sampling data from its cluster. The solve rate updates the bandit, while its negative (difficulty) guides Thompson Sampling, with additional downweighting for clusters whose hardness plateaus to encourage exploration (Thompson, 1933; Russo et al., 2020). Overall, SPaCe prioritizes examples that are challenging yet learnable, avoiding repeated training on already-solved problems and remaining effective under tight budgets in low-resource RFT (Le et al., 2025).

To evaluate our approach, we conduct extensive experiments on mathematical and logical reasoning tasks using various LLMs of different sizes. Results show that SPaCe significantly improves reasoning accuracy and robustness compared to both reinforcement learning and curriculum learning baselines. Notably, SPaCe also outperforms methods that rely on exhaustive search to select a single or a few training examples. Our analysis reveals how poorly designed curricula can get stuck in easy example regions, failing to leverage the diversity of the dataset.

In summary, our contributions are threefold: (1) We propose **SPaCe**, a novel two-stage framework that reduces the number of training examples and optimizes the RFT progress using MAB. (2) Our method is lightweight, significantly reducing the number of training examples while adding minimal computational overhead, well-suited for low-resource settings. (3) Our extensive experiments demonstrate that SPaCe consistently outperforms existing curriculum and data reduction strategies.

## 2 Related Work

### 2.1 Efficient Methods to Enhance Language Models.

The role of data in post-training LLMs remains an open research question. One line of work studies reinforcement fine-tuning (RFT). Several recent methods focus on curating high-quality mathematical datasets (Luo et al., 2025; Yu et al., 2025), but they do not explicitly investigate which data is most

effective for fine-tuning. More recently, alternative approaches have explored heuristic-based scoring methods, such as Learning Impact Measurement (Li et al., 2025) and variance-based data selection (Wang et al., 2025b). These methods improve data efficiency by enabling training on only a small subset of the data while still achieving strong reasoning performance. However, they typically require substantial precomputation, which limits their practicality in real-world settings, and they have not been thoroughly evaluated on small models with limited reasoning ability. Another line of work explores efficient post-training methods for LLMs, including alignment learning (Ji et al., 2024; Do et al., 2025a) and steering methods (Turner et al., 2025; Do et al., 2025b). In this paper, however, we focus specifically on methods related to SFT.

### 2.2 Curriculum Learning for LLMs.

Humans and animals learn more effectively when examples are presented in a meaningful order that gradually increases in complexity. Curriculum learning (Bengio et al., 2009) and performance-guided training progression (Le et al., 2022) have been applied to supervised and RL training. For LLMs, recent studies have explored how to organize training data to reduce computational cost and improve sample efficiency, though this area remains underdeveloped. Existing approaches include hand-crafted difficulty tiers (Wen et al., 2025; Luo et al., 2025; Song et al., 2025), which often require task-specific insights and manual tuning. More adaptive methods, such as AdaRFT (Shi et al., 2025), learn a training curriculum by dynamically adjusting a difficulty threshold to select examples. While promising, these methods still face limitations: repeated training on easy examples can lead to overfitting or poor generalization; difficulty heuristics may not transfer across tasks; and fixed sampling strategies may fail to adapt to evolving model capabilities. There remains a need for more principled curriculum strategies tailored to the scale and dynamics of LLM training.

## 3 Self-paced Reinforcement Fine-tuning

We aim to improve a policy  $\pi_\theta$  by adaptively presenting training samples while minimizing the required data. Focusing on tasks that are too easy or too hard is inefficient, offering little challenge or feedback. Instead, data assignment should adapt to the model’s evolving capabilities, presenting

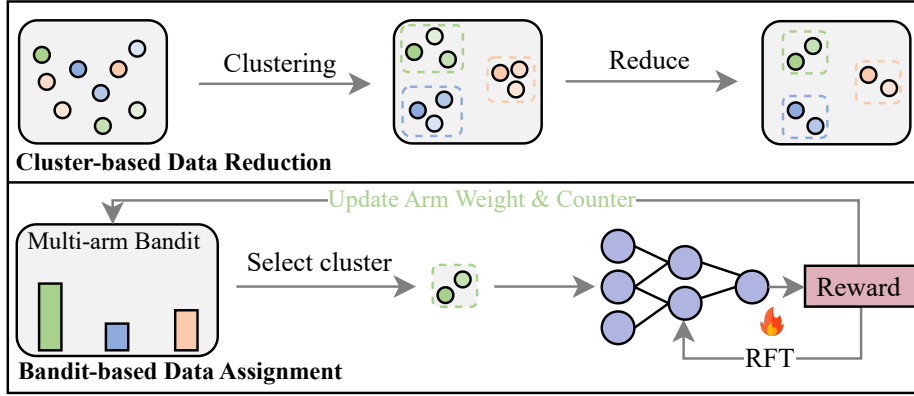


Figure 1: SPaCe Architecture. **Top:** In latent space, each initial training data’s feature vector is formed by concatenating its latent embedding with its difficulty score. The data is clustered into  $K$  groups, and from each cluster we select representative samples to ensure both coverage and diversity. **Bottom:** Each cluster is treated as an arm in a multi-armed bandit. At each step, Thompson Sampling is used to select a cluster, its representative examples are fed to the LLM to obtain rewards for RFT, and those rewards are used to update the bandit statistics.

examples that it is ready to learn at each stage. Threshold-based curricula require manual tuning and can be suboptimal, especially at the early training stage (Shi et al., 2025). To address this, we introduce **SPaCe**, a two-phase approach for self-paced optimization: (1) cluster-based data reduction and (2) bandit-based data assignment. SPaCe integrates with common RFT algorithms; we use GRPO (DeepSeek-AI, 2025) by default.

### 3.1 Cluster-based Data Reduction

#### 3.1.1 Data Clustering

RFT of large language models often presupposes access to abundant, high-quality supervision, which is costly or impractical in low-resource settings. We introduce a clustering-based data reduction procedure that lowers data requirements while preserving or improving training efficacy. The approach leverages two signals per training instance: (i) a latent representation and (ii) a scalar, per-example attribute available from the dataset. Grouping examples that are proximate in latent space and exhibit comparable attribute values yields a coherent partitioning well-suited to curriculum learning.

*Latent representation.* For each example, we obtain a latent representation using a pre-trained embedding model from SentenceTransformers (Reimers and Gurevych, 2019). We decouple these clustering embeddings from the RL policy to avoid representation drift during RL training. To mitigate the degradation of distance metrics in high dimensions, we apply Principal Component Analysis (PCA) for dimensionality reduction (Jolliffe,

2002).

*Per-example attribute.* Our framework supports the inclusion of an optional scalar attribute at the per-example level, alongside semantic embeddings. Let  $d_i$  denote this attribute, which can flexibly encode any task-specific signal. For clustering, we concatenate  $d_i$  with the latent embedding to form a joint representation. This design enables clustering to account not only for semantic similarity but also for structural or pedagogical cues that are important for curriculum construction.

*Clustering.* Let  $x_i$  denote the  $i^{\text{th}}$  training example and  $s_i$  its PCA-reduced latent vector. Before clustering, we standardize the coordinates of  $s_i$  and  $d_i$  to zero mean and unit variance, preventing either modality from dominating the distance metric. We then form the combined representation as follows:

$$e_i = \hat{s}_i \oplus \hat{d}_i, \quad (1)$$

where  $\hat{s}_i$  and  $\hat{d}_i$  are the standardized latent vector and scalar feature, respectively, and  $\oplus$  denotes concatenation. Finally, we apply  $k$ -means clustering to  $\{e_i\}$  to partition the dataset into  $K$  clusters (Lloyd, 1982), thereby ensuring coverage across the joint latent–attribute space. We perform clustering only once before training to define a stable set of bandit arms. Fixing the arms ensures that each arm has a consistent meaning over time, so the scheduler can accumulate and compare per-arm statistics (e.g., solve rates and no-improvement counts) throughout training; re-clustering would change the arm identities and invalidate these statistics unless additional bookkeeping is introduced.

### 3.1.2 Data Reduction

Prior work on scaling RFT shows that carefully selected subsets of training data can outperform fine-tuning on the full corpus (Li et al., 2025; Wang et al., 2025b). Motivated by this, the training set is partitioned into  $K$  clusters and a fixed quota  $l$  of examples is subsampled from each cluster to ensure balanced coverage and diversity. Let  $C_k$  denote the set of examples in cluster  $k$  with centroid  $\mu_k$  in the embedding space. For each example  $e_i \in C_k$ , its distance to the centroid is computed as  $\delta_i = \|e_i - \mu_k\|_2$ . Representative subsets are then selected via greedy farthest point sampling within each cluster: starting from the centroid, examples are iteratively chosen to maximize their distance from the already selected set. This preserves both centrality and geometric diversity within each cluster.

### 3.2 Bandit-based Data Assignment

Our key principle is to prioritize examples that are *currently challenging and still learning-relevant*, since model capability evolves during training. At each step  $t$ , a bandit *scheduler* selects a cluster  $c_t \in \{1, \dots, K\}$  (arms correspond to clusters) using Thompson Sampling. For each cluster  $k$ , we track cumulative reward  $R_k^{(t)}$  and pulls  $n_k^{(t)}$ . We define the online solve rate as  $\bar{r}_k^{(t)} = R_k^{(t)} / (n_k^{(t)} + \epsilon)$ , with  $\epsilon > 0$  for numerical stability. This online rate is recomputed every step and is different from the offline difficulty  $d_i$  used only for clustering. Difficulty is defined as  $h_k^{(t)} := -\bar{r}_k^{(t)}$ . Then, we use Thompson Sampling to draw a score  $\tilde{\mu}_k^{(t)}$  for each cluster and select the cluster with the largest draw:

$$\begin{aligned} \tilde{\mu}_k^{(t)} &\sim \mathcal{N}\left(h_k^{(t)}, \frac{1}{(n_k^{(t)} + \epsilon)}\right), \\ c_t &= \arg \max_{k \in \{1, \dots, K\}} \tilde{\mu}_k^{(t)}. \end{aligned} \quad (2)$$

In practice, we use a *progress-aware* variant expressed directly in terms of hardness. For each cluster  $k$ , we track whether its solve rate  $r_k^{(t)}$  remains *informative* for learning: if  $r_k^{(t)}$  does not increase by at least a small tolerance for  $T_{\text{consecutive}}$  consecutive updates, we mark the cluster as *stagnating* and subtract a small constant from its Thompson Sampling mean. This mechanism reduces the chance of repeatedly sampling clusters that remain hard but are no longer improving, thereby encouraging exploration among clusters of comparable hardness

without overriding the standard Thompson Sampling trade-off.

This procedure naturally balances exploration and exploitation: clusters with fewer observations are more likely to be explored, while clusters with consistently low solve rates (i.e., harder samples) are sampled more frequently. A batch of size  $B$  is then drawn from cluster  $c_t$  to train  $\pi_\theta$ . For each sample, we compute a binary correctness reward:

$$r_i = \begin{cases} 1, & \text{if the response is correct} \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The average batch reward is  $r_{\text{avg}} = \frac{1}{B} \sum_{i=1}^B r_i$ . This scalar signal updates the policy  $\pi_\theta$  via an RL algorithm (e.g., GRPO), by maximizing

$$\max_{\theta} \mathbb{E}_{q \sim D_{\text{train}}, a \sim \pi_\theta} r_{\text{avg}}, \quad (4)$$

where  $a$  is the sampled answer from  $\pi_\theta$  given question  $q$ . In addition,  $r_{\text{avg}}$  updates the bandit statistics for the selected cluster:

$$R_{c_t}^{(t+1)} = R_{c_t}^{(t)} + r_{\text{avg}}, \quad n_{c_t}^{(t+1)} = n_{c_t}^{(t)} + 1. \quad (5)$$

This dynamic update ensures that the sampling distribution adapts online to the evolving state of the model: clusters that remain challenging receive more attention, while clusters that become easy are sampled less frequently. The full SPaCe procedure is provided in Algorithm 1 (Appendix A.1). We further analyze the convergence behavior of the scheduler in the following proposition.

**Proposition 1.** *Under assumptions: (i) bounded rewards, LLM training with (ii) gradient clipping and (iii) decayed learning rate, the Thompson Sampling scheduler in SPaCe satisfies sublinear variation up to step  $T$ :  $V_T = O(\log T)$ . Consequently, as  $t \rightarrow \infty$ , the sampling distribution concentrates on clusters with maximal expected sampling score.*

*Proof.* See Appendix A.2.  $\square$

## 4 Experiments

We evaluate the proposed method on multiple LLMs. Full fine-tuning is conducted on *Qwen3-0.6B*, *DeepSeek-R1-Distill-Qwen-1.5B*, *Qwen2.5-0.5B-Instruct*, *Falcon3-1B-Instruct*, and *Llama3.2-1B-Instruct* using a single NVIDIA H100 GPU; notably, *Qwen3-0.6B* attains performance comparable to larger models (e.g., *Qwen2.5-Math-7B-Instruct*). We additionally fine-tune *Qwen3-8B-Base* with

| Method       | Qwen3-0.6B                 |                            |                            |                            | DeepSeek-R1-Distill-Qwen-1.5B |                            |                            |                            |
|--------------|----------------------------|----------------------------|----------------------------|----------------------------|-------------------------------|----------------------------|----------------------------|----------------------------|
|              | GSM8K                      | MATH500                    | AIME24                     | AIME25                     | GSM8K                         | MATH500                    | AIME24                     | AIME25                     |
| Base         | 78.0                       | 75.4                       | 10.0                       | <u>16.7</u>                | 70.2                          | 78.4                       | 20.0                       | 16.7                       |
| $\pi_1$      | 78.1                       | 75.4                       | <u>13.3</u>                | <u>16.7</u>                | 69.5                          | 76.4                       | 20.0                       | 20.0                       |
| $\pi_2$      | <u>79.0</u>                | 74.2                       | 6.7                        | 10.0                       | 68.0                          | 79.4                       | 23.3                       | 20.0                       |
| Ordered      | <u>77.9</u>                | 74.8                       | <u>13.3</u>                | 13.3                       | <u>71.0</u>                   | <u>80.6</u>                | <b>26.7</b>                | 23.3                       |
| SFT          | 77.6 <sub>0.5</sub>        | 74.8 <sub>0.4</sub>        | 7.8 <sub>1.9</sub>         | 14.4 <sub>5.1</sub>        | 70.4 <sub>0.9</sub>           | 79.7 <sub>0.8</sub>        | 15.6 <sub>1.9</sub>        | 18.9 <sub>1.9</sub>        |
| R1           | 77.9 <sub>1.0</sub>        | 71.6 <sub>1.3</sub>        | 8.9 <sub>5.1</sub>         | 12.2 <sub>5.1</sub>        | 70.7 <sub>1.1</sub>           | 80.2 <sub>0.9</sub>        | 18.9 <sub>1.9</sub>        | <u>24.4</u> <sub>3.9</sub> |
| AdaRFT       | 78.9 <sub>0.3</sub>        | <u>75.9</u> <sub>1.5</sub> | 12.2 <sub>3.9</sub>        | 14.4 <sub>3.9</sub>        | 69.9 <sub>0.2</sub>           | 76.5 <sub>1.5</sub>        | <b>26.7</b> <sub>3.3</sub> | 16.7 <sub>6.7</sub>        |
| <b>SPaCe</b> | <b>79.8</b> <sub>0.3</sub> | <b>78.2</b> <sub>1.0</sub> | <b>20.0</b> <sub>5.8</sub> | <b>18.9</b> <sub>5.1</sub> | <b>72.4</b> <sub>1.2</sub>    | <b>81.1</b> <sub>1.0</sub> | <b>25.6</b> <sub>3.9</sub> | <b>26.7</b> <sub>5.8</sub> |

Table 1: Results with Qwen3-0.6B and DeepSeek-R1-Distill-Qwen-1.5B base LLMs trained on the DeepScaleR-Uniform dataset across multiple benchmarks. We report extractive match scores (mean<sub>std</sub>) at the final training checkpoint, averaged over 3 seeds (except for the Base,  $\pi_1$ ,  $\pi_2$ , and Ordered baselines). Best results are highlighted in bold, and the second-best are underlined.

LoRA (Hu et al., 2021) on a single H200 GPU. Experiments cover mathematical and logical reasoning benchmarks: DeepScaleR subsets (Uniform, Easy, and Difficult) (10k each) (Luo et al., 2025), GSM8K (Cobbe et al., 2021), and Knights and Knaves (Xie et al., 2024). Latent embeddings for clustering are extracted using *Qwen3-Embedding-0.6B*. All datasets include difficulty annotations, either derived from solve rates of a moderate LLM (Shi et al., 2025) or provided as explicit labels (Xie et al., 2024); difficulty is used as the per-example clustering attribute due to its interpretability. The total number of training examples is  $l \times K$ , where  $l$  is the per-cluster quota and  $K$  the number of clusters (reported in Table 15). Across all settings,  $K \leq 10$ , yielding at most 100 training examples with  $l = 10$ . Each experiment is repeated with three random seeds and implemented using the Open-R1 codebase (Hugging Face, 2025). The method introduces negligible runtime overhead relative to the R1 baseline; detailed timing results are provided in Figure 15 in the appendix.

**Evaluation** We use five benchmarks that span different reasoning types and difficulty levels: **GSM8K** (Cobbe et al., 2021), consists of diverse grade school math problems; **MATH500**, a 500-sample subset of the MATH dataset (Hendrycks et al., 2021); **AIME24** and **AIME25**, comprising problems from the 2024 and 2025 American Invitational Mathematics Examination, respectively; and finally, the logical reasoning **K&K** test set consists of 700 samples, with 100 examples for different number of people in the question from 2 to 8 (Xie et al., 2024). We report the extractive match scores for all mathematical datasets, following Lighteval’s evaluation framework (Habib et al., 2023). For K&K dataset, we follow the evaluation protocols

established by the dataset authors (Xie et al., 2024).

**Baselines** **Base** refers to the pretrained model without any fine-tuning.  $\pi_1$  and  $\pi_2$  represent the baselines in **1-shot RLVR** paper, trained on one and two examples selected from the DeepScaleR dataset, respectively (Wang et al., 2025b). **SFT** denotes the supervised fine-tuning baseline. **Ordered** (Bengio et al., 2009) is a curriculum baseline in which training begins with easier examples and gradually progresses to harder ones. **R1** is the RL baseline trained with the standard GRPO algorithm without an SFT cold start, as in DeepSeek-R1 (DeepSeek-AI, 2025). **AdaRFT** is a curriculum learning approach that selects examples based on a difficulty threshold (Shi et al., 2025). We note that SFT, Ordered, R1, and AdaRFT baselines are trained on the full datasets. We also include the variance-based baseline **LIM** (Li et al., 2025), trained on MATH (Hendrycks et al., 2021); details and results are in Appendix A.6 due to its different training data.

## 5 Experimental Results

### 5.1 Mathematical Reasoning Benchmark

We fine-tune *Qwen3-0.6B* and *DeepSeek-R1-Distill-Qwen-1.5B* in a zero-shot setting and report their performance on four mathematical benchmarks. Despite their small size, they serve as strong backbones for mathematical reasoning. Especially, *Qwen3-0.6B* supports a *thinking mode* enabled via `<think>` `</think>` tags. Following (Wang et al., 2025b), we use a single seed for  $\pi_1$  and  $\pi_2$  baselines. For the *Ordered* baseline, we likewise use one seed, since the training example order is fixed. Additional training hyperparameters are given in Appendix A.17.

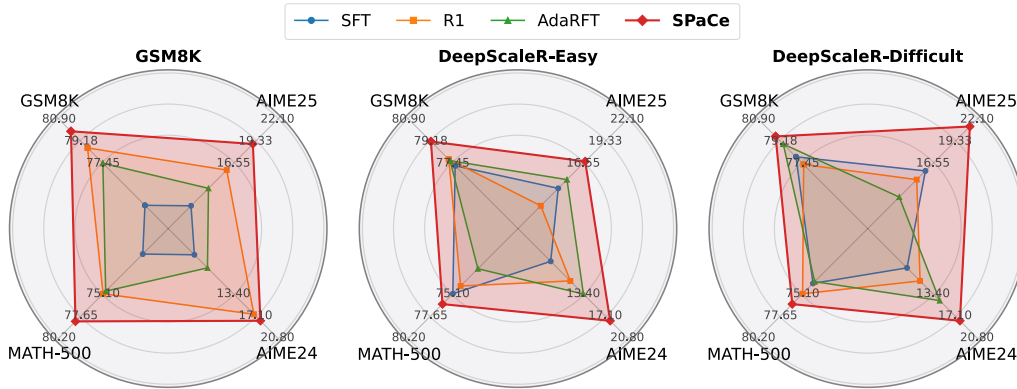


Figure 2: Results averaged over 3 training seeds using three training datasets with Qwen3-0.6B.

Table 1 presents test accuracies from the final training checkpoints using *Qwen3-0.6B* and *DeepSeek-R1-Distill-Qwen-1.5B*. SPaCe consistently achieves the highest performance across all benchmarks. On *Qwen3-0.6B*, SPaCe attains the best score on all four benchmarks, improving over the Base model by 1.8% on GSM8K and 2.8% on MATH500, with larger gains on the harder AIME sets (+5.6% on AIME24 and +2.2% on AIME25). On *DeepSeek-R1-Distill-Qwen-1.5B*, SPaCe improves over Base on GSM8K (+2.2%), MATH500 (+2.7%), AIME24 (+5.6%), and AIME25 (+10.0%), achieving the best result on three out of four benchmarks and remaining within 1.1% of the best method on AIME24. Overall, SPaCe provides consistent improvements relative to the included baselines, with particularly noticeable gains on the AIME benchmarks.

## 5.2 SPaCe Works With Various Datasets

### 5.2.1 Mathematical Datasets Training Results

We evaluate our method on three distinct training sets: (1) GSM8K; (2) DeepScaleR–Easy, a subset of DeepScaleR with primarily low-difficulty questions; and (3) DeepScaleR–Difficult, a subset with mainly high-difficulty questions. All experiments use *Qwen3-0.6B* as the backbone, and we adopt the same reward functions and hyperparameters as in Section 5.1. We compare against the top-3 baselines, excluding DeepScaleR-specific baselines  $\pi_1$  and  $\pi_2$ . As seen in Figure 2, across all settings, SPaCe consistently outperforms standard SFT, achieving gains of 2.9–5.5 % on GSM8K and up to 7.8 % on the AIME benchmarks. R1 generally ranks second, especially on the easier splits, while AdaRFT falls 1–2 % behind in most cases. Notably, when trained on the difficult subset (3), SPaCe attains a 2.3 %

improvement on MATH500 and more than doubles AIME24 accuracy relative to SFT. These results confirm that SPaCe not only enhances overall accuracy but also yields the greatest benefits on the most challenging training set.

### 5.2.2 K&K Training Results

In this dataset, we consider the number of people in each question as the per-example attribute  $d_i$  and use *Qwen3-0.6B* as the base LLM. The final answer is used to compute the accuracy reward, and the original evaluation protocol (Xie et al., 2024) is followed for consistency and comparability. As shown in Table 2, SPaCe consistently outperforms all baselines across difficulty levels, with especially clear gains on harder puzzles with 2–5 people, where reasoning demands are higher. It achieves the highest overall accuracy of 9.9%, versus 8.1% for R1 and 7.8% for AdaRFT, which shows relative improvements of 22.2% and 26.9%, respectively. AdaRFT’s accuracy drops in 2 of 7 settings, suggesting that a noisy curriculum can harm performance. Overall, these results show that SPaCe scales effectively to harder reasoning cases while remaining competitive on simpler ones, validating curriculum-guided selection for reasoning-focused training.

## 5.3 SPaCe Helps Diverse LLM Learners

We train on DeepScaleR-Uniform and evaluate *Qwen2.5-0.5B-Instruct*, *Falcon3-1B-Instruct*, *Llama3.2-1B-Instruct*, and *Qwen3-8B-Base*, which are compact to mid-size LLMs with strong reasoning, language, code, and math skills. We exclude *Qwen3-8B* (reasoning-enabled) due to the substantial compute from long `<think>` traces. We compare against the Base model and AdaRFT, the most consistent and second-best method in Sec-

| Method       | Number of People |                 |                 |                |                |         |                | Average        |
|--------------|------------------|-----------------|-----------------|----------------|----------------|---------|----------------|----------------|
|              | 2                | 3               | 4               | 5              | 6              | 7       | 8              |                |
| Base         | 32.0             | 10.0            | 8.0             | 2.0            | 0.0            | 0.0     | 0.0            | 7.4            |
| R1           | 31.7±1.5         | 11.3±0.6        | 8.3±0.6         | 4.0±1.0        | 0.7±0.6        | 0.0±0.0 | 1.0±0.0        | 8.1±0.3        |
| AdaRFT       | 31.7±0.6         | 10.7±2.1        | 7.3±0.6         | 3.7±0.6        | 0.3±0.6        | 0.0±0.0 | 1.0±0.0        | 7.8±0.2        |
| <b>SPaCe</b> | <b>34.3±1.2</b>  | <b>15.7±3.1</b> | <b>10.7±1.5</b> | <b>5.7±0.6</b> | <b>1.7±0.6</b> | 0.0±0.0 | <b>1.3±0.6</b> | <b>9.9±1.0</b> |

Table 2: Accuracy (%) by number of people in K&K puzzles with results reported as mean ± standard deviation (except for Base baseline) over 3 runs using *Qwen3-0.6B* as the base model. Bold denotes the best mean performance.

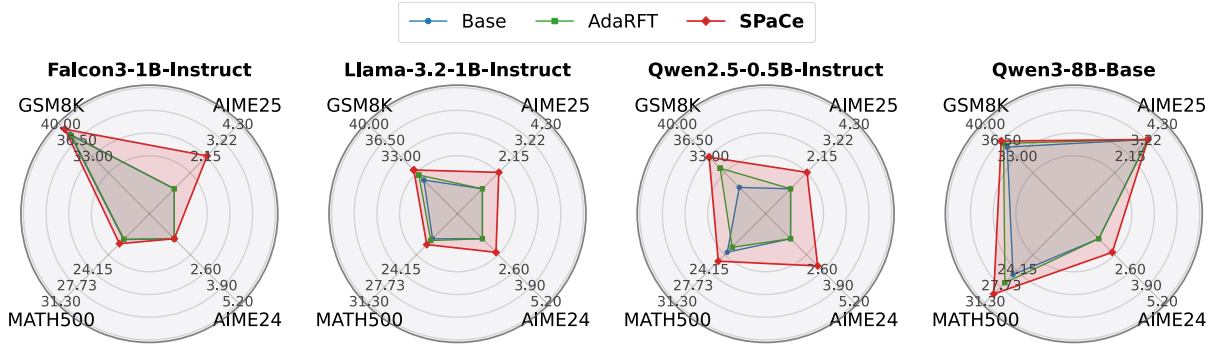


Figure 3: Results averaged over 3 training seeds using other LLMs across datasets.

tion 5.1. All models use the same zero-shot setup, except *Llama3.2-1B-Instruct*, which requires one in-context example per instance to yield valid correctness rewards (Le et al., 2025). Figure 3 shows SPaCe as the clear winner. Average gains are evident on GSM8K ( $\sim+3\%$ ) and MATH500 ( $\sim+2\%$ ). On the harder AIME splits, SPaCe turns near-zero baseline scores into consistent positives, reflecting better sample efficiency under sparse-reward RFT. We see complete or near-complete sweeps on the smaller models over Base and AdaRFT, indicating benefits in capacity-constrained settings; results on *Qwen3-8B-Base* remain strong despite not being used for RFT training. Overall, a bandit-driven, performance-aware curriculum generalizes across architectures and tasks with minimal protocol changes, delivering reliable gains under compute-conscious budgets.

## 6 Ablation Studies and Model Analyses

### 6.1 Multi-arm Bandit Analysis

**Empirical Convergence** SPaCe leverages the MAB framework to adaptively guide curriculum learning, making it important to characterize how the scheduler evolves during training. Figure 4 (Left) illustrates a heatmap of cluster solve rates alongside bandit selections over time using *Qwen3-0.6B* on the DeepScaleR-Uniform dataset partitioned into 7 clusters. Early in training (around step

200), the bandit behaves nearly uniformly, allocating samples across clusters with little preference. As the model accumulates experience, clear patterns emerge: clusters 1 and 2 exhibit higher solve-rate differentials, and the bandit correspondingly shifts toward sampling them more frequently, signaling that these clusters provide greater marginal learning benefit. From step 600 onward, this concentration intensifies, with clusters 1 and 2 dominating the selection distribution, indicating that the scheduler successfully adapts to focus training on regions of the data that remain most informative for continued performance improvement.

**Solve Rate Trends.** Clusters 1 and 2 deliver the largest gains, with solve rate rising from 20% at step 200 to 40% by step 1200, indicating moderate difficulty and strong learning signals. In contrast, cluster 3 is rarely selected and stays flat at around 70%, suggesting it is too easy to drive improvement. Clusters 4 and 5 show smaller gains under continued exploration, while cluster 0 provides little benefit. By the end of training, the bandit concentrates on clusters 1 and 2, which consistently yield the highest returns.

### 6.2 Data Reduction Analysis

**Average Sample Difficulty Comparison** We consider *Qwen3-0.6B* on DeepScaleR-Uniform with 7 clusters to examine selected example dif-

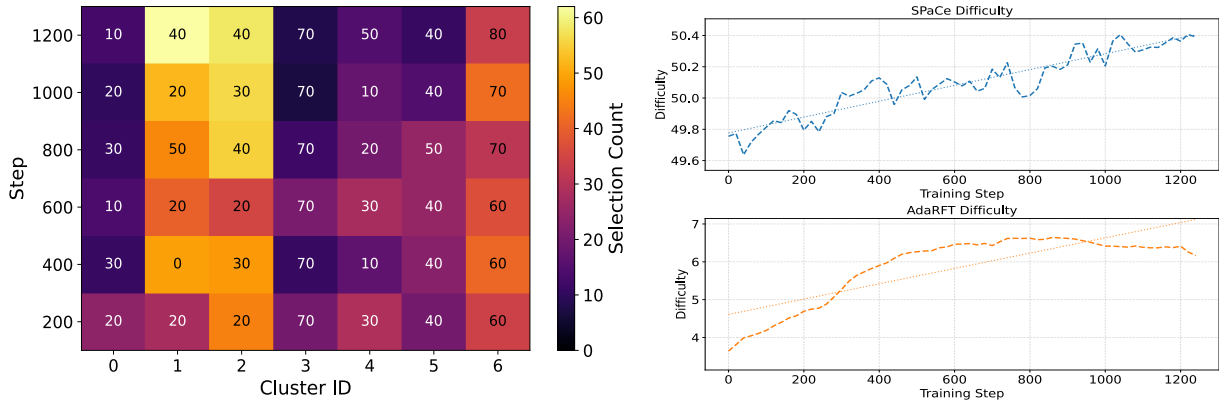


Figure 4: **Left:** Multi-arm bandit cluster selection (heatmap) with per-cluster solve rates (percentages) annotated in each cell across training steps. **Right: Top:** Difficulty of examples selected by SPaCe over time. **Bottom:** Difficulty of examples selected by AdaRFT over time.

| Method             | GSM8K           | MATH500         | AIME24          | AIME25          |
|--------------------|-----------------|-----------------|-----------------|-----------------|
| Base               | 78.0            | 75.4            | 10.0            | 16.7            |
| SPaCe <sup>-</sup> | 78.8±0.4        | 76.4±0.7        | 12.2±3.9        | 11.1±1.9        |
| SPaCe              | <b>79.8±0.3</b> | <b>78.2±1.0</b> | <b>20.0±5.8</b> | <b>18.9±5.1</b> |

Table 3: Mean  $\pm$  std over 3 seeds on DeepScaleR-uniform using Qwen3-0.6B. SPaCe<sup>-</sup> denotes the variant without data reduction. Best results in bold.

difficulties. Figure 4 (Right) compares the average difficulty of training examples chosen over time by SPaCe and AdaRFT. Although AdaRFT accesses the full dataset, its performance with LLMs struggles to reach medium and hard examples, due to a threshold mechanism that is highly sensitive and skews selection. In contrast, SPaCe favors medium to hard instances, avoiding the overemphasis on easy examples seen in AdaRFT. We attribute this to our clustering strategy, which captures both semantic diversity and difficulty: each cluster mixes a broad range of examples, enabling exploration of harder cases without sacrificing variety.

**Impact of Data Reduction** To assess the data reduction phase, we ablate selection by keeping all examples in each cluster. As shown in Table 3, this variant (SPaCe<sup>-</sup>) still outperforms the Base baseline but underperforms full SPaCe. Without reduction, the number of examples per cluster can exceed the batch size  $B$ , increasing variance and within-batch difficulty heterogeneity, which weakens the learning signal.

### 6.3 Clustering Effects

**Clustering Analysis** We further investigate the effect of training-data clustering. Figure 5 visu-

alizes the resulting clusters in a shared semantic space and highlights clear differences across clustering strategies. When clustering is based only on difficulty, examples with very different semantic content are often grouped together simply because they share similar hardness levels. This produces clusters that are less coherent in terms of task or content structure. In contrast, semantic-only clustering groups together examples that are topically similar, but the corresponding hardness regions remain highly overlapping, making it difficult to separate clusters by learning difficulty alone. Joint clustering, which combines semantic representations with per-example difficulty, achieves a better balance: the resulting clusters remain semantically localized while also exhibiting clearer separation along the difficulty dimension.

Figure 6 further supports this observation by showing the distribution of difficulty scores within each cluster. Semantic-only clusters tend to span broad and overlapping difficulty ranges, suggesting that semantic similarity alone is insufficient for forming well-structured curriculum units. By comparison, joint clustering produces tighter and more coherent difficulty bands, while still avoiding the degenerate behavior of purely difficulty-based clustering, where semantic diversity within each cluster becomes too large. Taken together, these findings suggest that concatenating embeddings with per-example difficulty provides a principled way to construct curriculum units that preserve semantic similarity while also controlling variation in difficulty. This balance is particularly desirable for curriculum learning, where both semantic coherence and gradual difficulty progression are important for effective training.

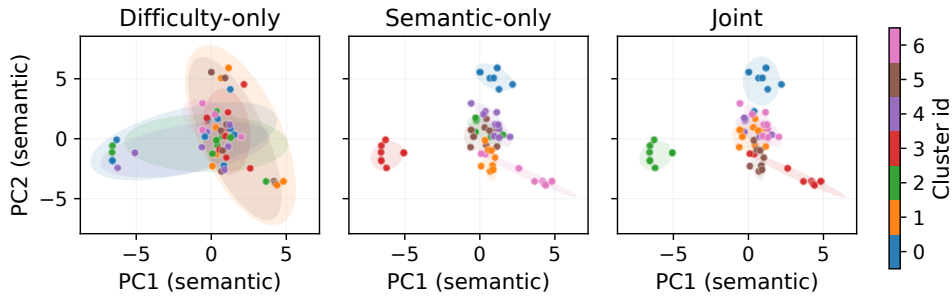


Figure 5: Clustering variants in semantic space with 7 clusters using DeepScaleR-Uniform.

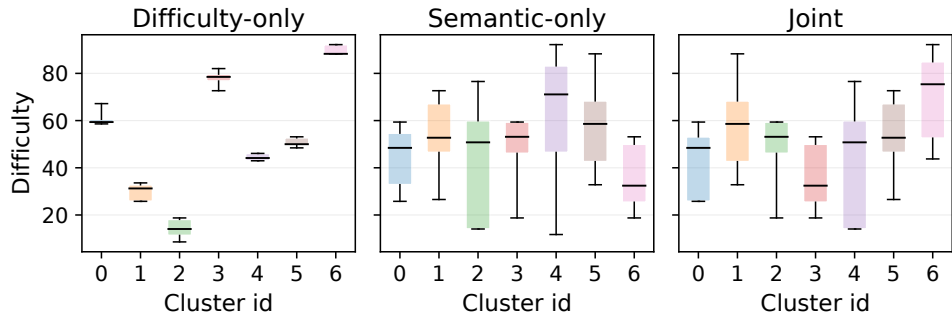


Figure 6: Difficulty of three clustering variants in semantic space with 7 clusters using DeepScaleR-Uniform.

## 6.4 Other Ablation Studies

We also ablate (i) paired statistical testing, (ii) number of clusters, (iii) diverse sample selection, (iv) embedding model choice, (v) number of PCA components, (vi) samples per cluster, (vii) removing difficulty effect, (viii) selected-sample difficulty, (ix) dataset distribution, (x) cluster properties and (xi) training time (Appendices A.4, A.5, A.7, A.8, A.9, A.10, A.11, A.13, A.14, A.15, A.16). Across these dimensions, the results consistently support the robustness of our approach, clarify trade-offs and hyperparameter sensitivities, and offer practical guidance for default settings.

## 7 Conclusion

We introduced **SPaCe**, a lightweight framework that enables efficient reasoning in language models through clustering and adaptive curriculum learning. SPaCe selects compact, diverse training subsets and dynamically adapts training focus based on model performance. Experiments show that SPaCe achieves competitive accuracy with significantly fewer samples. These results highlight the effectiveness of combining semantic clustering with performance-driven curricula to unlock reasoning in small models using minimal resources.

## Limitations

While SPaCe consistently improves training efficiency and performance across our evaluated settings, our experiments are limited to models with fewer than 8B parameters. Further evaluation on larger models would be valuable to better assess scalability and generality.

## References

- Art of Problem Solving. Art of problem solving wiki. [https://artofproblemsolving.com/wiki/index.php/Main\\_Page](https://artofproblemsolving.com/wiki/index.php/Main_Page). Accessed: 2025-09-19.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. **Curriculum learning**. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, page 41–48, New York, NY, USA. Association for Computing Machinery.
- Omar Besbes, Yonatan Gur, and Assaf Zeevi. 2014. Stochastic multi-armed-bandit problem with non-stationary rewards. *Advances in neural information processing systems*, 27.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). Preprint, arXiv:2501.12948.
- Van Dai Do, Quan Hung Tran, Ahmed Kirmani, Lu Zhang, and Hung Le. 2025a. [Sample efficient alignment learning with episodic control](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10601–10618, Suzhou, China. Association for Computational Linguistics.
- Van Dai Do, Quan Hung Tran, Svetha Venkatesh, and Hung Le. 2025b. [Dynamic steering with episodic memory for large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13731–13749, Vienna, Austria. Association for Computational Linguistics.
- Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. 2023. [RAFT: Reward ranked finetuning for generative foundation model alignment](#). *Transactions on Machine Learning Research*.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Google. 2024. [Introducing gemini 2.0: Our new ai model for the agentic era](#). Accessed: 2025-03-05.
- Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. [Lighteval: A lightweight framework for llm evaluation](#).
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#). Preprint, arXiv:2106.09685.
- Jian Hu. 2025. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*.
- Tianjin Huang, Ziquan Zhu, Gaojie Jin, Lu Liu, Zhangyang Wang, and Shiwei Liu. 2025. [SPAM: Spike-aware adam with momentum reset for stable LLM training](#). In *The Thirteenth International Conference on Learning Representations*.
- Hugging Face. 2025. [Open r1: A fully open reproduction of deepseek-r1](#).
- Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Juntao Dai, Tianyi Qiu, and Yaodong Yang. 2024. [Aligner: Efficient alignment by learning to correct](#). In *Advances in Neural Information Processing Systems*.
- I. T. Jolliffe. 2002. *Principal Component Analysis*. Springer.
- Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Salman Khan, and Fahad Shahbaz Khan. 2025. [Llm post-training: A deep dive into reasoning large language models](#). *arXiv preprint arXiv:2502.21321*.
- Hung Le, Majid Abdolshah, Thommen K George, Kien Do, Dung Nguyen, and Svetha Venkatesh. 2022. [Episodic policy gradient training](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7317–7325.
- Hung Le, Dai Do, Dung Nguyen, and Svetha Venkatesh. 2025. [Reasoning under 1 billion: Memory-augmented reinforcement learning for large language models](#). *arXiv preprint arXiv:2504.02273*.
- Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025. [Limr: Less is more for rl scaling](#). Preprint, arXiv:2502.11886.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. [Let’s verify step by step](#). *arXiv preprint arXiv:2305.20050*.
- Stuart Lloyd. 1982. [Least squares quantization in pcm](#). *IEEE Transactions on Information Theory*, 28(2):129–137.
- Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. [Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl](#). <https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL>. Notion Blog.
- OpenAI. 2024. [Hello gpt-4o](#).
- OpenAI. 2025. [Introducing gpt-5](#).
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. [Training language models to follow instructions with human feedback](#). *Advances in neural information processing systems*, 35:27730–27744.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Walter Rudin. 1976. *Principles of Mathematical Analysis*, 3rd edition. McGraw-Hill.
- Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. 2020. [A tutorial on thompson sampling](#). Preprint, arXiv:1707.02038.

- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *Preprint*, arXiv:1707.06347.
- Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. 2025. [Efficient reinforcement fine-tuning via adaptive curriculum learning](#). *Preprint*, arXiv:2504.05520.
- Mingyang Song, Mao Zheng, Zheng Li, Wenjie Yang, Xuan Luo, Yue Pan, and Feng Zhang. 2025. [Fastcurl: Curriculum reinforcement learning with stage-wise context scaling for efficient training rl-like reasoning models](#). *Preprint*, arXiv:2503.17287.
- Student. 1908. The probable error of a mean. *Biometrika*, 6(1):1–25.
- Richard S Sutton, Andrew G Barto, and 1 others. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.
- William R. Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.
- Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2025. [Steering language models with activation engineering](#). In *The Thirteenth International Conference on Learning Representations*.
- Guoxia Wang, Shuai Li, Congliang Chen, Jinle Zeng, Jiabin Yang, Tao Sun, Yanjun Ma, Dianhai Yu, and Li Shen. 2025a. [Adagc: Improving training stability for large language model pretraining](#). *Preprint*, arXiv:2502.11034.
- Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. 2025b. [Reinforcement learning for reasoning in large language models with one training example](#). *Preprint*, arXiv:2504.20571.
- Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. 2025. [Light-rl: Curriculum sft, dpo and rl for long cot from scratch and beyond](#). *Preprint*, arXiv:2503.10460.
- Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. 2024. [On memorization of large language models in logical reasoning](#).
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and

## A Appendix

### A.1 Algorithm for SPaCe

In this section, we provide the pseudo-code for SPaCe in Algorithm 1.

---

#### Algorithm 1 SPaCe

---

**Input:** Policy  $\pi_\theta$ , Dataset  $\mathcal{D}$ , Embedding model  $\phi$ , Clusters  $K$ , Batch size  $B$ , RL algorithm  $\mathcal{A}$ ,  $\epsilon > 0$ ,  $\delta > 0$ ,  $T_{\text{consecutive}}$ ,  $\gamma > 0$

**Output:** Trained policy  $\pi_\theta$

```

1: // Phase 1: Cluster-based Data Reduction
2: for each  $x_i \in \mathcal{D}$  do
3:    $e_i \leftarrow \text{PCA}(\phi(x_i)) \oplus \text{difficulty}(x_i)$ 
4: end for
5: Run K-means on  $\{e_i\}$  to form  $\{C_k\}_{k=1}^K$ ; pick
    $l$  diverse samples per cluster  $\Rightarrow \mathcal{D}_{\text{train}}$ 
6: // Phase 2: Bandit-driven Curriculum
7: for each  $k = 1, \dots, K$  do
8:    $R_k \leftarrow 0$ ,  $n_k \leftarrow 0$ ,  $\text{no\_improve}_k \leftarrow 0$ 
9: end for
10: while training not finished do
11:   for each  $k = 1, \dots, K$  do
12:      $h_k \leftarrow -\frac{R_k}{n_k + \epsilon}$ 
13:      $m_k \leftarrow h_k - \gamma \cdot \mathbb{I}[\text{no\_improve}_k \geq T_{\text{consecutive}}]$ 
14:      $\tilde{\mu}_k \sim \mathcal{N}\left(m_k, \frac{1}{n_k + \epsilon}\right)$ 
15:   end for
16:    $c_t \leftarrow \arg \max_k \tilde{\mu}_k$ ; sample  $X \subset C_{c_t}$ ,  $|X| = B$ 
17:    $G \leftarrow \pi_\theta(X)$ ; compute  $r_{\text{avg}} = \frac{1}{B} \sum_{i=1}^B r_i$ ,  $r_i \in \{0, 1\}$ 
18:    $\pi_\theta \leftarrow \mathcal{A}(\pi_\theta, X, G, r_{\text{avg}})$ 
19:    $R_{c_t} \leftarrow R_{c_t} + r_{\text{avg}}$ ,  $n_{c_t} \leftarrow n_{c_t} + 1$ 
20:    $\text{no\_improve}_{c_t} \leftarrow \mathbb{I}\left[\frac{R_{c_t}}{n_{c_t} + \epsilon} < \frac{R_{c_t} - r_{\text{avg}}}{(n_{c_t} - 1) + \epsilon} + \delta\right]$ 
21: end while
22: return  $\pi_\theta$ 

```

---

### A.2 Convergence of the Thompson Sampling Scheduler

We analyze the convergence of the Thompson Sampling scheduler used in SPaCe. Each data cluster is treated as an arm in a multi-armed bandit. At step  $t$ , let  $\pi_{\theta^{(t)}}$  denote the model with parameters  $\theta^{(t)}$ . The expected reward (solve rate) of cluster  $\mathcal{C}_k$  is defined as:

$$\mu_k^{(t)} = \mathbb{E}_{x \sim \mathcal{C}_k} [\text{Pr}(\pi_{\theta^{(t)}}(x) = \text{correct})]. \quad (6)$$

where  $\text{Pr}(\pi_{\theta^{(t)}}(x) = \text{correct})$  denotes the probability that the model produces a correct answer for input  $x$ .

We already have: (1) the model is trained using gradient clipping with threshold  $G_{\text{max}}$ ; (2) the learning rate  $\alpha_t$  follows a cosine decay schedule with warmup and is therefore non-increasing and vanishes as  $t \rightarrow \infty$ ; and (3) the expected rewards satisfy  $\mu_k^{(t)} \in [0, 1]$  for all clusters  $k$ . These three properties are directly enforced in the SPaCe implementation.

To complete the convergence analysis, we now bound the drift of each cluster’s expected reward. Define

$$f_k(\theta) = \mathbb{E}_{x \sim \mathcal{C}_k} [\text{Pr}(\pi_\theta(x) = \text{correct})], \quad (7)$$

where  $f_k$  is the *cluster-level reward surface* for arm  $k$ , and note that since each layer of our base model is continuously differentiable, so is  $f_k(\theta)$  (Goodfellow et al., 2016). We assume that, in practice, gradient clipping at norm  $G_{\text{max}}$  together with a bounded initialization prevents the parameters  $\{\theta^{(t)}\}$  from diverging excessively, effectively keeping them in some large but fixed ball  $\{\|\theta\| \leq R\}$ . Empirical studies on LLM training have repeatedly observed that clipped updates under cosine-decay schedules yield stable trajectories without catastrophic parameter growth (Wang et al., 2025a; Huang et al., 2025).

Under this assumption, the extreme-value theorem (Rudin, 1976) guarantees the existence of a constant  $H < \infty$  such that

$$\|\nabla_\theta f_k(\theta)\| \leq H \quad \forall \|\theta\| \leq R. \quad (8)$$

Moreover, each gradient step with cosine-decay learning rate  $\alpha_t$  and gradient clipping satisfies

$$\|\theta^{(t+1)} - \theta^{(t)}\| \leq \alpha_t G_{\text{max}}. \quad (9)$$

Applying the mean-value theorem (Rudin, 1976) then yields

$$\begin{aligned} |\mu_k^{(t+1)} - \mu_k^{(t)}| &= |f_k(\theta^{(t+1)}) - f_k(\theta^{(t)})| \\ &\leq H \|\theta^{(t+1)} - \theta^{(t)}\| \\ &\leq H G_{\text{max}} \alpha_t = \varepsilon_t, \end{aligned} \quad (10)$$

where  $\varepsilon_t \rightarrow 0$  as  $\alpha_t \rightarrow 0$ . Thus, we obtain the desired vanishing drift  $|\mu_k^{(t+1)} - \mu_k^{(t)}| \leq \varepsilon_t$  for every cluster  $k$ .

Let  $V_T = \sum_{t=1}^{T-1} \max_k |\mu_k^{(t+1)} - \mu_k^{(t)}|$  denote the total reward variation up to step  $T$ . The bound

above implies

$$V_T \leq \sum_{t=1}^{T-1} \varepsilon_t. \quad (11)$$

Let  $A_T := \sum_{t=1}^{T-1} \alpha_t$  denote the cumulative step size up to step  $T$ . Combining the drift bound with the definition of  $V_T$  gives

$$V_T \leq \sum_{t=1}^{T-1} \varepsilon_t = H G_{\max} \sum_{t=1}^{T-1} \alpha_t = H G_{\max} A_T. \quad (12)$$

For the cosine decay schedule with warmup used in SPaCe, the learning rate is non-increasing after warmup and reaches 0 at the end of the prescribed training horizon  $T_{\text{train}}$ ; we can equivalently take  $\alpha_t = 0$  for all  $t > T_{\text{train}}$ . Hence  $A_T$  is finite and bounded by

$$A_T \leq \sum_{t=1}^{T_{\text{train}}} \alpha_t = O(T_{\text{train}} \alpha_{\max}), \quad (13)$$

which implies that the total reward variation  $V_T$  is also bounded (and therefore sublinear as  $T \rightarrow \infty$  under the extension  $\alpha_t = 0$  for  $t > T_{\text{train}}$ ). This bounded variation budget matches the standard non-stationary bandit setting studied in prior work (Besbes et al., 2014), and supports the use of Thompson Sampling as a scheduler that tracks the best-performing cluster as training progresses and the drift  $\varepsilon_t$  becomes small. In particular, as  $\alpha_t \rightarrow 0$ , we have  $\varepsilon_t \rightarrow 0$ , so the cluster rewards become effectively stationary, and Thompson Sampling concentrates its selections on the cluster(s) with the highest current expected reward.

### A.3 Additional Results

In this section, we provide the full results for Qwen3-8B-Base in Table 4 and Meta-Llama3-8B in Table 5. These new experiments confirm that the gains of our method persist when scaling to the 8B regime, addressing the concern that our findings may be specific to small models.

### A.4 Paired Statistical Analysis

To verify the impact of SPaCe, we conduct paired statistical tests between SPaCe and baselines, each run with three random seeds. We use matched training seeds for SPaCe and the seed-controlled baselines (SFT, R1, and ADARFT) to enable a paired comparison. We report paired accuracy improvements (in percentage) of SPaCe over each baseline across multiple settings, using the Student’s

$t$ -distribution with  $df=2$  (Student, 1908). The comparisons using other LLMs are shown in Table 6. The comparisons on Knights and Knaves are shown in Table 7. The comparisons using other datasets as training data are shown in Table 8.

Table 6 reports paired accuracy gains of SPaCe over each baseline across three shared seeds. On **Qwen3-0.6B**, SPaCe yields *consistently significant* improvements on **GSM8K** and **MATH500**, with all 90% confidence intervals remaining strictly above zero, indicating reliable gains across baselines. On **DeepSeek-R1**, improvements are also robust on **MATH500**, while **GSM8K** gains are positive but less stable due to wider intervals. In contrast, results on **AIME24** and **AIME25** exhibit substantially larger variance and several intervals overlap zero, suggesting that while mean gains are often positive, statistical evidence is inconclusive under the three-seed setting.

On Knights and Knaves (Table 7), SPaCe consistently matches or improves upon both R1 and AdaRFT across depths 2–8, with the strongest mean gains concentrated at shallower depths (2–4). While confidence intervals are wider under three seeds, the improvements remain directionally positive throughout and become more stable at higher depths, where results converge as the task becomes more constrained. Overall, these findings suggest that SPaCe provides reliable benefits on this dataset, particularly for lower-depth instances where effective curriculum selection is most impactful.

Table 8 shows that SPaCe improves over both R1 and AdaRFT on DeepScaleR-Easy and DeepScaleR-Difficult, with consistently positive mean gains across all benchmarks. On the Easy split, improvements are particularly stable on **MATH500** (e.g.,  $+2.13 \pm 0.00$  over R1 and  $+3.40 \pm 0.78$  over AdaRFT), indicating reliable benefits under matched seeds. Notably, SPaCe continues to provide meaningful gains on the Difficult split, including improvements on **GSM8K** and **MATH500** and sizeable gains on **AIME25** (up to  $+6.67 \pm 4.89$  over R1). While the hardest benchmarks (AIME24/AIME25) exhibit larger variability under three seeds, the overall trend remains positive, suggesting that SPaCe generalizes across both easier and more challenging training regimes.

Finally, Table 9 shows results using other LLMs as backbones. Across four backbones, SPaCe consistently yields positive paired accuracy improvements over AdaRFT, indicating that its data as-

Table 4: Qwen3-8B-Base results (accuracy  $\pm$  std).

| Method              | GSM8K                            | MATH500                          | AIME24                          | AIME25                          |
|---------------------|----------------------------------|----------------------------------|---------------------------------|---------------------------------|
| Base                | 35.1                             | 25.0                             | 0.0                             | <b>3.3</b>                      |
| Method A            | 35.0                             | 28.0                             | <b>1.1</b>                      | 0.0                             |
| Method B            | 34.7                             | 28.2                             | 0.0                             | 0.0                             |
| Ordered             | 34.6                             | 26.0                             | 0.0                             | 0.0                             |
| SFT                 | 34.0 $\pm$ 0.7                   | 26.4 $\pm$ 0.6                   | 0.0 $\pm$ 0.0                   | 0.0 $\pm$ 0.0                   |
| R1                  | 35.5 $\pm$ 1.3                   | 27.0 $\pm$ 0.6                   | <b>1.1 <math>\pm</math> 1.9</b> | 0.0 $\pm$ 0.0                   |
| AdaRFT              | 35.9 $\pm$ 0.7                   | 26.8 $\pm$ 1.2                   | 0.0 $\pm$ 0.0                   | <b>3.3 <math>\pm</math> 0.0</b> |
| <b>SPaCe (Ours)</b> | <b>36.4 <math>\pm</math> 0.9</b> | <b>29.3 <math>\pm</math> 1.1</b> | <b>1.1 <math>\pm</math> 1.9</b> | <b>3.3 <math>\pm</math> 0.0</b> |

Table 5: Llama3-8B results (accuracy  $\pm$  std).

| Method              | GSM8K                            | MATH500                          | AIME24                          | AIME25                          |
|---------------------|----------------------------------|----------------------------------|---------------------------------|---------------------------------|
| Base                | 34.4                             | 26.4                             | 0.0                             | 3.3                             |
| Method A            | 50.3                             | 30.3                             | <b>3.3</b>                      | 0.0                             |
| Method B            | 51.0                             | 31.1                             | <b>3.3</b>                      | 0.0                             |
| Ordered             | 50.6                             | 32.2                             | 0.0                             | 0.0                             |
| SFT                 | 50.3 $\pm$ 1.2                   | 38.4 $\pm$ 0.9                   | <b>3.3 <math>\pm</math> 0.0</b> | 0.0 $\pm$ 0.0                   |
| R1                  | 51.6 $\pm$ 1.5                   | 34.0 $\pm$ 1.6                   | 0.0 $\pm$ 0.0                   | 0.0 $\pm$ 0.0                   |
| AdaRFT              | 51.2 $\pm$ 0.8                   | 36.1 $\pm$ 1.2                   | <b>3.3 <math>\pm</math> 0.0</b> | 3.3 $\pm$ 0.0                   |
| <b>SPaCe (Ours)</b> | <b>52.2 <math>\pm</math> 1.3</b> | <b>38.8 <math>\pm</math> 0.7</b> | <b>3.3 <math>\pm</math> 0.0</b> | <b>5.5 <math>\pm</math> 3.9</b> |

signment strategy generalizes well beyond a single base LLM. The gains are most pronounced on **GSM8K**, where every backbone benefits from a clear uplift, suggesting improved reasoning reliability under shared-seed evaluation. On **MATH500**, SPaCe remains beneficial and can be notably stable, with **Qwen3-8B-Base** showing a strong improvement accompanied by a tight confidence interval, consistent with dependable rather than noisy gains. Even on the more challenging **AIME** benchmarks, SPaCe maintains non-degrading behavior and achieves additional improvements on several backbones, reinforcing its robustness and transferability across diverse model families.

### A.5 Ablation On Number Of Clusters

In this section, we ablate on different number of clusters to see its effect on the final performance. Figure 7 shows how SPaCe’s performance depends on the number of clusters  $K$ . Accuracy peaks at a moderate value ( $K=7$ ), while very few ( $K=1$ ) or many ( $K=20$ ) clusters reduce performance. This reflects a trade-off between specialization and generalization: too few clusters collapse diverse examples into coarse groups, while too many fragment the data, leading to sparse sampling and unstable training signals. A moderate clustering level provides the best balance, enabling the ban-

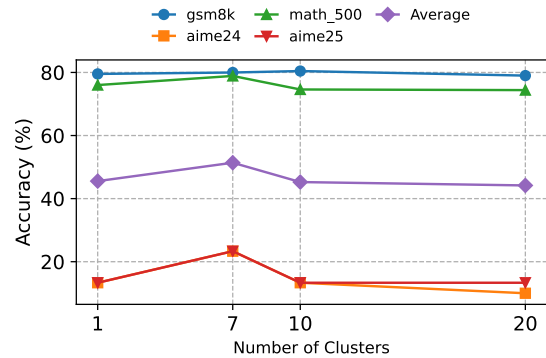


Figure 7: Results for 1 seed with Qwen3-0.6B and different number of clusters.

dit to exploit informative variation without over-fragmentation. These results highlight the importance of tuning  $K$ , as it directly affects how effectively the curriculum leverages per-example information.

### A.6 Additional Baseline: Learning Impact Measurement

In this section, we compare the performance of our method on mathematical reasoning tasks against a variance-based data selection approach, Learning Impact Measurement (LIM) (Li et al., 2025). While the original LIM paper reports results on

Table 6: Paired accuracy improvements (%) of **SPaCe** over each baseline, shown for two backbones. Entries are mean  $\pm$  90% confidence interval over three shared random seeds (Student’s  $t$ -distribution,  $df = 2$ ).

| Baseline | Qwen3-0.6B       |                  |                   |                  | DeepSeek-R1      |                  |                   |                    |
|----------|------------------|------------------|-------------------|------------------|------------------|------------------|-------------------|--------------------|
|          | GSM8K            | MATH500          | AIME24            | AIME25           | GSM8K            | MATH500          | AIME24            | AIME25             |
| SFT      | +2.18 $\pm$ 1.09 | +3.39 $\pm$ 0.99 | +10.01 $\pm$ 4.10 | +4.44 $\pm$ 9.77 | +2.01 $\pm$ 3.32 | +1.33 $\pm$ 0.85 | +10.00 $\pm$ 9.75 | +7.77 $\pm$ 8.59   |
| R1       | +1.79 $\pm$ 1.66 | +6.56 $\pm$ 1.94 | +8.89 $\pm$ 10.12 | +6.66 $\pm$ 6.94 | +1.73 $\pm$ 1.81 | +0.87 $\pm$ 0.50 | +6.66 $\pm$ 5.64  | +2.22 $\pm$ 8.59   |
| AdaRFT   | +0.83 $\pm$ 0.42 | +2.26 $\pm$ 2.01 | +5.56 $\pm$ 6.03  | +4.44 $\pm$ 2.92 | +2.54 $\pm$ 1.72 | +4.53 $\pm$ 4.12 | -1.11 $\pm$ 8.59  | +10.00 $\pm$ 20.30 |

Table 7: Paired improvements of **SPaCe** over baselines on the Knights and Knaves dataset using Qwen3-0.6B as the base LLM. Entries are mean  $\pm$  90% confidence interval over three shared random seeds (Student’s  $t$ -distribution,  $df = 2$ ).

| Baseline | 2                | 3                | 4                | 5                | 6                | 7                | 8                |
|----------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| R1       | +2.67 $\pm$ 4.25 | +4.33 $\pm$ 5.42 | +2.33 $\pm$ 2.58 | +1.67 $\pm$ 2.58 | +1.00 $\pm$ 0.00 | +0.00 $\pm$ 0.00 | +0.33 $\pm$ 0.97 |
| AdaRFT   | +2.67 $\pm$ 0.97 | +5.00 $\pm$ 6.08 | +3.33 $\pm$ 3.51 | +2.00 $\pm$ 0.00 | +1.33 $\pm$ 0.97 | +0.00 $\pm$ 0.00 | +0.33 $\pm$ 0.97 |

the MATH-Full dataset (Hendrycks et al., 2021), we also conduct clustering on the same dataset using our method (SPaCe) to ensure a fair comparison. It is worth noting that after data reduction, LIM retains approximately 1,400 training samples, whereas SPaCe selects only 5 representative clusters, corresponding to just 50 data points, which amounts to **merely 3% of the data used by LIM baseline**. Despite this drastic reduction, our method achieves competitive or superior results, demonstrating that SPaCe can reach high efficiency and effectiveness with a fraction of the training data. For these experiments, we perform training on 3 different seeds, and report in Table 10.

### A.6.1 LIM definition

LIM computes a per-sample score from its reward trajectory relative to the model’s average reward curve across epochs. Let  $r_i^k$  be the reward of sample  $i$  at epoch  $k$  and  $r_k = \frac{1}{N} \sum_{i=1}^N r_i^k$  the epoch-wise mean over all  $N$  samples for  $k = 1, \dots, K$ . The alignment score is

$$s_i = 1 - \frac{\sum_{k=1}^K (r_i^k - r_k)^2}{\sum_{k=1}^K (1 - r_k)^2}, \quad i = 1, \dots, N, \quad (14)$$

which normalizes the squared deviation of the sample trajectory from the epoch-wise mean. Data reduction is performed by thresholding:

$$\mathcal{D}_{\text{LIM}} = \{i : s_i > \theta\}. \quad (15)$$

### A.6.2 Results

We show the results on four mathematical reasoning datasets between our method and LIM using MATH as training data in Table 10. We select

Qwen2.5-0.5B-Instruct as the base LLM for training.

The results highlight the effectiveness of SPaCe compared to both the Base model and LIM. On GSM8K, SPaCe reaches 32.0%, which represents a relative improvement of +22% over the Base (26.3%) and still surpasses LIM (30.7%). On MATH500, SPaCe achieves 20.7%, outperforming LIM (20.3%) and the Base (20.0%), showing that our method yields more stable gains even on challenging competition-level problems. Notably, SPaCe is the only method that improves performance on the Olympiad benchmarks: it attains 1.1% on AIME24 and 3.3% on AIME25, while both the Base and LIM fail to make progress on these harder tasks.

Overall, these findings confirm that SPaCe not only provides consistent improvements on standard benchmarks such as GSM8K and MATH500, but also uniquely enhances generalization to the most challenging settings, where variance-based selection methods like LIM struggle. This demonstrates the robustness and efficiency of our cluster-based approach in leveraging limited training data for stronger downstream reasoning performance.

### A.7 Effect Of Diverse Sample Selection

We show the impact of sample selection strategies for the data reduction on the performance of different LLMs in Figure 8. Specifically, we compare our method against two baselines: *random*, which randomly selects training examples for the cluster, and *closest*, which selects the closest examples to the cluster center. As observed, selecting diverse examples with our method consistently yields the highest performance across four datasets. Inter-

Table 8: Paired accuracy improvements (%) of **SPaCe** over baselines on DeepScaleR-Easy and DeepScaleR-Difficult splits using Qwen3-0.6B-Instruct as the base LLM. Entries are mean  $\pm$  90% confidence interval over three shared random seeds (Student’s  $t$ -distribution,  $df = 2$ ).

| Baseline | DeepScaleR-Easy  |                  |                  |                  | DeepScaleR-Difficult |                  |                   |                   |
|----------|------------------|------------------|------------------|------------------|----------------------|------------------|-------------------|-------------------|
|          | GSM8K            | MATH500          | AIME24           | AIME25           | GSM8K                | MATH500          | AIME24            | AIME25            |
| R1       | +1.40 $\pm$ 1.36 | +2.13 $\pm$ 0.00 | +6.67 $\pm$ 9.77 | +5.56 $\pm$ 2.92 | +0.60 $\pm$ 0.96     | +0.40 $\pm$ 1.63 | +5.56 $\pm$ 14.14 | +6.67 $\pm$ 4.89  |
| AdaRFT   | +1.47 $\pm$ 0.91 | +3.40 $\pm$ 0.78 | +4.45 $\pm$ 5.78 | +2.22 $\pm$ 5.78 | +2.23 $\pm$ 1.15     | +2.07 $\pm$ 1.46 | +2.23 $\pm$ 9.22  | +7.78 $\pm$ 15.56 |

Table 9: Paired accuracy improvements (%) of **SPaCe** over **AdaRFT** using other base LLMs with DeepScaler-Uniform as the training data. Entries are mean  $\pm$  90% confidence interval over three shared random seeds (Student’s  $t$ -distribution,  $df = 2$ ).

| Baseline | Backbone              | GSM8K            | MATH500          | AIME24           | AIME25           |
|----------|-----------------------|------------------|------------------|------------------|------------------|
| AdaRFT   | Falcon3-1b-Instruct   | +2.09 $\pm$ 1.53 | +0.53 $\pm$ 1.28 | +0.00 $\pm$ 0.00 | +2.22 $\pm$ 3.24 |
| AdaRFT   | Llama-3.2-1B-Instruct | +1.04 $\pm$ 0.41 | +0.87 $\pm$ 2.70 | +0.00 $\pm$ 5.61 | +1.11 $\pm$ 3.24 |
| AdaRFT   | Qwen2.5-0.5B-Instruct | +2.40 $\pm$ 3.42 | +2.00 $\pm$ 3.36 | +2.20 $\pm$ 3.21 | +1.10 $\pm$ 3.21 |
| AdaRFT   | Qwen3-8B-Base         | +2.00 $\pm$ 2.03 | +2.47 $\pm$ 0.19 | +1.11 $\pm$ 3.24 | +0.00 $\pm$ 0.00 |

| Method | GSM8K                      | MATH500                    | AIME24                    | AIME25                    |
|--------|----------------------------|----------------------------|---------------------------|---------------------------|
| Base   | 26.3                       | 20.0                       | 0.0                       | 0.0                       |
| LIM    | 30.7 <sub>1.3</sub>        | 20.3 <sub>1.1</sub>        | 0.0 <sub>0.0</sub>        | 0.0 <sub>0.0</sub>        |
| SPaCe  | <b>32.0</b> <sub>0.8</sub> | <b>20.7</b> <sub>1.2</sub> | <b>1.1</b> <sub>1.9</sub> | <b>3.3</b> <sub>0.0</sub> |

Table 10: Mean<sub>std</sub> over 3 seeds on MATH benchmarks using Qwen2.5-0.5B-Instruct. Best in **bold**.

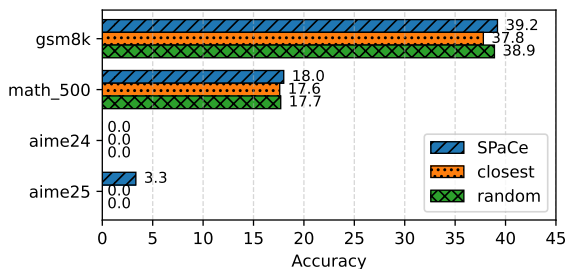


Figure 8: Comparison of selection strategies across datasets. Selecting diverse examples with SPaCe outperforms both random and closest baselines. Closest examples perform worse, likely due to reduced variety within each cluster.

estingly, the *closest* strategy performs worse than *random* in most cases. We hypothesize that this is because the examples nearest to the cluster center tend to be overly similar, thus failing to provide sufficient coverage and variation for effective learning.

### A.8 Impact Of Embedding Model Choice

To assess SPaCe’s robustness to different semantic embedding backbones, we compare its default sentence encoder with an alternative based on *Qwen2-1.5B-Instruct* (*Alibaba-NLP/gte-Qwen2-*

*1.5B-instruct*). We denote Qwen3<sup>1</sup> as the baseline using *Qwen3-Embedding-0.6B*, which is adopted in SPaCe, and Qwen2.5<sup>2</sup> as the baseline using *Alibaba-NLP/gte-Qwen2-1.5B-instruct*. Table 11 reports zero-shot performance on four math reasoning benchmarks. Results show that SPaCe yields nearly identical performance across both embedding models, with the Qwen3-based encoder showing a slight advantage. This plug-and-play flexibility demonstrates that any high-quality pre-trained encoder can be seamlessly integrated into our framework without retraining.

### A.9 Impact Of The Number Of PCA Component

In SPaCe, we first apply PCA to reduce the dimensionality of the latent vectors extracted from the pretrained Sentence-BERT model. By default we use 50 principal components; here, we vary this number to study its effect on final performance. Using Qwen3-0.6B as the base model and training on the DeepScaleR-uniform dataset, the results are shown in Table 12. We observe that smaller to moderate numbers of components (10 or 50) yield the best performance, whereas larger values (100 or 300) lead to a decline. We hypothesize that very high-dimensional embeddings overwhelm the difficulty signal prior to clustering, resulting in poorer downstream performance.

### A.10 Number Of Samples In Each Cluster

We vary the number of samples per cluster  $l$  to assess its impact, and present the results in Figure 9. As shown, performance peaks at our default setting

| Embedding            | GSM8K    | MATH500  | AIME24  | AIME25  |
|----------------------|----------|----------|---------|---------|
| Qwen3 <sup>1</sup>   | 32.9±0.9 | 22.0±0.7 | 2.2±1.9 | 1.1±1.9 |
| Qwen2.5 <sup>2</sup> | 31.6±0.6 | 21.0±1.2 | 1.1±1.9 | 0.0±0.0 |

Table 11: Results using DeepScaleR as training data for Qwen2.5-0.5B-Instruct, evaluated with two embedding models: Qwen3<sup>1</sup> and Qwen2.5<sup>2</sup>.

| Number of PCA Components | GSM8K | MATH500 | AIME24 | AIME25 | Average |
|--------------------------|-------|---------|--------|--------|---------|
| 10                       | 79.2  | 74.4    | 13.3   | 20.0   | 46.7    |
| 50                       | 80.0  | 78.9    | 23.3   | 13.3   | 48.9    |
| 100                      | 79.8  | 76.6    | 6.7    | 13.3   | 44.1    |
| 300                      | 79.0  | 75.8    | 16.7   | 10.0   | 45.4    |

Table 12: Results on 1 same seed on the DeepScaleR-uniform dataset using Qwen3-0.6B as the base model with different number of PCA components.

of  $l = 10$ . In contrast, using  $l = 1$  yields lower performance, likely because the selected examples lack sufficient diversity to provide a strong learning signal. Larger values of  $l \in [100, 300]$  also lead to degraded performance, which we attribute to increased randomness when sampling too many examples per arm, especially when  $l \gg B$ . Due to compute limits, we could not extensively tune  $l$  between 10–100, where better results might be possible. Overall, setting  $l \sim B$  achieves the most reasonable results.

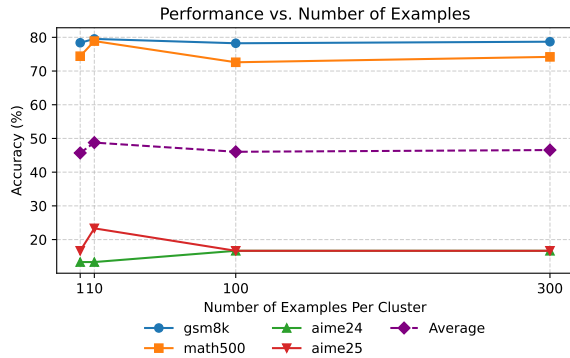


Figure 9: Results for 1 seed with Qwen3-0.6B and different number of samples per cluster.

### A.11 Effect of Removing Difficulty in Clustering on Performance

To isolate the contribution of the per-example difficulty attribute to cluster quality, the clustering pipeline is repeated after ablating this signal (i.e., omitting the difficulty attribute). Results are reported in Figure 10, which shows downstream performance of *Qwen3-0.6B* on each dataset under

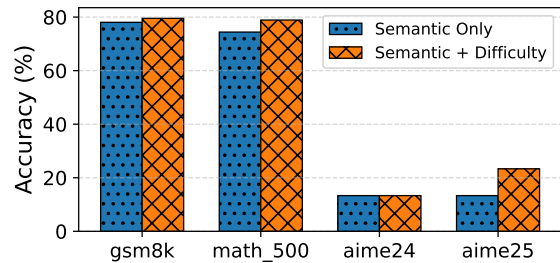


Figure 10: Effect of Removing Difficulty in Clustering on Performance.

this “no-difficulty” condition. Across benchmarks, performances consistently drop when the difficulty attribute is excluded, indicating that this signal is important for forming meaningful clusters and improving curriculum selection.

### A.12 Additional Related Work: Reinforcement Fine-Tuning for Language Models

Due to page limit in the main paper, we include the additional related work in the Reinforcement Fine-Tuning methods for Language Models here.

Language Models can be formulated as sequential decision-making agents, enabling the application of RL techniques for fine-tuning. Proximal Policy Optimization (PPO) (Schulman et al., 2017) has been widely adopted in early RLHF pipelines due to its balance between stability and sample efficiency. More recent work introduced actor-only alternatives such as REINFORCE++ (Hu, 2025) and Group Relative Policy Optimization (GRPO) (DeepSeek-AI, 2025), which eliminate the need

for value networks and have shown strong performance on large language models, particularly in reasoning tasks. By avoiding a separately trained critic, these approaches simplify optimization, reduce variance in policy updates, and mitigate instability caused by poorly estimated value functions. GRPO, in particular, has been successfully deployed in large-scale instruction tuning setups where explicit reward modeling is either impractical or misaligned with target behaviors. Instead of depending on hand-crafted reward models, GRPO leverages group-based relative comparisons across sampled trajectories, thereby aligning the optimization signal with preference-style supervision. This actor-only paradigm aligns naturally with recent trends in LLM alignment, where scalability, reduced computational overhead, and robustness to noisy feedback are critical. These methods represent a shift from critic-dependent RLHF pipelines toward lightweight, actor-centric algorithms that better match the scale and complexity of modern LLM training regimes.

### A.13 Selected Training Samples Difficulty

We analyze the selected training examples to understand how SPaCe constructs its curriculum. Figure 11 shows the difficulty distribution of questions selected on the DeepScaleR-uniform dataset. SPaCe consistently chooses examples across the full difficulty range—from easy (near 0) to hard (near 100)—ensuring balanced coverage. This diversity enables training on a broad range of problems, avoiding overfitting to simple or complex cases. Notably, this balance emerges without manual difficulty constraints, highlighting the effectiveness of SPaCe’s clustering and selection strategy.

### A.14 Deepscaler Subset Distributions

We provide the difficulty score distributions of three DeepScaleR subsets: DeepScaleR Uniform, DeepScaleR Easy, and DeepScaleR Difficult, as shown in Figure 12. Each subset exhibits distinct difficulty characteristics, reflecting the varying levels of challenge present in the data. The distributions are grouped into bins of size 10, allowing for a clear comparison of how problem difficulty varies across these subsets. In particular, the Uniform subset spans the entire difficulty range with roughly balanced coverage, making it suitable for general-purpose training and evaluation. By contrast, the Easy subset is concentrated heavily in the lower-difficulty bins, highlighting its role in

providing simpler problems for warm-up training or curriculum learning. Meanwhile, the Difficult subset skews strongly toward the higher-difficulty bins, offering more challenging samples that are valuable for stress-testing reasoning capabilities and benchmarking advanced methods. Together, these subsets provide complementary perspectives on model performance across a wide spectrum of difficulty levels, ensuring a more comprehensive assessment of reasoning ability.

### A.15 Cluter Analysis

In this section, to provide further insight into what happens during the clustering phase of our framework, we analyze several representative settings to examine both what is captured for training and how clustering shapes the overall behavior of our approach. Specifically, we investigate how problem embeddings are grouped and how these clusters align with meaningful curricular categories. For this purpose, we employ GPT-5’s API (OpenAI, 2025) to categorize each problem into one of the seven canonical subject areas defined by Hendrycks et al. (2021), namely: Prealgebra, Algebra, Number Theory, Counting & Probability, Geometry, Intermediate Algebra, and Precalculus. This taxonomy is consistent with the Art of Problem Solving (AoPS) curriculum (Art of Problem Solving), which provides a widely accepted structure for organizing mathematical problem-solving skills.

- **Prealgebra:** Covers arithmetic foundations, including fractions, decimals, percents, ratios, proportions, and basic number properties. It also introduces simple equations and word problems.
- **Algebra:** Focuses on symbolic manipulation and equations, such as linear and quadratic equations, inequalities, systems of equations, factoring, functions, and exponents. It marks the transition from arithmetic to general algebraic reasoning.
- **Number Theory:** Includes topics such as divisibility, prime numbers, greatest common divisors, modular arithmetic, congruences, and Diophantine equations. Problems emphasize reasoning about integer structure and properties.
- **Counting & Probability:** Encompasses combinatorics and elementary probability, includ-

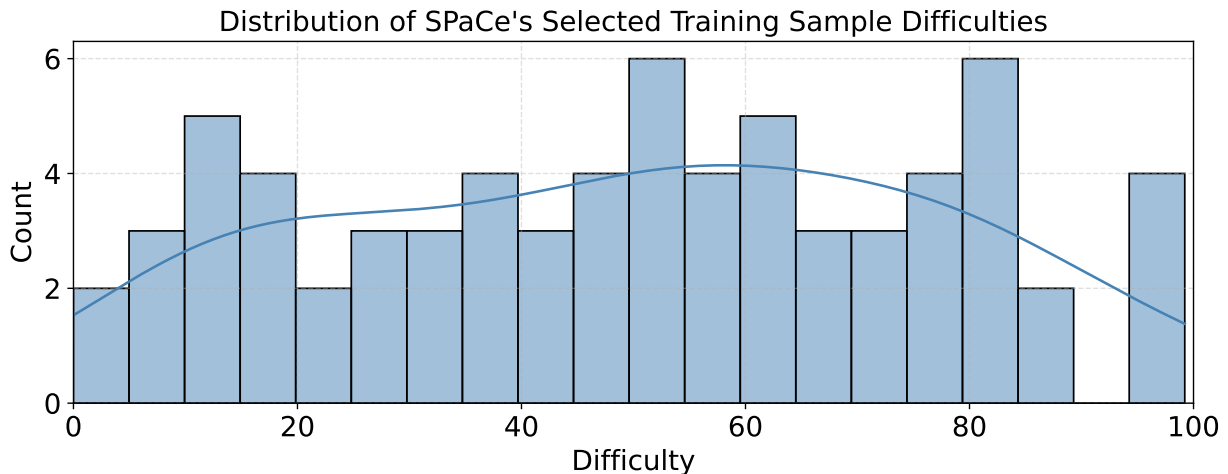


Figure 11: Difficulty distribution of training examples in SPaCe.

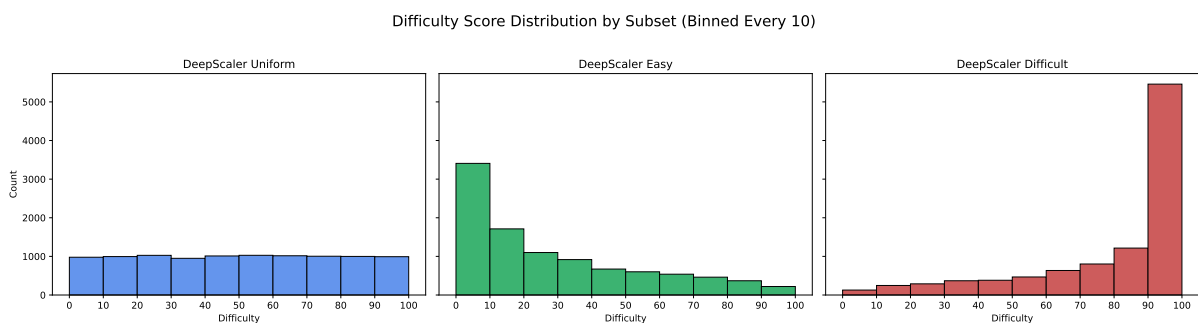


Figure 12: DeepScaleR subsets' difficulty distributions.

ing permutations, combinations, casework, binomial coefficients, expected value, and probabilistic reasoning.

- **Geometry:** Centers on Euclidean geometry of lines, angles, triangles, quadrilaterals, circles, and polygons. Topics include similarity, congruence, area, volume, coordinate geometry, and introductory trigonometric methods.
- **Intermediate Algebra:** Extends algebra with higher-level topics such as polynomials, rational functions, complex numbers, inequalities, logarithmic and exponential functions, and sequences/series.
- **Precalculus:** Prepares for calculus through trigonometry, advanced functions, polar/parametric representations, vectors, and deeper study of sequences and series.

### A.15.1 DeepScaleR Dataset

We analyze the *DeepScaleR* dataset under the configuration with  $K=7$  clusters induced by Qwen3-0.6B-Embeddings, consistent with Table 15. Figure 12 visualizes one representative

run. The resulting partition exhibits intuitive curricular structure: a majority of items fall into *Pre-algebra* (51.4%), with *Geometry* (18.6%) and *Algebra* (12.9%) also comprising substantial shares; by contrast, smaller categories such as *Number Theory* (1.4%) and *Intermediate Algebra* (1.4%) are comparatively scarce, with the remaining mass distributed across *Counting & Probability* and *Precalculus*.

Beyond mirroring topical prevalence in the underlying corpus, this distribution suggests that the embedding-driven clustering is aligned with both latent difficulty and high-level curricular distinctions. Practically, this yields two benefits for downstream training: (i) it avoids over-emphasizing any single subject area, providing balanced exposure across topics; and (ii) it simplifies scheduling, since strata can be sampled in a principled way (e.g., uniformly or by difficulty-aware policies) without ad-hoc reweighting to correct for cluster idiosyncrasies. In short, even though clusters are formed in representation space, they preserve pedagogically meaningful boundaries that support stable and fair curriculum design.

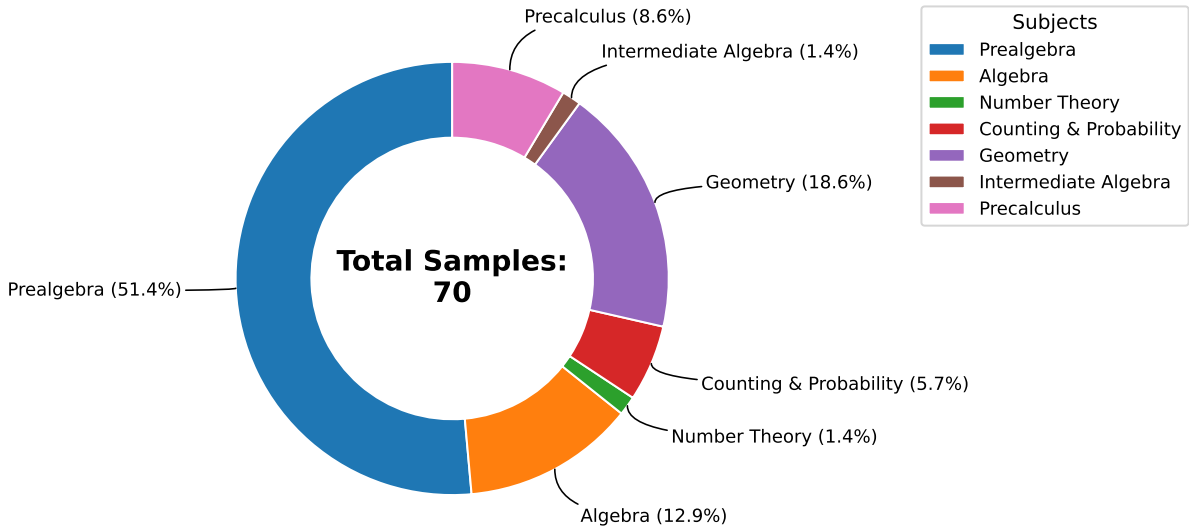


Figure 13: Selected data subjects (70 samples) using Qwen3-0.6B-Embedding. The data is clustered from DeepScaleR-uniform set.

### A.15.2 Knights-and-Knaves Dataset

We conduct an analogous analysis on the *Knights-and-Knaves* dataset using  $K=7$  clusters obtained with Qwen3-0.6B-Embeddings. The results are shown in Figure 14. Plotting the distribution of selected questions by the number of people per instance reveals two robust regularities across all three training seeds: (i) no 2-person questions are selected; and (ii) aside from the 7-person category, all remaining categories contain the same number of questions. The same symmetry appears in the other seeds, indicating that the clustering process is not only semantically coherent but also structurally consistent with a salient, coarse-grained attribute (the number of entities in the prompt).

These regularities have useful practical consequences. First, they provide balanced coverage over interaction sizes, preventing the curriculum from drifting toward a single conversational complexity. Second, they reduce confounds in subsequent evaluation and scheduling: because most categories are equalized, one can adopt simple, uniform sampling or layer a performance-aware scheduler on top without introducing artifacts from cluster imbalance. We reckon the persistent absence of 2-person items likely reflects a combination of dataset composition and our selection protocol’s preference for more discriminative examples.

### A.16 Training Time

To evaluate computational efficiency, we measure wall-clock training time when applying AdaRFT

and our proposed SPaCe across five representative base models: *Qwen3-0.6B*, *Falcon*, *Llama3*, *Qwen2.5*, and *Qwen3-8B*. These models span a range of parameter scales and architectures, providing a balanced testbed for assessing runtime behavior under different backbone choices. All runs are conducted under identical hardware and data conditions to ensure a fair comparison. We also note that both methods require precomputed hardness scores; we therefore follow the same hardness-estimation protocol as AdaRFT (Shi et al., 2025) for all methods and report only the training time here.

Figure 15 reports the results. Across all models, SPaCe consistently reduces wall-clock training time relative to AdaRFT, with per-model savings ranging from 2.1% to 10.8% (e.g., Qwen2.5: −66 minutes, −10.8%; Qwen3-8B: −85 minutes, −5.6%; Llama3: −40 minutes, −7.4%; Qwen3-0.6B: −23 minutes, −2.4%; Falcon: −14 minutes, −2.1%).

We attribute these savings to SPaCe’s clustering phase, which dynamically prioritizes clusters that yield higher learning signal over a reduced selection space, thereby avoiding wasted updates on redundant or low-yield samples. Although the absolute magnitude of savings depends on the underlying backbone, the improvements are consistent across diverse architectures, highlighting that SPaCe not only improves data efficiency but also offers a practical reduction in training cost without additional engineering or inference-time overhead.

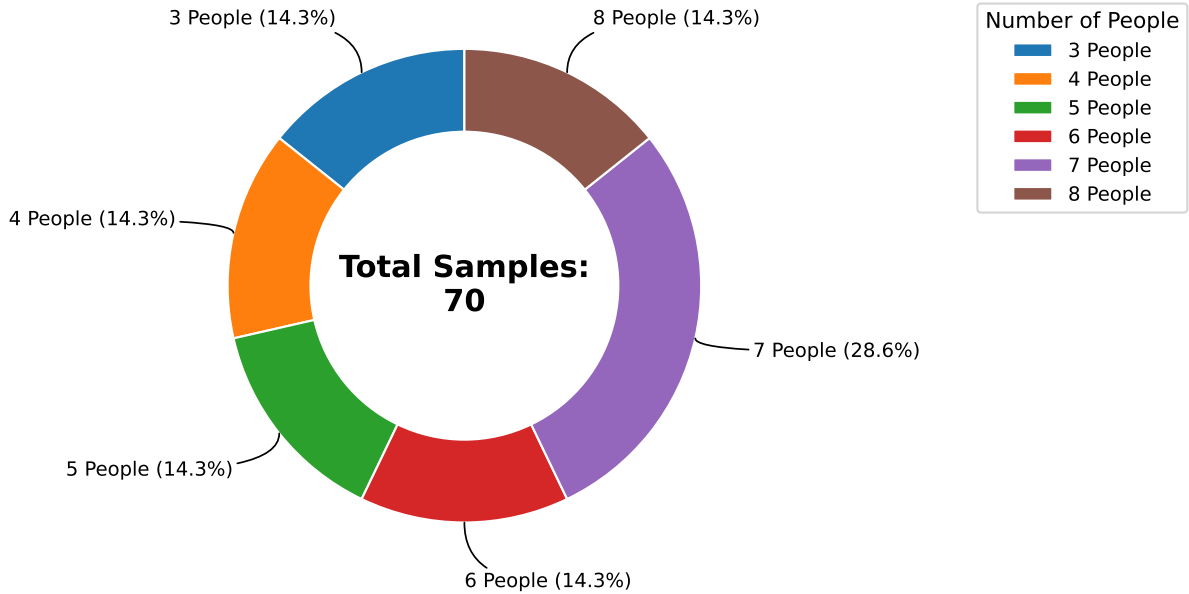


Figure 14: Selected data subjects (70 samples) using Qwen3-0.6B-Embedding. The data is clustered from Knights and Knaves set.

## A.17 SPaCe training details

### A.17.1 Training Hyperparameters

**General Training Parameters** In this section, we provide the training details of SPaCe in Table 13. These parameters apply for full training LLMs (which excludes the training of *Qwen3-8B-Base*).

**LoRA Training Parameters** In this section, we provide the LoRA training parameters for *Qwen3-8B-Base*. All the parameters used are reported in Table 14.

### A.17.2 Number of clusters

We provide the number of clusters used for different settings of SPaCe in Table 15. While the optimal number of clusters varies across datasets, it remains within a moderate range, consistent with our observations in Section 6.3.

### A.17.3 Model and data references

We list the links to the LLM models and datasets in Table 16.

### A.17.4 System Prompt

Following (Hugging Face, 2025), the system prompt asks the model to generate the answer with clear requirements, with reasoning and answer following the format, as described in Figure 16.

## A.18 Response Examples

We present several sampled responses of SPaCe in Table 17 and 18.

### A.19 Selected Example Analysis

In this section, we provide details on the selected examples by SPaCe. For each cluster, we show 2 selected examples selected by our method using *Qwen3-0.6B*, with the DeepScaleR-uniform dataset. We select the examples from the setting with total number of clusters equal 7. The examples are shown from Table 19 to Table 25.

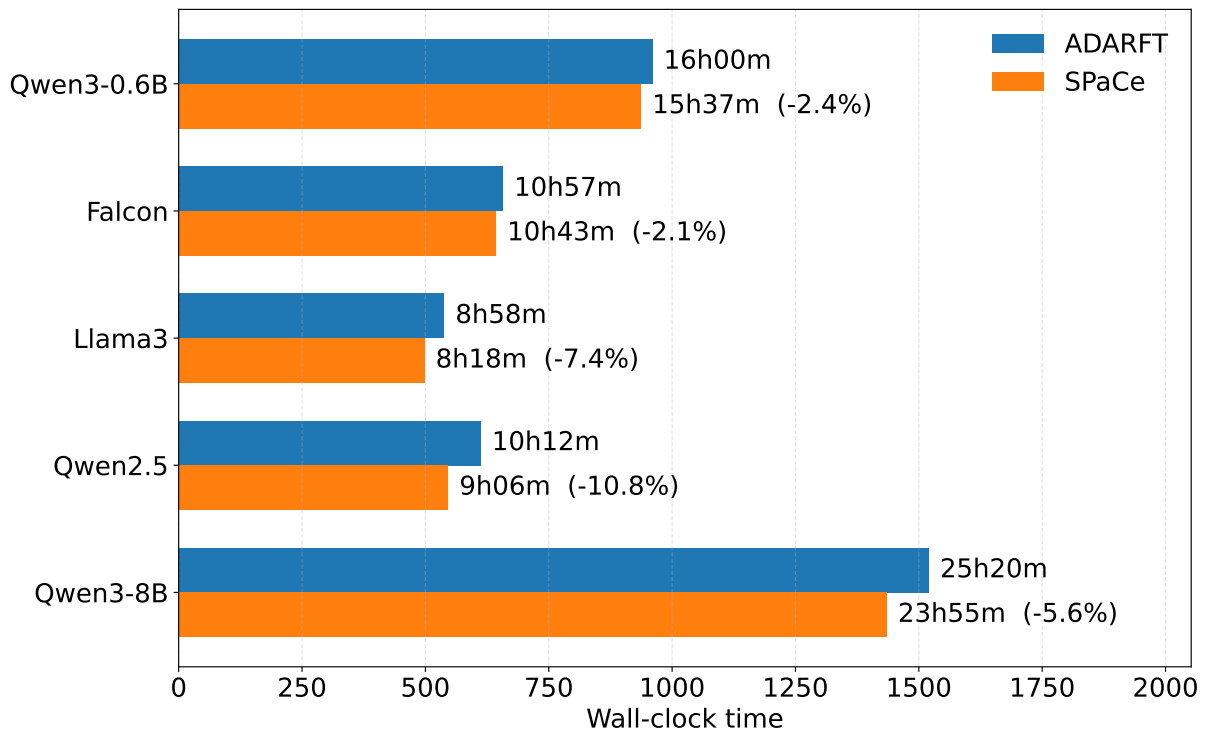


Figure 15: Training time comparison between AdaRFT and SPaCe.

**SYSTEM PROMPT**

You are a helpful assistant. Please reason step by step, and put your final answer within `\boxed{}`.

Figure 16: System prompt used in our experiments.

| Parameters   | Value     |
|--|-----------|
| Number of consecutive steps for penalizing ( $T_{consecutive}$ ) | 10        |
| Number of examples per cluster ( $l$ )                           | 10        |
| Number of PCA components   | 50        |
| Batch size (B)   | 8         |
| Number of generation per step (G)                                | 8         |
| Maximum completion length (L)                                    | 1200      |
| Initial learning rate ( $\alpha$ )                               | $5e^{-6}$ |
| Weight Decay   | 0.1       |
| Warmup Ratio   | 0.1       |
| lr_scheduler_type  | cosine    |
| Adam $\beta_1$   | 0.9       |
| Adam $\beta_2$   | 0.99      |
| bf16   | True      |
| Per device train batch size                                      | 8         |
| Gradient accumulation steps                                      | 8         |
| Max grad norm ( $G_{norm}$ )                                     | 0.1       |
| $\epsilon$   | $1e^{-6}$ |

Table 13: Parameters used in SPaCe.

| Parameters                   | Value                |
|------------------------------|----------------------|
| All Parameters               | 8,194,569,216        |
| Trainable Parameters ( $l$ ) | 3,833,856            |
| Trainable %                  | 0.05                 |
| Rank                         | 8                    |
| LoRA $\alpha$                | 16                   |
| Target Modules               | ["q_proj", "v_proj"] |
| LoRA Dropout                 | 0.1                  |
| Bias                         | None                 |

Table 14: Parameters used for Low-rank Adaptation (LoRA) Fine-tuning.

| Model                         | Train dataset        | Number of clusters |
|-------------------------------|----------------------|--------------------|
| Qwen3-0.6B                    | DeepScaleR-uniform   | 7                  |
|                               | DeepScaleR-easy      | 8                  |
|                               | DeepScaleR-difficult | 10                 |
|                               | GSM8K                | 10                 |
| DeepSeek-R1-Distill-Qwen-1.5B | DeepScaleR-uniform   | 7                  |
| Falcon3-1B-Instruct           | DeepScaleR-uniform   | 6                  |
| Llama-3.2-1B-Instruct         | DeepScaleR-uniform   | 8                  |
| Qwen2.5-0.5B-Instruct         | DeepScaleR-uniform   | 7                  |
| Qwen3-8B-Base                 | DeepScaleR-uniform   | 7                  |

Table 15: Number of clusters used for different settings of our method.

| <b>Models/Datasets</b>                  | <b>URL</b>  |
|---|---|
| Qwen3-Embedding-0.6B                    | <a href="https://huggingface.co/Qwen/Qwen3-Embedding-0.6B">https://huggingface.co/Qwen/Qwen3-Embedding-0.6B</a>   |
| Qwen2.5-0.5B-Instruct                   | <a href="https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct">https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct</a>   |
| Llama3.2-1B-Instruct                    | <a href="https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct">https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct</a>                                 |
| Falcon3-1B-Instruct                     | <a href="https://huggingface.co/tiiuae/Falcon3-1B-Instruct">https://huggingface.co/tiiuae/Falcon3-1B-Instruct</a>   |
| Alibaba-NLP/<br>gte-Qwen2-1.5B-instruct | <a href="https://huggingface.co/Alibaba-NLP/gte-Qwen2-1.5B-instruct">https://huggingface.co/Alibaba-NLP/gte-Qwen2-1.5B-instruct</a>                           |
| DeepScaleR                              | <a href="https://huggingface.co/datasets/agentica-org/DeepScaleR-Preview-Dataset">https://huggingface.co/datasets/agentica-org/DeepScaleR-Preview-Dataset</a> |
| GSM8K                                   | <a href="https://huggingface.co/datasets/openai/gsm8k">https://huggingface.co/datasets/openai/gsm8k</a>   |
| MATH-500                                | <a href="https://huggingface.co/datasets/HuggingFaceH4/MATH-500">https://huggingface.co/datasets/HuggingFaceH4/MATH-500</a>                                   |
| AIME24                                  | <a href="https://huggingface.co/datasets/math-ai/aime24">https://huggingface.co/datasets/math-ai/aime24</a>   |
| AIME25                                  | <a href="https://huggingface.co/datasets/math-ai/aime25">https://huggingface.co/datasets/math-ai/aime25</a>   |

Table 16: Models and Datasets Details.

Table 17: Qwen3-0.6B fine-tuned with SPaCe responses.

**Question**

Let  $f$  be the function defined by  $f(x) = ax^2 - \sqrt{2}$  for some positive  $a$ . If  $f(f(\sqrt{2})) = -\sqrt{2}$  then  $a =$ . Please reason step by step, and put your final answer within  $\boxed{\quad}$ .

**Answer**

To solve for  $a$ , we need to find the value of  $a$  such that  $f(f(\sqrt{2})) = -\sqrt{2}$ , where  $f(x) = ax^2 - \sqrt{2}$ . First, let's compute  $f(\sqrt{2})$ :

$$f(\sqrt{2}) = a(\sqrt{2})^2 - \sqrt{2} = a(2) - \sqrt{2} = 2a - \sqrt{2}$$

Now, we need to compute  $f(f(\sqrt{2}))$ :

$$f(f(\sqrt{2})) = f(2a - \sqrt{2}) = a(2a - \sqrt{2})^2 - \sqrt{2}$$

Let's expand the square term:

$$(2a - \sqrt{2})^2 = (2a)^2 - 2(2a)(\sqrt{2}) + (\sqrt{2})^2 = 4a^2 - 4a\sqrt{2} + 2$$

...

So the discriminant is zero, which means there is a repeated root:

$$a = \frac{2\sqrt{2}}{4} = \frac{\sqrt{2}}{2}$$

Therefore, the value of  $a$  is  $\frac{\sqrt{2}}{2}$ .

Let me double-check the calculations to ensure there are no errors. Starting from the beginning:

1.  $f(\sqrt{2}) = a(2) - \sqrt{2} = 2a - \sqrt{2}$
2. Then  $f(f(\sqrt{2})) = f(2a - \sqrt{2}) = a(2a - \sqrt{2})^2 - \sqrt{2}$
3. Expanding  $(2a - \sqrt{2})^2$  gives  $4a^2 - 4a\sqrt{2} + 2$
4. Multiplying by  $a$  and subtracting  $\sqrt{2}$  gives  $4a^3 - 4a^2\sqrt{2} + 2a - \sqrt{2}$
5. Setting equal to  $-\sqrt{2}$  leads to the equation  $4a^3 - 4a^2\sqrt{2} + 2a = 0$
6. Factoring gives  $2a(2a^2 - 2a\sqrt{2} + 1) = 0$ , leading to the quadratic with discriminant zero.

Yes, the steps seem correct. The discriminant being zero implies a repeated root, which gives  $a = \frac{\sqrt{2}}{2}$ .

Therefore, the answer is  $\boxed{\frac{\sqrt{2}}{2}}$ .

**\*\*Final Answer\*\***

$$\boxed{\frac{\sqrt{2}}{2}}$$

Table 18: Qwen3-0.6B fine-tuned with SPaCe responses.

|  |
|--|
| <p><b>Question</b><br/> A set <math>S</math> of points in the <math>xy</math>-plane is symmetric about the origin, both coordinate axes, and the line <math>y = x</math>. If <math>(2, 3)</math> is in <math>S</math>, what is the smallest number of points in <math>S</math>? Please reason step by step, and put your final answer within <math>\boxed{\quad}</math>.</p> <p><b>Answer</b><br/> To solve this problem, we need to determine the smallest number of points in the set <math>S</math>, given that the set is symmetric about the origin, both coordinate axes, and the line <math>y = x</math>. Additionally, the point <math>(2, 3)</math> is in <math>S</math>.</p> <p>—</p> <p>### Step 1: Understand the Symmetry Conditions<br/> A set of points that is symmetric about the <b>origin</b> means that for every point <math>(x, y)</math> in <math>S</math>, <math>(-x, -y)</math> is also in <math>S</math>.<br/> Symmetry about the <b>coordinate axes</b> means that if <math>(x, y)</math> is in <math>S</math>, then <math>(y, x)</math> is also in <math>S</math>.<br/> Symmetry about the <b>line <math>y = x</math></b> means that if <math>(x, y)</math> is in <math>S</math>, then <math>(y, x)</math> is also in <math>S</math>.</p> <p style="text-align: center;">...</p> <p>Is there a point that we might be missing?<br/> Let's check if <math>(3, 2)</math> is symmetric to itself. Yes, it is.<br/> So, with these three points, we have covered all the symmetries.<br/> But wait! What about the point <math>(2, 3)</math>? Is there a point symmetric to it that we haven't considered?<br/> No, because we already considered all the points under the given symmetries.<br/> So, the minimal number of points in <math>S</math> is 3.</p> <p>—</p> <p>### Final Answer:</p> <p style="text-align: center;"><math>\boxed{3}</math></p> |
|--|

Table 19: Cluster 0 examples. Only 2 examples are shown for illustration.

|  |
|--|
| <p><b>Example 1</b><br/> <b>Prompt</b><br/> (In the preliminaries of optimal method and experimental design) When using the 0.618 method to find the optimal amount to add in an experiment, if the current range of excellence is <math>[628, 774]</math> and the good point is 718, then the value of the addition point for the current experiment is _____. Please reason step by step, and put your final answer within <math>\boxed{\quad}</math>.</p> <p><b>Answer</b><br/> 684</p> |
| <p><b>Example 2</b><br/> <b>Prompt</b><br/> Calculate the probability that in a deck of 52 cards, the second card has a different suit than the first, and the third card has a different suit than the first and second. Please reason step by step, and put your final answer within <math>\boxed{\quad}</math>.</p> <p><b>Answer</b><br/> <math>\frac{169}{425}</math></p>  |

Table 20: Cluster 1 examples. Only 2 examples are shown for illustration.

|  |
|--|
| <p><b>Example 1</b></p> <p><b>Prompt</b></p> <p>Calculate: <math>\frac{\cos 190^\circ(1+\sqrt{3}\tan 10^\circ)}{\sin 290^\circ\sqrt{1-\cos 40^\circ}} = \underline{\hspace{2cm}}</math>. Please reason step by step, and put your final answer within <math>\boxed{\hspace{2cm}}</math>.</p> <p><b>Answer</b></p> <p><math>2\sqrt{2}</math></p>            |
| <p><b>Example 2</b></p> <p><b>Prompt</b></p> <p>Let <math>a_n</math> be the number of <math>n</math>-digit numbers formed using only digits 1 and 2 such that no two adjacent digits are both 2. Find <math>a_5</math>. Please reason step by step, and put your final answer within <math>\boxed{\hspace{2cm}}</math>.</p> <p><b>Answer</b></p> <p>13</p> |

Table 21: Cluster 2 examples. Only 2 examples are shown for illustration.

|   |
|---|
| <p><b>Example 1</b></p> <p><b>Prompt</b></p> <p>You have 5 red balls and 5 blue balls in a box. You randomly draw 4 balls without replacement. What is the probability that exactly 2 red balls are drawn? Please reason step by step, and put your final answer within <math>\boxed{\hspace{2cm}}</math>.</p> <p><b>Answer</b></p> <p><math>\frac{25}{63}</math></p> |
| <p><b>Example 2</b></p> <p><b>Prompt</b></p> <p>If <math>a</math> and <math>b</math> are real numbers such that <math>a^2 + b^2 = 1</math>, what is the maximum value of <math>ab</math>? Please reason step by step, and put your final answer within <math>\boxed{\hspace{2cm}}</math>.</p> <p><b>Answer</b></p> <p><math>\frac{1}{2}</math></p>                    |

Table 22: Cluster 3 examples. Only 2 examples are shown for illustration.

|   |
|---|
| <p><b>Example 1</b></p> <p><b>Prompt</b></p> <p>Solve for <math>x</math>: <math>\log_3(x^2 - 1) = 2</math>. Please reason step by step, and put your final answer within <math>\boxed{\hspace{2cm}}</math>.</p> <p><b>Answer</b></p> <p>4</p>           |
| <p><b>Example 2</b></p> <p><b>Prompt</b></p> <p>Evaluate the integral <math>\int_0^1 xe^x dx</math>. Please reason step by step, and put your final answer within <math>\boxed{\hspace{2cm}}</math>.</p> <p><b>Answer</b></p> <p><math>e - 2</math></p> |

Table 23: Cluster 4 examples. Only 2 examples are shown for illustration.

|   |
|---|
| <p><b>Example 1</b></p> <p><b>Prompt</b><br/>If <math>\sin x + \cos x = \sqrt{2}</math>, find the value of <math>\sin^4 x + \cos^4 x</math>. Please reason step by step, and put your final answer within <math>\boxed{\quad}</math>.</p> <p><b>Answer</b><br/><math>\frac{3}{4}</math></p> |
| <p><b>Example 2</b></p> <p><b>Prompt</b><br/>Find the sum of the series <math>\sum_{n=1}^{\infty} \frac{1}{n(n+1)}</math>. Please reason step by step, and put your final answer within <math>\boxed{\quad}</math>.</p> <p><b>Answer</b><br/>1</p>  |

Table 24: Cluster 5 examples. Only 2 examples are shown for illustration.

|   |
|---|
| <p><b>Example 1</b></p> <p><b>Prompt</b><br/>How many 4-digit numbers are there such that no two adjacent digits are the same? Please reason step by step, and put your final answer within <math>\boxed{\quad}</math>.</p> <p><b>Answer</b><br/>5832</p>   |
| <p><b>Example 2</b></p> <p><b>Prompt</b><br/>If <math>A = \{1, 2, 3, 4\}</math> and <math>B = \{3, 4, 5, 6\}</math>, what is <math>A \cup B</math>? Please reason step by step, and put your final answer within <math>\boxed{\quad}</math>.</p> <p><b>Answer</b><br/><math>\{1, 2, 3, 4, 5, 6\}</math></p> |

Table 25: Cluster 6 examples. Only 2 examples are shown for illustration.

|   |
|---|
| <p><b>Example 1</b></p> <p><b>Prompt</b><br/>What is the value of the determinant of the matrix <math>\begin{bmatrix} 1 &amp; 2 \\ 3 &amp; 4 \end{bmatrix}</math>? Please reason step by step, and put your final answer within <math>\boxed{\quad}</math>.</p> <p><b>Answer</b><br/>-2</p> |
| <p><b>Example 2</b></p> <p><b>Prompt</b><br/>Simplify: <math>(2x - 3)^2 - (x + 1)^2</math>. Please reason step by step, and put your final answer within <math>\boxed{\quad}</math>.</p> <p><b>Answer</b><br/><math>3x^2 - 14x + 8</math></p>   |