Contents

A Reproducibility

Our code repository is available at `https://anonymous.4open.science/r/Data_Efficiency_LLM_RL-1846/`.

B Discussion

B.1 Limitations and Future Work

Our adaptive difficulty prediction framework currently relies on randomly sampling a reference set of K questions at each selection step. While effective, the quality of the reference set can influence prediction performance. In principle, one could improve prediction performance by selecting a more diverse reference set that better covers the dataset. Building on this idea, a natural extension is to fix a shared set of K reference questions (with sufficient coverage) across training, re-evaluating their adaptive difficulty at each selection step.

Moreover, while we demonstrate the effectiveness of experience replay in the GRPO setting, our current strategy is relatively straightforward: we randomly replay rollouts associated with questions whose average reward across all rollouts is neither 0 nor 1. A promising direction for further improving efficiency is to incorporate more principled strategies, such as drawing inspiration from prioritized experience replay [19, 26].

B.2 Extended Related Work

RL fine-tuning of LLMs (with verifiable rewards) has recently attracted significant attention, driven in part by the success of DeepSeek-R1 [4]. Compared to the original GRPO algorithm [20], recent work has proposed several algorithmic improvements: DAPO [24] introduces techniques such as clip-higher, dynamic sampling, token-level policy gradient loss, and overlong reward shaping; Dr. GRPO [15] removes the length and standard deviation normalization terms to improve stability. Beyond these algorithmic enhancements, Vojnovic and Yun [21] and Mroueh [18] provide theoretical insights into GRPO, while Zeng et al. [25] and Yeo et al. [23] conduct large-scale empirical studies across models, sharing key design choices that enable RL fine-tuning.

In contrast, relatively little attention has been paid to data-centric approaches. LIMR [13] explores a static data selection strategy for RL fine-tuning by prioritizing samples based on their alignment with the policy's learning trajectory. However, it requires a full training run over the entire dataset beforehand, limiting its practicality. Our online data selection method `DOTS` is more efficient and applicable in realistic settings. In addition, prior work has not explored the use of rollout replay in GRPO, which we show can further reduce training costs.

C Proofs

Proof of Theorem 1. We restate Theorem 1 and provide a complete proof below.

Theorem 1 (Maximal Gradient Signal at 50% Success Rate)**.** *Consider a single question q, where G responses $\{o_i\}_{i=1}^{G}$ are sampled independently from the current policy $\pi_\theta(\cdot \mid q)$. Each response receives a binary reward $r_i \in \{0, 1\}$, sampled i.i.d. from a Bernoulli(p) distribution, where p represents the reward success rate. Define the group-relative advantage $\hat{A}_i$ as in Eq. 1. We consider the unclipped policy gradient estimator for this question without KL penalty:*

$$g = \sum_{i=1}^{G} \hat{A}_i \nabla_\theta \log \pi_\theta(o_i \mid q).$$

Assume that the likelihood gradients $\nabla_\theta \log \pi_\theta(o_i \mid q)$ are independent of the reward distribution parameter p and have bounded variance. Then, the expected squared norm of the gradient satisfies:

$$\mathbb{E}[\|g\|^2] \propto p(1-p) \cdot (1 - 1/G)\,,$$

and is maximized when $p = 0.5$.

Proof. Let $r_i \in \{0, 1\}$ be the binary reward for response o_i, sampled i.i.d. from a Bernoulli(p) distribution. Define the group-relative advantage as:

$$\hat{A}_i = r_i - \frac{1}{G}\sum_{j=1}^{G} r_j.$$

We aim to analyze the expected squared norm of the gradient estimator

$$g = \sum_{i=1}^{G} \hat{A}_i \nabla_\theta \log \pi_\theta(o_i \mid q).$$

Assume that the gradients $\nabla_\theta \log \pi_\theta(o_i \mid q)$ are independent of the rewards $\{r_i\}$ and are independent across i, with bounded second moment:

$$\mathbb{E}[\|\nabla_\theta \log \pi_\theta(o_i \mid q)\|^2] \leq C < \infty.$$

Because $\hat{A}_i$ and $\hat{A}_j$ are correlated, we compute the full second moment, where the expectation is taken with respect to π_θ:

$$\mathbb{E}[\|g\|^2] = \sum_{i,j=1}^{G} \mathbb{E}[\hat{A}_i \hat{A}_j] \cdot \mathbb{E}[\nabla_\theta \log \pi_\theta(o_i \mid q)^\top \nabla_\theta \log \pi_\theta(o_j \mid q)].$$

By assumption, the log-likelihood gradients are zero-mean, independent, and identically distributed across i:

$$\mathbb{E}[\nabla_\theta \log \pi_\theta(o_i \mid q)^\top \nabla_\theta \log \pi_\theta(o_j \mid q)] = \begin{cases} V, & i = j, \\ 0, & i \neq j. \end{cases}$$

So,

$$\mathbb{E}[\|g\|^2] = V \cdot \sum_{i=1}^{G} \mathbb{E}[\hat{A}_i^2].$$

We now compute $\mathbb{E}[\hat{A}_i^2]$. Let $\bar{r} := \frac{1}{G}\sum_{j=1}^{G} r_j$, then:

$$\mathbb{E}[\hat{A}_i^2] = \mathbb{E}[(r_i - \bar{r})^2] = \mathrm{Var}(r_i - \bar{r}) = \mathrm{Var}(r_i) + \mathrm{Var}(\bar{r}) - 2\,\mathrm{Cov}(r_i, \bar{r}).$$

Since $r_i \sim$ Bernoulli(p) and r_j are i.i.d.,

$$\mathrm{Var}(r_i) = p(1-p), \quad \mathrm{Var}(\bar{r}) = \frac{p(1-p)}{G}, \quad \mathrm{Cov}(r_i, \bar{r}) = \frac{p(1-p)}{G}.$$

Substitute in:

$$\mathbb{E}[\hat{A}_i^2] = p(1-p) + \frac{p(1-p)}{G} - 2 \cdot \frac{p(1-p)}{G} = p(1-p)\left(1 - \frac{1}{G}\right).$$

Therefore,

$$\mathbb{E}[\|g\|^2] = V \cdot G \cdot p(1-p)\left(1 - \frac{1}{G}\right),$$

which is maximized when $p = 0.5$.

□

D Details of Adaptive Difficulty Prediction Framework

D.1 Design and Implementation Details

The core of our adaptive difficulty prediction framework lies in obtaining proper embeddings to enable attention-based weighted prediction, as described in Section 4.1. To achieve this efficiently, we freeze the Qwen2.5-Math-1.5B-Instruct model as the backbone and augment it with a lightweight adapter and a calibration head.

The adapter is a GELU-activated MLP with three hidden layers, each containing 896 units and a dropout rate of 0.1. A LayerNorm is applied to the projection output to stabilize training. The calibration head is a two-layer MLP that takes the mean and standard deviation of reference set difficulties as input. The first output passes through a Softplus activation to yield the scale parameter $w^{(t)}$, while the second is transformed by a Tanh activation to produce a bounded bias term $b^{(t)}$, as defined in Section 4.1.

We collect training data from a set of LLMs that are disjoint from our policy models. These include Qwen2.5-Instruct and Qwen2.5-Math-Instruct series [22], Eurus-2-7B-PRIME [2], Mathstral-7B-v0.1[1], DeepSeek-R1-Distill-Qwen-1.5B [4], DeepScaleR-1.5B-Preview [16], and Qwen2.5-7B-SimpleRL-Zoo [25]. For each model, we sample query questions and reference questions from math datasets and compute their adaptive difficulty as supervision labels. Each training instance consists of a query question q, a reference set $\{(q_i, d_i)\}_{i=1}^{K}$ with known difficulty scores, and a ground-truth difficulty label d_q. Repeating this procedure across models yields the training dataset $\mathcal{D}_{\text{pred-train}}$.

We train the adapter and calibration head using the standard binary cross-entropy loss:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{|\mathcal{D}_{\text{pred-train}}|} \sum_{(q,\{(q_i,d_i)\},d_q)\in\mathcal{D}_{\text{pred-train}}} \left[d_q \log \hat{d}_{q,\text{cal}} + (1-d_q)\log(1-\hat{d}_{q,\text{cal}}) \right],$$

where $\hat{d}_{q,\text{cal}}$ is the calibrated predicted difficulty for the query question.

D.2 Qualitative Examples

Tab. 1 presents a qualitative example from the Qwen2.5-3B model, showing one unlabeled question alongside reference questions with the highest and lowest attention scores. The example demonstrates that our difficulty prediction framework assigns higher attention to reference questions that share key mathematical topics and structures (*e.g.*, rhombus, incircle), while down-weighting unrelated questions.

E Implementation Details

E.1 Training Datasets and Models

Our experiments involve three model sizes: Qwen2.5-Math-1.5B, Qwen2.5-3B, and Qwen2.5-Math-7B [22]. We adopt four open-source datasets of mathematical reasoning for RL fine-tuning:

- **MATH** [7]:This dataset contains 12,500 competition-level problems from sources such as AMC and AIME, spanning seven mathematical subjects and five difficulty levels. Following [13, 25], we merge the train and test splits and retain only Level 3–5 questions. These are guaranteed to have no overlap with the MATH500 benchmark to prevent data contamination.
- **DeepScaleR-40K** [16]: A collection of approximately 40,000 curated mathematical problems from AMC (pre-2023), AIME (1984–2023), Omni-MATH [3], and Still [17]. Deduplication is performed using embedding-based retrieval, and ungradable problems are filtered to ensure high-quality reward signals. We randomly sample 10,240 problems for training.
- **Open-Reasoner-Zero-57K (ORZ)** [9]: This dataset includes 57,000 high-quality reasoning problems sourced from AIME (up to 2023), AMC, MATH, Numina-MATH [12], and Tulu3 MATH [10]. Extensive cleaning via rule-based and LLM-based filters ensures evaluability and difficulty balance. We sample 8,192 problems for training.
- **DeepMath-103K** [6]: A large-scale dataset focused on high-difficulty mathematical problems, constructed with rigorous data decontamination procedures to support reliable benchmark evaluation. We also sample 8,192 problems for training.

E.2 RL Fine-tuning Details

Tab. 2 summarizes the hyperparameters used in our GRPO training. We adopt the same configuration across all experiments. Following [24, 9], we remove the KL regularization terms. For reward

[1] `https://huggingface.co/mistralai/Mathstral-7B-v0.1`

Table 1: **Qualitative example illustrating similarity-based attention mechanism in adaptive difficulty prediction.** The table shows one unlabeled question and its top- and bottom-ranked reference questions by attention score. High-attention references (red) typically share similar concepts and difficulty with the target question (*e.g.*, rhombus and incircle geometry), while low-attention references (blue) diverge in topic and are significantly easier.

Data Source: DeepScaleR

Unlabeled Question *[adaptive diff.* $= 1.000$*, predicted score* $= 0.907$*]*

In the rhombus $ABCD$, point Q divides side BC in the ratio $1:3$ starting from vertex B, and point E is the midpoint of side AB. It is known that the median CF of triangle CEQ is equal to $2\sqrt{2}$, and $EQ = \sqrt{2}$. Find the radius of the circle inscribed in rhombus $ABCD$.

#	Attention Score	Adaptive Diff.	Reference Question
1	0.487	1.000	Rhombus $ABCD$ has $\angle BAD < 90^\circ$. There is a point P on the incircle of the rhombus such that the distances from P to the lines DA, AB, and BC are 9, 5, and 16, respectively. Find the perimeter of $ABCD$.
2	0.093	1.000	Circle ω_1 with radius 3 is inscribed in a strip S having border lines a and b. Circle ω_2 within S with radius 2 is tangent externally to circle ω_1 and is also tangent to line a. Circle ω_3 within S is tangent externally to both circles ω_1 and ω_2, and is also tangent to line b. Compute the radius of circle ω_3.
			...
255	0.000	0.125	A package of milk with a volume of 1 liter cost 60 rubles. Recently, for the purpose of economy, the manufacturer reduced the package volume to 0.9 liters and increased its price to 81 rubles. By what percentage did the manufacturer's revenue increase?
256	0.000	0.125	Given $\tan\left(\alpha - \frac{\pi}{4}\right) = 2$, find the value of $\sin\left(2\alpha - \frac{\pi}{4}\right)$.

Table 2: **Detailed RL fine-tuning recipes.**

Optimizer	AdamW
Total Batch Size	512
Learning Rate	1e-6
LR Schedule	Constant
Weight Decay	0
Warm-up Ratio	0
Number of Steps	60
Max Prompt Length	1024
Max Rollout Length	3072/4096
Number of Rollouts Per Prompt	8
Rollout Sampling Temperature	0.6
Rollout Sampling Top-p	0.95
GPU Hardware	8x NVIDIA L40S/8x NVIDIA A100

computation, we use a simple rule-based function based solely on answer correctness, without incorporating any format-related signals. Specifically, a reward of 1 is assigned for exact matches with the reference answer, and 0 otherwise. Answer matching is implemented using the Math-Verify library[2]. We adopt a standard chain-of-thought (CoT) prompt template, provided in Tab. 3.

E.3 Implementation Details of DOTS and RR

We present the detailed hyperparameter settings of Algorithm 1 in Tab. 4. For DOTS, data selection is performed every two steps during RL fine-tuning.

[2] https://github.com/huggingface/Math-Verify

Table 3: **Prompt template used for RL fine-tuning and evaluation.** The placeholder <question> is replaced with the actual mathematical question during fine-tuning and evaluation. Special tokens "<|im_start|>" and "<|im_end|>" are omitted for clarity.

system
`Let's think step by step and output the final answer within \boxed{}.`
user
`<question>`
assistant

Table 4: **Hyperparameters of DOTS and RR.**

Target Difficulty α	0.5
Reference Set Size K	256
Data Sampling Temperature τ	1e-3
Fresh Rollout Fraction δ	0.5
Buffer Capacity C	256/512

E.4 Evaluation Details

Consistent with RL fine-tuning, we use a sampling temperature of 0.6, top-p of 0.95, and the same prompt template. We evaluate model performance on four commonly-used mathematical reasoning benchmarks and report the average accuracy to mitigate benchmark-specific variance. We exclude benchmarks with very few questions, such as AIME 24 (30 questions) and AMC 23 (40 questions), as their limited size leads to high evaluation variance and unreliable performance comparisons for smaller models [8].

- **GSM8K** [1]: A test set of 1,319 grade school math word problems from the GSM8K dataset, requiring multi-step arithmetic reasoning.
- **MATH500** [14]: A widely used subset of the MATH test split [7]. These problems are excluded from our MATH training data.
- **Minerva Math** [11]: A set of 272 undergraduate-level science and math questions from MIT OpenCourseWare.
- **OlympiadBench** [5]: A benchmark of 675 problems from international math olympiads and physics contests.

F Additional Experimental Results

F.1 Ablation Study on the Adaptive Difficulty Prediction Framework

Off-the-shelf embeddings fail to capture difficulty structure. We evaluate a baseline that directly uses frozen embeddings from the Qwen2.5-Math-1.5B-Instruct model without any training or calibration. In contrast, our framework incorporates trained adapter layers and a calibration head. As shown in Tab. 5, our framework consistently achieves significantly higher Pearson correlation with ground-truth adaptive difficulty across all settings. The poor performance of the off-the-shelf baseline highlights the necessity of further adapter layers and calibration for accurately predicting question difficulty.

DOTS **is robust to the size of reference set.** We further investigate the impact of the reference set size K in RL fine-tuning. Fig. 1 compares the performance of the original GRPO and the DOTS method under reference set sizes of 128 and 256, trained with Qwen2.5-Math-1.5B and Qwen2.5-3B on the DeepScaleR dataset. The results show that a reference set size of 128 yields RL performance comparable to that of 256. This indicates that our approach is robust to smaller reference sets, enabling more efficient rollout collection without sacrificing RL fine-tuning quality.

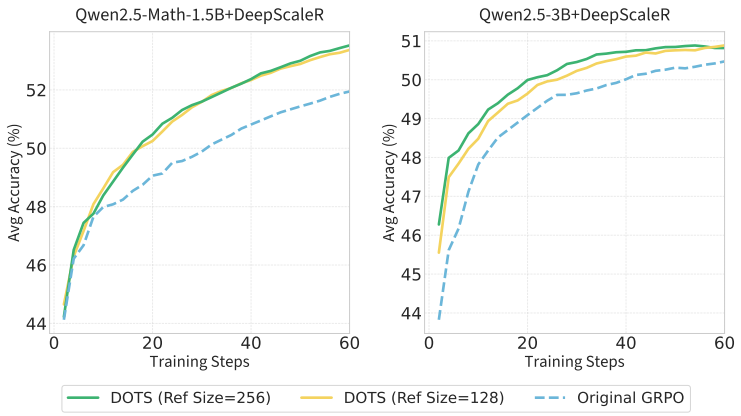

Figure 1: **Average accuracy curves of DOTS (Ref Size = 256), DOTS (Ref Size = 128), and Original GRPO on Qwen2.5-Math-1.5B and Qwen2.5-3B.** The curves show average performance aggregated over four benchmarks with exponential smoothing for visualization. Note that the x-axis is the number of **steps** (rather than time). The results show that a reference set size of 128 achieves performance comparable to that of 256, indicating the robustness of our method to smaller reference sets.

Table 5: **Ablation study on training with adapter and calibration.** Comparison of average Pearson correlation (ρ) between predicted scores and ground-truth adaptive difficulties, reported as mean $\pm$ standard deviation over 60 training steps. Results show that training with adapter layers and calibration significantly improves prediction performance.

Model	Dataset	Off-the-shelf Embedding	Our Framework (With Adapter Layers + Calibration)
Qwen2.5-Math-1.5B	MATH	0.2682 ± 0.0207	0.7843 ± 0.0243
	DeepScaleR	0.2064 ± 0.0518	0.7244 ± 0.0318
	ORZ	0.1598 ± 0.0266	0.7153 ± 0.0257
Qwen2.5-3B	DeepScaleR	0.2688 ± 0.0369	0.7789 ± 0.0191
	DeepMath	0.0671 ± 0.0168	0.7029 ± 0.0082
Qwen2.5-Math-7B	DeepScaleR	0.1983 ± 0.0254	0.7076 ± 0.0195

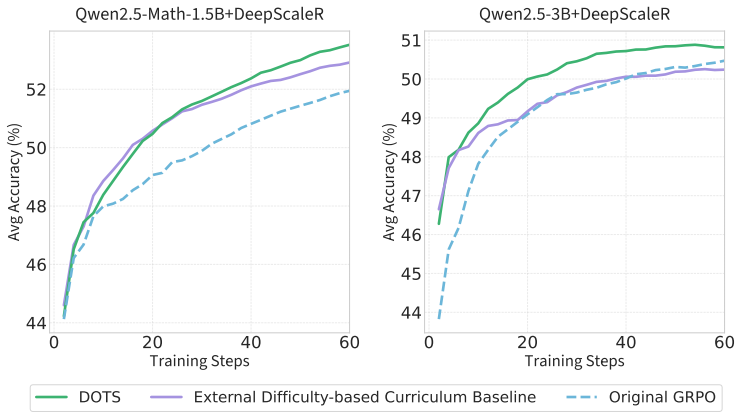

Figure 2: **Comparison between DOTS (ours) and an external difficulty-based curriculum baseline.** The curves show average performance aggregated over four benchmarks with exponential smoothing for visualization. Note that the x-axis is the number of **steps** (rather than time). Our method consistently outperforms the baseline.

F.2 Case Study: Online Data Selection Via External Difficulty-based Curriculum

We additionally implement an online data selection baseline that relies on external difficulty annotations. Specifically, we use the DeepScaleR dataset and label each question with GPT-4o-mini,

following the difficulty annotation prompt introduced in Luo et al. [16]. Each question is annotated 32 times, and the average score is used as its final difficulty.

We then follow a staged curriculum: in the first third of training steps, batches are sampled from the easiest third of the dataset; in the middle third, from the medium-difficulty third; and in the final third, from the hardest third. To ensure a fair comparison of online data selection strategies, we compare this baseline with DOTS (without RR). As shown in Fig. 2, our DOTS method consistently outperforms this baseline on both Qwen2.5-Math-1.5B and Qwen2.5-3B. We further discuss in the main text the limitations of such approaches, including the high cost of annotation and limited adaptability due to their reliance on fixed, hand-crafted curricula.

References

[1] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[2] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

[3] Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, et al. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*, 2024.

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

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

[6] Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

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

[8] Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility. *arXiv preprint arXiv:2504.07086*, 2025.

[9] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

[10] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[11] Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, 2022.

[12] Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. `https://huggingface.co/AI-MO/NuminaMath-CoT`, 2024. Report available at `https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf`.

[13] Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling. *arXiv preprint arXiv:2502.11886*, 2025.

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

[15] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025. *URL https://arxiv. org/abs/2503.20783*.

[16] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025. Notion Blog.

[17] Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, et al. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *arXiv preprint arXiv:2412.09413*, 2024.

[18] Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025.

[19] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay, 2016. URL `https://arxiv.org/abs/1511.05952`.

[20] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[21] Milan Vojnovic and Se-Young Yun. What is the alignment objective of grpo? *arXiv preprint arXiv:2502.18548*, 2025.

[22] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[23] Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

[24] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

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

[26] Shangtong Zhang and Richard S Sutton. A deeper look at experience replay. *arXiv preprint arXiv:1712.01275*, 2017.