# Not All Rollouts are Useful: Down-Sampling Rollouts in LLM Reinforcement Learning

**Yixuan Even Xu*** **Yash Savani*** **Fei Fang** **Zico Kolter**
Carnegie Mellon University
{yixuanx,ysavani,feif,zkolter}@cs.cmu.edu

## Abstract

Reinforcement learning (RL) has emerged as a powerful paradigm for enhancing reasoning capabilities in large language models, but faces a fundamental asymmetry in computation and memory requirements: inference is *embarrassingly parallel* with a minimal memory footprint, while policy updates require *extensive synchronization* and are memory-intensive. To address this asymmetry, we introduce **PODS** (Policy Optimization with Down-Sampling), a framework that strategically decouples these phases by generating numerous rollouts in parallel but updating only on an informative subset. Within this framework, we develop max-variance down-sampling, a theoretically motivated method that selects rollouts with maximally diverse reward signals. We prove that this approach has an efficient algorithmic solution, and empirically demonstrate that GRPO with PODS using max-variance down-sampling achieves superior performance over standard GRPO on the GSM8K benchmark.

## 1 Introduction

Reinforcement learning (RL) has emerged as a critical technique for enhancing the reasoning capabilities of large language models, substantially boosting performance on mathematical, coding, and general problem-solving benchmarks [1–4]. RL algorithms such as Proximal Policy Optimization (PPO) [5] and Group Relative Policy Optimization (GRPO) [6] are typically structured into two distinct phases: (1) an *inference phase*, in which rollouts are generated and evaluated, and (2) a *policy update phase*, in which the model is optimized based on those evaluations. These phases introduce a fundamental asymmetry in computational and memory demands: inference is embarrassingly parallel, with minimal communication overhead and a modest memory footprint, while policy updates require extensive synchronization across devices and substantially more memory to store optimizer states.

Despite advances in distributed training, this asymmetry remains a persistent bottleneck across computing environments. In *resource-constrained* settings, limited memory during the policy update phase forces practitioners to either process rollouts in inefficient microbatches or leave inference resources underutilized. In contrast, even in *resource-rich* environments with abundant compute and memory, policy updates encounter scalability limits due to the communication overhead of gradient synchronization—creating a throughput ceiling that inference can easily surpass. This computational imbalance presents a strategic opportunity: by rethinking how rollouts are selected for gradient updates, we can design more efficient RL algorithms that maximize parallel inference capacity while mitigating communication bottlenecks during policy update phases.

Our work provides a simple but powerful insight: *not all rollouts contribute equally to model improvement*. We introduce **PODS** (Policy Optimization with Down-Sampling), a frame-

Preprint. Under review.

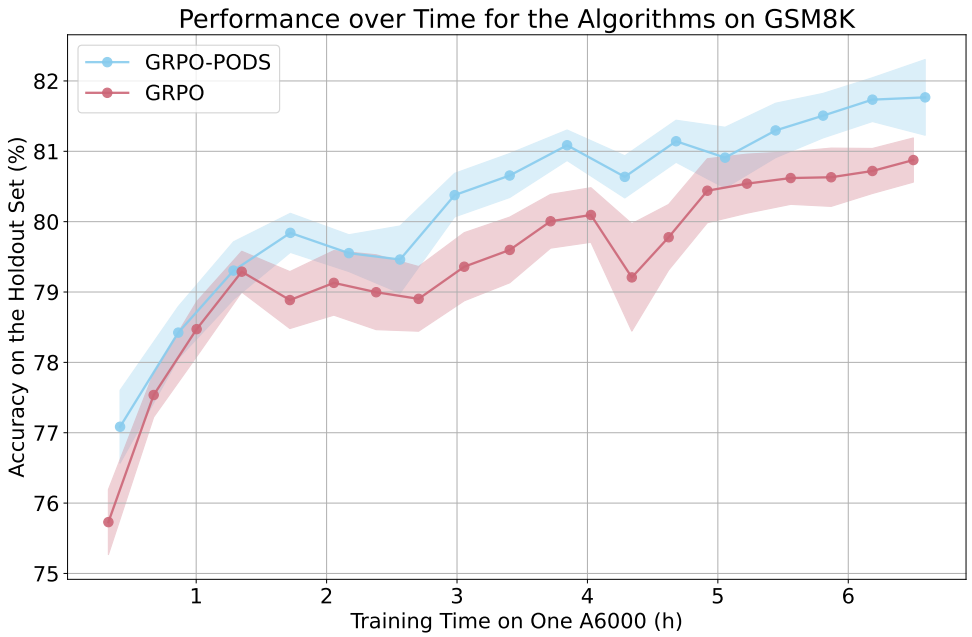


Figure 1: Performance of GRPO and GRPO-PODS with max-variance down-sampling on GSM8K. The x-axis shows the training time on one A6000, and the y-axis shows the accuracy on the test set. The shaded area represents 1.96 times the standard error of the mean.

work that generates many rollouts in parallel but updates the policy only on the most informative subset. By employing a strategic selection method, *max-variance sampling*, to encourage diverse contrastive signals, we achieve substantial efficiency gains over standard RL approaches. This method directly addresses the computational asymmetry that limits the scalability of RL for reasoning tasks across a wide range of hardware environments, enabling more efficient training for reasoning tasks.

## 2 Related Work

**Reinforcement learning for LLM reasoning.** Reinforcement learning (RL) has emerged as a powerful technique for enhancing large language models' reasoning capabilities on mathematical, coding, and general problem-solving tasks [1, 6, 7]. Classical algorithms like Proximal Policy Optimization (PPO) [5] established the foundation, while recent work has introduced innovations tailored to language models. Group Relative Policy Optimization (GRPO) [6] has gained popularity for reasoning tasks due to its simpler implementation, comparable performance to PPO, and elimination of separate critic networks. Since the release DeepSeek R1 [8], which employed large-scale RL with millions of samples, interest in reasoning-focused RL approaches has surged [9–12], among which value-based RL like PPO still played a significant role [13, 14]. Other approaches have explored Monte Carlo Tree Search [15, 16] and multi-agent methods [17]. Two recent methods share surface similarities with our approach: DAPO [18] introduces dynamic sampling by skipping problems with binary (0/1) accuracy during training, while VAPO [13] suggests generating more rollouts per problem rather than increasing batch size. However, neither proposes *down-sampling* rollouts generated during inference—our key innovation. Our approach complements these methods by addressing a fundamental computational efficiency bottleneck and can be combined with them to further improve reasoning performance.

**Down-sampling and data selection.** Modern machine learning relies heavily on large datasets, but the datasets collected from the real world are often noisy and unbalanced, and the computation cost of training on the entire dataset can be prohibitively high. As a

result, data selection and down-sampling techniques have been widely studied to improve the efficiency of training. Similar idea have been applied to clustering [19], regression [20–22], speech recognition [23, 24], computer vision [25, 26], reinforcement learning [27–29], foundation model training [30–32] and applications like advertising [33, 34]. To the best of our knowledge, our work is the first to apply down-sampling and data selection techniques to the rollout generation process in reinforcement learning for LLMs.

## 3 Down-Sampling Rollouts in GRPO

This section introduces our approach to address the computational asymmetry in reinforcement learning for large language models (LLMs). We begin by reviewing the standard GRPO algorithm in Section 3.1 and highlight its structure and computational requirements. We then present the PODS (Policy Optimization with Down-Sampling) framework in Section 3.2, a principled method for maximizing resource utilization during both the inference and policy update phases through strategic rollout selection. In Section 3.3, we then develop our theoretically motivated *max-variance down-sampling* method, which captures contrastive signals from both high- and low-reward examples. We prove that the method admits an elegant and computationally efficient solution, making it practical for real-world implementation. Our framework retains the core advantages of GRPO while improving the utilization of computational and memory resources in distributed training environments.

### 3.1 Preliminaries

Group Relative Policy Optimization (GRPO) is a reinforcement learning algorithm designed to enhance the reasoning capabilities of LLMs. Each training step in GRPO follows a structured two-phase process described below.

**Inference phase.** Let $\pi_\theta$ denote the policy parameterized by $\theta$, which defines a distribution over next-token probabilities given the previous tokens in a sequence. Given a single input prompt $p$ (e.g., a math problem), GRPO first generates a group of $n$ rollouts $\mathbf{o} = (o_1, o_2, \ldots, o_n)$ by autoregressively sampling from $\pi_\theta$. Each rollout is a complete token sequence excluding the prompt, representing a possible solution. Each rollout is then evaluated using a reward model $r_i = R(o_i)$, which scores the quality and correctness of the corresponding output $o_i$. This yields a reward vector $\mathbf{r} = (r_1, r_2, \ldots, r_n)$. From these rewards, we compute normalized advantage estimates: $a_i = (r_i - \mu)/\sigma$, where $\mu$ and $\sigma$ are the mean and standard deviation of the rewards respectively.

**Policy update phase.** After computing the advantages, the policy is updated by optimizing the GRPO objective $L_{\text{GRPO}}(\theta)$. Specifically, for each rollout $o_i$ with advantage $a_i$, we compute a loss for each token position $t$:

$$L_{\text{GRPO}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \min\left[\frac{\pi_\theta(o_{i,t} \mid p, o_{i,<t})}{\pi_{\theta_{\text{fixed}}}(o_{i,t} \mid p, o_{i,<t})}\cdot a_i,\ \text{clip}\left(\frac{\pi_\theta(o_{i,t} \mid p, o_{i,<t})}{\pi_{\theta_{\text{fixed}}}(o_{i,t} \mid p, o_{i,<t})}, 1-\epsilon, 1+\epsilon\right)\cdot a_i\right].$$

where $|o_i|$ is the number of tokens in $o_i$ and $\pi_{\theta_{\text{fixed}}}$ is a frozen copy of the policy used for importance weighting. This asymmetric loss embodies the *slow to adopt, quick to abandon* learning principle—limiting how aggressively the policy increases probabilities for tokens in high-reward rollouts while allowing more substantial reductions for low-reward sequences.

### 3.2 PODS Framework

We propose to *decouple the inference and training phases* of GRPO. Instead of using all the generated rollouts for policy updates, PODS generates a large number of $n$ rollouts in parallel but selectively trains on only a smaller subset $m < n$ according to a down-sampling rule $D$. This approach leverages parallel computation during inference while significantly reducing communication and memory overheads during the policy gradient update.

**Definition 3.1** (Down-sampling rule). *$D(\mathbf{o}, \mathbf{r}; m)$ is a function that takes as inputs $n$ rollouts $\mathbf{o} = (o_1, o_2, \ldots, o_n)$, their corresponding rewards $\mathbf{r} = (r_1, r_2, \ldots, r_n)$, and the target size $m$. It outputs a subset of indices $S \subseteq \{1, 2, \ldots, n\}$, where $|S| = m$, indicating which rollouts to retain for the policy update phase.*

Given a selected subset of indices $S$, we compute the advantage estimates using only the selected rollouts: $a_{S,i} = (r_i - \mu_S)/\sigma_S$, where $\mu_S$ and $\sigma_S$ are the mean and standard deviation of the rewards in the selected subset. The GRPO-PODS objective then becomes:

$$L_{\text{PODS}}(\theta, S) = \frac{1}{m} \sum_{i \in S} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min \left[ \frac{\pi_\theta(o_{i,t} \mid p, o_{i,<t})}{\pi_{\theta_{\text{fixed}}}(o_{i,t} \mid p, o_{i,<t})} \cdot a_{S,i}, \text{clip} \left( \frac{\pi_\theta(o_{i,t} \mid p, o_{i,<t})}{\pi_{\theta_{\text{fixed}}}(o_{i,t} \mid p, o_{i,<t})}, 1 - \varepsilon, 1 + \varepsilon \right) \cdot a_{S,i} \right].$$

Our framework can be summarized as follows:

**Algorithm 1** The PODS Framework for GRPO

**Input:** Models $\pi_\theta, \pi_{\theta_{\text{fixed}}}$, input prompt $p$, reward model $R$,
Number of rollouts $n$, target size $m$, down-sampling rule $D$

1: Individually sample $n$ rollouts $\mathbf{o} = (o_1, o_2, \ldots, o_n)$ using $\pi_{\theta_{\text{fixed}}}$ for prompt $p$
2: Compute rewards $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ using the reward model $R$
3: Down-sample a set of $m$ rollouts $S \leftarrow D(\mathbf{o}, \mathbf{r}; m)$
4: Update the policy using the GRPO-PODS objective $L_{\text{PODS}}(\theta, S)$

**Output:** An updated model $\pi_{\theta_{\text{updated}}}$

To illustrate this framework, we introduce two simple down-sampling strategies:

**Random down-sampling.** $(D_{\text{rand}})$ randomly selects $m$ indices from $\{1, 2, \ldots, n\}$, preserving the statistical properties of the original distribution. This approach is equivalent to standard GRPO with $m$ rollouts.

**Max-reward down-sampling.** $(D_{\text{maxr}})$ selects the $m$ rollouts with the highest rewards, focusing on examples that demonstrate the most desirable behavior. This allows the model to learn primarily from successful reasoning patterns.

### 3.3 Max-Variance Down-Sampling

While max-reward down-sampling effectively captures successful reasoning patterns, it neglects potentially valuable learning signals from poor-performing examples. We introduce *max-variance down-sampling* as an information-theoretically motivated approach that selects the most diverse and informative rollouts based on the reward distribution.

The max-variance down-sampling strategy $D_{\text{maxv}}$ selects the subset $S$ of $m$ rollouts that maximizes the variance of rewards: $\max_{|S|=m} \text{Var}(\{r_i | i \in S\})$. Intuitively, this approach captures the full spectrum of performance, providing stronger contrastive learning signals between successful and unsuccessful reasoning paths. Such intuition has recently been formally demonstrated from an optimization perspective [35]. A naive implementation would require examining $\mathcal{O}\left(\binom{n}{m}\right)$ possible subsets–computationally infeasible for practical values of $n$ and $m$. However, we prove that an elegant and efficient solution exists:

**Lemma 3.1.** *For a sorted list of rewards $r_1 \leq r_2 \leq \cdots \leq r_n$, the variance-maximizing subset of size $m$ always consists of the $k$ highest rewards and $(m - k)$ lowest rewards for some $k \in \{0, 1, \ldots, m\}$. That is,*

$$\text{Var}(\{r_1, \ldots, r_{m-k}\} \cup \{r_{n-k+1}, \ldots, r_n\}) = \max_{|S|=m} \text{Var}(\{r_i \mid i \in S\}).$$

**Proof of Lemma 3.1:** Let $S^* = \arg\max_{|S|=m} \mathrm{Var}(\{r_i \mid i \in S\})$ be the optimal subset of size $m$. We will show that if $S^*$ is not of the form $\{1, \ldots, m-k\} \cup \{n-k+1, \ldots, n\}$ for any $k$, then we can modify $S^*$ to obtain a new subset $S'$ of the same size with no smaller variance in rewards. By repeating this procedure, we can eventually reach a subset of this form.

Let $\mu_{S^*}$ be the mean of the rewards in $S^*$. Since the set $S^*$ does not take the form of $\{1, \ldots, m-k\} \cup \{n-k+1, \ldots, n\}$ for any $k$, there exists either (i) an element $i \in S^*$ such that $i > 1, r_i \leq \mu$ and $i-1 \notin S^*$, or (ii) an element $j \in S^*$ such that $j < n, r_j \geq \mu$ and $j+1 \notin S^*$. That is, there exists an element in $S^*$, such that another element further from $\mu^*$ is not in $S^*$. We will show that we can swap them without decreasing variance.

For the ease of notation, we will denote $\mathrm{Var}(\{r_i \mid i \in S\})$ as $\mathrm{Var}(S)$ in this proof.

For case (i), let $S' = (S^* \setminus \{i\}) \cup \{i-1\}$, and let $\mu'$ be the mean of the rewards in $S'$. Then

$$\begin{aligned}
\mathrm{Var}(S') - \mathrm{Var}(S^*) &= \left(\frac{1}{m}\sum_{t \in S'} r_t^2 - \mu'^2\right) - \left(\frac{1}{m}\sum_{t \in S^*} r_t^2 - \mu^2\right) \\
&= \frac{1}{m}(r_{i-1}^2 - r_i^2) - (\mu'^2 - \mu^2) \\
&= \frac{1}{m}(r_{i-1} - r_i)(r_{i-1} + r_i) - (\mu' - \mu)(\mu' + \mu) \\
&= \frac{1}{m}(r_{i-1} - r_i)[(r_{i-1} + r_i) - (\mu' + \mu)] \geq 0.
\end{aligned}$$

For case (ii), let $S' = (S^* \setminus \{j\}) \cup \{j+1\}$, we can similarly show that $\mathrm{Var}(S') - \mathrm{Var}(S^*) \geq 0$.

In either case, we have shown that we can modify $S^*$ to obtain a new subset $S'$ of the same size that has no smaller variance in rewards. We can repeat this process until we reach a subset of the form $\{1, \ldots, m-k\} \cup \{n-k+1, \ldots, n\}$ for some $k$. Thus, we conclude that there must exist one optimal subset of this form for some $k$. ■

Lemma 3.1 naturally leads to a practical algorithm for max-variance down-sampling. Moreover, it also provides a intuition for why maximizing variance is a good strategy: the optimal subset consists of the $k$ highest rewards and $(m-k)$ lowest rewards, which captures the contrastive signals of both positive and negative examples.

**Algorithm 2** Max-Variance Down-Sampling

**Input:** Number of rollouts $n$, target size $m$, rollouts $\{o_1, o_2, \ldots, o_n\}$, rewards $\{r_1, r_2, \ldots, r_n\}$

1: Sort the rollouts by reward and get the sorted indices $ind \leftarrow argsort(\{r_1, r_2, \ldots, r_n\})$
2: Let $S_{\text{ans}} \leftarrow \{1, \ldots, m\}$
3: **for** $k \in \{1, \ldots, m\}$ **do**
4: Let $S_{\text{this}} \leftarrow \{ind_1, \ldots, ind_{m-k}\} \cup \{ind_{n-k+1}, \ldots, ind_n\}$
5: Let $S_{\text{ans}} \leftarrow S_{\text{this}}$ **if** $\mathrm{Var}(\{r_i \mid i \in S_{\text{this}}\}) > \mathrm{Var}(\{r_i \mid i \in S_{\text{ans}}\})$
6: **end for**

**Output:** Selected indices $S_{\text{ans}}$ of rollouts

**Theorem 1.** *Algorithm 2 computes the max-variance down-sampling rule in $O(n \log n + m^2)$ time.*

Theorem 1 shows that the max-variance down-sampling rule can be computed efficiently in acceptably low time complexity. We note that it is possible to further reduce the time complexity to $O(n \log n + m)$ by using partial sum techniques to compute the variance of the selected rollouts. However, since the difference in time complexity of the down-sampling rule is negligible compared to the generation and training phases for the parameter settings of practical interest, we omit this optimization for simplicity.

## 4 Experiments

**Architectures, benchmarks, and baselines.** To evaluate the effectiveness of our method, we conduct experiments on the **GSM8K** dataset [36], comparing our proposed down-

sampling method with the original GRPO over training time. We use the `Qwen2.5-3B-Instruct` model [37] as the base and fine-tune it using both GRPO and GRPO-PODS, applying the max-variance down-sampling rule on the GSM8K training set. Model checkpoints are evaluated on the test set every 100 training steps.

**Training details.** For both GRPO and GRPO-PODS, we fine-tune the base model using LoRA [38] with rank 64, $\alpha = 64$, learning rate $5 \times 10^{-6}$, weight decay 0.1, maximum gradient norm 1.0, and a constant learning rate schedule with a warmup ratio of 0.01. We set the maximum sequence length to 1024. For GRPO, we use $n = 16$; for GRPO-PODS, we use $n = 32$ and $m = 16$, i.e., 32 rollouts are generated and 16 are selected for policy updates. We adopt a rule-based reward model that scores each rollout based on correctness and format. As suggested in [18], we omit the KL divergence term from the GRPO objective, as it is not necessary for reasoning tasks. Each algorithm is run 8 times with different random seeds. For each saved checkpoint, we evaluate the model on the test set 5 times and report average accuracy with standard error. Results are presented in Fig. 1.

**Results.** In Fig. 1, we show evidence that our proposed down-sampling method achieves better performance than the original GRPO method given the same amount of training time. For the parameter settings we run, we observe that GRPO-PODS with max-variance down-sampling consistently outperforms GRPO over the training time, with the performance separation being more evident as the training proceeds. This demonstrates that our PODS framework is effective in improving the performance of GRPO on reasoning tasks.

## 5 Conclusion

We presented **PODS**, a lightweight framework that addresses the inference-update asymmetry in RL for LLMs by generating large batches of rollouts in parallel and updating the policy on only a maximally informative subset selected with an efficient max-variance rule. This approach retains the embarrassingly parallel scalability of inference while reducing the communication and memory costs of training, and our theory shows the optimal subset can be found efficiently. Experiments on GSM8K with Qwen2.5-3B-Instruct confirm that GRPO-PODS clearly improves over GRPO under equal compute, demonstrating both the practicality and impact of selective updates. Because PODS is method-agnostic and hardware-friendly, it can be plugged into existing RLHF pipelines at any scale; future work will scale this line of research to more complex datasets and extend it to additional RL algorithms such as PPO.

## References


[1] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[2] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

[3] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[4] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

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

[6] 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.

[7] Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment. *arXiv preprint arXiv:2410.01679*, 2024.

[8] 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.

[9] Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, et al. An empirical study on eliciting and improving r1-like reasoning models. *arXiv preprint arXiv:2503.04548*, 2025.

[10] 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.

[11] Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

[12] 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.

[13] Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

[14] Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025.

[15] Zitian Gao, Boye Niu, Xuzheng He, Haotian Xu, Hongzhang Liu, Aiwei Liu, Xuming Hu, and Lijie Wen. Interpretable contrastive monte carlo tree search reasoning. *arXiv preprint arXiv:2410.01707*, 2024.

[16] Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*, 2024.

[17] Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

[18] 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.

[19] Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 291–300, 2004.

[20] Mu Li, Gary L Miller, and Richard Peng. Iterative row sampling. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 127–136. IEEE, 2013.

[21] Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM)*, 54(4):21–es, 2007.

[22] Kenneth L Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):1–30, 2010.

[23] Yuzong Liu, Rishabh K Iyer, Katrin Kirchhoff, and Jeff A Bilmes. Svitchboard ii and fisver i: high-quality limited-complexity corpora of conversational english speech. In *INTERSPEECH*, pages 673–677, 2015.

[24] Kai Wei, Yuzong Liu, Katrin Kirchhoff, and Jeff Bilmes. Unsupervised submodular subset selection for speech data. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4107–4111. IEEE, 2014.

[25] Vishal Kaushal, Rishabh Iyer, Suraj Kothawade, Rohan Mahadev, Khoshrav Doctor, and Ganesh Ramakrishnan. Learning from less data: A unified data subset selection and active learning framework for computer vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1289–1299. IEEE, 2019.

[26] William Bankes, George Hughes, Ilija Bogunovic, and Zi Wang. Reducr: Robust data downsampling using class priority reweighting. *Advances in Neural Information Processing Systems*, 37:82781–82810, 2024.

[27] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[28] Yuenan Hou, Lifeng Liu, Qing Wei, Xudong Xu, and Chunlin Chen. A novel ddpg method with prioritized experience replay. In *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 316–321. IEEE, 2017.

[29] Baturay Saglam, Furkan B Mutlu, Dogan C Cicek, and Suleyman S Kozat. Actor prioritized experience replay. *Journal of Artificial Intelligence Research*, 78:639–672, 2023.

[30] Sachin Goyal, Pratyush Maini, Zachary C Lipton, Aditi Raghunathan, and J Zico Kolter. Scaling laws for data filtering–data curation cannot be compute agnostic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22702–22711, 2024.

[31] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[32] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023.

[33] Xiaohui Bei, Nick Gravin, Pinyan Lu, and Zhihao Gavin Tang. Bidder subset selection problem in auction design. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3788–3801. SIAM, 2023.

[34] Nikolai Gravin, Yixuan Even Xu, and Renfei Zhou. Bidder selection problem in position auctions: A fast and simple algorithm via poisson approximation. In *Proceedings of the ACM Web Conference 2024*, pages 89–98, 2024.

[35] Noam Razin, Zixuan Wang, Hubert Strauss, Stanley Wei, Jason D Lee, and Sanjeev Arora. What makes a reward model a good teacher? an optimization perspective. *arXiv preprint arXiv:2503.15477*, 2025.

[36] 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.

[37] 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.

[38] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.