COMPUTE-EFFICIENT GRPO TRAINING

Rajat Ghosh, Vaishnavi Bhargava, Debojyoti Dutta
Nutanix
San Jose, CA 95110, USA
{rajat.ghosh, vaishnavi.bhargava, debojyoti.dutta}@nutanix.com

ABSTRACT

Reinforcement fine-tuning (RFT) methods such as Group Relative Policy Optimization (GRPO) and PPO are significantly more expensive than supervised fine-tuning due to on-policy sampling, repeated rollouts, multiple forward passes, and backpropagation through long sequences across multiple optimization epochs. These costs make post-training with reinforcement learning a major bottleneck for practitioners seeking to train or adapt large language models under limited computational budgets. In this work, we present an empirical study of GRPO post-training dynamics and identify a consistent early plateau in reward trajectories. Across four open-source models—Llama 3B/8B and Qwen 3B/7B—we observe that GRPO reward curves follow a highly regular, sigmoid-shaped pattern with three phases: slow initial progress, rapid improvement, and early saturation. We show that these dynamics are well captured by a simple parametric model conditioned on model size, initial reward, and normalized training progress, enabling reliable prediction of when marginal reward gains diminish.

A key practical finding is that, across all tested models, the majority of reward improvement occurs early in training: continuing GRPO beyond roughly 70–80% of a single epoch yields negligible gains while consuming a substantial fraction of total compute. Leveraging the proposed predictive model, practitioners can forecast saturation points early in training and select data-driven stopping criteria, substantially reducing GRPO compute without sacrificing final reward. Our results highlight predictable structure in GRPO training dynamics and suggest that lightweight, empirically grounded early-stopping strategies are an effective tool for managing post-training costs in reinforcement-based LLM fine-tuning.

1 INTRODUCTION

Frontier large language models (LLMs) demonstrate strong reasoning capabilities across mathematics, coding, and other complex domains. However, reproducing these capabilities within constrained organizational settings remains challenging due to the high computational cost of post-training. Post-training methods adapt pretrained models to reasoning-intensive tasks using curated data, typically through either supervised fine-tuning (SFT) or reinforcement fine-tuning (RFT). SFT primarily performs behavior cloning, training models to imitate a fixed demonstration distribution (Ouyang et al., 2022a; Touvron et al., 2023). In contrast, RFT methods—such as Reinforcement Learning from Human Feedback (RLHF) and more recent approaches like Group Relative Policy Optimization (GRPO)—introduce explicit reward signals that encourage exploration and optimization beyond the demonstrations (Ziegler et al., 2019; Bai et al., 2022; Yuan et al., 2024). Algorithms such as Proximal Policy Optimization (PPO) (Schulman et al., 2017a) and GRPO (Shao et al., 2024) are now central to large-scale reasoning systems, including DeepSeek-Math and DeepSeek-R1 (Shao et al., 2024; Guo et al., 2025). Despite their effectiveness, these methods impose substantial computational, memory, and engineering costs that limit their accessibility.

Several factors contribute to the high cost of RFT. First, RFT relies on on-policy data collection: the model must repeatedly generate fresh rollouts at each iteration to compute policy gradients. Unlike SFT, which operates on a static dataset, RFT repeatedly invokes expensive autoregressive generation from an evolving policy, often producing multiple candidate responses per prompt for comparative or group-based reward estimation (Bai et al., 2022; Yuan et al., 2024). For long-context reasoning tasks,

this sampling cost dominates total FLOPs, as observed in recent large-scale systems (Shao et al., 2024; Guo et al., 2025). Second, reward computation itself is costly. Human preference models, verifier models, or rule-based evaluators must be applied to every generated sample, and in multi-sample algorithms such as GRPO, reward-model inference alone can account for 20–40% of total post-training compute (Yuan et al., 2024). Third, stabilizing policy optimization for high-variance reasoning rewards often requires large batch sizes, multiple rollouts per prompt, KL regularization, and adaptive clipping, all of which further increase effective compute per update (Ziegler et al., 2019; Bai et al., 2022). Finally, RFT substantially increases system complexity: large deployments rely on tightly synchronized pipelines that interleave sampling, reward evaluation, and optimization, frequently using pipeline or expert parallelism across GPU clusters (Guo et al., 2025). Collectively, these factors make RFT approximately 3–$10\times$ more expensive than equivalently sized SFT runs, rendering reinforcement-based post-training a dominant cost bottleneck.

Given these costs, reducing unnecessary RFT computation without degrading final performance is a critical practical problem. Early stopping is a well-established technique in supervised learning, where training is halted once validation performance ceases to improve, preventing overfitting and wasted computation (Prechelt, 1998). A similar opportunity may exist in reinforcement fine-tuning. In GRPO and related methods, reward curves are commonly used to monitor training progress; however, these observable metrics often plateau well before the underlying policy optimization has fully converged (Bottou, 2012; Goodfellow et al., 2016; Nimmaturi et al., 2025). Nonetheless, reward curves are the dominant signal for monitoring training progress because they are directly aligned with the optimized objective and available at every update step. While imperfect and prone to early saturation, reward remains the only dense, low-latency signal that reflects the instantaneous state of policy optimization at scale. Prior work on predictive scaling laws (Nimmaturi et al., 2025) for GRPO has focused on accurately modeling reward trajectories across model sizes and training regimes. In contrast, our work is motivated by a different question: how can these predictable dynamics be exploited to make training more compute-efficient? We demonstrate that GRPO training exhibits consistent early saturation behavior that can be detected using only partial training signals, and we formalize this into a practical early stopping criterion. This reframes scaling laws from a descriptive tool into a prescriptive mechanism for controlling training cost, enabling practitioners to terminate runs near the onset of diminishing returns without sacrificing performance.

In this work, we analyze GRPO post-training dynamics with the goal of reducing unnecessary reinforcement fine-tuning compute. Our contributions are:

- **Quantitative characterization of GRPO learning phases.** We show that, across four open-source models (Llama 3B/8B and Qwen 3B/7B), GRPO training consistently follows a three-phase trajectory—slow start, rapid improvement, and early saturation—with approximately 70–80% of reward gains realized within the first fraction of a single training epoch.
- **A predictive parametric model for GRPO reward evolution.** We propose a simple parametric model that accurately fits GRPO reward trajectories and predicts the onset of saturation using only model size, initial reward, and normalized training progress, enabling early identification of diminishing returns.

2 METHODOLOGY

2.1 PRELIMINARY

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is a reinforcement learning algorithm designed to improve the efficiency of large language model (LLM) fine-tuning for reasoning tasks. GRPO is closely related to Proximal Policy Optimization (PPO) (Schulman et al., 2017b), but removes the need for an explicit value (critic) network by estimating baselines using *group-relative rewards*. This design significantly reduces computational and memory overhead while retaining the stability of PPO-style clipped policy updates. Formally, for each input query $q \sim P(Q)$, GRPO samples a group of G outputs

$$\{o_1, o_2, \ldots, o_G\} \sim \pi_{\theta_{\text{old}}}(O \mid q),$$

and assigns rewards $\{r_1, r_2, \ldots, r_G\}$. The group-relative advantage for output o_i is defined as

$$A_i = \frac{r_i - \text{mean}(\{r_1, \ldots, r_G\})}{\text{std}(\{r_1, \ldots, r_G\})}. \tag{1}$$

The policy π_θ is updated by maximizing the following objective:

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\min\left(\frac{\pi_\theta(o_i \mid q)}{\pi_{\theta_{\mathrm{old}}}(o_i \mid q)}A_i,\right.\right.$$
$$\left.\left.\mathrm{clip}\left(\frac{\pi_\theta(o_i \mid q)}{\pi_{\theta_{\mathrm{old}}}(o_i \mid q)}, 1-\epsilon,\ 1+\epsilon\right)A_i\right) - \beta D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})\right], \tag{2}$$

where ϵ controls the clipping range, β weights the KL regularization term, and π_{ref} is a fixed reference policy used to constrain policy drift. The KL divergence is defined as

$$D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}}) = \frac{\pi_{\mathrm{ref}}(o_i \mid q)}{\pi_\theta(o_i \mid q)} - \log\frac{\pi_{\mathrm{ref}}(o_i \mid q)}{\pi_\theta(o_i \mid q)} - 1. \tag{3}$$

GRPO supports both **outcome supervision**, where rewards are assigned only to final outputs, and **process supervision**, where intermediate reasoning steps are also rewarded.

Low-Rank Adaptation (LoRA) Hu et al. (2021) is a low-rank parameterization technique that eliminates the need for full-parameter fine-tuning. Recent work has explored LoRA for GRPO training (Wang et al., 2025).

2.2 Three-Phase GRPO Training

We hypothesize GRPO training follows a three-phase leaning curve because it is an online policy gradient algorithm. These three phases include Slow Reward Increase, Rapid Reward Increase, and Plateau phase.

- **Slow Reward Increase.** At the beginning of GRPO training, the policy remains close to the pre-trained initialization $\pi_{\theta_{old}}$. High-reward trajectories are relatively rare under this distribution, resulting in limited signal for policy improvement.
- **Rapid Reward Increase.** As training progresses, the policy gradually increases the probability of trajectories associated with higher rewards. This shifts the sampling distribution toward more successful responses, strengthening the gradient signal and accelerating learning.
- **Plateau.** Eventually the policy generates predominantly high-reward samples. As reward variance within each group decreases, the normalized GRPO advantages A_i diminish, resulting in smaller gradients and a saturation of learning progress.

Early Stopping Condition. The existence of a plateau suggests the possibility of early stopping. Equation 4 formalizes the early stopping condition by bounding the tail change ratio of the reward. The window size, k, is a critical hyperparameter which can be a fraction of the computational step in an epoch.

$$\frac{E[r_t] - E[r_{t-k}]}{E[r_t] - E[r_1]} \leq \delta \tag{4}$$

3 Experiment

To ensure reproducibility under limited computational budgets, we design all experiments to use minimal hardware while preserving scientific rigor. We combine Low-Rank Adaptation (LoRA) with Group Relative Policy Optimization (GRPO) to fine-tune open-source language models on publicly available reasoning datasets.

Model Selection. We evaluate four representative instruction-tuned models spanning two architectural families and multiple scales: **Llama family**: *Llama-3.2-3B* and *Llama-3.1-8B*; and **Qwen family**: *Qwen-2.5-3B* and *Qwen-2.5-7B*. This selection enables controlled analysis of architectural and scaling effects while remaining computationally feasible.

Dataset. We train on the MATH-LightEval subset of the OpenThoughts dataset (Guha et al., 2025; Hendrycks et al., 2021). This dataset provides high-quality, diverse reasoning problems and is fully public, ensuring reproducibility and comparability.

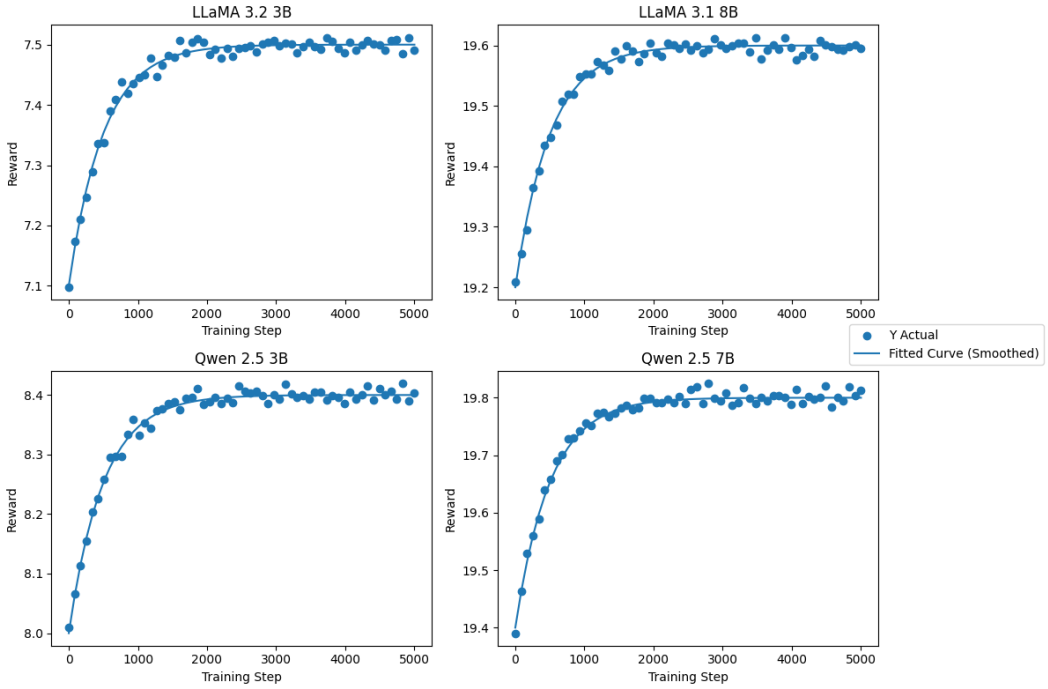

Figure 1: GRPO training reward convergence across all model configurations. All four models exhibit consistent sigmoid-shaped learning curves with similar phase transitions despite differences in parameter count and architectural family.

Training. We optimize all models using GRPO, which extends PPO-style objectives to group-based sampling. GRPO computes advantages using within-group baselines, eliminating the need for a separate critic model, directly optimizes task rewards, and enforces policy stability through clipped importance ratios. We apply LoRA to quantized models to enable parameter-efficient fine-tuning. This setup substantially reduces memory usage and training cost while preserving model capacity. Training only low-rank adapter parameters also enables modular control over reasoning behavior without modifying base model weights.

3.1 Research Questions

We seek to understand following research questions.

RQ1. Does GRPO training follow a three phase dynamics of slow reward increase, rapid reward increase, and plateau across different model sizes and architectures?

RQ2. Do training dynamics across different model sizes and architectures converge to a universal pattern?

4 Results

Figure 1 summarizes GRPO training dynamics across all four model configurations. Despite differences in model size and architectural family, all runs exhibit strikingly similar reward trajectories when plotted against normalized training progress. In particular, GRPO training consistently follows a smooth, sigmoid-shaped curve with well-aligned phase transitions across models.

4.1 Consistent Learning Phases Across Models (RQ1)

Across all settings, GRPO reward trajectories decompose into three distinct phases:

1. **Slow Reward Increase (0.0–0.1 normalized steps).** Reward improves slowly during early training, indicating that the policy is primarily adapting to the reward structure and stabilizing its behavior under the GRPO objective.
2. **Rapid Reward Increase. (0.1–0.2 normalized steps).** All models exhibit a sharp increase in reward over a narrow training window. This phase accounts for the majority of total reward gains and represents the most compute-efficient stage of GRPO training. The inflection point is tightly clustered across models, occurring near $t \approx 0.2$.
3. **Plateau (beyond 0.2 normalized steps).** After the inflection point, reward improvements diminish rapidly and trajectories converge to a stable plateau. Continued training in this regime yields marginal gains despite substantial additional compute.

A key empirical observation is that roughly 70–80% of cumulative reward improvement is achieved prior to the Plateau phase, even though that phase accounts for the majority of the total training compute. This pattern holds consistently across both Llama and Qwen families and across model scales.

4.2 Possible Universal Dynamics of Training Reward Evolution (RQ2)

As shown in Figure 1, the resulting curves closely match observed training dynamics, particularly in the saturation regime. Importantly, the fitted model predicts the onset of diminishing returns early in training, well before most computation is spent. There is a possibility of the existence of a universal GRPO reward law, as shown in 5:

$$R(t, M) = R_{\text{steady}} \left(1 - \exp\left(-\frac{t}{M^{0.3}}\right)\right) \tag{5}$$

Figure 2 shows the normalized GRPO training reward $R(t)/R_{\text{steady}}$ as a function of training steps on a logarithmic scale. Across all model families and sizes, the curves exhibit a consistent sigmoidal convergence pattern, characterized by rapid early-stage improvement followed by gradual saturation. By normalizing with respect to R_{steady}, we remove differences in absolute reward scale and reveal that the underlying learning dynamics are largely shared across architectures. While minor deviations are observed in the early phase, these differences are small and suggest only a weak dependence on model size. Overall, the empirical trends are consistent with the proposed scaling form $R(t, M) = R_{\text{steady}} \left(1 - \exp(-t/M^{0.3})\right)$, indicating that convergence behavior can be captured by a simple exponential family parameterized by model size.

Implications for Compute-Efficient GRPO Training. The consistency of GRPO reward dynamics across models suggests that early stopping decisions can be made in a principled, data-driven manner. Since the inflection point and saturation onset occur at similar normalized training progress across architectures and scales, continued training beyond approximately 70–80% of a single epoch is unlikely to yield meaningful reward improvements.

These results indicate that a substantial fraction of GRPO compute is spent in a regime of diminishing returns. By forecasting reward saturation using early training signals, practitioners can terminate training near the predicted plateau without sacrificing final performance, achieving significant reductions in reinforcement fine-tuning cost.

5 Discussion

5.1 Model Scale as the Primary Driver of Final Performance

While our results show consistent training dynamics across all models, final achievable reward scales strongly with model size. Larger models converge to substantially higher reward plateaus than their smaller counterparts within the same architectural family. This suggests that, under GRPO, reinforcement fine-tuning primarily amplifies the representational capacity already present in the base model rather than compensating for limited scale.

From a practical perspective, these findings imply diminishing returns from extended GRPO training on small models: once early saturation is reached, additional compute is unlikely to close the

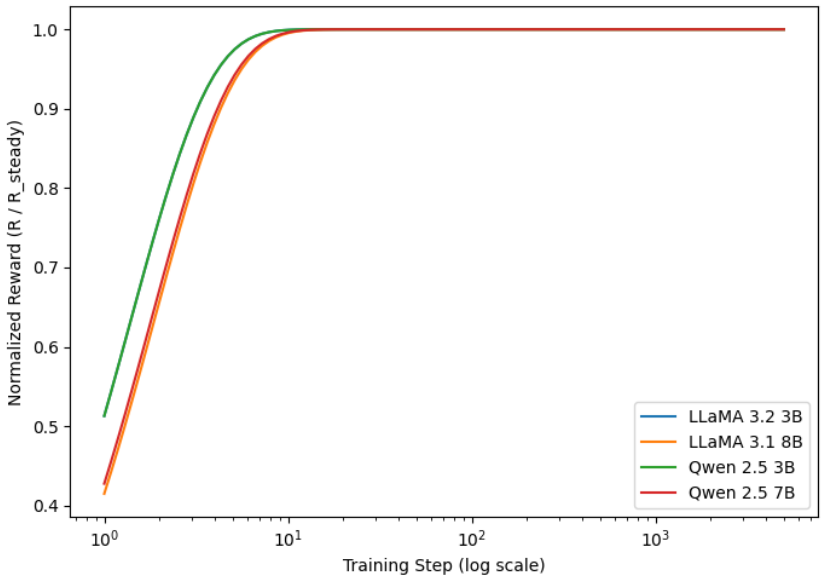

Figure 2: GRPO possibly exhibits a universal training law: normalized reward versus training steps (log scale) yields near-collapse of trajectories across model families and sizes onto a single curve.

performance gap relative to larger models. As a result, model size should be treated as a first-order design choice when planning reinforcement-based post-training.

5.2 Limited Influence of Architectural Family

Across both Llama and Qwen families, models of comparable size achieve similar final reward levels under GRPO. This consistency indicates that, for the reasoning tasks studied, architectural differences play a secondary role relative to parameter count once models are instruction-tuned and optimized with reinforcement learning.

This observation simplifies post-training decisions for practitioners: when compute budgets constrain model size, selecting between architectural families of similar scale is unlikely to yield large differences in GRPO outcomes. Instead, factors such as tooling maturity, inference efficiency, and deployment constraints may be more consequential.

5.3 Universality of GRPO Training Dynamics

Although final reward levels vary with model size, the relative progression of GRPO training is remarkably consistent across models. All configurations exhibit similar inflection points and saturation behavior when measured in normalized training progress. This suggests that GRPO induces a broadly universal optimization trajectory, largely independent of scale or architecture.

Such universality is important because it enables transferable heuristics across models. Observations made early in training on one configuration can be used to anticipate later-stage behavior on others, reducing the need for exhaustive trial-and-error experimentation.

5.4 Implications for Compute-Efficient Post-Training

The predictability of GRPO reward dynamics enables principled reductions in reinforcement fine-tuning cost. Since saturation occurs at similar normalized progress across models, continued training beyond this point primarily expends compute in a regime of diminishing returns. Early stopping based on predicted saturation therefore offers a straightforward mechanism for avoiding unnecessary computation.

More broadly, these results suggest a shift in how reinforcement fine-tuning compute should be allocated. Rather than over-training a single configuration, practitioners may achieve better overall outcomes by terminating training early and reallocating resources toward model scaling, hyperparameter exploration, or data curation.

5.5 Practical Guidance for Model Selection and Training

Taken together, our findings provide several actionable guidelines for GRPO-based post-training: (i) prioritize model scale over architectural family when targeting higher final performance, (ii) monitor early training dynamics to forecast saturation and determine stopping points, and (iii) treat reinforcement fine-tuning as a compute-sensitive refinement stage rather than an open-ended optimization process.

These insights reinforce the value of lightweight, predictive approaches to managing post-training compute and highlight the opportunity for more systematic, cost-aware reinforcement fine-tuning pipelines.

6 Related Work

Reinforcement Fine-Tuning for Large Language Models. Reinforcement fine-tuning (RFT) has become a central component of modern large language model (LLM) alignment and reasoning systems. Early work on Reinforcement Learning from Human Feedback (RLHF) established the use of policy optimization algorithms such as Proximal Policy Optimization (PPO) to align pretrained language models with human preferences and task-specific objectives (Ziegler et al., 2019; Ouyang et al., 2022b). Subsequent systems demonstrated that reinforcement-based post-training can substantially improve reasoning, instruction following, and safety, but at significant computational cost (Bai et al., 2022; Touvron et al., 2023).

Recent work has focused on improving the efficiency and stability of RFT. Group Relative Policy Optimization (GRPO) removes the need for an explicit critic by using group-relative baselines, reducing memory and optimization overhead while retaining PPO-style stability guarantees (Yuan et al., 2024). GRPO and related variants have been successfully applied to large-scale reasoning models, including DeepSeek-Math and DeepSeek-R1, where reinforcement learning plays a critical role in achieving strong mathematical and symbolic reasoning performance (Shao et al., 2024; Guo et al., 2025). Despite these advances, reinforcement fine-tuning remains substantially more expensive than supervised fine-tuning due to on-policy sampling, repeated rollouts, and long-context generation.

Compute Efficiency and Cost Reduction in Post-Training. The high cost of reinforcement-based post-training has motivated a growing body of work on improving training efficiency. Prior efforts have explored parameter-efficient fine-tuning methods such as LoRA to reduce memory and optimization costs during both supervised and reinforcement learning stages (**?**). Other work has emphasized systems-level optimizations, including more efficient inference engines, memory management, and parallelization strategies to reduce the overhead of rollout generation and reward evaluation (Kwon et al., 2023).

While these approaches reduce constant factors, they do not address the question of whether all reinforcement fine-tuning steps are necessary. In practice, many large-scale training pipelines run RFT for fixed budgets or epochs, often without principled criteria for determining when additional training yields diminishing returns. Our work complements systems-level optimizations by focusing on the *temporal structure* of GRPO training itself, identifying predictable saturation behavior that can be exploited to avoid unnecessary computation.

Learning Curves, Scaling Laws, and Early Stopping. Understanding learning dynamics through empirical scaling laws has a long history in machine learning. Power-law relationships between model size, data, compute, and performance have been extensively studied in supervised and self-supervised learning (Kaplan et al., 2020). In reinforcement learning, prior work has observed diminishing returns and characteristic learning phases in policy optimization, though these analyses typically focus on final performance rather than training efficiency (Schulman et al., 2017a).

Early stopping is a classical technique in supervised learning, commonly used to prevent overfitting and reduce wasted computation once validation performance plateaus (Prechelt, 2002). However, applying early stopping to reinforcement fine-tuning is non-trivial due to noisy reward signals, delayed credit assignment, and the interaction between reward maximization and policy regularization. As a result, early stopping remains underexplored in large-scale RFT settings.

Our work differs from prior learning-curve analyses by providing an empirical characterization of GRPO reward dynamics across multiple architectures and scales, showing that reward trajectories exhibit consistent, sigmoid-shaped structure and saturate early in normalized training progress. We further demonstrate that this structure can be captured by a simple parametric model, enabling reliable prediction of diminishing returns early in training. This positions our approach as a lightweight, empirically grounded alternative to fixed-budget reinforcement fine-tuning.

Positioning of This Work. In contrast to prior work that primarily aims to improve the final performance of reinforcement-trained LLMs, our focus is on *compute efficiency without sacrificing reward*. By analyzing GRPO training dynamics directly, rather than proposing new objectives or algorithms, we identify a practical opportunity for early stopping that is immediately applicable to existing GRPO pipelines. Our findings suggest that a substantial fraction of reinforcement fine-tuning compute can be eliminated through principled stopping criteria, complementing ongoing efforts in parameter-efficient training and systems optimization.

7 Conclusion and Future Work

In this work, we presented an empirical study of GRPO post-training dynamics with the goal of reducing unnecessary reinforcement fine-tuning compute. Across multiple open-source models spanning different sizes and architectural families, we showed that GRPO reward trajectories exhibit highly regular structure and saturate early in training. Despite large differences in final reward levels, the relative progression of learning follows a consistent pattern across models, enabling reliable prediction of when marginal gains diminish.

By fitting a simple parametric model to early training signals, we demonstrated that the onset of saturation can be forecasted before most computation is consumed. This enables principled early stopping strategies that significantly reduce GRPO compute without sacrificing final reward. Our findings suggest that reinforcement fine-tuning for reasoning models is often over-extended in practice and that lightweight, predictive approaches can substantially improve post-training efficiency.

This study opens several directions for future research. First, while our analysis focuses on GRPO, it would be valuable to examine whether similar early saturation dynamics and predictive structures arise in other reinforcement fine-tuning algorithms, including PPO-based RLHF and DPO-style preference optimization. Second, extending the analysis to larger model scales and more diverse task domains—such as code generation, tool use, and long-horizon planning—would help assess the generality of the observed dynamics beyond mathematical reasoning benchmarks. Third, integrating predictive stopping criteria directly into training pipelines remains an open challenge. Adaptive training schedules that dynamically adjust rollout budgets, KL regularization, or batch sizes in response to early saturation signals may further improve compute efficiency beyond simple early termination. Finally, reward is not the only informative training signal; other metrics, including training and validation losses as well as KL divergence, may also exhibit predictive dynamics and warrant systematic investigation.

Acknowledgment

We thank Datta Nimmaturi for discussions for an earlier version of the work (Nimmaturi et al., 2025), when he was at Nutanix. We also like to acknowledge Johnu George.

REFERENCES

Yuntao Bai, S. Jones, Kamal Ndousse, D. Almeida, David Farhi, Jared Kaplan, Jan Leike, John Schulman, Long Ouyang, Jeff Wu, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Léon Bottou. Stochastic gradient descent tricks. *Neural Networks: Tricks of the Trade*, pp. 421–436, 2012.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models, 2025. URL https://arxiv.org/abs/2506.04178.

Jinlin Guo, Ming Qiu, Yifan Gu, Junyang Chen, Yuxiao Wang, Xiaohua Hu, Wei Wang, Yang Yu, Lei Zhang, Xiaodong Li, et al. Deepseek-r1: Scaling reinforcement learning for large language model reasoning. *arXiv preprint arXiv:2501.02000*, 2025.

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. *arXiv preprint arXiv:2103.03874*, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

Datta Nimmaturi, Vaishnavi Bhargava, Rajat Ghosh, Johnu George, and Debojyoti Dutta. Predictive scaling laws for efficient grpo training of large reasoning models, 2025. URL https://arxiv.org/abs/2507.18014.

Long Ouyang, Jeff 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. *arXiv preprint arXiv:2203.02155*, 2022a.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022b. URL https://arxiv.org/abs/2203.02155.

Lutz Prechelt. Early stopping—but when? *Neural Networks: Tricks of the Trade*, pp. 55–69, 1998.

Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pp. 55–69. Springer, 2002.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1–11, 2017a. arXiv:1707.06347.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017b. URL https://arxiv.org/abs/1707.06347.

Ziang Shao, Yifei Liu, Kai Zhang, Weilin Zhao, Wenqiang Luo, Yikang Zhang, Shuhe Wang, Jiajun Zhang, Peng Zhou, Yang Chen, et al. Deepseekmath: Boosting llms for mathematical reasoning with reinforcement learning. *arXiv preprint arXiv:2402.03300*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Tim Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Shangshang Wang, Julian Asilis, Omer Faruk Akgul, Enes Burak Bilgin, Ollie Liu, and Willie Neiswanger. Tina: Tiny reasoning models via lora. *arXiv preprint arXiv:2504.15777*, 2025.

Hao Yuan, Berf Anahtarci, H. Inan, A. Zeng, L. Hou, M. Chen, S. Zhang, C. Wu, and C. Zhang. Grpo: Group relative policy optimization for efficient llm alignment. *arXiv preprint arXiv:2402.08714*, 2024.

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.