Understanding Data Influence in Reinforcement Finetuning

Haoru Tan[1] **Xiuzhe Wu**[5†] **Sitong Wu**[3] **Shaofeng Zhang**[4]

Yanfeng Chen[2] **Xingwu Sun**[2] **Jeanne Shen**[5] **Xiaojuan Qi**[1†]

[1]The University of Hong Kong [2]Hunyuan Team, Tencent

[3]The Chinese University of Hong Kong [4]University of Science and Technology of China

[5]Stanford University

hrtan@eee.hku.hk, stone-wu@link.cuhk.edu.hk

Abstract

Reinforcement fine-tuning (RFT) is essential for enhancing the reasoning and generalization capabilities of large language models, but its success heavily relies on the quality of the training data. While data selection has been extensively studied in supervised learning, its role in reinforcement learning, particularly during the RFT stage, remains largely underexplored. In this work, we introduce RFT-Inf, the first influence estimator designed for data in reinforcement learning. RFT-Inf quantifies the importance of each training example by measuring how its removal affects the final training reward, offering a direct estimate of its contribution to model learning. To ensure scalability, we propose a first-order approximation of the RFT-Inf score by backtracking through the optimization process and applying temporal differentiation to the sample-wise influence term, along with a first-order Taylor approximation to adjacent time steps. This yields a lightweight, gradient-based estimator that evaluates the alignment between an individual sample's gradient and the average gradient direction of all training samples, where a higher degree of alignment implies greater training utility. Extensive experiments demonstrate that RFT-Inf consistently improves reward performance and accelerates convergence in reinforcement fine-tuning.

1 Introduction

Reinforcement Fine-Tuning (RFT) [1–4] has emerged as a powerful technique for refining the capabilities of large language models (LLMs) [1, 2, 5, 3] by leveraging reward-driven optimization [6, 7]. Unlike Supervised Fine-Tuning (SFT) [8–10], which guides model behavior through direct supervision [11], RFT enables models to learn more robust reasoning strategies and generalize beyond seen data [1, 11, 12]. Recent studies have confirmed the critical role of RFT in advancing the reasoning capabilities of LLMs, particularly in complex tasks requiring multi-step inference and decision-making.

Background. A fundamental aspect of successful RFT lies in the construction and utilization of high-quality training data [1, 13, 14]. However, current understanding of data's role in RFT remains

[†]Corresponding Authors

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

largely heuristic or empirical, often guided by reward trends observed during training [13, 15] or the perceived difficulty of training examples [16, 17]. Despite its importance, there remains a lack of principled, quantitative methods for assessing the impact of individual samples on RFT outcomes. Addressing this gap is essential for designing more efficient and effective training pipelines.

Our Method. In this work, we propose RFT-Inf for quantifying the influence of the individual training sample in reinforcement fine-tuning. The core idea behind RFT-Inf is to measure how the removal of a single sample influences the final training reward, providing a direct estimate of its contribution to the model's learning process. While a brute-force approach would require re-training the model after removing each sample, such computation is prohibitively expensive. To address this, we introduce a scalable, first-order approximation of the RFT-Inf score derived from backtracking through the optimization process and applying temporal differentiation to the sample-wise influence term, along with a first-order Taylor approximation to adjacent time steps. This results in a lightweight gradient-based estimator with linear complexities that measures the alignment between a sample's gradient and the average gradient direction of all training samples across different training stages, see Eq.(5). A higher alignment corresponds to a higher RFT-Inf score, indicating greater sample importance. Intuitively, if a sample's gradient is closely aligned with the overall optimization direction, it is more representative and beneficial to training. Removing such a sample would degrade performance, highlighting its importance. We further provide a theoretical worst-case error bound for our first-order approximation, offering insights into its reliability.

We validate RFT-Inf across multiple benchmarks, demonstrating its effectiveness in identifying high-impact samples and improving final model performance. Using RFT-Inf for data selection yields significant performance gains over baselines that rely on full datasets or heuristic selection strategies. Notably, in mathematical reasoning tasks, we only require about **20%** data selected by our data influence estimator to achieve more stable training and superior results compared to using the entire dataset. Compared to various heuristic or rule-based data selection methods, our approach significantly outperforms them in both performance and generalization.

2 Related Work

Reinforcement Fine-Tuning. Recent breakthroughs in large language models (LLMs) [2, 1, 3] have been significantly driven by Reinforcement Fine-Tuning (RFT) [6, 7, 1, 18, 19], a technique that refines LLM behavior using reward-guided optimization. Unlike Supervised Fine-Tuning (SFT) [8–10], which aligns model outputs with labeled responses [11], RFT employs reinforcement learning to adapt models based on feedback signals. Typically, RFT relies on rule-based reward functions and reinforcement learning algorithms such as Proximal Policy Optimization (PPO) [6]. For instance, DeepSeek-R1 [1] adopts Group Relative Policy Optimization (GRPO) [7], using binary rewards to indicate answer correctness in tasks like mathematics [20] and coding [21], achieving impressive performance. Several studies suggest that RFT enhances cognitive abilities such as reflection and self-correction [22, 1], while also improving generalization across tasks [11]. Much of the current RFT research focuses on algorithmic improvements. For example, VinePPO [23] addresses limitations in PPO's value networks for complex reasoning tasks by introducing unbiased Monte Carlo estimates for better credit assignment. Other works aim to simplify GRPO, e.g., by removing the KL-divergence term to achieve more robust empirical results [18, 24].

Most recently, some emerging research explores data-centric approaches to improve RFT [13, 14, 18, 15]. For example, LIMR [13] selects samples by analyzing changes in reward trends, while the Historical Variance Score [15] prioritizes data with high reward variability, suggesting that such samples are more impactful. Other works incorporate sample difficulty as a selection criterion to strike a balance between learning efficiency and model robustness [16, 17]. Despite these advances, existing work on data selection for RFT is largely heuristic and lacks principled methods for quantifying the influence of individual training samples. As a result, most methods rely on empirical intuition rather than theoretical insight. This highlights an urgent need for data-centric frameworks that enable quantitative analysis of sample importance during RFT, ultimately guiding more effective training.

Data Influence Analysis. Analyzing the impact of individual training samples is a longstanding problem in machine learning [25, 26]. The canonical approach involves removing a sample and re-training the model to observe the effect—an idea that is precise but computationally prohibitive [26–28]. To overcome this, many works propose theoretical estimators that approximate sample

influence without full retraining. A seminal method by Koh and Liang [26] estimates sample influence using the inverse Hessian-gradient product, assuming the model parameters change smoothly under small data perturbations. Building on this, subsequent work has improved the efficiency of Hessian computations through decomposition techniques [29–32], incorporated group-level effects [33, 34], or integrated influence estimation with training procedures [35, 36].

However, these approaches are predominantly designed for supervised learning and often depend on key assumptions: a well-defined, twice-differentiable objective function, and convexity of the loss landscape [26, 37, 38]. Such conditions facilitate tractable approximation but do not hold in reinforcement learning, where the optimization objective is to maximize expected reward rather than minimize a loss. Consequently, applying these influence functions directly to reinforcement learning is challenging. Moreover, data influence research in supervised learning has been leveraged for tasks such as data selection [26, 39–43] and attribution [29, 44–49], but similar progress in reinforcement learning remains limited. This disconnect motivates the development of new, scalable influence estimation tools tailored specifically for RFT scenarios.

3 Preliminaries

This section introduces the foundational concepts of reinforcement fine-tuning (RFT) [2, 1, 14, 7, 50, 3] as applied to large language models (LLMs) [51, 52]. We consider a training dataset $\mathcal{Z} = z_i = (s_i, y_i)_{i=1}^N$ composed of structured question–answer pairs, where each sample z_i consists of a prompt or question s_i and a corresponding deterministic ground-truth answer y_i, e.g., a math problem and its correct solution. Let π_θ denote the policy model parameterized by θ, representing the LLM. In this formulation, the input state s corresponds to the prompt, and the action a is the model's generated response. A scalar reward r measures the quality of the response a for the given prompt s. The objective of RFT is to fine-tune the LLM using reinforcement learning to produce correct and high-quality answers [6, 7]. A commonly adopted reward scheme directly compares the model output with the ground truth [1], assigning a reward of $r = 1$ if the output matches the correct answer and $r = -1$ otherwise.

Objective Function. The standard RFT objective is to optimize the policy π_θ by maximizing the expected advantage:

$$\mathcal{J}(\theta) = \mathbb{E}_{(s,a)}\Big[A(s,a)\Big], \tag{1}$$

where the advantage function is defined as $A(s,a) = r(s,a) - v$, representing the relative merit of taking action a in state s compared to a baseline value v. Optimization proceeds via stochastic gradient ascent: $\theta_{t+1} = \theta_t + \eta_t \nabla_\theta J(\theta_t)$, where η_t is the learning rate and the gradient of $\mathcal{J}(\theta)$ is derived as: $\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{(s,a)}\Big[A(s,a) \cdot \nabla_\theta \log \pi_\theta(a|s)\Big]$.

Modern RFT implementations typically enhance this vanilla formulation to improve stability and performance. We now briefly review two widely used variants: Proximal Policy Optimization (PPO) [6] and Group Relative Policy Optimization (GRPO) [7, 1], which serve as the foundation for many current RFT advancements [18, 12, 50, 53].

Proximal Policy Optimization (PPO) [6] introduces a clipping mechanism to stabilize training by constraining policy updates. Its objective is formulated as:

$$\mathcal{J}^{\text{PPO}}(\theta) = \mathbb{E}_{(s,a)}\left[\min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}A(s,a), \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}, 1-\epsilon, 1+\epsilon\right)A(s,a)\right)\right], \tag{2}$$

where ϵ is a clipping hyperparameter. $\pi_\theta(a|s)$ is the probability of taking action a in state s according to the current policy with parameters θ, and $\pi_{\theta_{\text{old}}}(a|s)$ is the probability of taking action a in state s according to the old policy with parameters θ_{old}.

Group Relative Policy Optimization (GRPO) [7, 1] extends PPO by defining advantages through relative comparisons within a group of sampled responses for the same prompt. For each prompt s, a group of outputs $a_1, \ldots, a_G$ is sampled from the old policy $\pi_{\theta_{\text{old}}}$. The GRPO objective is:

$$\mathcal{J}^{\text{GRPO}}(\theta) = \\ \mathbb{E}_{(s,a_i)}\left[\frac{1}{G}\sum_{i=1}^{G}\min\left(\frac{\pi_\theta(a_i|s)}{\pi_{\theta_{\text{old}}}(a_i|s)}A(s,a_i), \text{clip}\left(\frac{\pi_\theta(a_i|s)}{\pi_{\theta_{\text{old}}}(a_i|s)}, 1-\epsilon, 1+\epsilon\right)A(s,a_i)\right) - \beta\text{KL}\left(\pi_\theta \| \pi_{\text{ref}}\right)\right], \tag{3}$$

where ϵ is the clipping hyperparameter and β is the coefficient controlling the weight of the KL-divergence regularization to ensure the optimized model does not deviate excessively from the reference model (*e.g.*, the initial policy model); and $A(s, a_i) = \frac{r_i - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G)}$ is the advantage, where r_i corresponds to the reward of each output. GRPO eliminates the need for a separate value network, reducing computational cost and simplifying training. For more details, we refer readers to the original GRPO papers [7, 1].

4 Method

In this section, we introduce our proposed method. We begin by defining the data influence estimator and presenting a first-order approximation for efficient computation, along with a theoretical analysis of the approximation error. We then describe how this score can be used to select high-impact samples to improve the performance of RFT.

4.1 Data Influence in Reinforcement Finetuning

We propose a principled approach to assess the importance of each training sample by measuring its influence on the final performance of reinforcement fine-tuning through evaluating how model performance changes when the sample is removed from training.

Definition. Let $\mathcal{Z}$ denote the training dataset consisting of samples $z = (s, y)$, where s is a prompt and y is its deterministic answer (e.g., a math problem and its unique solution). Let $\mathcal{Z}/z$ denote the dataset excluding sample z, and let $\theta^*_{\mathcal{Z}}$ and $\theta^*_{\mathcal{Z}/z}$ denote the parameters obtained by training on $\mathcal{Z}$ and $\mathcal{Z}/z$, respectively. Assuming a reward function that assigns $r = 1$ if the model output matches the ground truth and $r = -1$ otherwise [1, 14], we define the data influence estimator of a sample z as:

$$\mathcal{D}(z) = \mathcal{J}\left(\theta^*_{\mathcal{Z}}\right) - \mathcal{J}\left(\theta^*_{\mathcal{Z}/z}\right), \tag{4}$$

where $\mathcal{J}(\cdot)$ denotes the expected reward. This score is directional; it not only quantifies the importance of a sample but also distinguishes whether the sample is beneficial or detrimental to training. A large positive value of $\mathcal{D}(z)$ indicates that removing the sample significantly reduces the final reward, suggesting that the sample is beneficial. Conversely, a negative value implies that the sample is harmful, as removing it would increase the reward. Values near zero indicate negligible influence. Thus, this metric provides a direct and interpretable measure of each sample's contribution to RFT.

Efficient First-order Approximation. Precisely Direct computation of $\mathcal{D}(z)$ is impractical, as it requires retraining the model for each sample. To address this issue, we introduce a first-order approximation of the RFT-Inf score by backtracking the score defined in Eq. (4) through the optimization process. We apply temporal differentiation and a first-order Taylor approximation to adjacent time steps. The detailed derivation is provided in the supplementary material. Below is the approximation with linear complexities:

$$\hat{\mathcal{D}}(z) = \sum_t \frac{2\eta_t}{N} \left\langle \mathcal{G}_z^{(t)},\ \mathcal{G}_{\mathcal{Z}}^{(t)} \right\rangle, \tag{5}$$

where $\left\langle :, : \right\rangle$ is the inner-product operator, $\mathcal{Z}/z$ represents the dataset $\mathcal{Z}$ excluding sample $z : (s, y)$, and η_t is the learning rate at time step t. T and N denote the maximum number of time steps and the training set size, respectively. $\mathcal{G}_z^{(t)}$ represents the policy gradient contributed by sample $z : (s, y)$ at time step t. Typically, for most RL algorithms, we have $\mathcal{G}_z^{(t)} = \hat{A}(s, a)\nabla \log \pi_{\theta_t}(a|s)$, where $\hat{A}(s, a)$ is the advantage function. In the case of GRPO [7], $\mathcal{G}_z^{(t)} = \sum_i^G \hat{A}(s, a_i)\nabla \log \pi_{\theta_t}(a_i|s)$, where G is the group size. The other gradient term $\mathcal{G}_{\mathcal{Z}}^{(t)}$ is defined as the gradient over all samples, that is, $\mathcal{G}_{\mathcal{Z}}^{(t)} = \sum_{z \in \mathcal{Z}} \mathcal{G}_z^{(t)}$.

This approximation evaluates the alignment between a sample's gradient and the overall training direction. A higher value indicates that the sample's influence aligns with the global training trajectory, marking it as representative and valuable. This can be interpreted as follows: if the sample's gradient $\mathcal{G}_z^{(t)}$ exhibits a high degree of alignment with the average gradient vector $\mathcal{G}_{\mathcal{Z}}^{(t)}$ for all time steps, then

Algorithm 1: Data Selection Pipeline

Require: A dataset $\mathcal{Z} = \{(s_i, y_i)\}_{i=1}^{N}$ and selection budget δ; A large language model π_θ and a reinforcement fine-tuning algorithm `RFT`

1: Train the model π_θ for E epochs on $\mathcal{Z}$ using `RFT` and save checkpoints: $\{\theta^1, \ldots, \theta^E\} \leftarrow \texttt{RFT}(\pi_\theta, \mathcal{Z}, E)$
2: **for** each sample $z_i = (s_i, y_i) \in \mathcal{Z}$ **do**
3: Calculate the data influence estimator $\hat{\mathcal{D}}(z_i)$ according to Eq. (5)
4: **end for**
5: Select the top-δ samples based on their data influence estimators to form the new subset $\mathcal{Z}_{\text{new}}$
6: **return** The subset model $\mathcal{Z}_{\text{new}}$

optimizing the network using sample z will have a similar effect optimizing with the full training set. This indicates that the sample z is a representative and important sample. Therefore, this sample is regarded as a more representative and significant one.

4.2 Error Analysis

First, we provide a worst-case error bound for the proposed approximation, demonstrating its robustness under some mild assumptions:

Proposition 1. *Under the assumptions that the log-likelihood function* $\log \pi_\theta(a|s)$ *exhibits* ℓ*-Lipschitz continuity and that the adventure value is upper-bounded by* A_{max}*, the approximation error of Eq.* (5) *is bounded as follows:*

$$\left|\mathcal{D}(z) - \hat{\mathcal{D}}(z)\right| \leq \mathcal{O}\Big(\Big[\frac{\eta_{\max}(4N+4)}{N}(\ell A_{\max})^2 + 2\eta\ell^2 A_{\max}\Big]T\Big), \tag{6}$$

where T *denotes the maximum number of iterations and* N *is the number of all training data.*

The above proposition formalizes the relationship between the approximation error and influencing factors, particularly the training duration T. To mitigate errors in practice, we adopt a strategy of early stopping during the surrogate training phase, avoiding full convergence. Specifically, we limit the number of update epochs to a small value (e.g., two epochs) to ensure the effective execution of surrogate training while also controlling the theoretical maximum error. Remarkably, the resulting error bound remains tighter than many known bounds for influence functions in supervised learning [41, 54], despite the added challenges posed by the reinforcement learning setting.

4.3 Complexity Analysis

Then, we discuss the computational complexity of our estimator. Eq. (5) offers an efficient approximation of the data influence estimator with linear computational complexity. Further, rather than summing over all training steps, we adopt a Monte Carlo sampling strategy [55, 56], selecting a small number of evenly spaced time steps to estimate the full summation since checkpoints from nearby iterations will be similar. Let the number of selected checkpoints be C. By employing this strategy, the total computational complexity for scoring the entire dataset is $\mathcal{O}\Big(NE + NC\Big)$, where NE is the cost of surrogate training, and N represents the number of training samples and C denotes the number of checkpoints (which also corresponds to the number of sampled checkpoints, as we select one checkpoint for each epoch). This represents a substantial efficiency gain over the naive approach, which requires retraining the model for each sample and incurs a cost of $\mathcal{O}\Big((N-1)E\Big)$ per sample, resulting in $\mathcal{O}\Big(N(N-1)E\Big)$ overall. Leveraging practical implementations such as PyTorch's efficient gradient operations [57], or computing only final-layer gradients [55, 58], further enhances scalability.

4.4 Implementation Details

We now describe how to use the proposed data influence estimator to identify and select high-impact samples for reinforcement fine-tuning. The full pipeline is outlined in Algorithm 1. We are given a

structured dataset $\mathcal{Z} = \{z_i = (s_i, y_i)\}_{i=1}^{N}$ consisting of question–answer pairs, where each sample z_i comprises a prompt or question s_i and a corresponding deterministic answer y_i. We are also provided with a pre-trained large language model, serving as the policy model π_θ, and a reward evaluation mechanism. A common and effective approach to reward calculation involves directly comparing the generated answer to the ground truth y; for instance, a reward of $r = 1$ is assigned if the generated answer matches the ground truth, and $r = -1$ otherwise.

As shown in Algorithm 1: the policy model first undergoes a small number of reinforcement fine-tuning (RFT) epochs, namely E epochs, on the full training dataset. We refer to this training phase as surrogate training. The choice of reinforcement learning algorithm is flexible; for instance, one may use PPO [6], GRPO [7], or any other compatible method. At the end of each epoch, a model checkpoint is saved for later use. To reduce training costs, several practical strategies can be employed. These include setting a small number of epochs (e.g., $E = 2$), or applying parameter-efficient fine-tuning techniques such as LoRA [59] instead of full model fine-tuning, as recommended by recent studies [60, 31]. These techniques have been empirically shown to significantly reduce computational overhead without negatively impacting performance. After surrogate training, the saved checkpoints are used to compute per-sample gradients as well as the overall gradient across the dataset. The data influence estimator for each sample is then computed according to Eq. (5). Using the computed data influence estimators, we perform data selection based on a predefined budget δ, which specifies the size of the target subset. The top-δ samples with the highest data influence estimators are selected to form a refined training set, denoted as $\mathcal{Z}_{\text{new}}$. This curated subset is then used for the formal reinforcement fine-tuning (RFT) phase to obtain the final model parameters π_θ^*.

5 Experiments

We conducted comprehensive experiments to evaluate the effectiveness of our proposed method. Sec. 5.1 is the main experiment. Then, Sec. 5.2 provides detailed ablation studies to analyze the impact of key components in our approach.

5.1 Main Experiments

Models, Datasets, and Benchmarks. We utilized the dataset released by DeepScaleR [14], which is a comprehensive mathematical dataset compiled from multiple sources, with duplicates removed and data cleaned. This dataset includes AIME problems from 1984 to 2023 and AMC problems before 2023, along with questions from the Omni-MATH [61] and STILL [62] datasets, featuring problems from various national and international mathematics competitions. This training dataset contains approximately 40,000 math problem-answer pairs. To evaluate the reasoning abilities of the models, we utilize five different mathematics benchmarks: AIME24 [20], MATH-500 [63], AMC23 [64], Minerva [65], and OlympiadBench [66]. Our experiments encompass a variety of model configurations, including DeepSeek-R1-Distill-Qwen-1.5B [1], DeepSeek-R1-Distill-Qwen-7B [1], and Llama-3.2-3B-Instruct [67]. Unless otherwise specified, the default algorithm used is GRPO, with a group size set to 8.

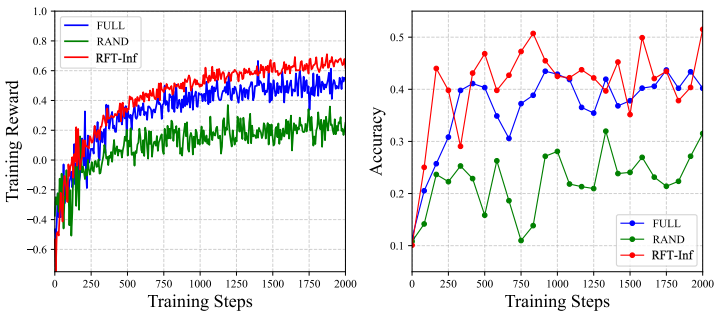

Figure 1: The reward and accuracy curve during learning on the subset selected by RFT-Inf (20% selection ratio). This experiment is conducted on DeepSeek-R1-Distill-Qwen-1.5B, and the selected benchmark is AIME24 [20].

Setups. The experiments were conducted using the PyTorch framework on two high-performance computing servers, each equipped with eight NVIDIA H200 GPUs. For our approach, we made the following settings: during the surrogate training phase, we performed LoRA training with a LoRA

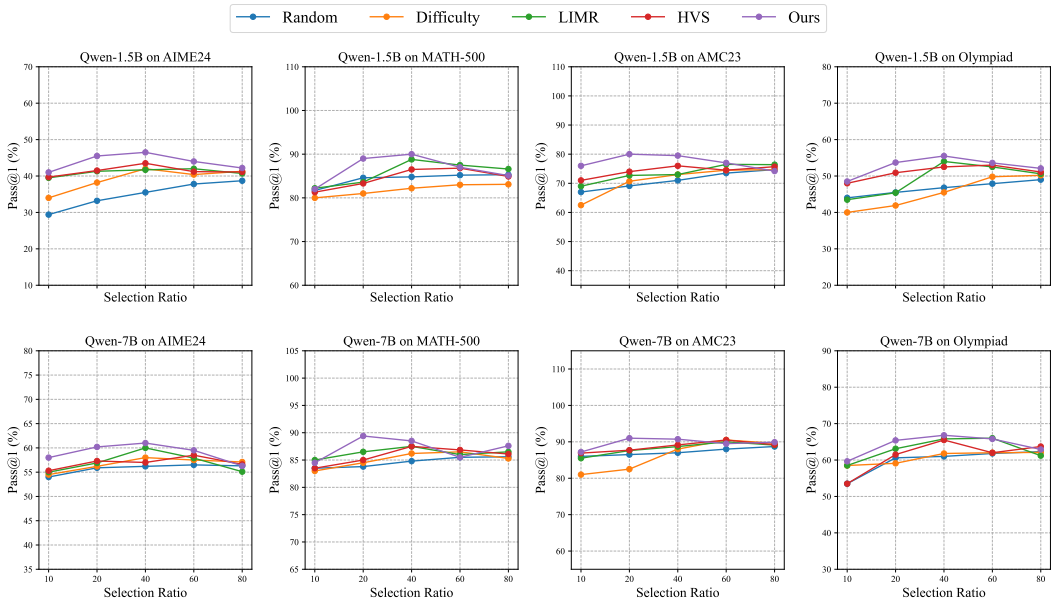

Figure 2: This figure presents the zero-shot pass@1 performance (%) of our proposed data selection method ("Ours") against several baselines (Random, Difficulty-based, LIMR, HVS) across four mathematical reasoning benchmarks (AIME24, MATH-500, AMC23, Olympiad) and two model sizes: Qwen-1.5B (DeepSeek-R1-Distill-Qwen-1.5B) and Qwen-7B (DeepSeek-R1-Distill-Qwen-7B). The performance is plotted as a function of the data Selection Ratio (from 10% to 80%), representing the budget constraint.

rank set to 16 and a total of 2 training epochs. We optimized the network using the AdamW optimizer with a constant learning rate of $1 \times e^{-6}$ and a weight decay of 0.1.

Baselines. We compare our method against several widely used baselines: (1) Random selection, where a subset of data is chosen uniformly at random. (2) Difficulty-driven selection, where the difficulty of each sample is estimated based on model performance. Specifically, we run inference for each sample with multiple models (DeepSeek-R1-Distill-Qwen-1.5B/7B/32B [1]), and use the average pass rate as a proxy for difficulty [17]. Based on this, we select those samples with median difficulty [68]. (3) LIMR [13], which determines sample importance by measuring the alignment between a sample's reward and the overall reward trend during training. (4) Historical Variance Score (HSV) [15], which quantifies sample importance by computing the variance in its reward values over the course of training.

To evaluate our method, we selected two models: DeepSeek-R1-Distill-Qwen-1.5B [1] and DeepSeek-R1-Distill-Qwen-7B [1]. Here, the model we use for scoring is consistent with the final trained model. Both models are derived from the Qwen series and have been distilled using DeepSeek-R1, demonstrating a solid foundational reasoning ability. We summarize the relevant experimental results in Fig. 2. Clearly, our method outperforms various baseline approaches under the most constrained budgets, particularly at very low selection ratios. Specifically, at the 20% selection ratio setting: for the 1.5B model type, our method achieves an accuracy of 45.5% in AIME24, outperforming all baseline methods, including LIMR [13] and HVS [15]. On MATH-500, our score of 89.0% is notably higher than the best baseline score of 86.2% from Full Data. In the case of the 20% selection ratio and the 7B model, our method further excels, achieving a score of 60.2% on AIME24, which is significantly higher than the next best baseline score of 57.3% from HVS [15].

5.2 Further study

In this subsection, we explore several key factors, including the number of checkpoints and the training method (either LoRA rank or full-parameter training), also along with the case study.

Number of Checkpoints. We employed a Monte Carlo strategy when approximating the estimator in Eq. (5), considering checkpoints from only a few time steps. In practice, we observed that as the

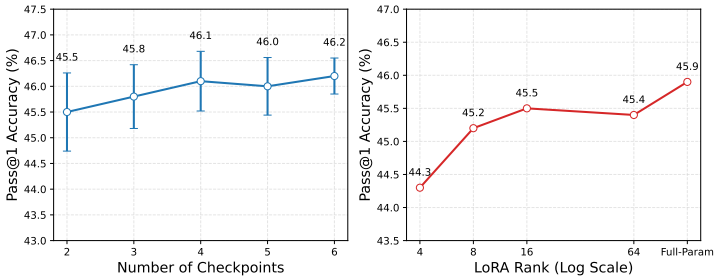

Figure 3: Ablative experimental results with DeepSeek-R1-Distill-Qwen-1.5B as the model and AIME24 as the benchmark. The performance metric is zero-shot Pass@1 Accuracy. (a) Pass@1 accuracy versus the number of checkpoints used, showing that increasing checkpoints stabilizes the training (decreasing standard deviation). (b) Pass@1 accuracy versus LoRA rank (log scale), demonstrating that performance generally improves with higher LoRA ranks, approaching the Full-Parameter result.

number of checkpoints increased, the final performance improved, see Fig. 3. This is understandable, as it effectively increases the sampling steps, thereby enhancing approximation accuracy. However, this also results in a significant decrease in scoring speed, and the improvement shows a trend of stagnation. Therefore, we opted for a relatively small number of checkpoints, specifically two, with each corresponding to one epoch.

LoRA-Rank. Our method requires training a surrogate model before scoring, using LoRA training with a rank of 16. LoRA (Low-Rank Adaptation) is a technique that efficiently fine-tunes large language models by incorporating low-rank decomposition into the weight updates, enabling effective adaptation with fewer parameters and reduced computational cost. This means that when approximating the estimator in Eq.(5), we do not utilize all model parameters. We tested the impact of using LoRA training and the size of the LoRA rank on final performance, see the results in Fig. 3. In conclusion, we found that using LoRA significantly improves efficiency compared to full-parameter training, without negatively impacting the performance of the data selected based on the final data influence estimator. Additionally, we discovered that even with a LoRA rank of 4, the final performance remains quite satisfactory.

Case study. In the following, we provide an illustrative examination of several cases, showing their assigned RFT-Inf scores and the resulting importance rankings (in descending order). For benchmarking, we include the data importance rankings generated by the alternative HVS [15] strategy. To establish an objective metric for task complexity, we use the pass rate, measured by the number of correct outputs from eight QwQ-32B [5] samples. Lower pass-rates correlate with higher question difficulty. Our analysis reveals a crucial distinction in data valuation: A high degree of concordance is found between RFT-Inf and HVS rankings for data points associated with extreme difficulty levels. For example, in both Case 1 and Case 4, which represent the spectrum's edges, both methods assign significantly low importance ranks, implying a general agreement that the marginal utility of such data is negligible. Crucially, the two ranking strategies exhibit substantial disparity when evaluating data points of intermediate or moderate difficulty. This disagreement in identifying the most valuable samples within the 'middle ground' of difficulty constitutes the fundamental driver behind the varying empirical performance observed in different data selection paradigms.

Case 1

Problem: Each vertex of a regular octagon is independently colored either red or blue with equal probability. The probability that the octagon can then be rotated so that all of the blue vertices end up at positions where there were originally red vertices is $\frac{m}{n}$, where m and n are relatively prime positive integers. What is $m+n$?

`Pass-rate-by-QwQ` [5]: $0/8$;
`Rank-by-RFT-DA`: **40245**;
`Rank-by-HVS` [15]: **39076**.

Case 2

Problem: Among the 900 residents of Aimeville, there are 195 who own a diamond ring, 367 who own a set of golf clubs, and 562 who own a garden spade. In addition, each of the 900 residents owns a bag of candy hearts. There are 437 residents who own exactly two of these things, and 234 residents who own exactly three of these things. Find the number of residents of Aimeville who own all four of these things.

`Pass-rate-by-QwQ` [5]: $5/8$;
`Rank-by-RFT-DA`: **3205**;
`Rank-by-HVS` [15]: **27855**.

Case 3

Problem: Let ABC be a triangle inscribed in circle ω. Let the tangents to ω at B and C intersect at point D, and let $\overline{AD}$ intersect ω at P. If $AB = 5$, $BC = 9$, and $AC = 10$, AP can be written as the form $\frac{m}{n}$, where m and n are relatively prime integers. Find $m + n$.

`Pass-rate-by-QwQ` [5]: $4/8$;
`Rank-by-RFT-DA`: **1129**;
`Rank-by-HVS` [15]: **30724**.

Case 4

Problem: Jen enters a lottery by picking 4 distinct numbers from $S = \{1, 2, 3, \cdots, 9, 10\}$. 4 numbers are randomly chosen from S. She wins a prize if at least two of her numbers were 2 of the randomly chosen numbers, and wins the grand prize if all four of her numbers were the randomly chosen numbers. The probability of her winning the grand prize given that she won a prize is $\frac{m}{n}$ where m and n are relatively prime positive integers. Find $m + n$.

`Pass-rate-by-QwQ` [5]: $8/8$;
`Rank-by-RFT-DA`: **39275**;
`Rank-by-HVS` [15]: **40236**.

6 Conclusion

In this study, we present RFT-Inf, a novel approach for data selection in reinforcement fine-tuning, aimed at finding those beneficial training examples while eliminating noisy or harmful ones. Central to RFT-Inf is a sample-level influence analysis that assesses the significance of each training example by evaluating how its exclusion impacts the final training reward, providing a clear indication of its contribution to model learning. To enhance scalability, we introduce a lightweight, gradient-based estimator with theoretical guarantees. Comprehensive experiments demonstrate that RFT-Inf consistently improves reward performance and accelerates convergence in reinforcement fine-tuning.

Limitations. While our method demonstrates strong performance across benchmarks, the experimental scale validated in this paper is still limited. In real-world scenarios, both the data and model sizes are significantly larger. In the future, we plan to secure additional computational resources to validate performance on a larger experimental scale.

7 Acknowledge

This work has been supported by the National Key R&D Program of China (Grant No. 2022YFB3608300), Hong Kong Research Grant Council - Early Career Scheme (Grant No. 27209621), General Research Fund Scheme (Grant No. 17202422, 17212923, 17215025), Themebased Research (Grant No. T45-701/22-R), and Shenzhen Science and Technology Innovation Commission (SGDX20220530111405040). Part of the described research work is conducted in the JC STEM Lab of Robotics for Soft Materials funded by The Hong Kong Jockey Club Charities Trust.

References

[1] 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.

[2] OpenAI. Learning to reason with llms, 2024.

[3] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[4] 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.

[5] Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, 2025.

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

[7] 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.

[8] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

[9] 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.

[10] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*, 2025.

[11] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

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

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

[14] 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. 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-19681902c1468005bed8ca303013a4e2, 2025. Notion Blog.

[15] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025.

[16] 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.

[17] Yunjie Ji, Sitong Zhao, Xiaoyu Tian, Haotian Wang, Shuaiting Chen, Yiping Peng, Han Zhao, and Xiangang Li. How difficulty-aware staged reinforcement learning enhances llms' reasoning capabilities: A preliminary experimental study. *arXiv preprint arXiv:2504.00829*, 2025.

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

[20] AIME. American invitational mathematics examination. https://artofproblemsolving.com/wiki/index.php/AIME-Problems-and-Solutions, 2024.

[21] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

[22] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

[23] 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.

[24] Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. Gpg: A simple and strong reinforcement learning baseline for model reasoning, 2025.

[25] R Dennis Cook. Assessment of local influence. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 48(2):133–155, 1986.

[26] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, 2017.

[27] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2018.

[28] Vitaly Feldman and Zhang Chiyuan. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in neural information processing systems*, 2020.

[29] Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023.

[30] Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[31] Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. *arXiv preprint arXiv:2310.00902*, 2023.

[32] Yegor Klochkov and Yang Liu. Revisiting inverse hessian vector products for calculating influence functions, 2024.

[33] Pang Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In *Advances in neural information processing systems*, 2019.

[34] Samyadeep Basu, Xuchen You, and Soheil Feizi. On second-order group influence functions for black-box predictions. In *International Conference on Machine Learning*, 2020.

[35] SungYub Kim, Kyungsu Kim, and Eunho Yang. Gex: A flexible method for approximating influence via geometric ensemble. *Advances in Neural Information Processing Systems*, 36, 2024.

[36] Myeongseob Ko, Feiyang Kang, Weiyan Shi, Ming Jin, Zhou Yu, and Ruoxi Jia. The mirrored influence hypothesis: Efficient data influence estimation by harnessing forward passes. In *CVPR*, 2024.

[37] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gerard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. *Journal of Machine Learning Research*, 38:192–204, 2015.

[38] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems*, pages 2933–2941, 2014.

[39] Ruoxi Jia, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, and Dawn Song. Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification? In *IEEE / CVF Computer Vision and Pattern Recognition Conference*, 2021.

[40] Siyi Tang, Amirata Ghorbani, Rikiya Yamashita, Sameer Rehman, Jared Dunnmon, James Zou, and Daniel L. Rubin. Data valuation for medical imaging using shapley value and application to a large-scale chest x-ray dataset. *Science Report*, 11:8366, 2021.

[41] Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data cleansing for models trained with sgd. *Advances in Neural Information Processing Systems*, 32, 2019.

[42] Anshuman Chhabra, Bo Li, Jian Chen, Prasant Mohapatra, and Hongfu Liu. Outlier gradient analysis: Efficiently identifying detrimental training samples for deep learning models, 2024.

[43] Zhuoming Liu, Hao Ding, Huaping Zhong, Weijia Li, Jifeng Dai, and Conghui He. Influence selection for active learning. In *International Conference on Computer Vision*, 2021.

[44] Hugh Chen, Scott M Lundberg, and Su-In Lee. Explaining a series of models by propagating shapley values. *Nature communications*, 13(1):4512, 2022.

[45] Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*, 2023.

[46] Zheng Dai and David K Gifford. Training data attribution for diffusion models. *arXiv preprint arXiv:2306.02174*, 2023.

[47] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. *arXiv preprint arXiv:2202.00622*, 2022.

[48] Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. In *Advances in neural information processing systems*, 2023.

[49] Haoru Tan, Sitong Wu, Xiuzhe Wu, Wang Wang, Bo Zhao, Zeke Xie, Gui-Song Xia, and Xiaojuan Qi. Understanding data influence with differential approximation, 2025.

[50] Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

[51] AI@Meta Llama Team. The llama 3 herd of models, 2024.

[52] Qwen Team. Qwen2.5: A party of foundation models, September 2024.

[53] Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, Shimiao Jiang, Shiqi Kuang, Shouyu Yin, Chaohang Wen, Haotian Zhang, Bin Chen, and Bing Yu. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm, 2025.

[54] Jiachen T Wang, Dawn Song, James Zou, Prateek Mittal, and Ruoxi Jia. Capturing the temporal dependence of training data influence. *arXiv preprint arXiv:2412.09538*, 2024.

[55] Garima Pruthi, Frederick Liu, Sundararajan Mukund, and Satyen Kale. Estimating training data influence by tracing gradient descent. *arXiv preprint arXiv:2002.08484*, 2020.

[56] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pages 2242–2251. PMLR, 2019.

[57] Adam Paszke, Sam Gross, Soumith Chintala, G Chanan, E Yang, Zachary Devito, Zeming Lin, Alban Desmaison, L Antiga, A Lerer, and et.al. Automatic differentiation in pytorch. In *Advances in neural information processing systems Workshop*, 2017.

[58] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. In *International Conference on Learning Representations*, 2023.

[59] 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. In *International Conference on Learning Representations*, 2022.

[60] Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

[61] 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.

[62] Jinhao Jiang, Zhipeng Chen, Yingqian Min, Jie Chen, Xiaoxue Cheng, Jiapeng Wang, Yiru Tang, Haoxiang Sun, Jia Deng, Wayne Xin Zhao, Zheng Liu, Dong Yan, Jian Xie, Zhongyuan Wang, and Ji-Rong Wen. Enhancing llm reasoning with reward-guided tree search. *arXiv preprint arXiv:2411.11694*, 2024.

[63] HuggingFaceH4 Team. Math-500: A subset of the math dataset. https://huggingface.co/datasets/HuggingFaceH4/MATH-500, 2024. Accessed: 2025-05-15.

[64] Qwen Team. Amc23 benchmark dataset. https://github.com/QwenLM/Qwen2.5-Math, 2024. Accessed: 2025-05-15.

[65] David Lewkowycz, Ethan Dyer, Guy Gur-Ari, Igor Babuschkin, Jeffrey Wu, Tom Henighan, Jared Kaplan, et al. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*, 2022.

[66] Yeyu Feng, Yiyang Gu, Ziyang Liu, Jiayi Wang, Qihao Zhu, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

[67] Meta AI. Llama 3.2 3b instruct. https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct, 2024. Accessed: 2025-05-15.

[68] Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *International Conference on Learning Representations*, 2023.

[69] Haoru Tan, Chuang Wang, Sitong Wu, Tieqiang Wang, Xu-Yao Zhang, and Cheng-Lin Liu. Proxy graph matching with proximal matching networks. In *The Annual AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[70] Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6(1):147–160, 1994.

[71] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[72] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[73] Haoru Tan, Sitong Wu, and Jimin Pi. Semantic diffusion network for semantic segmentation. In *Neural Information Processing Systems (NeurIPS)*, 2023.

[74] Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*, 2024.

[75] Sitong Wu, Haoru Tan, Yukang Chen, Shaofeng Zhang, Jingyao Li, Bei Yu, Xiaojuan Qi, and Jiaya Jia. Mixture-of-scores: Robust image-text data quality score via three lines of code. In *International Conference on Computer Vision (ICCV)*, 2025.

[76] Haoru Tan, Sitong Wu, Zhuotao Tian, Yukang Chen, Xiaojuan Qi, and Jiaya Jia. Saco loss: Sample-wise affinity consistency for vision-language pre-training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[77] Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, 2024.

[78] Sitong Wu, Tianyi Wu, Haoru Tan, and Guodong Guo. Pale transformer: A general vision transformer backbone with pale-shaped attention. In *The Annual AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[79] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A coreset approach. In *International Conference on Learning Representations*, 2018.

[80] Peizhao Li and Hongfu Liu. Achieving fairness at no utility cost via data reweighing with influence, 2022.

[81] Haoru Tan, Chuang Wang, Sitong Wu, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Ensemble quadratic assignment network for graph matching. *International Journal of Computer Vision (IJCV)*, 2024.

[82] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2019.

[83] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

[84] Samyadeep Basu, Phillip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. In *International Conference on Learning Representations*, 2021.

[85] Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: A survey. *arXiv preprint arXiv:2212.04612*, 2022.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please see the abstract and introduction section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see the relevant section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Please see the supplemental material.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please see the experimental section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and data will be made publicly available upon acceptance through peer review.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please see the experimental section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Please see the experimental section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please see the experimental section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We strictly follow the relevant Guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please check the relevant section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The training algorithm involved in this article will not cause the originally safe model to become dangerous.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: This paper does not use existing assets

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: This article only uses LLM to help check for spelling mistakes in very small quantities.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Impact Statement

This paper introduces a novel influence analysis algorithm to advance RFT training. It has potential positive societal effects, such as improving understanding of data roles in developing robust systems and possibly reducing data bias. However, there are concerns that the negative impacts could be misused for inhumane social surveillance, which legislative bodies worldwide should address seriously.

B Mathematical Proof: Approximation

Let θ^t denote the learned parameter at the $t-$th iteration on the full dataset and θ^t_{-z} denote the learned parameter at the $t-$th iteration on the dataset without sample z, we use $\mathcal{D}^t(z)$ to denote the empirical expected reward at the t-th step,

$$\mathcal{D}^t(z) = \mathcal{J}(\theta^t) - \mathcal{J}(\theta^t_{-z}),$$

where $t \leq T$, and $\mathcal{D}(z) = \mathcal{D}^T(z)$. Note that the network on the full set and that on the subset $\mathcal{Z}/z$ started from the same initialization, hence, $\mathcal{M}^0(z) = 0$. Let's start with the identical equation below,

$$\begin{aligned}\mathcal{D}(z) &= \Big(\mathcal{D}(z) - \mathcal{D}^{T-1}(z)\Big) + \Big(\mathcal{D}^{T-1}(z) - \mathcal{D}^{T-2}(z)\Big) + ... + \Big(\mathcal{D}^1(z) - \mathcal{D}^0(z)\Big) + \mathcal{D}^0(z) \\ &= \Big(\mathcal{D}(z) - \mathcal{D}^{T-1}(z)\Big) + \Big(\mathcal{D}^{T-1}(z) - \mathcal{D}^{T-2}(z)\Big) + ... + \Big(\mathcal{D}^1(z) - \mathcal{D}^0(z)\Big) \\ &= \Delta\mathcal{D}^T + \Delta\mathcal{D}^{T-1} + ... + \Delta\mathcal{D}^1.\end{aligned} \tag{7}$$

Let's take one single item $\Delta\mathcal{D}^t$ as an example,

$$\Delta\mathcal{D}^t = \mathcal{D}^t(z) - \mathcal{D}^{t-1}(z) = \Big[\mathcal{J}(\theta^t) - \mathcal{J}(\theta^{t-1})\Big] - \Big[\mathcal{J}(\theta^t_{-z}) - \mathcal{J}(\theta^{t-1}_{-z})\Big] \tag{8}$$

By using the first-order Taylor approximation and according to the gradient-ascent algorithm in RFT, we have that,

$$\mathcal{J}(\theta^t) - \mathcal{J}(\theta^{t-1}) \approx \nabla\mathcal{J}(\theta^{t-1})\Big(\theta^t - \theta^{t-1}\Big) = \eta^t||\nabla\mathcal{J}(\theta^{t-1})||^2,$$

where the policy gradient $\nabla\mathcal{J}(\theta^{t-1}) = E_{(s,a)}(A(a,s)\nabla\log\pi_{\theta^t}(a|s))$. Hence, we have the equation for the differential term $\Delta\mathcal{D}^t$, that is,

$$\Delta\mathcal{D}^t = \eta^t||\nabla\mathcal{J}(\theta^{t-1})||^2 - \eta^t||\nabla\mathcal{J}(\theta^{t-1}_{-z})||^2.$$

Given a specific sample z, we present a unified empirical expected reward formulation for RFT:

$$\mathcal{J}_\epsilon = \frac{1}{N}\sum_{s\neq z} A(a_s, s) + (\frac{1}{N} + \epsilon)\cdot A(a_z, z),$$

where ϵ is a coefficient. Hence, we have the expected reward on the full training set by setting $\epsilon = 0$ and the subset without sample z by (approximately) setting $\epsilon = \frac{-1}{N}$, which is a convention in influence analysis for supervised learning [26, 33, 34, 31, 41, 58, 29, 46]. We can treat the $(\frac{1}{N} + \epsilon)\cdot A(a, z)$ as the new reward for sample z. Hence, the policy gradient given $\mathcal{J}_\epsilon$ is:

$$\nabla\mathcal{J}_\epsilon = \frac{1}{N}\sum_{s\neq z} A(a_s, s)\nabla\log\pi_\theta(a_s|s) + (\frac{1}{N} + \epsilon)\cdot A(a_z, z)\nabla\log\pi_\theta(a_z|z).$$

Here, we use the Taylor approximation again to approximate $\nabla\mathcal{J}(\theta^{t-1}_{-z})$ with $\nabla\mathcal{J}(\theta^{t-1})$, which is also a convention in influence analysis for supervised learning [26, 33, 34, 31, 41, 58, 29, 46],

$$\nabla\mathcal{J}(\theta^{t-1}_{-z}) \approx \nabla\mathcal{J}(\theta^{t-1}) + \frac{\partial\nabla\mathcal{J}^{t-1}_\epsilon}{\partial\epsilon}\Big|_{\epsilon=0}\Big((\epsilon = \frac{-1}{N}) - (\epsilon = 0)\Big) = \nabla\mathcal{J}(\theta^{t-1}) - \frac{1}{N}g^{t-1}_z, \tag{9}$$

where $g^{t-1}_z = \nabla\log\pi_{\theta^{t-1}}(a_z|z)A(a_z, z)$.

For other variants, the formulation for g_z^{t-1} also needs to be adapted, for example, the group relative formulation for GRPO [7]:

$$g_z^{t-1} = \frac{1}{G}\sum_{i\leq G} \nabla \log \pi_{\theta^{t-1}}(a_{z,i}|z)\hat{A}(a_{z,i},z),$$

where $a_{z,i}$ means the i-th answer from the model given the input question/prompt z, $\hat{A}(a_{z,i},z) = \left(\frac{\pi_\theta(a_i|s)}{\pi_{\theta_{\text{old}}}(a_i|s)}A(s,a_i), \text{clip}\left(\frac{\pi_\theta(a_i|s)}{\pi_{\theta_{\text{old}}}(a_i|s)}, 1-\epsilon, 1+\epsilon\right)A(s,a_i)\right) - \frac{\beta}{G}\text{KL}\left(\pi_\theta||\pi_{\text{ref}}\right)$.

Hence, we have,

$$\begin{aligned}
\Delta\mathcal{D}^t(z) \approx& \eta^t||\nabla\mathcal{J}(\theta^{t-1})||^2 - \eta^t||\nabla\mathcal{J}(\theta^{t-1}_{-z})||^2 \\
\approx& \eta^t||\nabla\mathcal{J}(\theta^{t-1})||^2 - \eta^t\left\langle \nabla\mathcal{J}(\theta^{t-1}) - \frac{1}{N}g_z^{t-1} \quad , \quad \nabla\mathcal{J}(\theta^{t-1}) - \frac{1}{N}g_z^{t-1}\right\rangle \\
=& \frac{2\eta^t}{N}\left\langle \nabla\mathcal{J}(\theta^{t-1}), g_z^{t-1}\right\rangle - \frac{\eta^t}{N^2}\left\langle \nabla g_z^{t-1}, g_z^{t-1}\right\rangle \\
\approx& \frac{2\eta^t}{N}\left\langle \nabla\mathcal{J}(\theta^{t-1}), g_z^{t-1}\right\rangle,
\end{aligned} \tag{10}$$

where $\left\langle :,: \right\rangle$ is the inner-product operator. By substituting the approximation for $\mathcal{D}^t(z)$ into Eq.(7), we have that,

$$\mathcal{D}(z) = \Delta\mathcal{D}^T + \Delta\mathcal{D}^{T-1} + ... + \Delta\mathcal{D}^1 \approx \sum_t \frac{2\eta^t}{N}\left\langle \nabla\mathcal{J}(\theta^t), g_z^t\right\rangle, \tag{11}$$

where $0 \leq t < T$.

C Mathematical Proof: Error Bound

We provide a worst-case error bound for the proposed approximation, demonstrating its robustness under some mild assumptions.

Proposition 1. *Under the assumptions that the log-likelihood function $\log \pi_\theta(a|s)$ exhibits ℓ-Lipschitz continuity and that the adventure value is upper-bounded by A_{max}, the approximation error is bounded as follows:*

$$\left|\mathcal{D}(z) - \hat{\mathcal{D}}(z)\right| \leq \mathcal{O}\Big(\Big[\frac{\eta_{\max}(4N+4)}{N}(\ell A_{\max})^2 + 2\eta\ell^2 A_{\max}\Big]T\Big), \tag{12}$$

where T denotes the maximum number of iterations and N is the number of all training data.

Our estimator is based on the time-domain differential operation over $\mathcal{D}(z) = \sum_t \Delta\mathcal{D}^t(z)$, and performing Taylor approximation on $\Delta\mathcal{D}^t(z)$. Hence, the overall error is bounded by

$$|\mathcal{D}(z) - \hat{\mathcal{D}}(z)| \leq T|\mathcal{D}^t(z) - \hat{\mathcal{D}}^t(z)|.$$

We set d as the upper bound of the norm of the policy gradient. Hence, we have $d \leq A_{\max}\ell$, where ℓ is the Lipschitz constant of the log-likelihood function $\log \pi_\theta(a|s)$ with ℓ-Lipschitz continuity.

According to Eq.(10), the error comes from the following parts: The first approximation term, $\Delta\mathcal{D}^t(z) \approx \eta^t||\nabla\mathcal{J}(\theta^{t-1})||^2 - \eta^t||\nabla\mathcal{J}(\theta^{t-1}_{-z})||^2$, using the first-order Taylor approximation and the gradient-ascent algorithm in RFT, the error is bounded by

$$\begin{aligned}
&\left|\Delta\mathcal{D}^t(z) - \left[\eta^t||\nabla\mathcal{J}(\theta^{t-1})||^2 - \eta^t||\nabla\mathcal{J}(\theta^{t-1}_{-z})||^2\right]\right| \\
&= \left|\left[\mathcal{J}(\theta^t) - \mathcal{J}(\theta^{t-1})\right] - \left[\mathcal{J}(\theta^t_{-z}) - \mathcal{J}(\theta^{t-1}_{-z})\right] - \left[\eta^t||\nabla\mathcal{J}(\theta^{t-1})||^2 - \eta^t||\nabla\mathcal{J}(\theta^{t-1}_{-z})||^2\right]\right| \\
&\leq 2\ell\eta d + 2\eta d^2.
\end{aligned} \tag{13}$$

The second approximation term in approximating $\nabla\mathcal{J}(\theta^{t-1}_{-z})$ with $\nabla\mathcal{J}(\theta^{t-1})$, that is, $\nabla\mathcal{J}(\theta^{t-1}_{-z}) \approx \nabla\mathcal{J}(\theta^{t-1}) - \frac{1}{N}g^{t-1}_z$. We have

$$\Big|\ \eta^t||\nabla\mathcal{J}(\theta^{t-1}_{-z})||^2 - \eta^t||\nabla\mathcal{J}(\theta^{t-1}) - \frac{1}{N}g^{t-1}_z||^2\Big| \leq 2\eta d^2 + \frac{2\eta}{N}d^2 + \frac{\eta}{N^2}d^2. \tag{14}$$

The last error term comes from ignoring $\frac{\eta^t}{N^2}\Big\langle \nabla g^{t-1}_z, g^{t-1}_z \Big\rangle$ in the final approximation in Eq.(10), this error is bounded by $\frac{\eta}{N^2}d^2$.

Hence, the overall error is bounded by

$$\begin{aligned}
|\mathcal{D}(z) - \hat{\mathcal{D}}(z)| \leq T|\mathcal{D}^t(z) - \hat{\mathcal{D}}^t(z)| &\leq T\Big(2\ell\eta d + 2\eta d^2 + 2\eta d^2 + \frac{2\eta}{N}d^2 + \frac{\eta}{N^2}d^2 + \frac{\eta}{N^2}d^2\Big) \\
&= T\Big((4\eta + \frac{2\eta}{N} + \frac{2\eta}{N^2})d^2 + 2\ell\eta d\Big) \\
&\leq T\Big((4\eta + \frac{4\eta}{N})d^2 + 2\ell\eta d\Big) \\
&= T\Big(\frac{\eta_{\max}(4N+4)}{N}d^2 + 2\ell\eta d\Big) \\
&\leq \Big[\frac{\eta_{\max}(4N+4)}{N}(\ell A_{\max})^2 + 2\eta\ell^2 A_{\max}\Big]T
\end{aligned} \tag{15}$$