LLM推論へのミニマリスト的アプローチ :拒絶から強化のためのサンプリング

ウェイ•ション †ジャールイ•ヤオ † ユーフイ•シュー ‡ボー•パン ‡レイ•ワン ‡

ドイェン•サフー ‡ジュンナン•リー ‡ナン•ジャン †トン•ジャン †カイミン•ション ‡

ハンゼ•ドン ‡

‡ Salesforce AIリサーチ †イリノイ大学アーバナシャンペーン校

抽象的な

強化学習（RL)は、複雑な推論タスクにおける大規模言語モデル（LLM)のファインチューニングにおいて、広く普及している手法となっている。近年の手法の中でも、GRPOはDeepSeek-R1などのモデルの学習において実証的な成功を収めているが、その有効性の源泉は未だ十分に解明されていない。本研究では、GRPOを強化学習的なアルゴリズムの観点から再考し、その中核となる構成要素を分析する。

驚いたことに、正の報酬が与えられたサンプルのみでトレーニングする単純な拒否サンプリング ベースライン RAFT が、GRPO や PPO よりも競争力のあるパフォーマンスを生み出すことがわかりました。アブレーション研究により、GRPO の主な利点は、報酬の正規化ではなく、完全に間違った応答を持つプロンプトを破棄することから生じることが明らかになりました。この洞察に基づき、完全に間違ったサンプルと完全に正しいサンプルの両方をフィルターするポリシー勾配の最小限の拡張である Reinforce-Rej を提案します。Reinforce-Rej は KL の効率と安定性を改善し、より複雑な RL アルゴリズムの軽量でありながら効果的な代替手段として機能します。RAFT は堅牢で解釈可能なベースラインであると主張し、将来の進歩では、無差別に負のサンプルに依存するのではなく、負のサンプルを組み込むためのより原理的な設計に焦点を当てるべきであると提言します。私たちの研究結果は、トレーニング後の報酬ベースの LLM における将来の研究への指針となります。

1 はじめに

検証可能な報酬を用いた大規模言語モデル（LLM)の微調整という文脈において、強化学習（RL)アルゴリズムを研究する。本研究では、OpenAIのO1モデル（Jaech et al., 2024)やDeepSeek-R1（DeepSeek-AI et al., 2025)といったモデルのリリースを受けて近年大きな注目を集めている数学的推論タスクに焦点を当てる。LLM学習後の学習において、最も主流のアプローチは近似ポリシー最適化（PPO）(Schulman et al., 2017; Bai et al., 2022; Ouyang et al., 2022)である。しかし、PPOは、バニラReinforceアルゴリズム（Williams and Peng, 1991)に加えて、追加の批評ネットワークを必要とするため、計算オーバーヘッドとアルゴリズムの複雑さの両方が生じる。一方、LLMの決定論的な遷移特性は、比較的低い分散で問題を単純化するため、PPOの高度な構成要素の多くはこの設定では不要となる可能性があります。この観察結果から、学習後LLMのためのよりシンプルかつ効果的な強化学習アルゴリズムの設計への関心が高まっています。

最近の研究では、ReMax (Li et al., 2023)、RLOO (Ahma-dian et al., 2024; Kool et al., 2019)、GRPO (Shao et al., 2024)、Reinforce++ (Hu, 2025) など、Reinforceスタイルのアプローチが再検討されています。同時に、他の手法では、方策勾配を超えた様々な方向性が模索されています。報酬ランク付け微調整 (RAFT) (Anthony et al., 2017; Dong et al., 2023) は、プロンプトごとにn個の応答を反復的に生成し、誤った回答を除外し、残りの承認済みサンプルに対してLLMを微調整します。直接的な選好に基づく手法、例えば

*HDとWXはこの作業に等しく貢献しました。連絡先はhanze.dong@salesforce.comとwx13@illinois.eduです。

SFT ベースの対照学習 (Slic-HF) (Zhao et al., 2023) や DPO (Rafailov et al., 2023) は、ペアワイズ比較データセットに基づいて対照目的を最適化します。

これらの中でも、GRPOはDeepSeek-R1の学習における成功（DeepSeek-AI et al., 2025)により、数学推論タスクにおけるLLMの強化に最も広く使用されているアルゴリズムの一つとして際立っています。しかし、そのアルゴリズムの詳細は未だほとんど文書化されておらず、その採用が固有の利点によるものなのか、それとも先行研究で使用された手法との連続性によるものなのかは不明です。対照的に、RAFTは最もシンプルで解釈しやすいベースラインの一つとして確立されており、そのミニマルな設計にもかかわらず、先行研究において一貫して良好な実験的パフォーマンスを示しています。

このプロジェクトでは、(1) RAFT（LLM文献では拒否サンプリングとも呼ばれる)を再検討します。これは、LLM後トレーニング用の最も基本的なRLアルゴリズムであると言えるでしょう。(2) Vanilla Reinforce（古典的なポリシー勾配アルゴリズム)は、批評家モデルを排除することでPPOの簡易版として機能します。(3) GRPO（Reinforceアルゴリズムのバリアント)は、プロンプトごとにn個の応答をサンプリングし、各プロンプト内の平均と標準偏差を使用してサンプル報酬を正規化することで相対的な利点を計算します。

GRPO (Reinforce) と RAFT の主な違いは、負のサンプルを処理する方法にあります。GRPO はトレーニング中に承認された例と拒否された例の両方を混合しますが、RAFT は正のサンプルのみに依存します。
一般的に、ネガティブシグナルを活用する強化学習（RL)手法は、ポジティブサンプルのみを使用するSFT類似アルゴリズムよりも大幅に優れた性能を発揮すると考えられていますが、私たちの予備実験では、その性能差は驚くほど小さく、RAFT類似アプローチは学習初期段階（例えば最初の100〜200回の反復)においてより速い収束を示すことが確認されました。さらに分析を進めた結果、完全に誤った応答を含むプロンプトなど、特定の種類のネガティブシグナルは、モデル性能に著しく悪影響を与える可能性があることが明らかになりました。一方、報酬正規化などの他の手法は、影響が最小限に抑えられるようです。

これらのダイナミクスをより深く理解するために、Qwen（Yang et al., 2024)モデルとLLaMA（Grattafiori et al., 2024)モデルの両方を用いて、複数のReinforceバリアントを対象に、個々の設計選択を分離したアブレーション研究を実施しました。その結果、以下の重要な知見が明らかになりました。

1. 正のサンプルのみを用いる単純な棄却サンプリングベースラインであるRAFTを再検討した結果、その性能は最先端の強化学習手法であるGRPOと遜色なく、驚くほど小さなギャップと、学習初期段階におけるより速い収束速度を示すことが分かりました。さらに詳細な分析を行うと、正のサンプルのみを用いて学習するRAFTは、方策エントロピーの急激な減少を招き、探索を制限し、最終的にはGRPOに追い抜かれることが明らかになりました。

2. 様々なReinforce法を用いた一連の制御実験を通して、方策オン法において、サンプルされたすべての応答が不正解であるプロンプトで訓練すると、パフォーマンスが著しく低下する可能性があることが分かりました。さらに、GRPOが標準的なReinforce法に対してパフォーマンス向上を実現しているのは、主にこれらの有害なプロンプトを暗黙的にフィルタリングしているためであることも分かりました。対照的に、プロンプト内の平均値と標準偏差による報酬正規化手法は、その影響は最小限にとどまります。

3. RAFTとReinforceの両方の研究を踏まえ、Reinforceの新しい変種であるReinforce-rejを研究します。Reinforce-rejは、すべて正解またはすべて不正解のプロンプトを選択的に除外します。この手法はGRPOと同等の最終的なパフォーマンスを示し、優れたKL効率を示します。

これらの洞察は、報酬ベースのLLMにおけるアルゴリズム設計よりもサンプル選択の重要性を強調している。
トレーニング後。このプロジェクトのコードはhttps://github.com/RLHFlow/Minimal-RLで入手できます。

2 方法

表記法。プロンプトが与えられた場合、LLMは、プロンプトを応答a上の分布π(a|x)にマッピングできる方策として表記されます。また、r(x, a) ∈ {−1, 1}を、プロンプトと応答のペアにスカラーフィードバックを割り当てる2値報酬関数として表記します。これは検証者1によって実装できます。収集されたプロンプトと応答のペアのデータセットをDと表記します。各プロンプトxに対して、n個の候補応答a1, · · ·を生成でき、それに対応する報酬はr1, · · ·です。 、そして、

、rn.

1https://github.com/huggingface/Math-Verify

応答におけるt番目のトークンをa = (a1, ・・・, a|a|)とし、 st(θ) = $\frac{\pi\theta(a_t|x,a_{1:t-1})}{\pi\theta_{old}(a_t|x,a_{1:t-1})}$ を表すトークンtの重要度サンプリング比。また、報酬のベースラインをmean(r1, ・・・, rn) 、その標準偏差をstd(r1, ・・・, rn)と定義する。ここで、LLM後学習に用いられる代表的なアルゴリズムをいくつか見ていく。

RAFT。RAFTアルゴリズムは、文献では棄却サンプリング微調整法（Touvron et al., 2023; Yuan et al., 2023）とも呼ばれています。本稿では、Dong et al. (2023) の定式化に従い、以下の3つのステップから構成されます。

- データ収集。プロンプトのバッチ{x1, ・・・, xM}に対して、参照モデル（たとえば、現在のモデル）からプロンプトごとにn個の応答をサンプリングし、各xiに対する候補応答{ai,1, ・・・, ai,n}を取得します。
- データのランク付け（棄却サンプリング）。各プロンプトxiについて 、各応答{ri,1, ・・・, ri,n}の報酬値を2値報酬関数r(x, a)を用いて計算し、最も高い報酬値（典型的にはr = 1)を持つ応答のみを保持する。得られた正のサンプル集合はデータセットDに集約される。
- モデルの微調整。現在のポリシーπは、
  選択されたデータセット:

$$L_{ラフト}(\theta) = \sum_{(x,a)\in D} \log \pi_\theta(a|x). \tag{1}$$

密接に関連するアルゴリズムは STaR (Zelikman et al.、2022) であり、これも自己生成された CoT 応答に基づいてトレーニングします。
これに対し、STaRは各反復において、現在のモデルではなく、元の事前学習済みモデルから再学習を行います。また、RAFTで使用される棄却サンプリングとは異なり、STaRは貪欲デコードを使用し、1つの応答のみを生成します。さらに、STaRは、難しい問題に対してCoT応答を生成するために、プロンプトに回答を提供することも提案しています。

方策勾配と強化学習。ここでは、単純化のために行動全体を取り上げてこの考え方を説明し、後ほど自己回帰モデルに拡張します。方策勾配アルゴリズムは、以下の学習目標を達成するように設計されています。

$$J(\theta) = J(\pi_\theta) = E_{x\sim d_0} E_{a\sim\pi_\theta(\cdot|x)} r(x, a) , \tag{2}$$

ここで、θはニューラルネットワークのパラメータです。ポリシーアセントを用いてポリシーネットワークを更新することができます。

$$\theta' \leftarrow \theta + \beta \cdot \nabla_\theta J(\theta),$$

ここで、 ∇θJ(θ)は文献では政策勾配と呼ばれています。政策勾配は以下のように表されます。

$$\frac{\partial J(\theta)}{\partial \theta} = E_{x\sim d_0} E_{a\sim\pi_\theta(\cdot|x)} \left[r(x, a) \cdot \frac{\partial \log \pi_\theta(a|x)}{\partial \theta} \right]$$

実際には、RAFTのパイプラインと同様に、 πθoldを用いて軌道をリプレイバッファDに収集し、これらのサンプルを用いて確率的方策勾配を計算してπθoldを更新します。しかし、厳密な方策オントレーニングでは、勾配上昇の1ステップ後に新しいデータを収集する必要があります。トレーニングを高速化するために、通常はミニバッチ方式で複数のステップを実行し、重要度サンプリング手法を用いて分布を補正します。具体的には、目的関数は次のように書き換えられます。

$$J(\theta) = J(\pi_\theta) = E_{x\sim d_0} E_{a\sim\pi_{\theta_{old}}(\cdot|x)} r(x, a) \cdot \frac{\pi_\theta(a|x)}{\pi_{\theta_{old}}(a|x)} \tag{3}$$

そして、 πθoldによって収集された軌跡{x, a, r}のバッチを用いて、上記の重要度サンプリングのトリックを用いて複数のステップを更新することができる。しかし、重要度サンプリングは、分布が

πθとπθoldは離れすぎています。学習を安定させるために、PPOのクリッピング手法も活用できます。最終的に、損失関数は次のようになります。

$$L_{強化}(\theta) = \frac{1}{|だ|} \sum_{x,a \in D} 分 \frac{\pi_\theta(a|x)}{\pi_{\theta old}(a|x)} r(x,a)、clip\left(\frac{\pi_{\theta old}(a|x)}{\pi_\theta(a|x)}, 1-\quad, 1+\quad\right) \cdot r(x,a)。 \tag{4}$$

LLMは自己回帰的なため、通常は各トークンをアクションと見なします。したがって、損失をトークンレベルの対応部分に拡張することができます。

$$L_{強化}(\theta) = \frac{1}{|だ|} \sum_{x,a \in D} \frac{1}{|あ|} \sum_{t=1}^{|あ|} 最小 st(\theta)、clip(st(\theta)、1-\quad、1+\quad) \cdot r(x,a)、 \tag{5}$$

だしst(θ) = であり、 at はa の t 番目のトークンです。πθold (at|x,a1:t−1)

GRPOは式(5)と同様の損失関数を採用するが、r(x, a)を応答aのt番目のトークンに対するアドバンテージ関数At(x, a)に置き換える。具体的には、各プロンプトxに対して、GRPOはn > 1個の応答をサンプリングし、i番目の応答のt番目のトークンについて以下のアドバンテージを計算する。

$$At(x, a_i) = \frac{r_i - 平均(r_1, \cdots r_n)}{std(r_1, \cdots, r_n)}。$$

mean(r1, · · · rn)はRLの文献ではベースラインと呼ばれることが多く、確率的勾配の分散を減らすのに役立ちます。

（反復)DPO。DPOアルゴリズムは、一対比較データセット{(x, a+, a-)}（aはプロンプトxに対する2つの応答)に基づきます。次に、DPOは以下の対比損失を最適化します。

$$L_{DPO}(\theta) = -\log \sigma\left(\beta \log \frac{\pi_\theta(a^+|x)}{\pi_{ref}(a^+|x)} - \beta \log \frac{\pi_\theta(a^-|x)}{\pi_{ref}(a^-|x)}\right)、$$

ここで、β > 0であり、 πrefは通常、初期チェックポイントとして設定されます。オリジナルのDPOアルゴリズムは、オフラインおよびオフポリシーデータで学習します。その後の研究（Liu et al., 2023; Xiong et al., 2023; Xu et al., 2023; Hoang Tran, 2024; Dong et al., 2024)では、中間チェックポイントを反復的に使用して新しい応答を生成し、選好シグナルにラベルを付け、自己生成したオンポリシーデータで学習することで、モデルのパフォーマンスを大幅に向上できることが示されています。

RAFT++。RAFTは、各反復処理でリプレイバッファに対して複数のステップを実行する際にポリシー外となる可能性のあるハイブリッドアルゴリズムと見なすこともできます。自然な拡張として、元のRAFTに重要度サンプリングとクリッピングの手法を適用し、同様の損失関数を得ます。

$$L_{ラフト++}(\theta) = \frac{1}{|だ|} \sum_{x,a \in D} \frac{1}{|あ|} \sum_{t=1}^{|あ|} \min st(\theta)、clip(st(\theta)、1-\quad、1+\quad) \cdot I\, r(x,a) = \operatorname{argmax} r(x, a_i)\ 、 \tag{6}$$

ここで、インジケーターは、最も高い報酬を持つ応答 (肯定的なサンプル) のみをトレーニングすることを保証します。

3 実験のセットアップ

このプロジェクトでは、数学的推論タスクに焦点を当てています。実装は主にverl (Sheng et al., 2024) フレームワークに基づいています。

モデル	アルゴリズム	Math500	ミネルバ	数学オリンピック	ベンチ平均
Qwen2.5-数学-7Bベース	ベース	41.3	11.0	18.6	23.6
	RAFT	77.4	34.4	37.8	49.9
	RAFT++ 反復	80.5	35.8	41.2	52.5
	DPO 75.7 強化	80.6 GRPO	PPO 30.5	38.3	48.2
			36.1	42.1	52.9
		81.6	36.7	43.3	53.9
		79.6	34.8	41.1	51.8
LLaMA-3.2-3B-命令	ベース	27.3	8.3	6.5	14.0
	ラフト	46.6	16.8	15.5	26.3
	ラフト++	48.8	16.9	16.8	27.5
	強化する	42.9	14.8	12.5	23.4
	GRPO	51.0	18.9	18.4	29.4
	PPO	47.7	17.3	16.1	27.0

表1: Math500を含む3つのベンチマークにおけるさまざまなアルゴリズムのパフォーマンス（Hendrycks et al.、2021年）、Minerva Math（Lewkowycz他、2022年）、Olympiad Bench（He他、2024年）など。報告された精度は平均@16、温度1.0、最大世代長4096トークンです。

データセットとモデル。Numina-Math (Beeching et al., 2024)のプロンプトセットを使用してモデルをトレーニングします。
約86万問の数学問題とラベル付き正解から構成されています。
Numina-Mathは、中国の高校数学演習から米国および国際数学オリンピックまでを網羅しています。
競争問題。一般性を確保するために、Qwen2.5-Math-7B-baseとLLaMA-3.2-3B-instructの両方で実験を行った。これらのモデルのデフォルトのチャットテンプレートを使用し、CoTプロンプト「さあ、
段階的に考えて、最終的な答えを \boxed{} 内に出力します。

ハイパーパラメータ。Verlフレームワークで推奨されているハイパーパラメータ設定のほとんどに従います。
Reinforce、GRPO、PPOの学習では、RAFTとRAFT++のハイパーパラメータも同様です。
GRPOスクリプトと同様です。具体的には、学習率1×10－6のAdamW最適化器を使用します。
反復ごとに1024のプロンプトをサンプリングし、RAFTとGRPOのプロンプトごとにn = 4の応答を生成します。
訓練ミニバッチサイズは512に設定されている。モデルは訓練中に最大4096トークンを生成することができる。
トレーニング。より詳細なスクリプトはGitHubリポジトリで入手できます。反復的なDPOのベースラインとして、
Zhang et al. (2025) で開発されたコードベースを使用します。

評価。モデルの推論能力をMath500（Hendrycks et al., 2021）、Minerva
数学（Lewkowycz et al., 2022）、オリンピックベンチ（He et al., 2024）。人気のAIME2024は含まれていません。
30問しか出題されていないため、ベンチマークとしては不向きです。予備実験では、
このベンチマークでは、検討したすべてのアルゴリズムにおいて非常にノイズが多い。評価には主にaverage@16を使用する。
私たちのモデルでは、温度1.0のプロンプトごとに16の応答を生成し、平均精度を使用します。
メトリックとして。モデルは最大4096トークンを生成できます。

コードはhttps://github.com/RLHFlow/Minimal-RL で入手できます。

4つの主な結果

RAFT と RAFT++ は、驚くほど小さなパフォーマンスギャップでディープ RL 手法に近づきます。
様々なアルゴリズムを用いて訓練されたモデルのテスト精度を表1にまとめた。最初の観察結果は
RAFT（およびその変種であるRAFT++)は、おそらく最も単純なアルゴリズムであり、競争力のある
反復 DPO やディープ RL ベースのアプローチなどのより複雑な方法と比較したパフォーマンス。
具体的には、Qwen2.5-Math-7B-baseでは、バニラRAFTは平均精度49.9%に達し、

反復DPO（48.2%)とPPO（51.8%)に近づいています。追加の重要度サンプリングとクリッピング手法により、RAFT++はバニラRAFTよりもさらに改善され、平均精度52.5%を達成しました。この結果は、最先端の深層強化学習（Deep RL)手法であるGRPO（最良モデルで平均精度53.9%)に驚くほど近いものです。LLaMA-3.2-3B-instructモデルでも同様の傾向が見られ、異なるモデル間でRAFTとRAFT++の堅牢性を示しています。RL手法は負のフィードバックを利用できるため、より強力であると考えられることが多いため、これらの結果はやや直感に反するものです。

興味深いことに、LLaMAベースの設定では、ReinforceのパフォーマンスはRAFT++よりも大幅に低く、平均正解率は23.4%であるのに対し、RAFT++は27.5%でした。考えられる理由の一つは、最終的な回答の正答率のみに基づいてネガティブサンプルを定義するのは粗すぎる可能性があり、RLトレーニングにおけるネガティブフィードバックの使用によるメリットが制限される可能性があることです。また、RLトレーニングの現状におけるネガティブサンプルの役割を調査するため、より多くのアブレーション研究も実施する予定です。

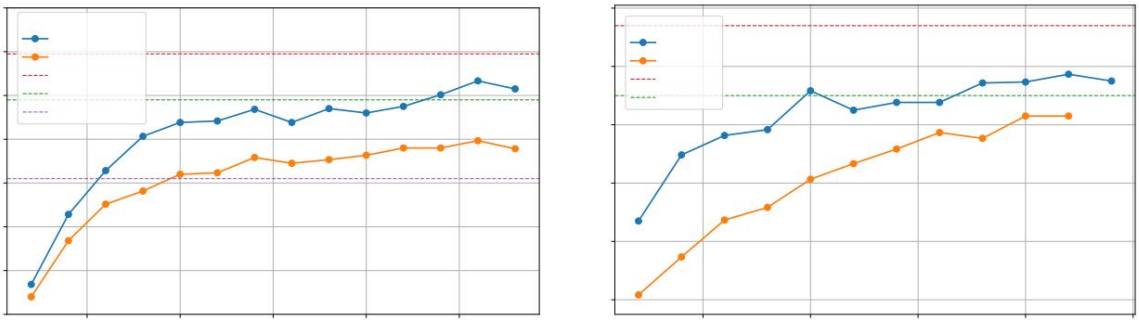

図1 :Qwen2.5-Math-7B-base（左)とLLaMA-3.2-3B-instruct（右)から初期化したRAFTとRAFT++の学習ダイナミクス。Y軸は16点平均の精度で、MATH500、Minerva Math、Olympiad Benchでさらに平均化されています。参考までに、GRPO、PPO、Iterative DPOの最適モデルもプロットしています。

分布補正とクリッピングは、バニラRAFTを改善します。表1は、リプレイバッファ内の分布シフトを補正するために重要度サンプリングを適用すると、RAFTの最終的なテスト精度が向上し、RAFT++と呼ばれるより強力なバリアントにつながることを示しています。さらに、図1では、RAFTとRAFT++の学習ダイナミクスを示しています。どちらの手法も、オンライン更新を通じてモデルの推論能力を着実に向上させることができ、RAFT++はバニラRAFTよりも収束が速く、最終的な精度も高くなります。

アブレーション研究の一環として、クリッピングなしで重要度サンプリングを適用する中間的なバリアントも評価しました。図2に示すように、このバリアントは通常のRAFTよりもパフォーマンスが劣っています。これは、クリッピングはめったに発生しないため不要であると示唆するAhmadianら (2024) の知見と矛盾しています。クリッピングはまれにしか発生しないものの、 π_θが1から大きく逸脱した場合に発生すると仮定しています。このような場合、無制限の更新は方策勾配法の根底にある方策オン仮定に著しく違反し、不安定性とパフォーマンスの低下につながる可能性があります。 $\overline{\pi_{\theta_{old}}}$

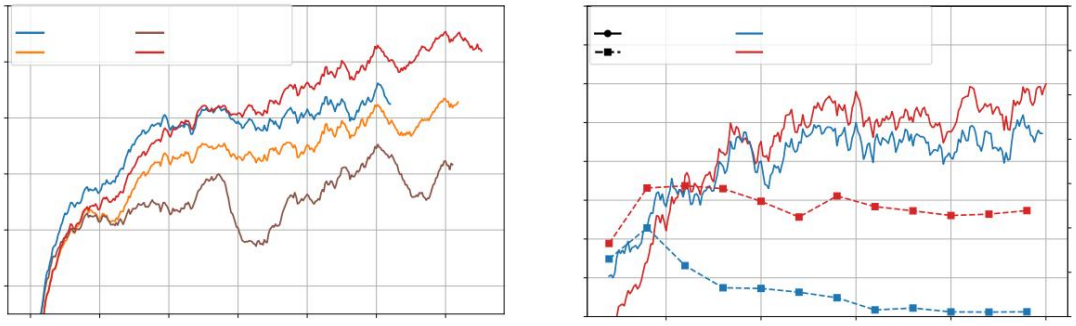

図2 :左 :RAFT、RAFT++、クリッピングなしのRAFT++（RAFTと重要度サンプリング）、およびGRPOのトレーニング報酬曲線（Qwen2.5-Math-7B-baseで初期化）。右 :RAFT++と、クリッピング強化トリックを適用したRAFT++のトレーニング報酬曲線（LLaMA-3.2-3B-instructで初期化）。
元の報酬を(1 + r)/2で変換し、結果の値が訓練データの精度と一致するようにします。また、曲線を滑らかにするために、ウィンドウサイズ20の移動平均を適用します。

RAFT++ は初期段階の収束が速いですが、後期のトレーニングでは GRPO に追い抜かれます。
図 2 からは、RAFT++ は GRPO と比較して初期段階の学習が速いこともわかります。
さらに、反復回数100回付近で学習ダイナミクスに明確な転換点が見られ、この時点以降、成長率が顕著に低下します。最終的に、学習後期において、最終的なモデルテスト精度において、RAFT++はGRPOに追い抜かれます。RAFT++のこの速度低下の原因と、このプロセスにおける欠損ネガティブサンプルの役割を調査するために、アブレーション実験も実施する予定です。

4.1 アブレーション研究

このサブセクションでは、これまでに示した主要な調査結果の背後にある根本的な理由を理解することを目指します。
この目的のために、私たちは以下の質問に答えるために設計された一連のアブレーション研究を実施しています。

1. RAFT++からReinforce（GRPOを含む）へ: なぜRAFT++は初期段階では高速なのに、最終的には
   訓練の後半で明らかに優れたパフォーマンスを発揮しましたか ?ネガティブサンプルはどのような役割を果たしますか ?
2. Vanilla Reinforce から GRPO へ: GRPO の優れたパフォーマンスの重要な要因は何ですか?

正のサンプルのみから学習すると、収束とエントロピーの崩壊が速くなります。図3に示すように、まずRAFT++とGRPOについて、初期ポリシーからのポリシーエントロピーとKLダイバージェンスを調べます。重要な観察結果は、正のサンプルのみでトレーニングするRAFT++は、GRPOと比較してポリシーエントロピーの減少がはるかに速いことです。この傾向は、QwenモデルとLLaMAモデルの両方で一貫しています。エントロピーが低いレベルで安定すると、RAFT++のパフォーマンス向上は著しく遅くなります。これは、低エントロピーポリシーでは多様な推論パスが生成されにくくなるため、探索が減るためだと考えています。同時に、初期トレーニング中のRAFT++では初期ポリシーからのKLダイバージェンスがより急速に増加し、テスト精度における初期の優位性を反映しています。

しかし、継続的な調査が不足しているため、RAFT++ はすぐに停滞し、GRPO は改善を続け、最終的にはそれを上回ります。

これらの結果は、負のサンプルが探索を維持し、分布の崩壊を防ぐ上で重要な役割を果たしていることを示唆している。この探索の利点は、RAFT++とReinforceやGRPOなどの強化学習ベースの手法との間のパフォーマンスギャップの一因となっている可能性が高い。方策エントロピーと報酬学習の関係をさらに調査するために、Yu et al. (2025) の「clip higher」手法を組み込んだ。この手法は、下限に 1 = 0.1、上限に 2 = 0.2という非対称なクリッピング範囲を使用する。この手法をLLaMA-3.2-3B-instructモデルに適用し、

図2の右図は、訓練報酬曲線と方策エントロピー曲線を示しています。Yu et al. (2025) の知見と一致して、より大きな　2を使用すると、オンライン訓練全体にわたって方策エントロピーが安定化します。その結果、この強化されたRAFT++バリアントは、訓練の後期段階でオリジナルのRAFT++よりも優れた性能を発揮します。

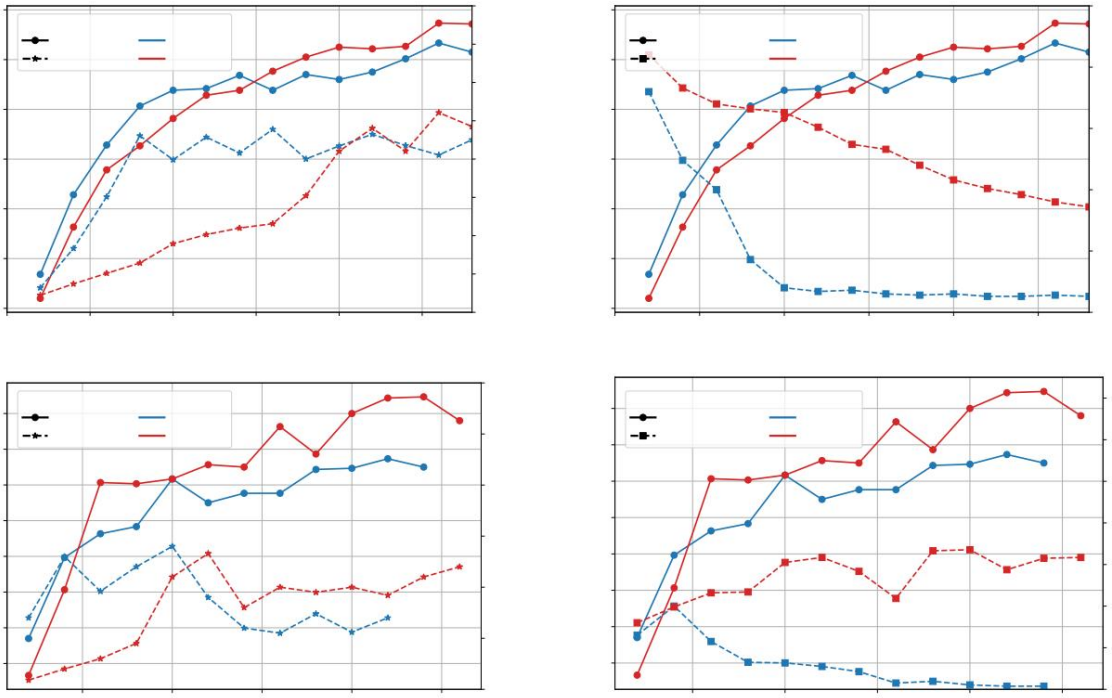

図3: Qwen2.5-Math-7B-base（1行目）とLLaMA-3.2-3B-instruct（2行目）から初期化されたRAFT++とGRPOの学習ダイナミクス。左列にはKL損失、右列には方策エントロピー損失もプロットしています。

ReinforceからGRPOへ：GRPOの成功における鍵となる役割は何でしょうか？GRPOとRAFTの主な違いは、負のサンプルの使用と報酬正規化の適用という2つの点にあります。各要素の寄与を分離し、それぞれの効果をより深く理解するために、私たちは一連の対照実験を設計し、それらの影響を体系的に評価しました。具体的には、以下のアルゴリズムを検討します。

1. 強化：式（5）で導入された通常のもの。
2. 強化 + 平均ゼロ: 各プロンプト内の平均報酬を減算します。
3. すべての正解を強化 + 削除: 回答が完全に正しいプロンプトを除外します。
4. すべての間違いを強化 + 削除: 応答が完全に間違っているプロンプトをフィルターで除外します。
5. 両方を強化 + 削除: 完全に正しいプロンプトと完全に間違ったプロンプトの両方を削除します。
6. 強化 + 両方を削除 + 正規化された標準偏差: 完全に正しいプロンプトと完全に間違ったプロンプトの両方を削除することに加えて、正規化のために各プロンプト内の報酬をその標準偏差でさらに分割します。

図4に示すように、「強化学習 + 全ての誤学習サンプルを削除」というバリアントは、報酬の点でバニラの強化学習よりも大幅なパフォーマンス向上を達成しており、誤学習サンプルが強化学習の学習プロセスにおいて特に有害であることを明確に示しています。これは、誤学習サンプルの大きな分散と誤解を招く勾配が更新を支配し、学習を誤らせる可能性があるためと考えられます。一方、正解サンプルのみを削除する「強化学習 + 全ての正解サンプルを削除」は、あまり効果がありません。一方、全ての正解サンプルと誤学習サンプルの両方を削除すると、エントロピー損失がより安定し、報酬がわずかに向上するため、探索の維持に役立つことが示唆されます。

また、「強化 + 平均ゼロ」バリアントのような正規化のみでは、KLダイバージェンスが増大し、報酬が改善されないことも確認されており、潜在的な不安定性を示唆しています。さらに、標準偏差の正規化（「強化 + 両方削除 + 標準偏差正規化」）を適用しても、単に不良サンプルを削除する場合と比べてほとんど追加の効果が得られないことから、分散の正規化はパフォーマンスに大きく貢献していないことが示唆されます。

これらの結果を総合すると、GRPOの核となる強みは、正規化そのものではなく、低品質（特に誤った）サンプルを棄却することにあることが浮き彫りになります。正しいサンプルと誤ったサンプルの両方を棄却する変種（「強化 + 両方を棄却」）をReinforce-Rejと呼び、これはLLMにおける報酬ベースの方策最適化のための、簡略化されながらも競争力のあるベースラインとして機能します。

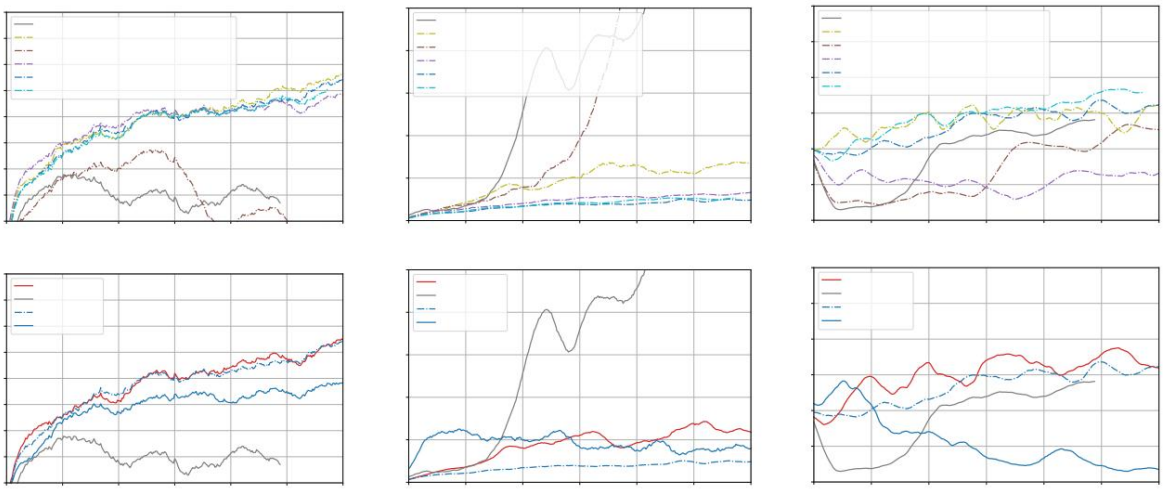

図4: LLaMA-3.2-3B-instructを用いたGRPO型アルゴリズムとReinforce型アルゴリズムの構成要素に関するアブレーション研究。GRPOを他のReinforceベースのバリアントと比較し、誤ったサンプルの除去、正しいサンプルの除去、および正規化の適用の効果を分離しています。誤ったサンプルの除去（「すべての誤ったサンプルを除去する」）は報酬の増加が最も大きく、その有害な影響が浮き彫りになっています。一方、正しいサンプルの除去の報酬は依然として不十分です。平均ゼロ正規化はKL損失を増加させ、学習を不安定にします。

標準偏差による正規化は、追加的な効果は最小限に抑えられます。「強化 + 両方除去」という選択肢は、報酬、KL安定性、エントロピー正則化のバランスを良好に保っています。元の報酬を(1 + r)/2を用いて変換することで、結果の値が訓練データの精度と一致するようにします。

また、曲線を滑らかにするために、ウィンドウ サイズ 20 の移動平均も適用します。

5 結論

我々は、LLM後学習における強化学習アルゴリズムの設計空間を、棄却サンプリングの観点から再検討した。本研究では、正の報酬を与えられたサンプルのみを用いた単純な棄却ベースの手法であるRAFTが、驚くほど強力なベースラインとして機能し、PPOや反復DPOといったより洗練された手法を上回る、あるいは同等の性能を示すことを示した。我々は、RAFTをさらに改良するために、以下の手法を組み込んだ。

重要度サンプリングとクリッピングにより、シンプルで安定したトレーニング パイプラインを維持しながら、ほぼ最先端のパフォーマンスを実現する RAFT++ が実現しました。

広範なアブレーションを通して、GRPOの主な利点は報酬の正規化ではなく、完全に正しい応答と完全に間違った応答の両方を含むプロンプトを破棄することにあることを明らかにしました。この知見に基づき、完全に間違った応答と完全に正しい応答の両方を含むサンプルをフィルタリングする、最小限のポリシー勾配変種であるReinforce-Rejを提案しました。Reinforce-RejはKL効率とエントロピー安定性を向上させ、報酬ベースのファインチューニングにおける探索の役割を浮き彫りにします。

私たちの研究結果は、RLベースのLLM訓練におけるネガティブサンプルの有用性は、これまで考えられていたよりも微妙であることを示唆しています。将来の手法では、生のネガティブフィードバックに頼るのではなく、サンプルの質を組み込むためのより選択的かつ原理的なメカニズムを検討する必要があります。RAFTとReinforce-Rejは、訓練後の報酬駆動型LLMに関する将来の研究において、軽量で解釈可能かつ効果的なベースラインとして推奨されます。

参考文献

Ahmadian, A., Cremer, C., Gall´e, M., Fadaee, M., Kreutzer, J., Pietquin, O., Ust¨un, A., および Hooker, S. (2024). 基本に立ち返る：LLMSにおける人間からのフィードバックからの学習のための強化スタイルの最適化の再検討。arXivプレプリント arXiv:2402.14740。

Anthony, T., Tian, Z., Barber, D. (2017). ディープラーニングとツリー探索によるファスト&スロー思考. 神経情報処理システムの進歩、30。

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. (2022). 人間からのフィードバックからの強化学習を用いた、親切かつ無害なアシスタントのトレーニング.arXivプレプリント arXiv:2204.05862.

Beeching, E.、Huang, SC、Jiang, A.、Li, J.、Lipkin, B.、Qina, Z.、Rasul, K.、Shen, Z.、Soletskyi, R.、および Tunstall, L. (2024)。 Numinamath 7b コット。 https://huggingface.co/AI-MO/NuminaMath-7B-CoT。

DeepSeek-AI、Guo, D.、Yang, D.、Zhang, H.、Song, J.、Zhang, R.、Xu, R.、Zhu, Q.、Ma, S.、Wang, P.、Bi, X.、Zhang, X.、Yu, X.、Wu, Y.、Wu, ZF、Gou, Z.、Shao, Z.、Li, Z.、Gao, Z.、Liu, A.、Xue、 B.、Wang、B.、Wu、B.、Feng、B.、Lu、C.、Zhao、C.、Deng、C.、Zhang、C.、Ruan、C.、Dai、D.、Chen、D.、Ji、D.、Li、E、Lin、F.、Dai、F.、Luo、F.、Hao、G.、Chen、G.、Li、G.、Zhang、H.、Bao、H.、徐、H、王、 H.、Ding、H.、Xin、H.、Gao、H.、Qu、H.、Li、H.、Guo、J.、Li、J.、Wang、J.、Chen、J.、Yuan、J.、Qiu、J.、Li、J.、Cai、JL、Ni、J.、Liang、J.、Chen、J.、Dong、K.、Hu、K.、Gao、K.、Guan、 K.、Huang、K.、Yu、K.、Wang, L.、Zhang, L.、Zhao, L.、Wang, L.、Zhang, L.、Xu, L.、Xia, L.、Zhang, M.、Zhang, M.、Tang, M.、Li, M.、Wang, M.、Li, M.、Tian, N.、Huang, P.、Zhang, P.、Wang, Q.、Chen、 Q.、Du、Q.、Ge、R.、Zhang、 R.、Pan、R.、Wang、R.、Chen、RJ、Jin、RL、Chen、R.、Lu、S.、Zhou、S.、Chen、S.、Ye、S.、Wang、S.、Yu、S.、周、S.、Pan、S.、Li、SS、周、S.、Wu、S.、Ye、S.、Yun、T.、Pei、T.、Sun、T.、Wang、 T.、Zeng、W.、Zhao、W.、Liu, W.、Liang, W.、Gao, W.、Yu, W.、Zhang, W.、Xiao, WL、An, W.、Liu, X.、Wang, X.、Chen, X.、Nie, X.、Cheng, X.、Liu, X.、Xie, X.、Liu, X.、Yang, X.、Li, X.、Su, X.、リン、X.、リー、XQ、ジン、 X.、Shen、X.、Chen、X.、Sun、X.、Wang、X.、Song、X.、Zhou、X.、Wang、X.、Shan、X.、Li、YK、Wang、YQ、Wei、YX、Zhang、Y.、Xu、Y.、Li、Y.、Zhao、Y.、Sun、Y.、Wang、Y.、Yu、Y.、Zhang、Y.、Shi、Y.、 Xiong、Y.、He、Y.、Piao、Y.、Wang、Y.、Tan、Y.、Ma、Y.、Liu、Y.、Guo、Y.、Ou、Y.、Wang、Y.、Gong、Y.、Zou、Y.、彼、Y.、Xiong、Y.、Luo、Y.、You、Y.、Liu、Y.、周、Y.、Zhu、YX、Xu、Y.、Huang、 Y.、リー、Y.、鄭義、朱裕、馬毅、唐勇、趙勇、燕勇、任、ZZ、仁、沙、傅、徐、謝、張、邯、邢、馬、燕、呉、呉、顧、朱、劉、李、 Xie,Z.、Song,Z.、Pan,Z.、Huang,Z.、Xu,Z.、Zhang,Z.、およびZhang,Z.

(2025). Deepseek-r1: 強化学習によるLLMにおける推論能力のインセンティブ化。

Dong, H., Xiong, W., Goyal, D., Zhang, Y., Chow, W., Pan, R., Diao, S., Zhang, J., SHUM, K., Zhang, T. (2023). RAFT: 生成的基盤モデルのアライメントのための報酬ランク付けファインチューニング. Transactions on Machine Learning Research.

Dong, H., Xiong, W., Pang, B., Wang, H., Zhao, H., Zhou, Y., Jiang, N., Sahoo, D., Xiong, C., Zhang, T. (2024). Rlhfワークフロー：報酬モデリングからオンラインrlhfへ.arXivプレプリント arXiv:2405.07863.

Grattafiori, A.、Dubey, A.、Jauhri, A.、Pandey, A.、Kadian, A.、Al-Dahle, A.、Letman, A.、Mathur, A.、Shelten, A.、Vaughan, A.、他。 （2024年）。ラマ 3 の群れのモデル。 arXiv プレプリント arXiv:2407.21783。

He, C.、Luo, R.、Bai, Y.、Hu, S.、Thai、ZL、Shen, J.、Hu, J.、Han, X.、Huang, Y.、Zhang, Y.、他。 （2024年）。Olympiadbench: オリンピックレベルのバイリンガルマルチモーダル科学問題で AGI を促進するための挑戦的なベンチマーク。arXiv プレプリント arXiv:2402.14008。

Hendrycks, D.、Burns, C.、Kadavath, S.、Arora, A.、Basart, S.、Tang, E.、Song, D.、および Steinhardt, J. (2021)。数学データセットを使用した数学の問題解決の測定。arXiv プレプリント arXiv:2103.03874。

ホアン•トラン、クリス•グレイズ、BH (2024). シュノーケル•ミストラル•ペアrm-dpo. https://huggingface.co/snorkelai/シュノーケル-ミストラル-ペアRM-DPO。

Hu, J. (2025). Reinforce++: 大規模言語モデルのアラインメントのためのシンプルで効率的なアプローチ. arXiv プレプリントarXiv:2501.03262.

ジェイク、A.、カライ、A.、レラー、A.、リチャードソン、A.、エルキシュキー、A.、ロウ、A.、ヘリヤー、A.、マドリー、A.、ボイテル、A.、Carney, A., et al. (2024). Openai o1 システムカード. arXiv プレプリント arXiv:2412.16720.

Kool, W., van Hoof, H., Welling, M. (2019). 強化サンプルを4つご購入いただくと、ベースラインが無料で付いてきます！

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., 他 (2022). 言語モデルを用いた定量的推論問題の解決. 神経情報処理システムの進歩、35:3843–3857。

Li, Z., Xu, T., Zhang, Y., Yu, Y., Sun, R., Luo, Z.-Q. (2023). Remax: 大規模言語モデルのアラインメントのためのシンプルで効果的かつ効率的な強化学習手法 .arXiv e-prints, pages arXiv–2310.

Liu, T., Zhao, Y., Joshi, R., Khalman, M., Saleh, M., Liu, PJ, Liu, J. (2023). 統計的棄却サンプリングは選好最適化を改善する .arXivプレプリント arXiv:2309.06657.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). 人間のフィードバックを用いた指示に従う言語モデルの学習. Advances in Neural Information Processing Systems, 35:27730–27744.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, CD, and Finn, C. (2023). 直接選好最適化：言語モデルは実は報酬モデルである .arXivプレプリント arXiv:2305.18290.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O. (2017). 近接政策最適化アルゴリズム。 arXiv プレプリント arXiv:1707.06347。

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Zhang, M., Li, Y., Wu, Y., Guo, D. (2024). Deepseekmath: オープン言語モデルにおける数学的推論の限界を押し広げる. arXiv プレプリント arXiv:2402.03300.

Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., および Wu, C. (2024)。Hybridflow: 柔軟かつ効率的な rlhf フレームワーク。arXiv プレプリント arXiv: 2409.19256。

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: オープンな基盤と微調整されたチャットモデル. arXiv プレプリント arXiv:2307.09288.

Williams, RJとPeng, J. (1991). コネクショニスト強化学習アルゴリズムを用いた関数の最適化. Connection Science, 3(3):241–268.

Xiong, W., Dong, H., Ye, C., Wang, Z., Zhong, H., Ji, H., Jiang, N., Zhang, T. (2023). 人間のフィードバックからの反復的選好学習：KL制約下におけるRLHFの理論と実践の橋渡し.

Xu, J., Lee, A., Sukhbaatar, S., Weston, J. (2023). あるものは他のものより恥ずかしい :好み
ペアワイズクリンジ損失による最適化。arXiv プレプリント arXiv:2312.16682。

Yang, A.、Yang, B.、Zhang, B.、Hui, B.、Zheng, B.、Yu, B.、Li, C.、Liu, D.、Huang, F.、Wei, H.、他。 （2024年）。
Qwen2.5技術レポート。arXivプレプリントarXiv:2412.15115。

Yu, Q.、Zhang, Z.、Zhu, R.、Yuan, Y.、Zuo, X.、Yue, Y.、Fan, T.、Liu, G.、Liu, L.、Liu, X.、他。 (2025年)。
Dapo: 大規模なオープンソースの LLM 強化学習システム。arXiv プレプリント arXiv:2503.14476。

Yuan, Z., Yuan, H., Li, C., Dong, G., Tan, C., Zhou, C. (2023). 学習におけるスケーリング関係
大規模言語モデルによる数学的推論。arXiv プレプリント arXiv:2308.01825。

Zelikman, E., Wu, Y., Mu, J., Goodman, N. (2022). Star: Bootstrapping Reasoning with Reasoning.
神経情報処理システムの進歩、35:15476–15488。

Zhang, H., Yao, J., Ye, C., Xiong, W., Zhang, T. (2025). Online-dpo-r1: 効果的な推論を解き放つ
PPO オーバーヘッドなし。

Zhao, Y., Joshi, R., Liu, T., Khalman, M., Saleh, M., Liu, PJ (2023). Slic-hf: 人間によるフィードバックを用いたシーケンス尤度キャリブレーション. arXiv プレプリント arXiv:2305.10425.