On the Policy Gradient Foundations of Group Relative Policy Optimization: Credit Assignment, Gradient Sparsity, and Rank Collapse

Amritansh Mishra [1] **Supriyo Chakraborty** [1] **Berkcan Kapusuzoglu** [1]

Abstract

Group Relative Policy Optimization (GRPO) eliminates the learned critic in PPO by using the mean reward of grouped rollouts as a baseline. We provide a rigorous derivation of GRPO from first principles of the policy gradient theorem, revealing a fundamental credit assignment failure: under output-only reward, every token in a rollout receives identical advantage, collapsing token-level credit to a single scalar. We prove this induces gradient sparsity that intensifies over training, and demonstrate empirically via SVD analysis of GRPO gradients on Nemotron-4B/GSM8K that the gradient matrix has effective rank ≈ 2 regardless of group size $R \in \{2, 4, 8\}$. We formalize this as an intrinsic rank-2 structure arising from the zero-sum constraint on advantages and derive conditions under which GRPO's baseline is optimal. Our results characterize when GRPO's simplicity is theoretically justified and identify the credit assignment bottleneck as the key limitation for multi-step reasoning.

1. Introduction

Policy gradient methods have become central to post-training of large language models, both for aligning with human preferences (Ouyang et al., 2022) and for improving reasoning capabilities (DeepSeek-AI, 2025; Shao et al., 2024). The standard approach uses Proximal Policy Optimization (PPO) (Schulman et al., 2017), which requires a learned value function (critic) to estimate advantages doubling memory and compute footprint.

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) offers an elegant alternative for each prompt, sample R completions, compute their rewards, and use the group mean as the baseline. This eliminates the critic entirely. GRPO has been successfully deployed in DeepSeek-R1 (DeepSeek-AI, 2025) and other systems, but its theoretical properties-particularly around *credit assignment* and its links to Policy Gradient Methods remain poorly understood.

In this paper, we show that the GRPO gradient is fundamentally a policy gradient estimator, and derive it step by step from the policy gradient theorem. By making explicit the assumptions under which GRPO is formulated—output-only reward and a group-mean baseline—we reveal a structural consequence: **GRPO assigns identical credit to every token in a sequence**. This uniform credit assignment is not merely a simplification but a fundamental property that shapes the gradient's spectral characteristics.

Contributions.

1. We derive GRPO as a special case of the REINFORCE estimator with a group-mean baseline, making all approximations explicit (Section 2).
2. We prove that output-only reward induces a *uniform credit assignment* where every token receives the same advantage, and characterize the resulting gradient sparsity (Section 3).
3. We empirically demonstrate that the GRPO gradient matrix collapses to an effective rank ≈ 2 independent of group size R, and formalize how the zero-sum constraint drives this behavior.
4. We validate our theory empirically: SVD analysis of GRPO gradients on Nemotron-4B/GSM8K confirms effective rank ≈ 2 for $R \in \{2, 4, 8\}$ (Section 7).
5. We also explain multi turn limitations of GRPO.

Scope. We focus on the policy gradient core of GRPO the advantage estimator and gradient structure deliberately setting aside clipping and KL regularization, which affect the optimization step but not the gradient signal's fundamental properties.

[1]Capital One. Correspondence to: Amritansh Mishra <amritansh.mishra@capitalone.com>.

Published as a paper at the 1st FoGen workshop, ICML 2026, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

2. From Policy Gradients to GRPO

We derive GRPO from first principles, following the chain: value functions → policy gradient theorem → REINFORCE with baseline → advantage estimation → GRPO.

2.1. MDP Formulation for Text Generation

We model autoregressive generation as an episodic MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \rho_0)$. A state s_t is the concatenation of the prompt and tokens generated so far. An action $a_t \in \mathcal{A}$ (the vocabulary) appends a token. The transition P is deterministic (concatenation). The initial distribution ρ_0 is the prompt distribution.

Definition 2.1 (Value Functions)**.** For policy π_θ with parameters $\theta \in \mathbb{R}^d$:

$$v^\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s], \quad G_t = \sum_{k=0}^{T-t-1} \gamma^k r_{t+k+1}, \tag{1}$$

$$q^\pi(s,a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a], \tag{2}$$

$$A^\pi(s,a) = q^\pi(s,a) - v^\pi(s). \tag{3}$$

2.2. Policy Gradient Theorem

The objective $J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[G_0]$ has gradient (Sutton et al., 1999):

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T-1} \gamma^t q^{\pi_\theta}(S_t, A_t) \nabla_\theta \log \pi_\theta(A_t \mid S_t)\right]. \tag{4}$$

2.3. The REINFORCE Estimator

The REINFORCE algorithm (Williams, 1992) estimates the policy gradient by replacing the expected return with a single Monte Carlo sample. Given a trajectory $\tau = (s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T)$, the estimator is:

$$\hat{g}_{\text{RF}} = \sum_{t=0}^{T-1} \gamma^t \hat{G}_t \nabla_\theta \log \pi_\theta(A_t \mid S_t), \tag{5}$$

where $\hat{G}_t = \sum_{k=0}^{T-t-1} \gamma^k r_{t+k+1}$ is the empirical return from time t. The key insight is that $\hat{G}_t$ is a Monte Carlo estimate of $q^\pi(S_t, A_t)$: by definition, $q^\pi(s,a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$, so a single rollout from state s_t after taking action a_t gives an unbiased (but high-variance) sample of the action-value function. This substitution replacing q^π with the realized return from the trajectory is what makes REINFORCE a practical algorithm, at the cost of high variance.

2.4. Variance Reduction via Baselines

To reduce variance, we can subtract any state-dependent baseline $b(S_t)$ from the return. This does not introduce bias because:

$$\mathbb{E}_{\pi_\theta}\left[b(S_t)\nabla_\theta \log \pi_\theta(A_t \mid S_t)\right] = b(S_t)\sum_a \nabla_\theta \pi_\theta(a \mid S_t) = b(S_t)\cdot\nabla_\theta 1 = 0. \tag{6}$$

Therefore we can write:

$$\hat{g}_{\text{RF}} = \sum_{t=0}^{T-1} \gamma^t (\hat{G}_t - b(S_t)) \nabla_\theta \log \pi_\theta(A_t \mid S_t). \tag{7}$$

The variance-minimizing choice is $b^*(s) = v^\pi(s)$. To see why this yields the advantage, note that the return $\hat{G}_t$ can be decomposed via the Bellman equation. In the one-step (TD) form:

$$\hat{G}_t = r_{t+1} + \gamma \hat{G}_{t+1} = r_{t+1} + \gamma v^\pi(S_{t+1}). \tag{8}$$

Subtracting the baseline $b(S_t) = v^\pi(S_t)$:

$$\hat{G}_t - v^\pi(S_t) \approx r_{t+1} + \gamma v^\pi(S_{t+1}) - v^\pi(S_t) = \delta_t, \tag{9}$$

which is the temporal difference (TD) error. In expectation, this equals the advantage: $\mathbb{E}[\hat{G}_t - v^\pi(S_t) \mid S_t = s, A_t = a] = q^\pi(s,a) - v^\pi(s) = A^\pi(s,a)$. Thus the policy gradient becomes:

$$\nabla_\theta J(\theta) = \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t A^\pi(S_t, A_t) \nabla_\theta \log \pi_\theta(A_t \mid S_t)\right]. \tag{10}$$

This is the advantage actor-critic gradient. The challenge is that computing $v^\pi(s)$ exactly requires a learned critic which is precisely what GRPO eliminates.

2.5. Deriving GRPO

GRPO arises from two specializations of the advantage-based policy gradient (Equation (10)).

Specialization 1: Output-only reward (ORM). In LLM training with an outcome reward model, reward is given only at the terminal state: $r(s_t, a_t) = 0$ for all $t < T-1$, and $r(s_{T-1}, a_{T-1}) = R_{\text{terminal}}$. Setting $\gamma = 1$, consider what happens to the advantage at each token position.

The return at any time step t is:

$$\hat{G}_t = \sum_{k=t}^{T-1} r(s_k, a_k) = R_{\text{terminal}} \quad \forall\, t \in \{0, \ldots, T-1\}, \tag{11}$$

since all intermediate rewards are zero. Now consider the value function: since the only reward comes at the end and $\gamma = 1$, the value of *any* intermediate state s_t along a trajectory is the expected terminal reward conditioned on being in that state. But crucially, once we condition on the full trajectory (as in REINFORCE), the realized return is the same R_{terminal} regardless of which token position t we evaluate.

This means the TD error at every position becomes:

$$\delta_t = r_{t+1} + v^\pi(s_{t+1}) - v^\pi(s_t) = 0 + v^\pi(s_{t+1}) - v^\pi(s_t) \tag{12}$$

and the full Monte Carlo advantage $\hat{G}_t - v^\pi(s_t) = R_{\text{terminal}} - v^\pi(s_t)$. Since all states share the same initial prompt s_0 and the value function under output-only reward satisfies $v^\pi(s_0) = \mathbb{E}_\pi[R_{\text{terminal}} \mid s_0]$, the advantage estimate for the entire trajectory collapses to a single scalar:

$$A_t = R_{\text{terminal}} - v^\pi(s_0) \quad \forall\, t. \tag{13}$$

Every token receives the same advantage. The first token, the critical reasoning step, and filler tokens all get identical credit.

Specialization 2: Group-mean baseline. The remaining challenge is estimating $v^\pi(s_0)$. Instead of learning a critic, GRPO estimates it by sampling R rollouts from the same prompt and computing the sample mean:

$$\hat{v}_R(s_0) = \frac{1}{R}\sum_{i=1}^{R} r_i, \tag{14}$$

where r_i is the terminal reward of rollout i. This is a valid Monte Carlo estimate of $v^\pi(s_0) = \mathbb{E}_\pi[R_{\text{terminal}} \mid s_0]$.

Substituting both specializations into Equation (7), the GRPO gradient for a single prompt is:

$$\boxed{\hat{g}_{\text{GRPO}} = \frac{1}{R}\sum_{i=1}^{R} \underbrace{(r_i - \hat{v}_R)}_{\hat{A}_i} \sum_{t=0}^{T_i-1} \nabla_\theta \log \pi_\theta(a_t^{(i)} \mid s_t^{(i)}).} \tag{15}$$

This derivation makes explicit that GRPO is REINFORCE with a specific baseline choice, operating under the output-only reward assumption. The key consequence is immediate: $\hat{A}_i$ does not depend on t.

Note: The full GRPO objective also includes importance sampling clipping (analogous to PPO) and a KL divergence penalty against a reference policy. We omit these terms here as they affect the optimization step size but not the fundamental structure of the gradient signal—the credit assignment and rank properties we analyze depend only on the advantage estimator.

3. Credit Assignment Failure

We now formalize the credit assignment implications of Equation (15). Recall that in GRPO, we sample R rollouts from a single prompt s_0 and compute one shared baseline $\hat{v}_R(s_0) = \frac{1}{R}\sum_{i=1}^{R} r_i$. The advantage for rollout i is then $\hat{A}_i = r_i - \hat{v}_R(s_0)$—a scalar that measures how much better (or worse) rollout i performed relative to the group. Note that $\hat{A}_i \neq 0$ in general; it is positive for above-average rollouts and negative for below-average ones.

The critical observation is what happens *within* each rollout:

Theorem 3.1 (Uniform Token-Level Credit)**.** *Under output-only reward with $\gamma = 1$, the advantage assigned to every token position t within rollout i is the same scalar:*

$$\hat{A}_i(t) = r_i - \hat{v}_R(s_0) = \hat{A}_i \quad \forall\, t \in \{0, \ldots, T_i - 1\}. \tag{16}$$

That is, GRPO assigns a single rollout-level credit signal uniformly across all token positions. The per-token gradient contribution is:

$$\hat{g}_i(t) = \hat{A}_i \cdot \nabla_\theta \log \pi_\theta(a_t^{(i)} \mid s_t^{(i)}), \tag{17}$$

*where $\hat{A}_i$ depends on the group of R rollouts but **not** on which token position t within rollout i we are at.*

Proof. Under ORM, the return at any position t within rollout i is $G_t^{(i)} = R_{\text{terminal}}^{(i)} = r_i$ (since all intermediate rewards are zero and $\gamma = 1$). The baseline $\hat{v}_R(s_0) = \frac{1}{R}\sum_{j=1}^{R} r_j$ is computed once from the R group rewards and shared across all positions. Therefore $\hat{A}_i(t) = G_t^{(i)} - \hat{v}_R(s_0) = r_i - \hat{v}_R(s_0) = \hat{A}_i$, independent of t. □

Remark 3.2 (Credit Assignment Interpretation). The advantage $\hat{A}_i$ is non-zero—it captures whether rollout i is better or worse than the group average. However, this credit signal cannot distinguish *which tokens* within rollout i were responsible for the outcome. A rollout that succeeds because of a single correct reasoning step in the middle has every token—including irrelevant preamble and filler—reinforced equally. GRPO provides *inter-rollout* credit (which rollouts are good) but no *intra-rollout* credit (which tokens within a rollout are good).

3.1. Gradient Factorization

Since $\hat{A}_i$ is the same for every token in rollout i, we can factor it out of the inner sum in Equation (15). Define the *trajectory score* of rollout i as the sum of all per-token log-probability gradients:

$$\psi_i = \sum_{t=0}^{T_i-1} \nabla_\theta \log \pi_\theta(a_t^{(i)} \mid s_t^{(i)}) \in \mathbb{R}^d. \tag{18}$$

This is a single vector that represents the direction in parameter space that would increase the likelihood of the entire sequence i. The GRPO gradient then simplifies to:

$$\hat{g}_{\text{GRPO}} = \frac{1}{R}\sum_{i=1}^{R} \hat{A}_i \cdot \psi_i. \tag{19}$$

Each rollout contributes one scalar weight ($\hat{A}_i$, how good the rollout was) times one direction (ψ_i, how to make it more

likely). If GRPO had token-level advantages, we could not factor this way—different tokens would pull in different directions with different magnitudes. The uniform credit is what enables this rank-reducing factorization.

4. Structural Constraints on Advantages

The factorized form Equation (19) reveals that the GRPO gradient is entirely determined by the advantage vector $(\hat{A}_1, \ldots, \hat{A}_R)$ and the trajectory scores $(\psi_1, \ldots, \psi_R)$. We now characterize structural properties of the advantage vector that constrain the gradient.

Theorem 4.1 (Zero-Sum and Correlation Structure)**.** *The GRPO advantages satisfy:*

(i) ***Zero-sum:*** $\sum_{i=1}^{R} \hat{A}_i = 0$. *The advantages within a group always cancel—positive and negative rollouts exactly balance.*

(ii) ***Variance:*** $\mathrm{Var}(\hat{A}_i) = \frac{R-1}{R}\sigma_r^2$, *where* $\sigma_r^2 = \mathrm{Var}_{\pi_\theta}(r \mid s_0)$ *is the reward variance under the current policy.*

(iii) ***Negative correlation:*** $\mathrm{Cov}(\hat{A}_i, \hat{A}_j) = -\frac{\sigma_r^2}{R}$ *for* $i \neq j$. *Making one rollout's advantage more positive necessarily pushes others more negative.*

(iv) ***ϵ-Effective rollouts:*** *Define a rollout i as ϵ-effective if $|\hat{A}_i| = |r_i - \hat{v}_R| > \epsilon$. The expected fraction of ϵ-effective rollouts is bounded by:*

$$\rho_\epsilon = \mathbb{P}\left(|\hat{A}_i| > \epsilon\right) \leq \frac{(R-1)\sigma_r^2}{R\epsilon^2}. \tag{20}$$

Only ϵ-effective rollouts contribute meaningful gradient signal; the rest produce near-zero updates.

Proof. (i) $\sum_i \hat{A}_i = \sum_i (r_i - \hat{v}_R) = \sum_i r_i - R \cdot \frac{1}{R}\sum_i r_i = 0$.

(ii) Write $\hat{A}_i = r_i - \frac{1}{R}\sum_j r_j = \frac{R-1}{R} r_i - \frac{1}{R}\sum_{j \neq i} r_j$. Since the r_j are i.i.d.: $\mathrm{Var}(\hat{A}_i) = \left(\frac{R-1}{R}\right)^2 \sigma_r^2 + \frac{R-1}{R^2}\sigma_r^2 = \frac{R-1}{R}\sigma_r^2$.

(iii) From $\mathrm{Var}(\sum_i \hat{A}_i) = 0$: $R\,\mathrm{Var}(\hat{A}_i) + R(R-1)\,\mathrm{Cov}(\hat{A}_i, \hat{A}_j) = 0$, giving $\mathrm{Cov}(\hat{A}_i, \hat{A}_j) = -\sigma_r^2/R$.

(iv) By Chebyshev's inequality: $\mathbb{P}(|\hat{A}_i| > \epsilon) \leq \mathrm{Var}(\hat{A}_i)/\epsilon^2 = \frac{R-1}{R}\sigma_r^2/\epsilon^2$.

□

Corollary 4.2 (Gradient Signal Decay)**.** *The magnitude of the gradient $\|\hat{g}_{GRPO}\|$ scales with σ_r—the reward standard deviation under the current policy. As training progresses and the policy converges (i.e., it gets most problems right or most wrong for a given prompt), $\sigma_r \to 0$ and the advantages $\hat{A}_i \to 0$ for all i simultaneously. When all rollouts from a prompt receive the same reward, $\hat{A}_i = 0$ exactly and the prompt contributes zero gradient.*

The zero-sum constraint means GRPO can only learn from prompts where the R rollouts produce *diverse* rewards—some correct, some incorrect. Prompts that are too easy (all correct) or too hard (all wrong) yield $\hat{A}_i = 0$ for every rollout and contribute nothing to learning. This makes GRPO a natural *curriculum*: it automatically focuses on prompts at the boundary of the policy's capability, where reward variance is highest.

5. Sparse Gradient Updates and Rank Collapse

We now discuss how the GRPO gradient structure encourages sparse parameter updates and empirically leads to low-rank gradients.

5.1. Why Updates Are Sparse

From Equation (19), the gradient is $\hat{g} = \frac{1}{R}\sum_i \hat{A}_i \psi_i$. Two structural properties conspire to make this sparse:

(1) Zero-sum advantages concentrate signal. Since $\sum_i \hat{A}_i = 0$, the gradient is determined entirely by the *contrast* between rollouts. Consider binary reward ($r \in \{0, 1\}$) as in GSM8K: if k of R rollouts are correct, then the correct rollouts have $\hat{A}_i = 1 - k/R$ and incorrect ones have $\hat{A}_i = -k/R$. When k is close to 0 or R (easy/hard prompts), the advantages become small and the gradient vanishes. Only prompts where some rollouts succeed and some fail produce meaningful signal.

(2) Uniform credit concentrates in few directions. Because $\hat{A}_i$ is a single scalar for all tokens in rollout i, the gradient for each rollout points in one direction ψ_i. If trajectory scores are similar (e.g., rollouts from the same prompt share syntactic structure), the R directions ψ_i cluster. We hypothesize that this clustering, combined with the zero-sum advantages, is what empirically concentrates the gradient into a low-dimensional (often rank-2) subspace.

6. Multi-Turn Limitation

The derivation in Section 2 reveals why GRPO is inherently a single-turn method. Recall the standard advantage at state s_t:

$$A^\pi(s_t, a_t) = q^\pi(s_t, a_t) - v^\pi(s_t) = r_{t+1} + \gamma v^\pi(s_{t+1}) - v^\pi(s_t). \tag{21}$$

This requires bootstrapping from the *next state's value* $v^\pi(s_{t+1})$—which is what a learned critic provides in PPO.

GRPO has no critic. It uses the group-mean baseline $\hat{v}_R(s_0)$, which estimates the value of the *initial state* s_0 only. In the

single-turn ORM setting this is fine: with $\gamma = 1$ and $r_t = 0$ for $t < T - 1$, the advantage at every position reduces to $R_{\text{terminal}} - v^\pi(s_0)$, and $\hat{v}_R(s_0)$ is an unbiased estimate of $v^\pi(s_0)$.

But in multi-turn settings with per-step rewards $r_1, \ldots, r_H$, the true advantage at stage h requires $v^\pi(s_h)$—the value of the *current* state, not the initial state. GRPO still subtracts $\hat{v}_R(s_0)$ from the total return:

$$\hat{A}_{\text{GRPO}}(h) = \sum_{h'=1}^{H} r_{h'} - \hat{v}_R(s_0). \tag{22}$$

This conflates past rewards (already received, not actionable) with future rewards (what the agent can still influence), introducing bias:

$$\text{Bias}_h = \sum_{h'=1}^{h-1} r_{h'} + v^\pi(s_h) - v^\pi(s_0), \tag{23}$$

which grows linearly with stage h. The first term is accumulated past reward; the second is the value mismatch from using $v^\pi(s_0)$ instead of $v^\pi(s_h)$.

In short: GRPO cannot perform the bootstrapping step $\gamma v^\pi(s_{t+1})$ because it has no value function. This makes it exact for single-turn generation (where bootstrapping is unnecessary) but introduces growing bias for multi-turn problems where the value of states changes along the trajectory.

7. Experiments

We validate our theoretical predictions empirically using GRPO training of Nemotron-4B on GSM8K.

7.1. Setup

We train Nemotron-4B (NVIDIA, 2024) with GRPO on GSM8K (mathematical reasoning, binary correctness reward). We vary the group size $R \in \{2, 4, 8\}$ across 3 random seeds. Using a custom gradient SVD tracker integrated into our FSDP training pipeline, we explicitly capture the fully reduced flattened gradients ($\nabla_\theta J$) for tracked linear layers (e.g., `q_proj`, `v_proj`) every 50 optimizer steps. For each tracked layer at each checkpoint, we stack the per-rollout gradient contributions into a matrix $G \in \mathbb{R}^{R \times d}$ and compute its truncated SVD.

Metrics:

- **Effective rank**: $r_{\text{eff}} = \left(\sum_i \sigma_i\right)^2 / \sum_i \sigma_i^2$
- **Top-1 fraction**: $\sigma_1^2 / \sum_i \sigma_i^2$ (variance explained by first component)
- **Reward std**: $\sigma_r(k)$ at training step k

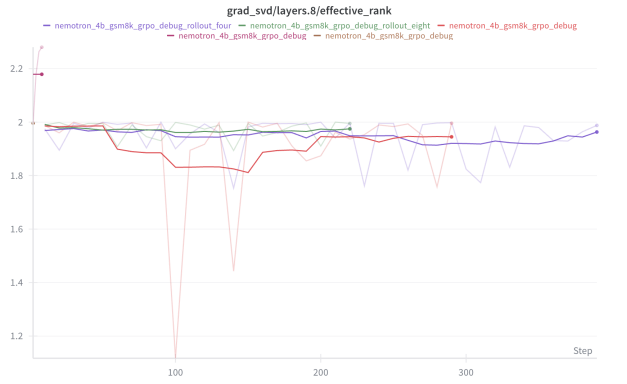

Figure 1. Effective rank of the GRPO gradient matrix across training steps for $R \in \{2, 4, 8\}$. Despite the maximum possible rank being higher, effective rank remains ≈ 2 throughout training.

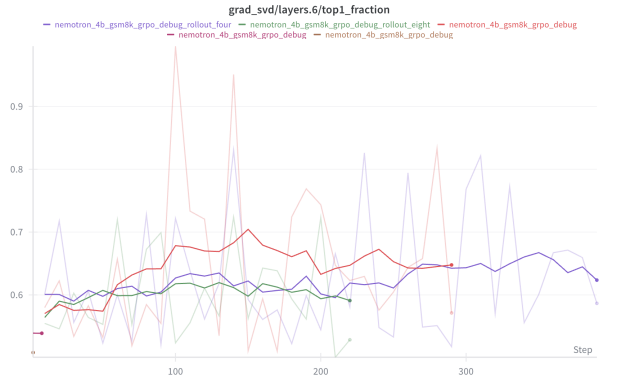

Figure 2. Top-1 singular value fraction ($\sigma_1^2 / \sum_i \sigma_i^2$) across training. $R = 4$ shows the highest concentration, while $R = 8$ diffuses energy across more components. This non-monotonic pattern is explained by the interplay between advantage concentration and trajectory score diversity.

7.2. Results

Effective rank ≈ 2 for all R. Across all group sizes, the effective rank of the gradient matrix remains approximately 2 (Table 1). This holds despite the theoretical maximum rank being larger for groups $R \geq 2$.

Interpretation. The rank-2 structure means that **each GRPO gradient step moves parameters along only $\sim$2 directions**, regardless of whether 2 or 8 rollouts are computed. The additional rollouts for $R > 2$ refine the *magnitude* and *direction* of these two components but do not add new dimensions of movement. This has direct implications for training efficiency: increasing R provides diminishing returns in terms of gradient informativeness.

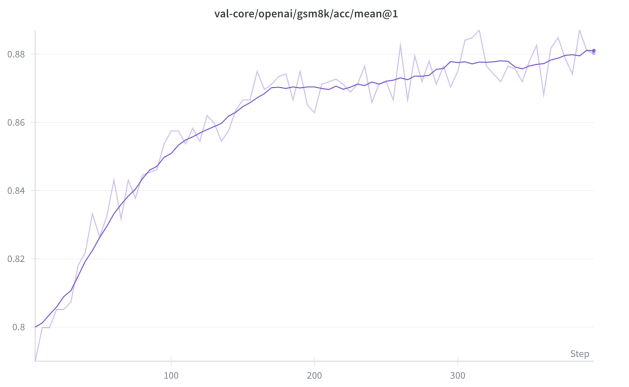

Figure 3. Training accuracy on GSM8K for different group sizes. Despite rank-2 gradient structure for all R, larger groups still improve learning speed by providing better estimates of the advantage direction $\bar{\psi}_+ - \bar{\psi}_-$.

Table 1. Spectral properties of GRPO gradient matrices (Nemotron-4B, GSM8K). Effective rank stays ≈ 2 while max possible rank grows with R.

R	MAX RANK	EFF. RANK	TOP-1 FRAC.
2	2*	$\sim$2	0.6–0.7
4	3	$\sim$2	0.7–0.8
8	7	$\sim$2	0.5–0.6

*For $R = 2$, while advantages are zero-sum ($\hat{A}_1 = -\hat{A}_2$), the trajectory scores ψ_1 and ψ_2 are linearly independent, making the maximum possible rank 2.

8. Discussion and Future Work

Connection to memorization vs. generalization. The uniform credit assignment (Theorem 3.1) means GRPO reinforces entire sequences indiscriminately. From the memorization/generalization perspective, this biases toward *template-level learning*: sequences that consistently achieve high reward are reinforced as holistic units, rather than having their compositional structure decomposed. This may explain why GRPO-trained models can reproduce successful reasoning patterns but struggle to compose them in novel ways.

Implications for multi-step reasoning. The multi-turn bias and rank-2 collapse together suggest that GRPO is poorly suited for learning *which step* in a reasoning chain matters. Process reward models (PRMs) that provide per-step credit are a natural remedy, but our analysis shows that even with dense reward, the group-relative baseline introduces structural limitations (negative correlations, zero-sum constraints) that constrain the gradient's expressiveness.

Future directions. (1) *Adaptive baselines*: Can leave-one-out (RLOO) or learned token-level baselines break the rank-2 bottleneck? (2) *Rank-aware training*: If effective rank predicts training saturation, can we use it as a signal to increase R or switch to denser rewards? (3) *Progressive rank collapse over training*: Our theory predicts effective rank should decrease as $\sigma_r \to 0$; longitudinal measurement would validate Theorem 4.2.

9. Related Work

Policy gradients and baselines. The policy gradient theorem (Sutton et al., 1999), REINFORCE (Williams, 1992), and optimal baselines (Weaver & Tao, 2001) provide the foundation. GAE (Schulman et al., 2015) offers a bias-variance tradeoff via temporal-difference mixing.

GRPO and RLHF. GRPO was introduced in DeepSeek-Math (Shao et al., 2024) and used in DeepSeek-R1 (DeepSeek-AI, 2025). RLOO (Ahmadian et al., 2024) uses a leave-one-out baseline that eliminates the $O(1/R)$ bias. DPO (Rafailov et al., 2023) avoids RL entirely.

Credit assignment in LLMs. Process reward models (Lightman et al., 2023) address token-level credit. Recent work on advantage-weighted regression and return-conditioned training explores alternative credit mechanisms.

Gradient rank in deep learning. Low-rank gradient structure has been observed in supervised learning (Li et al., 2018) and exploited in efficient fine-tuning (LoRA, Hu et al., 2021). Our work identifies a *theoretical* reason for low rank specific to GRPO's policy gradient structure.

10. Conclusion

We have shown that GRPO, derived from first principles of the policy gradient theorem, suffers from a fundamental credit assignment limitation: output-only reward forces identical advantage across all tokens, collapsing the gradient into a rank-2 structure independent of group size. This is validated empirically on Nemotron-4B/GSM8K. Our results formalize when GRPO's simplicity is theoretically justified (single-turn, high initial σ_r) and where it breaks down (multi-step reasoning requiring fine-grained credit). Breaking the rank-2 bottleneck—via denser rewards, token-level baselines, or hybrid approaches—is an important direction for scaling RLHF to complex reasoning tasks.

References

Ahmadian, A., Cremer, C., Gallé, M., Fadaee, M., Kreutzer, J., Pietquin, O., Üstün, A., and Hooker, S. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback in LLMs. *arXiv preprint arXiv:2402.14740*, 2024.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

NVIDIA. Nemotron-4 15b technical report. *arXiv preprint arXiv:2402.16819*, 2024.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 2022.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., and Guo, D. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pp. 1057–1063, 1999.

Weaver, L. and Tao, N. The optimal reward baseline for gradient-based reinforcement learning. In *Uncertainty in Artificial Intelligence*, pp. 538–545, 2001.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.

A. Proofs

A.1. Proof of Theorem 4.1 (Full Details)

Zero-sum: $\sum_{i=1}^{R} \hat{A}_i = \sum_i (r_i - \hat{v}_R) = \sum_i r_i - R\hat{v}_R = \sum_i r_i - \sum_i r_i = 0$.

Variance: Write $\hat{A}_i = r_i - \frac{1}{R}\sum_j r_j = \frac{R-1}{R} r_i - \frac{1}{R}\sum_{j \neq i} r_j$. Then:

$$\mathrm{Var}(\hat{A}_i) = \left(\frac{R-1}{R}\right)^2 \mathrm{Var}(r_i) + \frac{1}{R^2}\sum_{j \neq i} \mathrm{Var}(r_j) \tag{24}$$

$$= \frac{(R-1)^2}{R^2}\sigma_r^2 + \frac{R-1}{R^2}\sigma_r^2 = \frac{(R-1)^2 + (R-1)}{R^2}\sigma_r^2 = \frac{(R-1)R}{R^2}\sigma_r^2 = \frac{R-1}{R}\sigma_r^2. \tag{25}$$

Covariance: From $\mathrm{Var}(\sum_i \hat{A}_i) = 0$:

$$R\,\mathrm{Var}(\hat{A}_i) + R(R-1)\,\mathrm{Cov}(\hat{A}_i, \hat{A}_j) = 0 \implies \mathrm{Cov}(\hat{A}_i, \hat{A}_j) = -\frac{\mathrm{Var}(\hat{A}_i)}{R-1} = -\frac{\sigma_r^2}{R}.$$

Sparsity bound: $\mathrm{Var}(r_i - \hat{v}_R) = \mathrm{Var}(\hat{A}_i) = \frac{R-1}{R}\sigma_r^2$. By Chebyshev: $\mathbb{P}(|\hat{A}_i| > \epsilon) \leq \frac{(R-1)\sigma_r^2}{R\epsilon^2}$.

A.2. Hoeffding Inequality Analysis

The baseline $\hat{v}_R(s_0) = \frac{1}{R}\sum_{i=1}^{R} r_i$ is the arithmetic mean of R i.i.d. random variables with $r_i \in [0,1]$ and $\mathbb{E}[r_i] = v^\pi(s_0)$. By Hoeffding's inequality for bounded i.i.d. random variables:

$$\mathbb{P}\left(\left|\frac{1}{R}\sum_{i=1}^{R} r_i - \mathbb{E}[r_i]\right| > \epsilon\right) \leq 2\exp\left(\frac{-2R^2\epsilon^2}{\sum_{i=1}^{R}(b_i - a_i)^2}\right) = 2\exp\left(\frac{-2R^2\epsilon^2}{R \cdot 1^2}\right) = 2\exp(-2R\epsilon^2).$$

Interpretation by group size:

R	$\mathbb{P}(\lvert\hat{v}_R - v^\pi\rvert > 0.3)$	$\mathbb{P}(\lvert\hat{v}_R - v^\pi\rvert > 0.2)$	$\mathbb{P}(\lvert\hat{v}_R - v^\pi\rvert > 0.1)$
2	≤ 1.34 (vacuous)	≤ 1.70 (vacuous)	≤ 1.92 (vacuous)
4	≤ 0.93 (vacuous)	≤ 1.44 (vacuous)	≤ 1.85 (vacuous)
8	≤ 0.44	≤ 1.05 (vacuous)	≤ 1.71 (vacuous)
16	≤ 0.10	≤ 0.55	≤ 1.47 (vacuous)

The bound becomes non-trivial only for larger R or larger ϵ. This reflects the fact that with few rollouts ($R = 2, 4$), the baseline has substantial variance (σ_r^2/R), and the advantages are noisy estimates of the true $r_i - v^\pi(s_0)$.

Connection to advantage quality: The advantage error for rollout i is:

$$\hat{A}_i - (r_i - v^\pi(s_0)) = v^\pi(s_0) - \hat{v}_R(s_0).$$

So this directly bounds the advantage estimation error: with probability $\geq 1 - 2e^{-2R\epsilon^2}$, every rollout's advantage is within ϵ of its true value $r_i - v^\pi(s_0)$.

A.3. Multi-Turn Bias Derivation

In multi-turn with H stages, the true advantage at stage h is:

$$A^\pi(s_h, a_h) = \mathbb{E}_\pi\left[\sum_{h'=h}^{H} r_{h'} \mid s_h, a_h\right] - v^\pi(s_h).$$

GRPO, lacking a value function for intermediate states, uses the same estimate for all stages:

$$\hat{A}_{\text{GRPO}}(h) = \underbrace{\sum_{h'=1}^{H} r_{h'}}_{\text{total return}} - \underbrace{\hat{v}_R(s_0)}_{\text{baseline from initial state}}.$$

The key issue: GRPO subtracts $\hat{v}_R(s_0) \approx v^\pi(s_0)$ when it should subtract $v^\pi(s_h)$. Taking expectations conditioned on (s_h, a_h):

$$\mathbb{E}[\hat{A}_{\text{GRPO}}(h) \mid s_h, a_h] = \underbrace{\sum_{h'=1}^{h-1} r_{h'}}_{\text{past rewards (fixed)}} + \underbrace{\mathbb{E}\left[\sum_{h'=h}^{H} r_{h'} \mid s_h, a_h\right]}_{=q^\pi(s_h,a_h)} - v^\pi(s_0) \tag{26}$$

$$= \sum_{h'=1}^{h-1} r_{h'} + A^\pi(s_h, a_h) + v^\pi(s_h) - v^\pi(s_0). \tag{27}$$

Therefore:

$$\text{Bias}_h = \mathbb{E}[\hat{A}_{\text{GRPO}}(h)] - A^\pi(s_h, a_h) = \underbrace{\sum_{h'=1}^{h-1} r_{h'}}_{\text{past reward leakage}} + \underbrace{v^\pi(s_h) - v^\pi(s_0)}_{\text{value mismatch}}.$$

The bias has two sources:

1. **Past reward leakage**: Rewards from stages $1, \ldots, h-1$ are included in the return but are not attributable to action a_h. This grows as h increases.
2. **Value mismatch**: GRPO bootstraps from $v^\pi(s_0)$ instead of $v^\pi(s_h)$. In the standard advantage (Equation (21)), we subtract $v^\pi(s_t)$—the value of the current state. Without a critic, GRPO cannot compute $v^\pi(s_h)$ for $h > 0$.

Bounding: $|\,\text{Bias}_h\,| \leq (h-1)r_{\max} + |v^\pi(s_h) - v^\pi(s_0)| \leq (h-1)r_{\max} + 2V_{\max}$.

For single-turn ($H = 1$): $\text{Bias}_1 = 0$ (no past rewards, no value mismatch). GRPO is exact.