SEED-GRPO: Semantic Entropy Enhanced GRPO for Uncertainty-Aware Policy Optimization

Minghan Chen **Guikun Chen** **Wenguan Wang** **Yi Yang**
ReLER Lab, CCAI, Zhejiang University

Abstract

Large language models (LLMs) exhibit varying levels of confidence across input prompts (questions): some lead to consistent, semantically similar answers, while others yield diverse or contradictory outputs. This variation reflects LLM's uncertainty about the input prompt, a signal of how confidently the model understands a given problem. However, vanilla Group Relative Policy Optimization (GRPO) **treats all prompts equally** during policy updates, ignoring this important information about the model's knowledge boundaries. To address this limitation, we propose SEED-GRPO (**S**emantic **E**ntropy **E**nhance**D** GRPO), which explicitly measures LLMs' uncertainty of the input prompts semantic entropy. Semantic entropy measures the diversity of meaning in multiple generated answers given a prompt and uses this to modulate the magnitude of policy updates. This uncertainty-aware training mechanism enables dynamic adjustment of policy update magnitudes based on question uncertainty. It allows more conservative updates on high-uncertainty questions while maintaining the original learning signal on confident ones. Experimental results on five mathematical reasoning benchmarks (AIME24 **56.7**, AMC **68.7**, MATH **83.4**, Minerva **34.2**, and OlympiadBench **48.0**) demonstrate that SEED-GRPO achieves new state-of-the-art performance in average accuracy, validating the effectiveness of uncertainty-aware policy optimization.

1 Introduction

Reinforcement learning (RL) emerges as a critical tool for fine-tuning large language models (LLMs) [1, 2, 3, 4, 5, 6, 7, 8, 6, 9, 10, 11] to improve reasoning and accuracy on complex tasks. Leading systems such as OpenAI's GPT-4o and o1 [12], Google's Gemini [13], Anthropic's Claude 3 Opus [14], and DeepSeek [15, 1, 2] all rely on RL techniques to enhance their capabilities beyond what is possible with supervised learning alone. These models demonstrate remarkable proficiency in domains requiring sophisticated reasoning, with RL serving as the key mechanism.

Recent advances like Group Relative Policy Optimization (GRPO) [1] leverage multiple sampled answers per input prompt to compute relative rewards and advantages within each group, leading to significant gains in reasoning performance. GRPO eliminates the need for a critic model by using the average reward of a group as a baseline, and achieves strong performance on complex reasoning tasks like math and code generation.

Despite recent progress, GRPO [1, 2] and its variants [3, 7, 16, 17, 18, 19] **assign equal importance to all training prompts** during optimization, ignoring the varying levels of confidence that LLMs demonstrate across different input prompts. However, this signal serves as a crucial probe for model uncertainty, reflecting how well the model understands a given prompt. Uncertainties in responses reveal meaningful information about the model's internal knowledge boundaries. Failing to leverage this uncertainty restricts the ability of policy optimization methods to adaptively focus learning on examples that lie within the model's current capabilities.

Preprint. Under review.

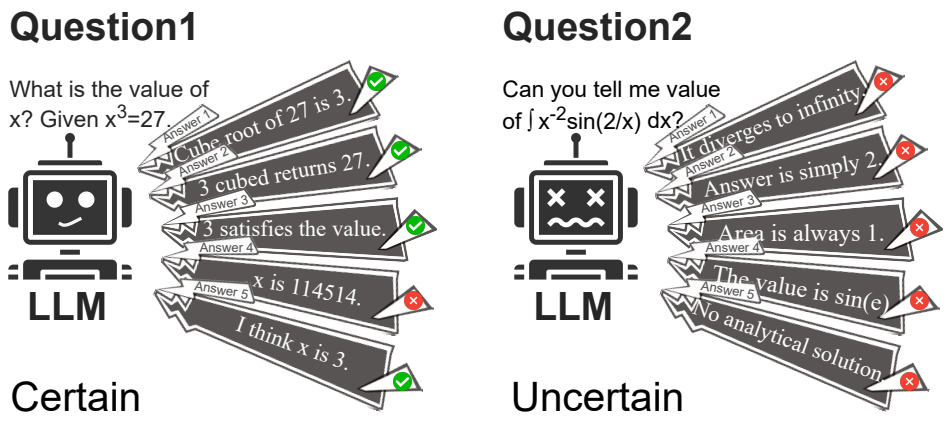

Figure 1: Intuitive explanation of semantic entropy. For Question 1, although the 6 responses have slight syntactic variations, 5 of them convey the same meaning, indicating low semantic entropy and high model certainty. For Question 2, the 6 responses can be clustered into 6 distinct meaning classes, resulting in high semantic entropy and indicating significant model uncertainty.

Intuitively, when an LLM model generates highly diverse answers to a given question, it indicates fundamental uncertainty about how to solve the problem. Prior work [6, 20, 21, 22, 23, 24] observes that when an LLM generates semantically diverse answers to the given prompt, it typically indicates the problem lies beyond the model's current reasoning capability. In other words, such uncertainty may reflect that the question lies outside the model's current reasoning comfort zone, and large policy updates in these cases may result in unstable learning rather than meaningful improvement. Conversely, when responses demonstrate semantic consistency (*i.e.*, low uncertainty), the model has a coherent understanding of the problem domain, making it safer to update.

This leads us to propose an uncertainty-aware approach to policy optimization: instead of applying uniform updates across all examples, we adaptively adjust the magnitude of updates based on the model's uncertainty for each training prompt. This framework effectively implements a dynamic learning rate mechanism that automatically calibrates according to the model's current capabilities and the problem difficulty. Similar to curriculum learning [25], this method provides an adaptive learning signal for each problem, reducing the update magnitude for questions that present significant challenges to the model's current reasoning abilities, while maintaining robust training signals for problems where the model demonstrates greater confidence.

In this paper, we introduce SEED-GRPO (**S**emantic **E**ntropy **E**nhance**D** GRPO), an uncertainty-aware policy optimization algorithm that explicitly quantifies **prompt-specific uncertainty** using semantic entropy [23, 20]. Semantic entropy is an entropy-based [26] metric that captures the diversity of meanings among responses to the same input prompt. Figure 1a illustrates this concept. For Question 1, the model produces five correct answers and one incorrect one. While the correct answers vary in syntax and presentation, they all share the same underlying meaning (*i.e.*, $x = 3$), forming a single semantic cluster. This leads to low semantic entropy and indicates low uncertainty. In contrast, for Question 2, the model generates six incorrect and semantically diverse responses, resulting in high entropy and thus high uncertainty. This suggests the model lacks a coherent reasoning path for that problem, possibly because it falls outside the model's current capabilities. This observed correlation between semantic entropy and problem difficulty provides a principled foundation for uncertainty-aware learning. By leveraging semantic entropy as a proxy for model uncertainty, SEED-GRPO dynamically calibrates policy updates based on how confidently the model responds to each question.

Our contributions can be summarized as follow: **i)** We propose using *semantic entropy* to quantify model uncertainty at the prompt level, and empirically observe that low entropy correlates with questions within the model's current capabilities (yielding correct predictions), while high entropy signals problems that likely exceed its abilities (resulting in inconsistent and incorrect outputs). **ii)** We introduce SEED-GRPO, an uncertainty-aware policy optimization algorithm that dynamically calibrates updates based on measured semantic entropy. **iii)** We demonstrate that SEED-GRPO achieves strong performance across five mathematical reasoning benchmarks (AIME24, AMC,

MATH, Minerva, and OlympiadBench), establishing new state-of-the-art results in average accuracy and supporting the effectiveness of semantic entropy-guided optimization.

2 Related Work

Reasoning LLMs. Recent efforts enhance the reasoning capabilities of Large Language Models (LLMs) through both innovative prompting techniques and sophisticated fine-tuning strategies. Chain-of-Thought [27] prompting encourages models to generate intermediate reasoning steps, resulting in performance improvements on complex mathematical and logical reasoning tasks. Subsequent research extends this approach through frameworks such as Tree-of-Thoughts [22] and self-consistency CoT [21], which strategically aggregate multiple reasoning paths to enhance reliability and accuracy. LIMO [28] demonstrates exceptional results by employing Supervised Fine-Tuning (SFT) to train specialized reasoning models. Meanwhile, alternative approaches to reinforcement learning beyond GRPO show promising outcomes, as evidenced by the effectiveness of Open-Reasoner-Zero [10], KIMI K1.5 [5], and ReST-MCTS* [29].

Group Relative Policy Optimization and Variants. DeepSeek first proposes Group Relative Policy Optimization (GRPO) [1, 2] to train reasoning LLMs. This RL algorithm is a variant of Proximal Policy Optimization (PPO) [30] that eliminates the need for value models, which are difficult to train and consume computational resources. GRPO achieves strong performance across numerous reasoning benchmarks spanning mathematics, coding, and question answering domains. Open-R1 [31] represents Hugging Face's fully open-source implementation of GRPO. SRPO [19] introduces history resampling, which preserves valuable problems in storage for reuse during later training stages. DAPO [16] proposes dynamic sampling that filters completely correct and completely incorrect samples to ensure effective training. Dr.GRPO [3] identifies length bias issues and presents an improved version to address this challenge. Concurrent work on EMPO [6] similarly incorporates semantic entropy, however, they directly incorporate semantic entropy as an optimization objective, whereas our work leverages semantic entropy to measure uncertainty and integrates this uncertainty into advantage calculations. To the best of our knowledge, we are the first to incorporate uncertainty-aware policy optimization into the GRPO framework.

3 SEED-GRPO: Uncertainty-Aware Policy Optimization

3.1 Motivation: Uncertainty-Aware Learning

The fundamental insight behind our approach is that when a model generates divergent responses to the same prompt across multiple attempts, such variation often reflects high uncertainty, suggesting that the task potentially exceeds the model's current capabilities (§1). SEED-GRPO leverages this insight through a principled mechanism: For questions where the model exhibits high semantic entropy (high uncertainty), we adaptively downscale the advantages during policy updates, resulting in more conservative learning steps. This prevents the model from overfitting to potentially noisy rewards on prompts it cannot yet reliably solve. For questions where the model demonstrates low semantic entropy (high certainty), we maintain the original advantages.

This design echoes the principle of curriculum learning [25], where learning progresses from easier to harder examples. However, rather than relying on static difficulty heuristics, SEED-GRPO employs semantic entropy as a dynamic, model-specific uncertainty signal to calibrate learning pressure.

3.2 SEED-GRPO Illustration via Math Reasoning Example

To illustrate the core mechanics of SEED-GRPO, consider a math problem q (prompt) such as:

> "What is the value of x? Given $x^3 = 27$."

Using an LLM $\pi_{\theta_{\text{old}}}$, we sample a group of G responses $\{ o_1, o_2, \ldots, o_G \} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)$, as shown in Fig. 2, responses are also referred as rollout samples. Each response $o_i \in \mathbb{R}^{l_i \times dim}$ is a token sequence of length l_i and token dimension *dim*. These sequences contain detailed step-by-step reasoning and conclude with a boxed final answer. While such sequences are sometimes referred to as "trajectories" in traditional reinforcement learning (*e.g.*, PPO [30]), we avoid this terminology for clarity.

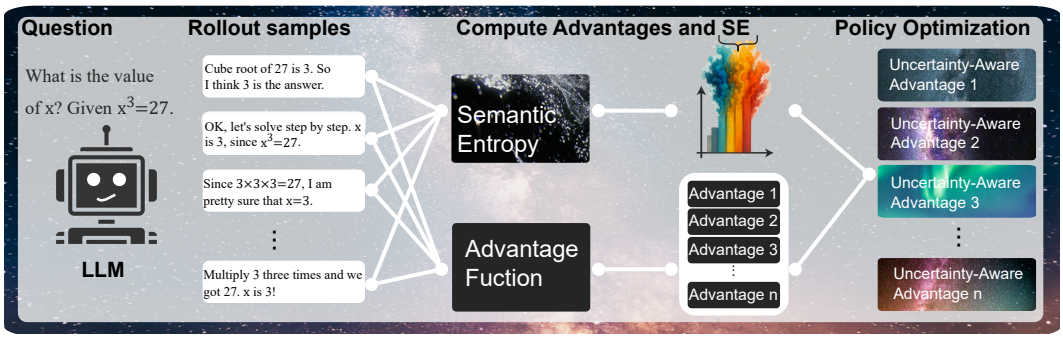

Figure 2: The SEED-GRPO framework incorporating semantic entropy for uncertainty-aware reinforcement learning. The framework samples multiple responses from a pre-trained LLM, computes semantic entropy to measure model uncertainty, and modulates the advantage function accordingly to enable more conservative updates for high-uncertainty questions.

Each o_i is an independently sampled text sequence. Some may contain correct solution paths, while others may contain logical or arithmetic errors. We extract the final answers and compute rewards: $r_i = 1$ if o_i is correct, and $r_i = 0$ otherwise. Note that, in SEED-GRPO there is no reward model, these rewards are obtained by comparing with ground truth labels using specific verification rules. The average group reward $\bar{r} = \frac{1}{G}\sum_{i=1}^{G} r_i$ serves as a baseline, and advantages can be calculated:

$$A_i = r_i - \bar{r}, \quad A_i \in \mathbb{R}. \tag{1}$$

In SEED-GRPO, the advantage A_i is broadcast across all tokens in the response o_i. For instance, if o_3 consists of 50 tokens and its advantage is 0.5, then each of those 50 tokens is associated with the same scalar advantage during training. Once the advantages are computed, policy updates are performed using the clipped surrogate objective inspired by PPO:

$$\mathcal{L}_i(\theta) = \min\Big(\text{ratio}_i(\theta)\, A_i,\ \text{clip}\big(\text{ratio}_i(\theta),\, 1-\epsilon,\, 1+\epsilon\big)\, A_i\Big), \qquad \epsilon = 0.2, \tag{2}$$

where $\text{ratio}_i(\theta) = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}$ is the importance sampling ratio between the current and old policies.

The overall training objective for question q is the mean over all G samples:

$$\mathcal{L}(\theta) = \frac{1}{G}\sum_{i=1}^{G} \mathcal{L}_i(\theta). \tag{3}$$

We then update the model parameters by maximizing this objective.

To incorporate uncertainty into the learning process, we measure the **semantic entropy** $\text{SE}(q)$ [23, 20] of the generated answer group (rollout samples Fig. 2). Semantic entropy quantifies the degree of semantic diversity across the generated responses. It captures whether the outputs consistently converge on a single reasoning path or instead diverge into multiple, potentially conflicting solutions.

Intuitively, semantic entropy measures how diverse and inconsistent the model's rollout samples are in terms of their meaning. As illustrated in Fig. 1, for a given mathematical problem, an LLM may generate multiple answers. For Question 1, although there are 6 responses with different syntactic structures and phrasing, 5 of them express essentially the same meaning $x = 3$, indicating that the model is highly certain about its answer. Conversely, for Question 2, the 6 responses fall into 6 distinct meaning classes, resulting in high semantic entropy, which suggests that the model struggles with this more challenging problem and lacks confidence in its outputs.

Before computing semantic entropy, we first group the G sampled responses $\{o_1, o_2, \ldots, o_G\}$ into a set of meaning clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$, where $C_k = \{o_i : \text{meaning}(o_i) = k\}$:

The semantic entropy is theoretically defined as:

$$\text{SE}(q) = -\sum_{c} \Big(\big(\sum_{o_i \in c} p(o_i \mid q)\big) \log \big[\sum_{o_i \in c} p(o_i \mid q)\big]\Big), \tag{4}$$

where c indexes the semantic equivalence classes, and $p(o_i \mid q)$ is the probability of generating response o_i given question q under the current policy model $\pi_{\theta_{\text{old}}}$.

However, in practice, an LLM can generate an unbounded number of diverse responses to a given question q, potentially spanning a vast—and unknown—set of meaning classes. It is infeasible to enumerate all possible semantic clusters.

Therefore, we approximate the semantic entropy using the Monte Carlo method proposed in [20, 23]. Before computing semantic entropy, we first group the G sampled responses $\{o_1, o_2, \ldots, o_G\}$ into a set of meaning clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$, where $C_k = \{o_i : \text{meaning}(o_i) = k\}$. The clustering method will be detailed in §4.1. Each cluster C_k represents a semantically coherent subset of responses that share the same meaning, despite possible differences in wording or reasoning steps.

Based on observed meaning clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$, derived from a finite number of G sampled responses, the semantic entropy of prompt q is estimated as:

$$\text{SE}(q) \approx -\frac{1}{K} \sum_{k=1}^{K} \log p(C_k \mid q), \tag{5}$$

where $p(C_k \mid q) = \sum_{o_i \in C_k} \pi_{\theta_{\text{old}}}(o_i \mid q)$ denotes the total probability mass assigned to the k-th observed cluster. This formulation aligns with the Rao–Blackwellized Monte Carlo estimate proposed in [20, 23], which approximates semantic entropy over sampled outputs and observed clusters.

Semantic entropy is non-negative and measures model's uncertainty on the given prompt. When all G responses convey the same meaning (K=1), the entropy reaches its minimum value of 0, indicating complete certainty. Conversely, when each response belongs to a distinct semantic cluster ($K = G$), the entropy reaches its maximum value, signaling extreme uncertainty where the model produces entirely different answers each time. Given a fixed number of responses G, the maximum possible semantic entropy can be calculated as: $\text{SE}_{\max} = \log G$. For instance, if $G = 8$ and all eight responses fall into different semantic clusters, the maximum semantic entropy would be approximately **2.07**.

This semantic entropy allows us to quantify the uncertainty of the model for each question. Higher entropy indicates greater semantic diversity in the model's responses, suggesting that the model is uncertain about the correct answer. Lower entropy indicates greater consensus among responses, suggesting higher confidence in the model's answers. We leverage this uncertainty measurement to modulate the advantage in the reinforcement learning objective. The key insight is that model updates should be more conservative for questions where the model exhibits high uncertainty. Our uncertainty-aware advantage modulation function is defined as:

$$\hat{A}_i = A_i \cdot f\big(\alpha \cdot \text{SE}(q)/\text{SE}_{\max}(q)\big), \tag{6}$$

where α is a hyperparameter controlling sensitivity. When semantic entropy is high, we interpret it as model uncertainty and scale down the advantage to produce more conservative updates. The function f can take various forms, such as linear, exponential, or focal styles, influencing how uncertainty affects the advantage scaling. We conduct ablation studies on f in detail (§4.3).

This modulation attenuates advantages for high-entropy questions, effectively reducing the magnitude of parameter updates for problems where the model is uncertain. Intuitively, this approach makes the training process more cautious about learning from feedback on questions where the model lacks confidence, mitigating the risk of overfitting to potentially noisy or misleading reward signals.

3.3 Discussion and Analysis

1) Why not use vanilla information entropy to measure uncertainty? Shannon entropy [26], while widely used, can yield misleading estimates of uncertainty in the context of language model outputs. Specifically, when a model generates several responses that differ in phrasing, syntax, or word choice but convey the same underlying meaning, vanilla entropy will still report a high value. This is because it operates on surface-level token distributions and is agnostic to semantic equivalence. Consequently, it overestimates uncertainty in cases where the model is, in fact, semantically consistent. In contrast, semantic entropy clusters responses based on meaning rather than form. This makes semantic entropy a more faithful and robust indicator of a model's uncertainty on the input prompt.

2) What benefits does uncertainty-aware advantage bring to policy optimization?

Incorporating uncertainty into the advantage computation allows SEED-GRPO to modulate the learning process adaptively. To better understand this mechanism, we present a simplified gradient analysis of the policy update. For clarity, we consider the loss function without clipping:

$$\mathcal{L}_i(\theta) = \text{ratio}_i(\theta) \cdot \hat{A}_i = \frac{\pi_\theta(o_i \mid q)}{\pi_{\theta_{\text{old}}}(o_i \mid q)} \cdot \hat{A}_i. \tag{7}$$

The gradient is computed as:

$$\nabla_\theta \mathcal{L}_i(\theta) = \nabla_\theta \log \pi_\theta(o_i \mid q) \cdot \text{ratio}_i(\theta) \cdot \hat{A}_i. \tag{8}$$

Accordingly, the policy update becomes (with the global learning rate η):

$$\theta \leftarrow \theta + \eta \cdot \nabla_\theta \log \pi_\theta(o_i \mid q) \cdot \text{ratio}_i(\theta) \cdot \underbrace{\left[A_i \cdot f\left(\alpha \cdot \text{SE}(q)/\text{SE}_{\max}(q)\right)\right]}_{\hat{A}_i}. \tag{9}$$

By integrating semantic uncertainty into $\hat{A}_i$, this formulation effectively scales the gradient for each input based on the model's uncertainty. This uncertainty-aware advantage computation effectively creates a question-specific adaptive learning rate.

As shown in Eq. 9, the policy update is governed by four components: the global learning rate η, the log-probability gradient, the importance sampling ratio, and the advantage term. By incorporating the uncertainty-dependent factor $f(\cdot)$, which is non-negative, SEED-GRPO effectively modulates the update magnitude in proportion to the model's uncertainty. This can be viewed as dynamically adjusting the effective learning rate on a per-question basis.

This mechanism creates an implicit curriculum learning effect: the model naturally takes larger learning steps on problems it can confidently solve, while proceeding more cautiously on challenging ones where the reward signal may be less reliable. This approach helps prevent overfitting to noise in difficult problems while allowing efficient learning from well-understood ones.

4 Experiments

4.1 Experimental Setup

Train Datasets. Our training dataset is MATH [32] Level 3-Level 5, following the same setting of Dr.GRPO [3].

Table 1: Dataset statistics.

Dataset	#Questions	Level
Train Datasets		
MATH (L3–L5)	8.5k	–
Test Datasets		
AIME24	30	Olympiad
AMC	83	Intermediate
MATH500	500	Advanced
Minerva	272	Graduate
OlympiadBench	675	Olympiad

Test Datasets. We evaluate our method on five mathematical reasoning benchmarks: **i)** AIME24 contains 30 high-school level olympiad problems from the American Invitational Mathematics Examination 2024; **ii)** AMC includes 83 problems from the AMC series, consisting mostly of multiple-choice questions of intermediate difficulty; **iii)** MATH500 is a randomly selected subset of 500 problems from the original MATH [32] dataset, covering algebra, geometry, and number theory; **iv)** Minerva (MIN) [33] comprises 272 questions introduced by the Minerva benchmark mostly requiring multi-step reasoning; **v)** OlympiadBench (OLY) [34] includes 675 high-difficulty math problems.

Model. Following previous works [2, 3], we use Qwen2.5-Math [35] 1.5B, 7B, and DeepSeek-R1-Distill-Qwen-7B [2] as our base models. We choose Dr.GRPO [3] as the default baseline algorithm.

Competitor. We compare against state-of-the-art methods including Dr.GRPO [3], DeepSeek-R1-Zero-Qwen [1], RAFT++ [8], GPG [17], DAPO [16], SimpleRL-Zoo [36], SRPO [19], Eurus [37], OpenReasoner-Zero [10], and QwQ-preview [38].

Evaluation Metrics. To maintain consistency with prior research [3, 36], we primarily employ the Pass@1 metric for comparative analysis [39]. The pass@k metric evaluates whether, among k responses to a given problem, at least one solution passes the test criteria. The Pass@1 scenario, where only a single response is generated, presents a more challenging setting. For the uncertainty function $f(\cdot)$, we default choose Linear function with $\alpha = 0.02$, more ablation studies are in §4.3.

Implementation Details. Our training configuration and hyperparameter settings follow Dr.GRPO [3]. Specifically, we limit the maximum output to 3,000 tokens, and when calculating advantages, we do not normalize by the group reward standard deviation. Similarly, during loss computation, we do not divide by generation length. For semantic entropy clustering, we employ a straightforward approach that only considers whether the final answers generated by the model are identical. Additional implementation details are provided in the supplementary materials.

Table 2: Pass@1 performance comparison across multiple mathematical reasoning benchmarks. Results marked with $^+$ are reported as the mean $\pm$ standard deviation across 3 runs under the same default experimental setting (§4.1). Our other results report the best performance.

Method	AIME24	AMC	MATH	MIN.	OLY.	Avg.
Baseline methods						
Qwen2.5-Math-base **1.5B**	16.7	43.4	61.8	15.1	28.4	33.1
Qwen2.5-Math-base **7B**	0.2	45.8	69.0	21.3	34.7	38.2
Dr.GRPO **1.5B**	20.0	53.0	74.2	25.7	37.6	42.1
Dr.GRPO **7B**	43.3	62.7	80.0	30.1	41.0	51.4
RAFT++ **7B**	-	-	80.5	35.8	41.2	-
OpenReasoner-Zero **7B**	13.3	47.0	79.2	31.6	44.0	43.0
Eurus **7B**	16.7	62.7	83.8	36.0	40.9	48.0
SimpleRL-Zoo **7B**	26.7	60.2	78.2	27.6	40.3	46.6
GPG **7B**	33.3	65.0	80.0	34.2	42.4	51.0
SRPO **32B**	44.3	-	-	-	-	-
DAPO **32B**	50.0(Avg@32)	-	-	-	-	-
DeepSeek-R1-Zero-Qwen **32B**	46.7	-	-	-	-	-
QwQ-preview **32B**	50.0	-	90.6	-	-	-
Our methods						
SEED-GRPO **1.5B** (Linear, α=0.02)	23.3	50.6	75.4	26.8	41.3	43.5
SEED-GRPO **7B** (Linear, α=0.02)$^+$	$43.3_{\pm 3.4}$	$64.67_{\pm 4.9}$	$82.2_{\pm 1.4}$	$35.03_{\pm 1.6}$	$45.2_{\pm 2.2}$	$54.73_{\pm 2.0}$
SEED-GRPO **7B** (Linear, α=0.02)	46.7	69.9	83.0	36.7	46.8	56.6
SEED-GRPO **7B** (Linear, α=0.02, G=16)	**56.7**	68.7	**83.4**	34.2	48.0	58.2
SEED-GRPO **7B** (Linear, α=0.02, R1-Distill)	50.0	**78.3**	**91.6**	**38.6**	**61.5**	**64.0**

Table 3: Training configuration and performance comparison of mathematical reasoning methods.

Method	#Train Data	#Prompt Batch Size	#Rollouts(G)	#Steps	AIME24	MATH
Baseline methods						
Dr.GRPO **7B**	8.5k	128	8	400	43.3	80.0
SimpleRL-Zoo **7B**	7.5k	1024	8	150	26.7	78.2
DAPO **32B**	17k	512	16	5.5k	50.0(Avg@32)	-
Our methods						
SEED-GRPO **7B**	8.5k	128	8	384	40.0	81.4
SEED-GRPO **7B**	8.5k	128	8	928	46.7	83.0
SEED-GRPO **7B**	8.5k	128	16	360	**56.7**	**83.4**

4.2 Quantitative Comparison Results

Table 2 presents a comprehensive evaluation of our SEED-GRPO approach against established mathematical reasoning methods across multiple benchmarks. Our method demonstrates consistent and substantial improvements over strong baseline systems. Under the Qwen-Math-base setting, SEED-GRPO 1.5B shows significant average improvements compared to the Qwen-Math-base1.5B model, achieving 43.5% average score across all benchmarks.

For our default configuration (§4.1), SEED-GRPO 7B (Linear, α=0.02) achieves an excellent average score of 56.6% across all benchmarks, representing a significant improvement of **5.2%** over the Dr.GRPO 7B baseline. Notably, SEED-GRPO 7B even surpasses SRPO 32B on the challenging AIME24 benchmark (46.7% vs. 44.3%), despite having only a fraction of the parameters. This configuration particularly excels on the AMC benchmark with a score of 69.9%, surpassing all other 7B parameter models with the same initial base architecture.

Our experiments further validate the effectiveness of increasing the number of rollouts G per query. As shown in Table 2, simply doubling G from 8 to 16 leads to a **+1.6**% gain on average score, and a dramatic **+10**% jump on AIME24 (from 46.7% to 56.7%). This enhanced configuration achieves an average score of 58.2% across all benchmarks, outperforming several 32B models including SRPO, DAPO, DeepSeek-R1-Zero-Qwen, and QwQ-preview. Importantly, these results come at a significantly lower computational cost compared to training large 32B models.

Notably, in the DeepSeek-R1-Distill-Qwen-7B setting, our SEED-GRPO (7B, R1-Distill) achieves the best overall performance, with an impressive average score of 64.0% on Pass@1. It outperforms all 7B and even 32B models across key benchmarks like AIME24, MATH, and OlympiadBench.

Table 3 compares performance across different training configurations. Compared to baseline methods, our SEED-GRPO achieves superior results with similar or even reduced training data size and computational steps. In particular, with $8.5k$ training data and a batch size of 128, by increasing the number of rollouts to 16, the AIME24 score improved to 56.7% and the MATH score reached 83.4%, surpassing all other 7B models.

(a) Method Comparison

Method	AIME	AMC	MATH	MIN	OLY	Avg.
Baseline **7B**	0.2	45.8	69.0	21.3	34.7	38.2
Dr.GRPO **7B**	43.3	62.7	80.0	30.1	41.0	51.4
SEED-GRPO	46.7	69.9	83.0	36.7	46.8	56.6

(b) SE Weight α

α	AIME	AMC	MATH	MIN	OLY	Avg.
0.01	46.7	60.2	80.6	33.5	42.7	52.7
0.02	46.7	**69.9**	**83.0**	**36.7**	**46.8**	**56.6**
0.03	**50.0**	61.4	83.0	34.2	44.4	54.6

(c) Weight Fuction $f(\cdot)$

Func.	AIME	AMC	MATH	MIN	OLY	Avg.
Focal	43.3	65.1	**84.4**	35.3	**47.6**	55.1
Exponential	43.3	66.3	82.0	35.7	44.3	54.3
Linear	46.7	**69.9**	83.0	**36.7**	46.8	**56.6**

(d) #Rollouts (G)

G	AIME	AMC	MATH	MIN	OLY	Avg.
8	46.7	**69.9**	83.0	36.7	46.8	56.6
10	50.0	61.4	**84.0**	**37.5**	**48.1**	56.2
16	**56.7**	68.7	83.4	34.2	48.0	**58.2**

(e) Base Models

Method	AIME	AMC	MATH	MIN	OLY	Avg.
Qwen2.5 ***1.5B***						
Base	16.7	43.4	61.8	15.1	28.4	33.1
Dr.GRPO **1.5B**	20.0	53.0	74.2	25.7	37.6	42.1
SEED-GRPO	23.3	50.6	75.4	26.8	41.3	43.5
Qwen2.5 ***7B***						
Base	0.2	45.8	69.0	21.3	34.7	38.2
Dr.GRPO	43.3	62.7	80.0	30.1	41.0	51.4
SEED-GRPO	**56.7**	68.7	83.4	34.2	48.0	58.2
R1-Distill ***7B***						
Base	10.0	26.2	80.0	30.1	41.0	51.4
SEED-GRPO	50.0	**78.3**	**91.6**	**38.6**	**61.5**	**64.0**

Table 4: SEED-GRPO ablations across five math reasoning benchmarks.

It is worth highlighting that our SEED-GRPO 7B (Linear, α=0.02) achieves superior performance to several 32B models AIME24, demonstrating the effectiveness of our approach. While DAPO reports a higher Avg@32 score of 50.0%, our method focuses on the more challenging Pass@1 metric.

4.3 Ablation Study

Method Comparison. Table 4(a) compares SEED-GRPO with the initial base model Qwen2.5-Math-base 7B and the baseline Dr.GRPO 7B. It's important to note that both SEED-GRPO and Dr.GRPO start from the same Qwen2.5-Math-base 7B, using identical hyperparameters. Particularly, SEED-GRPO achieves a remarkable 13.4% improvement over the baseline on AIME (from 43.3% to 46.7%). On average, SEED-GRPO outperforms Dr.GRPO by 5.2% confirming the effectiveness of uncertainty-aware policy optimization.

Semantic Entropy Weight. We investigate the impact of the semantic entropy weight parameter α in Table 4(b), which controls how much influence uncertainty has on the training process. Our results indicate that a medium weight value of $\alpha = 0.02$ yields the best overall performance with an average accuracy of 56.6%. Interestingly, a higher weight ($\alpha = 0.03$) improves performance on the challenging AIME benchmark but slightly decreases performance on other tasks. This suggests that more difficult tasks may benefit from stronger uncertainty weighting, while simpler tasks require less emphasis on uncertainty. Setting α too low (0.01) consistently underperforms, confirming that some degree of uncertainty modeling is beneficial across all benchmarks.

Weight Function. In Table 4(c), we evaluate different functional forms for incorporating semantic entropy into our training objective. We compare linear, exponential, and focal weighting functions. The linear weighting function achieves the best overall performance with an average accuracy of 56.6%, outperforming both alternatives. While the focal function excels on particular benchmarks like MATH (84.4%) and OLY (47.6%), it performs less consistently across all tasks. The exponential function shows competitive but generally lower performance, suggesting that more aggressive uncertainty penalization may not be optimal. These results indicate that a simple linear relationship between semantic entropy and policy updates provides the most robust learning signal.

Number of Rollouts. Table 4(d) examines how the number of sampled solutions per query (G) affects model performance. Increasing G from 8 to 16 improves the average accuracy from 56.6% to 58.2%, with particularly gains on the challenging AIME benchmark (from 46.7% to 56.7%). This improvement demonstrates that a larger sample size enables more accurate estimation of semantic entropy. However, the performance with $G = 10$ shows mixed results, performing best on some benchmarks (MATH, MIN, OLY) but worse on others (AMC), suggesting task-specific optimal

sampling strategies. Overall, our findings support using larger rollout numbers when computational resources permit, with diminishing returns likely beyond $G = 16$.

Base Models. Table 4(e) shows SEED-GRPO's effectiveness across different base models. When applied to Qwen2.5-1.5B, SEED-GRPO improves average performance by 10.4 percentage points (from 33.1% to 43.5%). The improvement is even more substantial for Qwen2.5-7B, with a 20.0 percentage point gain (from 38.2% to 58.2%). This demonstrates that SEED-GRPO's benefits scale with model size, suggesting that larger models can better leverage uncertainty information during training. We also evaluated SEED-GRPO on the R1-Distill 7B model, achieving strong performance on AMC (71.2%) and AIME (46.7%), though complete results were not available for all benchmarks. Figure 4(f) illustrates the performance trend during training, showing that SEED-GRPO consistently outperforms baseline methods throughout the training process, with the performance gap widening as training progresses.

5 Limitation and Future Work

Limitation. Our current implementation of SEED-GRPO focuses solely on utilizing the final answers for semantic clustering in mathematical reasoning tasks, without considering the intermediate reasoning steps. This design choice offers simplicity and proves effective for problems with unique, well-defined answers. However, this approach has several limitations. First, for open-ended problems without unique answers, our current semantic entropy calculation may not adequately capture the diversity of valid reasoning paths. Second, while focusing on final answers works well for mathematical domains with clear correctness criteria, it may be insufficient for domains requiring nuanced evaluation of the reasoning process itself.

Future Work. SEED-GRPO focuses on mathematical reasoning tasks, where the final answer can be explicitly verified. A promising direction for future work is to extend SEED-GRPO to other domains such as multimodal tasks (image-text VQA, images or videos understanding), code generation, and open-ended textual question answering. These domains often require more nuanced semantic understanding and may benefit even more from uncertainty-aware policy optimization.

Another promising avenue is to refine semantic entropy estimation by incorporating intermediate reasoning steps, rather than relying solely on final answers. This would enable more fine-grained uncertainty modeling along the reasoning trajectory, potentially leading to better training dynamics. Future work could incorporate external models such as commercial LLMs (GPT-4o and o1 [12], Gemini [13], Claude 3 [14]) or open-source models (RoBERTa [40], SentenceTransformer [41, 42]) as additional meaning clustering models.

Lastly, while SEED-GRPO currently applies semantic entropy during training, it would be valuable to explore test-time compute strategies based on entropy signals. For example, semantic entropy could be used to dynamically adjust generation strategies, such as increasing rollout count or triggering fallback mechanisms when the model is uncertain. Such extensions would allow uncertainty-aware reasoning not only during learning but also at inference time.

6 Conclusion

We introduce SEED-GRPO, an uncertainty-aware reinforcement learning framework that enhances Group Relative Policy Optimization (GRPO) by integrating semantic entropy into the training objective. By incorporating a measure of semantic uncertainty into the training objective, our method ensures that policy updates are adaptively scaled according to the model's confidence level for each prompt. This leads to more conservative updates on challenging prompts where the model exhibits high uncertainty, while maintaining effective learning on problems the model can confidently solve. Extensive experiments across five challenging mathematical reasoning benchmarks (AIME24 **56.7**, AMC **68.7**, MATH **83.4**, Minerva **34.2**, and OlympiadBench **48.0**) show that SEED-GRPO consistently outperforms strong baselines, including Dr.GRPO [3] and 32B-scale competitors [16, 10] achieving new state-of-the-art performance using only a 7B model. Our extensive ablation studies confirm the effectiveness of uncertainty-aware policy optimization.

References

[1] 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.

[2] 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.

[3] 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.

[4] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[5] 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.

[6] Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025.

[7] Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342*, 2025.

[8] Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

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

[10] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

[11] Yixuan Even Xu, Yash Savani, Fei Fang, and Zico Kolter. Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. *arXiv preprint arXiv:2504.13818*, 2025.

[12] OpenAI. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[13] Gemini Team, Rohan Anil, Sebastian Borgeaud, et al. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[14] Anthropic. The claude 3 model family: Opus, sonnet, haiku. 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

[15] DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

[16] 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.

[17] Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*, 2025.

[18] Jixiao Zhang and Chunsheng Zuo. Grpo-lead: A difficulty-aware reinforcement learning approach for concise mathematical reasoning in language models. *arXiv preprint arXiv:2504.09696*, 2025.

[19] Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, et al. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *arXiv preprint arXiv:2504.14286*, 2025.

[20] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

[21] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

[22] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*, 2023. URL https://openreview.net/forum?id=5Xc1ecxO1h.

[23] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *ICLR*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

[24] Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*, 2024.

[25] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009.

[26] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[27] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35:24824–24837, 2022.

[28] 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.

[29] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. ReST-MCTS*: LLM self-training via process reward guided tree search. In *NeurIPS*, 2024. URL https://openreview.net/forum?id=8rcFOqEud5.

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

[31] Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

[32] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

[33] Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *NeurIPS*, 2022. URL https://openreview.net/forum?id=IFXTZERXdM7.

[34] Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, Yikai Zhang, Yuqing Yang, Ting Wu, Binjie Wang, Shichao Sun, Yang Xiao, Yiyuan Li, Fan Zhou, Steffi Chern, Yiwei Qin, Yan Ma, Jiadi Su, Yixiu Liu, Yuxiang Zheng, Shaoting Zhang, Dahua Lin, Yu Qiao, and Pengfei Liu. Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent AI. In *NeurIPS*, 2024. URL https://openreview.net/forum?id=ayF8bEKYQy.

[35] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[36] 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.

[37] Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, et al. Advancing llm reasoning generalists with preference trees. *arXiv preprint arXiv:2404.02078*, 2024.

[38] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[39] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[40] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[41] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 11 2019. URL https://arxiv.org/abs/1908.10084.

[42] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2020. URL https://arxiv.org/abs/2004.09813.