Enhancing Mathematical Reasoning Ability of Nuclear Domain-Specific Large Language Model using Reinforcement Learning

Byeongjae Kim[a], Solji Park[a], Yonggyun Yu[b,c*]

MOTIVATION

1. Problem: General-purpose LLMs lack the specialized mathematical reasoning and domain knowledge crucial for safety-critical nuclear applications (e.g., reactor simulation, emergency decision support).
2. Objective: To enhance the mathematical reasoning of 'AtomicGPT' (a nuclear-specific LLM) to bridge the gap and meet domain-specific demands.
3. Method: Apply Reinforcement Learning (RL) to AtomicGPT to improve its mathematical reasoning capabilities.
4. Expected Outcome: More reliable and autonomous virtual reactor operations, with enhanced decision support in complex or emergency situations.

METHODS & RESULTS

1. Key Technical Methodologies

Enhancing AI Math Reasoning with Advanced Techniques:

- **Reinforcement Learning (RL):** The AI learns by trial and error, guided by rewards for good performance.
- **GRPO (Group Relative Policy Optimization):** An efficient RL method where the AI generates multiple responses and learns by comparing them within the group to identify better solutions.
- **LoRA (Low-Rank Adaptation):** A resource-saving technique used with GRPO to effectively fine-tune large AI models.

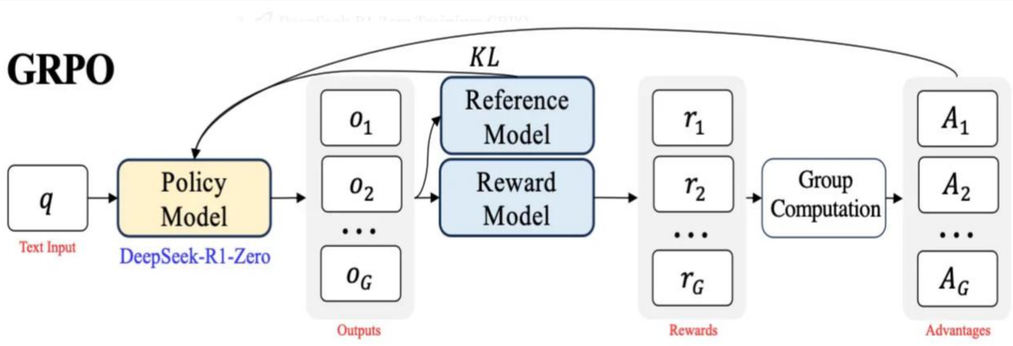


Fig 1. GRPO Reinforcement Learning Process. (Source: Medium, https://pub.towardsai.net/grpo-and-deepseek-r1-zero-9e81f15c6ba2)

2. Model and GRPO configuration

1) Training Model configuration and training dataset

- **Base Model: gemma2-Korean-AtomicGPT-9B**
- **Training Dataset:** openai/gsm8k
- **Training Process:**
  - Stage 1: Reinforcement Learning (GRPO) with LoRA **(on a single A100 GPU)**
- **Technique:** Applied **LoRA and higher *max_sequence_length* (to support long infernce)

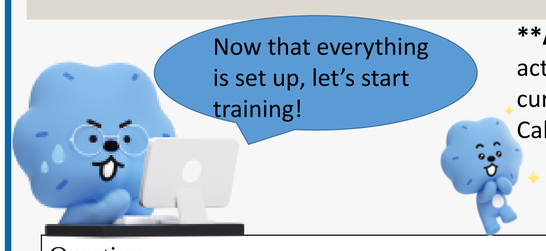
2) Design Reward functions

Need for Reward Functions :

- Evaluate model behavior and set the learning direction.
- Optimize the model's internal parameters by first calculating an ****advantage** from this evaluation (reward), and then using this advantage to guide the optimization..
- Optimized parameters define the model's specific role and behavior.

Types of designed reward functions :

1. Correctness_reward _func : Evaluates the accuracy of the final answer.
2. Int_reward_func : Checks if the answer is a clean integer.
3. Soft _format_func : Rewards perfect adherence to required
4. Strict_format_func : Rewards general adherence to format-thing
5. Count xml_func : Assesses appropriate use and quantity of XML tags.



****Advantage (A) :** A value indicating how much better a specific action (or a specific output oi) is than the average, given the current state (or question q).
Calculated based on **rewards**; the calculation formula is as follows:

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \ldots, r_G\})}{\text{std}(\{r_1, r_2, \ldots, r_G\})}$$

Question:
Janelle had 26 green marbles. Then she bought 6 bags of blue marbles. There were 10 marbles in each bag. She created a gift of 6 green marbles and 8 blue marbles and gave it to a friend. How many marbles does Janelle have now?
Expected Answer:
72
Full Response:
To find the total number of blue marbles Janelle bought, multiply the number of blue marbles per bag by the number of bags:
10 marbles/bag * 6 bags = 60 blue marbles.

The total number of marbles Janelle had initially was 26 green marbles + 60 blue marbles = 86 marbles.

After Janelle gave her friend a gift of 6 green marbles and 8 blue marbles, she had 86 - 6 - 8 = 72 marbles left.
Extracted Answer:
After Janelle gave her friend a gift of 6 green marbles and 8 blue marbles, she had 86 - 6 - 8 = 72 marbles left
Expected Number: 72.0, Extracted Number: 72.0
✓ CORRECT! Reward: 1.5

Table Ⅰ. Training logs of GRPO-based reinforcement learning

3. The process of creating a model and evaluation

Optimal Model Selection Method:

- **Monitoring:** Tracked the model's average reward throughout training.
  - *Average Reward = Total Accumulated Reward / Total Training Steps*
- **Result:** Achieved a peak average reward of 2.6605 at training step 750.
- **Selection:** The model from this step was ultimately adopted as the most effective policy.

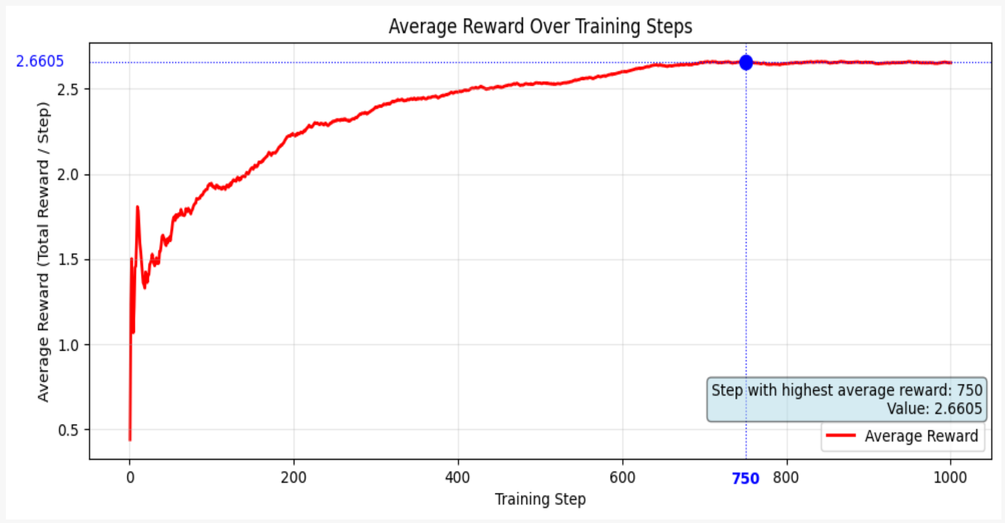


Fig 2. illustrates the trend of average reward across training steps, with the maximum value observed at step 750.

4. Benchmark results

- **Our Model:** gemma2-Korean-AtomGPT-9B (with GRPO RL)
- **Evaluation Dataset:** Math-500(100 problems), aqua-rat
- **Evaluation Metrics:**
  - Math-500 : Evaluated using Gemini 1.5 pro (automatic judging)
  - Aqua-rat : Exact-match accuracy (model answer vs gold answer)
- **Comparison:** gemma2-Korean-AtomicGPT-9B

	AtomicGPT-GRPO RL	AtomicGPT
Math-500 (pass@1)	47	41
Aqua-rat (pass@1)	53.96	51.5

Table Ⅱ : AtomicGPT-GRPO RL, AtomicGPT benchmarks performed with Math-500 and Aqua-rat as evaluation datasets utilizing the pass@1 metric.

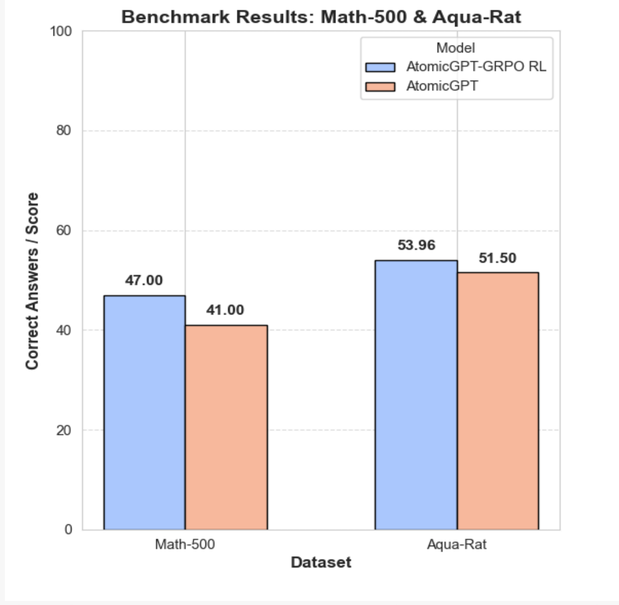


Fig. 3. Benchmarks of AtomicGPT- GRPO RL and AtomicGPT with Math-500 (100 questions with Gemini-1.5 pro to evaluate the correct answers), Aqua-rat dataset

CONCLUSIONS

Our research confirms that GRPO enhances the mathematical reasoning of LLMs while requiring fewer resources. This improved reasoning capability offers promising potential for advanced applications, such as controlling virtual nuclear reactor simulators. With its deep expertise in the nuclear domain, AtomicGPT is expected to manage complex operational scenarios more effectively and perform precise control actions within simulations—potentially outperforming general-purpose LLMs.

[a]KOREATECH [b]Korea Atomic Energy Research Institute [c]University of Science & Technology

This work was supported in part by Korea Atomic Energy Research Institute R&D Program under Grant KAERI-524540-24.