SEDIR: SELF-DISTILLED REASONER FOR SELF-IMPROVING LLMS

Alkın Ünlü
alkinunl@gmail.com

ABSTRACT

In recent months, OpenAI o1 has shown promising progress in solving complex reasoning tasks by synthesizing long chain-of-thoughts (CoT) before giving a final answer. This approach has demonstrated the potential to enhance performance on reasoning and coding tasks by increasing test-time compute. Existing open-source approaches remain limited by the need for human labeling, distilled datasets, or grounded verifiers, however a open-ended self-improving framework has yet to be fully explored with open-ended reasoning tasks.

This paper introduces ***SeDiR***, a novel framework for enabling fully open-ended self-improvement in reasoning LLMs. By leveraging the diversity of data at both pretraining and post-training stages, ***SeDiR*** iteratively generates and scores high-quality reasoning traces without requiring human intervention or seed data. This is a report of replicating o1 like reasoning capabilities with open-ended self-improving systems.

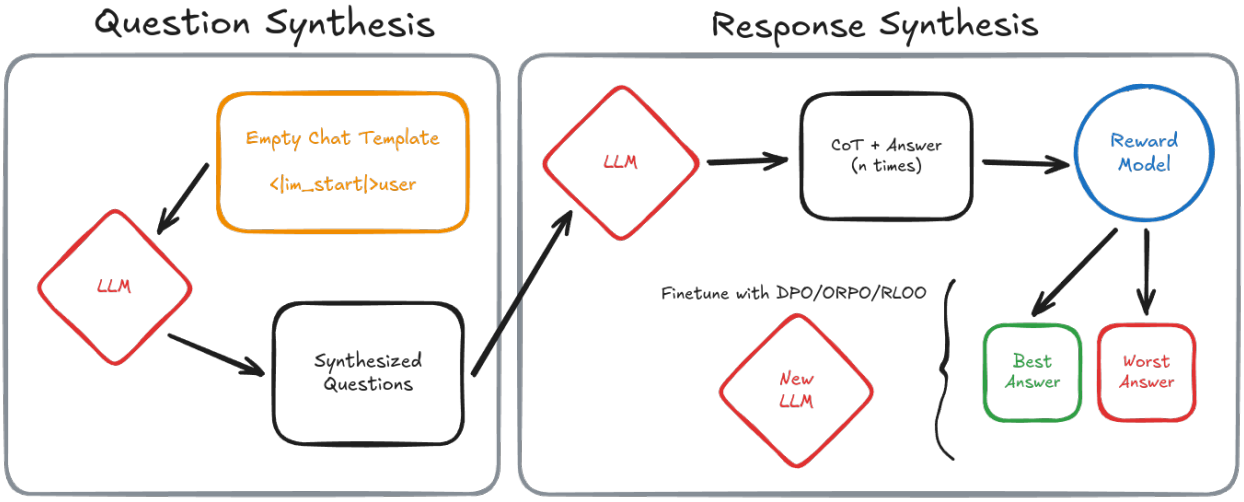

Figure 1: A simplified view of ***SeDiR***. While other approaches require grounded or supervised data, our approach doesnt require any human labeling or seed data to improve its reasoning abilities.

1 Introduction

Recently, OpenAI announced o1, a large language model (LLM) that synthesizes long internal thoughts before giving a final answer. This method is proven to be really effective in reasoning, coding and other tasks that require deep reasoning abilities. By bootstrapping already knowladgable LLMs with this ability, we can build systems that are closer to system-2 thinking while still doing system-1 thinking ultimately.

OpenAI and other teams achieved superb performance in tasks like math and coding by simply distilling the o1 model. We posit that a distilled model with enough data can outperform the teacher model in tasks that require deep reasoning

and not much knowladge. As o1-mini can outperform o1-preview in math, coding, and logic tasks, but lacks behind the o1-preview in more knowladge intensive tasks. This suggests that simple distillation rules arent feasible in tasks that require deep reasoning.

For example, in basic distilling principles, the student model cant outperform the teacher model in any task, however this is not the case for these models that can mimic reasoning. This finding is a hint that the limiting factor for these new systems is the scale of data, not scale of the model parameters. As a result, our hypothesis is that self-feeding open-ended systems can achieve superior performance to any other model as the system isnt limited to any external data.

We argue that an open-ended reinforcement-learning (RL) based method is a promising approach for self-improving reasoning LLMs. However, RL has some limitations, such as the need for grounded verifiers, which are not available with open-ended tasks. In this paper, we propose a new framework that can self-imprıove reasoning LLMs by iteratively generating and scoring high-quality samples to then finetune itself. We find that the reward model will not be a bottleneck, as the CoT will also inputted with final answer, allowing the reward model to interpret the steps proposed by the model. It is also great to highlight that distillation can only outperform the teacher model in tasks that require deep thinking.

2 Related Work

The field of self-improving language models has seen significant advancements in recent years. A number of frameworks have been proposed to enhance the capabilities of large language models (LLMs) through self-generated data, self-play, self-reward and so on. However, these frameworks are limited in their ability to be both: fully open-ended and steerable.

Self-Rewarding Language Models Recent work has shown that LLMs can be used to self-critique and generate high-quality, diverse finetuning data for themselves. While these approaches have shown promising results, they lack the ability to be steerable, grounded and performant.

Self-Play Recent approaches also have been proposed to improve the capabilities of LLMs with the help of a self-play mechanism to strictly adhere to the inputted dataset. While these approaches have shown promising results, they are limited to the human-generated data that is inputted to them.

SeDiR falls under the category of fully open-ended but steerable systems that can recursively improve reasoning, refine and synthesize their own training data.

3 Methodology

3.1 System Overview

The ***SeDiR*** framework consists of two main components: **Question Synthesis** and **Response Synthesis**. The **Question Synthesis** stage generates diverse and steerable questions, while the **Response Synthesis** stage leverages a reward model to select high-quality examples, enabling iterative refinement in reasoning. Figure 1 provides a schematic overview of the overall system.

3.2 Question Synthesis

The thia stage starts with an empty chat template as the input for the LLM (e.g., `<|im_start|>user`). With this template, the LLM synthesizes a random user instruction. We find that capable model can generate pretty promising questions at temperature of 1, however smaller and less capable models may require lower temperatures to generate coherent questions.

- **Prompt Initialization**: An empty template is used to prompt the LLM to generate questions.
- **Question Generation**: The model generates questions with temperature of 1.
- **Question Steerability**: We can set the system prompt to steer the generated questions (e.g., `The user has a question about reasoning, try your best to answer it.`).

To generate question with the LLM, we use the following template:

```
<|im_start|>system
You are a helpful assistant.
Try your best to answer the users reasoning question.<|im_end|>
<|im_start|>user
```

With these prompts we can generate quality, diverse and steerable questions in an open-ended fashion.

3.3 Response Synthesis

In this stage, we sample n number of answers with reasoning traces for each synthesized question. After sampling, we use our reward model to pick the highest-scoring and the lowest-scoring answer to generate a preference pair.

- **Answer Generation**: For each synthesized question, the model generates multiple candidate CoT and answers.
- **Reward-Based Selection**: The reward model scores each candidate CoT and answer, selecting the highest and lowest scoring answers. We find that the reward model is cannot be a bottleneck, as CoT will also be inputted with final answer, enabling our reward model to interpret the steps proposed by the model. This helps the reward model to follow the CoT and generate high-quality rewards.

4 Experimental Setup

4.1 Setup

With the base model being `Qwen2.5-1.5B-Instruct`, we sample 600 data points from the latest iteration of our model. We finetune the latest iteration of our model with Unsloth SFT.

4.2 Evaluation Metrics

The GPQA score evaluates the model's question-answering performance by measuring factors such as accuracy, relevance, and diversity of generated answers.

4.3 Baseline

We initialized ***SeDiR*** with `Qwen2.5-1.5B-Instruct` as the baseline model. This model serves as the starting point for iterative self-improvement, with initial GPQA scores of 16.41.

5 Results

5.1 Iterative Improvement on GPQA

Table 1 shows the GPQA scores across five iterations, demonstrating consistent improvement with each iteration.

Iteration	GPQA Score
0	16.41
1	28.03
2	30.05
3	32.32
4	34.09

Table 1: GPQA scores across iterations, with 0 being the baseline.

6 Conclusion

We presented ***SeDiR***, a novel framework for the open-ended self-improvement of LLMs through open-ended question-answer synthesis. Over four iterations, ***SeDiR*** demonstrated a 17.68-point improvement on the GPQA benchmark with no signs of plateau, highlighting the future of self-synthesized data in driving model enhancement.

References

[1] Author A., Title of Paper A. Journal Name, Year.
[2] Author B., Title of Paper B. Journal Name, Year.
[3] Author C., Title of Paper C. Journal Name, Year.
[4] Author D., Title of Paper D. Journal Name, Year.