# Scientific Reasoning with LLMs

Stuttgart Team: Florian Dreyer (3667877) Ekaterina Kolos (3688474)
Daria Matiash (3668740)

November 13, 2024

## 1 Introduction

Scientific machine reasoning is the application of AI techniques to simulate scientific reasoning processes, such as data interpretation, hypothesis generation, and causal inference. It facilitates scientific inquiry and discovery by enabling machines to independently analyze scientific data, generate hypotheses, and make predictions. For our project, we intend to compare different approaches around generative large language models to generate solutions for school science problems. On an existing benchmark with human solutions, we are planning to compare: a pure LLM with zero-shot and few-shot prompting (and possibly some more fine-grained prompting techniques), an agentic approach with ability to query external resources like textbooks in physics, maths etc, and soft-prompting techniques. We then plan to attempt knowledge distillation to a smaller LLM.

## 2 Background and Related Work

**Prompting techniques** A prompt is a command a user gives to a generative LLM used to guide its output for a downstream task, which contains several of the following: a role (persona) the LLM has to follow (e.g. "You are a helpful assistant"), a precise task description, examples (exemplars) for analogous learning (few-shot prompting), requirements on the process of reasoning, style instructions, output format requirements, additional information for in-context learning, emotion prompting components (highlighting why the task or a particular requirement is important). Prompts can be combined into sequences or graphs (with complex branching and parallelism). Of particular interest are small requirements on the reasoning process that can significantly improve performance when added to the prompt, such as "Let's think step by step" [**kojima2022large**]. Chain-of-thought prompts improve reasoning capabilities by asking the model to speak its process of thinking out loud, with variations including self-ask, step-back prompting, thread-of-thought, least-to-most prompting (decomposing then solving), plan-and-solve prompting, tree-of-thought, recursion-of-thought and many more [**schulhoff2024prompt**]. Soft prompting is an alternative to fine-tuning where the model is frozen while only a small number of "soft" prompt parameters are trained [**lester2021prompt**]. These parameters are used to guide the model in the right direction.

**Agents** An important step forward in using LLMs are agent-based architectures [**lin2024swiftsage**] [**ghafarollahi2024sciagents**]. They allow the LLM to act as an intelligent being capable of planning a solution of a complex task, while resorting to external resources on the way to acquire in-context the knowledge it does not have from pre-training. The retrieved information, results of invoking tools, such as calculators, and of interactions with the environment are appended to the prompt (or, in a more advanced scenario, to the summarized memory [**ge2023context**]). Optionally, a reflexion step (explicit reasoning on all the accumulated information) is added before allowing further generation of the final response.

**Knowledge distillation (KD)** allows to obtain smaller models capable of successfully following the behavior of larger teacher models. This is particularly useful for privacy reasons (mobile AI apps) and when access to large models in the cloud is not guaranteed (e.g. in cars). The teacher model can be used to intelligently select examples on which the student model is trained (dataset distillation) [**yu2023dataset**], or provide negative samples to show the student what incorrect answers or reasoning paths it should avoid to improve task accuracy [**li2024turning**] (c.f. contrastive CoT). Training small models on a CoT reasoning path of a larger model was also shown to be a way to obtain a small student model replicating reasoning capabilities of teacher on downstream tasks [**magister2022teaching**], which is close to *response-based KD* where the student model mimics the output of the teacher. Alternatively, with *feature-based KD* the knowledge from specific layers of the teacher model is distilled into the student model [**sepahvand2022teacher**],

while the student model's structure may be a quantized, simplified, or condensed version of the teacher's architecture [**gou2021knowledge**].

In our project, we plan to develop an extended version of CoT knowledge distillation, where both the reasoning paths and the essence of the external knowledge the teacher agent has retrieved for the tasks are acting as training data for the student model. This way the teacher agent "defines" the "protocols" of solving complex tasks, while the student model tries to "follow" the learnt "instructions" and memorize important information for similar tasks. If we succeed, such small models could be a valuable asset in practice in enterprise environments, to encode important protocols and parts of project documentation, replacing expensive intelligent agents for specific tasks.

# 3 Methodology

We postulate the following research questions: RQ1: Can we build a LLM agent to improve the LLMs performance on science problems? RQ2: Does Soft Prompting improve the performance of the LLM (agent)? RQ3: Can an LLM agent's behavior be distilled into a single model?

**RQ1** We plan to start by prompting a multimodal model with reasoning capabilities with a zero-shot and few-shot prompting settings. Preliminary list of foundation models to be compared as baselines is: CLIP, GPT4o (GPT4-V), T5, BigGAN, Gemini; small models: TinyCLIP, T5-Small; Llama models. This simple generation will be compared with an agent-based approach using the same models, which will now include augmented retrieval of information on the scientific task from domain-specific texts.

**RQ2** We plan to use Soft Prompting as one of the parameter-efficient fine-tuning techniques on the LLM used to guide it towards better reasoning. To achieve this we will add learnable prompt parameters to the base LLM we use. While training these parameters the rest of the model will be frozen [**lester2021prompt**].

**RQ3** In order to distill the knowledge from the system obtained at previous steps, we plan to do the model distillation with Chain-of-Thought Prompting for Reasoning approach. Following techniques described in [**magister2022teaching**] [**wei2022chain**], the student model will be fine-tuned using these CoT responses together with any additional information the agent has retrieved from external sources to produce intermediate reasoning steps. At the same time, the teacher model generates multiple CoT responses, and the student learns from the aggregate (self-consistent) reasoning paths. The foundation model to serve as student LLM should be small, multimodal, capable of reasoning and capable of acquiring new distilled knowledge. A preliminary list of candidate models: small Llama and LLaVA models, MiniGPT-4, small FLAVA models, distilUNITER, lxmert, distilled ViLT.

# 4 Dataset

This study is based on the ScienceQA dataset [**lu2022learn**]. The dataset includes a variety of science-related multimodal multiple-choice questions together with annotations of the answers that provide relevant lectures and explanations in Nature Science, Language Science and Social Science.

Data points are structured as follows:

```
{'image': <PIL.PngImagePlugin.PngImageFile image mode=RGB size=200x202>,
 'question': 'Is diorite a mineral or a rock?',
 'choices': ['rock', 'mineral'],
 'answer': 0,
 'hint': 'Diorite has the following properties:\nno fixed crystal structure\nnaturally occurring\nnot
 'task': 'closed choice',
 'grade': 'grade8',
 'subject': 'natural science',
 'topic': 'earth-science',
 'category': 'Rocks and minerals',
 'skill': 'Identify rocks and minerals',
 'lecture': 'Minerals are the building blocks of rocks. A rock can be made of one or more minerals.\nM
 'solution': 'The properties of diorite match the properties of a rock. So, diorite is a rock.'}
```

For some datapoints, the image data is missing, while the answer can be deduced from text alone (e.g. questions on 'subject': 'language science', 'topic': 'figurative-language', 'skill': 'Interpret figures of speech' (e.g. irony recognition).

For other datapoints, however, the image data can be missing, although required to make an answer, e.g.

```
{'image': None,
 'question': 'What information supports the conclusion that Rick inherited this trait?',
 'choices': ["Rick's coworker also has curly hair.",
  "Rick's biological father has curly hair.",
  'Rick and his biological parents have brown hair.'],
 'answer': 1,
 'hint': 'Read the description of a trait.\nRick has curly hair.',
 'task': 'closed choice',
 'grade': 'grade7',
 'subject': 'natural science',
 'topic': 'biology',
 'category': 'Genes to traits',
 'skill': 'Inherited and acquired traits: use evidence to support a statement',
 'lecture': "Organisms, including people, have both inherited and acquired traits. Inherited and acqui
 'solution': ''}
```

Apart from the image field, missing values can occur in lecture and solution fields. The statistics in table 1 give an overview on the amount of problematic datapoints.

| missing | frequency |
|---|---|
| solution | 9.3% |
| lecture | 15.7% |
| solution and/or lecture | 24.2% |
| image | 50.6% |

Table 1: Defect datapoints in dataset

**Retained data** As 50.6% of the datapoints are missing an image, we decide to process both text-only and text+image datapoints similarly with multimodal models, attaching the image to the prompt if it is available. For now, we will only work with datapoints with non-empty solution AND with non-empty lecture. All the other datapoints are disregarded, which results in **3216** datapoints kept in validation data.

**Metrics** We'll evaluate model's performance with question answering accuracy domain-wise in order to have a fair comparison with leaderboards. The reasoning steps can be evaluated with semantic similarity metrics (e.g. adopted from machine translation) such as BLEU, METEOR, ROUGE, BLANC, and BERTScore and more.

# 5 Experiments

## 5.1 Zero-shot and Few-shot Benchmarking

We start by zero-shot benchmarking existing multimodal models in three simple prompt settings, partly repeating with the official evaluation suite:

1. **question - choices - hint - image - task**, with the following prompt:

   ```
   {f'Question: {question}\n Task: {task}\n Choices: {choices}\n Hint: {hint}'}
   ```

2. **question - choices - hint - image - task + lecture**, with the following prompt:

   ```
   {f'Question: {question}\n Task: {task}\n Choices: {choices}\n Hint: {hint}\n
   Lecture: {lecture}'}
   ```

3. **question - choices - hint - image - task + lecture + solution**, with the following prompt:

   ```
   {f'Question: {question}\n Task: {task}\n Choices: {choices}\n Hint: {hint}\n
   Lecture: {lecture}\n Solution: {solution}'}
   ```

For the first two the model has to output **answer - solution**, for the third - **answer** (the easiest set up meant to ensure the model can deduce the correct answer from an already provided correct solution).

We evaluate: OpenAI, Gemini, MistralAI, as well as small local models.

| | |
|---|---|
| Model name | llava-1.5-7b-hf |
| Size | 7B |
| Published | 09.2023 |
| Type | SLM |
| Chat model | yes |
| Parent models | LLaMA, Vicuna |
| Data | GPT-generated multimodal instruction-following data |

Table 2: Model Comparison: Benchmarked Models

#### 5.1.1 Local Models from HuggingFace

**Pre-experiments** We were not successful with processing with small non-chat models. For the given text + image, the models would return short non-meaningful answers, like "Yes" to a "X or Y" question, or "no idea" to a completely defined task with all relevant context included. These were multimodal models below 1B on Huggingface, not tuned to be used in a chat scenario (and prepending a "You are a helpful assistant..." to the beginning of the prompt was not enough to fix this). We are not sure how to use those models for our purposes as of now.

**MiniGPT-4** This is a small multimodal model designed from ?? [**zhu2023minigpt**]. It offers to build the model based on Llama or Vicuna. **Unsuccessful experiment**: We tried to resort to an unofficial implementation: `wangrongsheng/MiniGPT-4-LLaMA-7B`. Loading the model was not successful (no preprocessor_config.json). Another issue: using BlipForConditionalGeneration and BlipProcessor: "You are using a model of type llama to instantiate a model of type blip. This is not supported for all configurations of models and can yield errors.". Model size: 10GB. Processor: 3.6GB.

**LlaVa** The exact model used is: `llava-hf/llava-1.5-7b-hf` (and there is a larger sister 13b model). It is based on LLaMA and (?) Vicuna fine-tuned on GPT-generated multimodal instruction-following data. It expects inputs as conversation, allowing the preprocessor to convert textual input into format "USER: request ASSISTANT: " where the assistant has to fill in the rest. We encountered the following issues while applying this model version on ScienceQA data:

- although prompted like larger models, this model returned answer strings and not choice index; this may additionally raise an issue of the returned answer not being entirely part of the choice list;
- the model currently does not accept inputs without an image (text-only); according to a huggingface discussion[1], the authors claim to have accidentally removed this possibility in a recent update. We worked around this by producing placeholder empty 10x10 white images and passing them as part of the input when the image in the dataset was missing. We also appended the suffix "Ignore image" to the end of the prompt, because the model would otherwise try to hallucinate a relation between the text and the image (e.g. "Why is the sky blue" + placeholder image results in "The sky is blue on the picture because ...");
- the model does not really seem to support batch processing; although officially the technical possibility exists, it has been reported online [2] that the model would only take into account the first image in the batch, which makes the processing useless. We did not risk to attempt the supported or custom batching and processed the filtered validation dataset 1 datapoint by 1, which took 10+ hours on GPU for inference.

There is another available model LLaVANext, which recently start to support batching, and should have better performance.

**BLIP-2**

The metrics used for evaluation at this point are: accuracy, `BLEU-1`, `BLEU-4`, `ROUGE`, similarity.

[1] https://huggingface.co/llava-hf/llava-1.5-7b-hf/discussions/38
[2] https://huggingface.co/llava-hf/llava-1.5-13b-hf/discussions/10

# 6 Approximate Timeline

1. **Now - mid November**:
   - Preliminary baseline tests: compare "pure LLM"s relying only on commonsense reasoning / only on pretraining (zero-shot, few-shot).
   - Stronger baseline – same LLM with better prompting techniques like CoT/Self-Ask.
2. **mid November - mid December**:
   - Construct agentic / RAG pipeline with "big smart" model + additional resources.
   - Optional: soft-prompting for better workflow to distill from.
3. **mid December - mid January**:
   - Knowledge distillation experiments.