Transferring Reasoning Capabilities between LLMs operating via Curriculum Learning Policy

Anonymous authors
Paper under double-blind review

Abstract

In-context *reasoning methods*, exemplified by Chain-of-Thought (CoT) *(et alia.,)* empower the reasoning abilities of large language models (LLMs), eliciting them to solve complex reasoning tasks step-by-step. Nevertheless, the capacities to deliver robust CoT explanations arise only in models with billions of parameters, representing a barrier to entry for many users forced to operate on a smaller model scale, i.e., Small Language Models (SLMs). Even though many companies are releasing LLMs of the same family with a reduced number of parameters, these models sometimes produce misleading answers and are unable to deliver accurate step-wise reasoned answers. This paper proposes a method to transfer step-wise reasoning over SLMs by operating via Instruction-tuning (IT) on synthetic demonstrations delivered in a pedagogically motivated manner. In particular, firstly, we propose aligning step-wise reasoning capabilities via IT using Demonstrations "taught" by LLMs teacher to SLMs students. Then, we operate via *Curriculum Learning*, a pedagogically motivated learning method that improves the IT phase. We analyse the impact on the downstream performances of four question-answering benchmarks. The results show that SMLs can be instructed to reason via Demonstrations delivered by LLMs. We move a step further in research: conceiving SLMs as human learners, we expose them to a CL teaching-based approach, obtaining better results on downstream performances.

1 Introduction

In-context *reasoning methods*, exemplified by Chain-of-Thought (CoT) *(et alia.,)* empower the reasoning abilities of large language models (LLMs), eliciting them to solve complex reasoning tasks step-by-step. These methods enable large language models (LLMs) to deliver multi-step, controlled reasoning Kojima et al. (2023); Wei et al. (2022), achieving outstanding results in commonsense Bubeck et al. (2023), symbolic Gaur & Saunshi (2023), and mathematical Liu et al. (2023) reasoning tasks. LLMs achieve all these results with at least several billions of parameters, such as GPTs OpenAI (2024), Llamas Touvron et al. (2023); Grattafiori et al. (2024) and Mistral MistralAI (2023).

In contrast, Small Language Models (SLMs) break down the problems and deliver step-wise answers less effectively. Although these models are highly functional across different tasks, the CoT prompting mechanism consistently benefitted only models at a certain threshold scale (e.g., with more than 60B parameters Wei et al. (2023)). SLMs are crucial in fostering research since these are smaller versions of LLMs that are often open-source and accessible to most researchers, e.g., Llama-3-1b and Llama-3-8b Grattafiori et al. (2024). However, these SLMs produce illogical answers when prompted under the CoT framework.

To this end, we propose an approach to align the reasoning abilities of the SMLs (students) with the LLMs (teachers) via Instruction-tuning-CoT (*IT-CoT*), that is, an instruction tuning over synthetic CoT Demonstrations delivered from larger models (see Figure 1).

Concerning the foundation teacher-student approach Magister et al. (2023); Ho et al. (2023b); Li et al. (2023), we move a step further by introducing the *IT* via CoT, and, concerning Ranaldi & Freitas (2024a), we improve the strategies to expose the student to examples in a reasonable, pedagogically-motivated order using Curriculum Learning Bengio et al. (2009). Hence, starting from the idea that humans acquire first elemental concepts and then, gradually, more complex ones, Bengio et al. (2009) proposed Curriculum Learning (CL) and demonstrated its benefits in several tasks. We adopt this idea to reorder the *IT* Demonstrations in a meaningful way. Hence, we evaluate the reasoning chains that are answers delivered by teachers via CoT prompting to elicit student learning.

This leads to the target research questions, which are the focus of this paper: *RQ1 - How does IT-CoT via synthetic Demonstrations impact the reasoning abilities of students' models? And what is the effect of step-wise Demonstrations delivered with the CoT process? RQ2 - How important is reasoning chain valuation to facilitate the presentation of demonstrations during IT?*

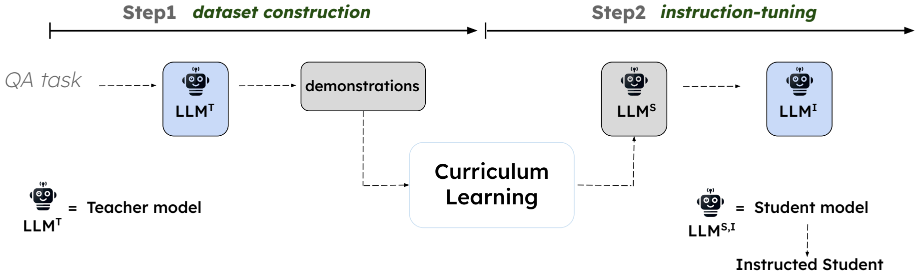

Figure 1: In Instruction-tuning, the smaller models instruct themselves using the synthetic demonstrations generated by the larger models. Hence, we elicit a larger model to deliver step-wise reasoned answer solutions. Moreover, we evaluate the reasoning chain using Curriculum Learning metrics to facilitate the instruction phase and expose the Demonstrations in a meaningful way.

We answer these questions by selecting Llama-3-1b, Llama-3-8b Grattafiori et al. (2024) as students and Llama-3-70b, and GPT-4 as teachers. We perform a comprehensive analysis using four question-answering benchmarks. We use Llama-3-70 and GPT-4 to deliver Answers at the core of the Demonstrations (see Figure 1) to instruct Llama-3-1 and -13. Furthermore, we evaluate the complexity of the reasoning chains in generated answers to expose the students to Demonstrations delivered by teachers. Hence, we propose a metric based on informativeness comprehensibility used as a pivot in the *IT* phase.

Behind a wide analysis, we show that the *IT* and *IT-CoT* approaches on Demonstrations instruct students, and they outperform baseline SLMs in all proposed benchmarks. In addition, the students exposed to the Demonstrations via the CL approach outperformed students instructed via non-CL.

Our findings can be summarized as follows:

i) The *IT* of SLM students via Demonstrations delivered by an LLM teacher outperformed the baselines in terms of downstream performance. The SLMs instructed via Demonstrations consistently outperformed the baselines defined by non-tuned SLMs on the four proposed question-answering benchmarks.

ii) The CL-based *IT* approach outperforms standard *IT*. Llama-3-1 and Llama-3-8, instructed via the CL method, outperform the instructed models without CL.

iii) Finally, the CL method favours the alignment of CoT abilities within the family. Llama-3-1 and Llama-3-8 were exposed to CL Demonstrations produced by Llama-3-70, which outperformed students instructed by GPT-4 teachers in other SMLs as well.

2 Method

We propose three steps to align the reasoning abilities of smaller Language Models using further knowledge generated by larger Language Models, as shown in Figure 1. In the first part, there is an annotation phase where the large language models (LLMs) systematically prompt generate outputs (Section 2.1). The outputs will be the core of Demonstrations used during the Instruction-tuning *IT* phase from the smaller Language

Models, presented in Section 2.2. However, the Curriculum Learning approach is behind the *IT* phase, where the Demonstrations are reorganized following our measure introduced in Section 2.3.

2.1 Teacher Model

As teacher model, we selected the largest Llama version Grattafiori et al. (2024), that is, Llama-3-70b, and in terms of comparison, GPT-4[1] OpenAI (2024). We selected GPT-4 because it generates high-quality data with and without the CoT prompting approach, as shown in Fu et al. (2023). Meanwhile, Llama-3-70b because it has smaller versions that can be used as students of the same family (presented in Section 2.2), and these smaller versions obtain remarkable results despite the reduced number of parameters.

Hence, we proposed the following prompt in a zero-shot scenario:

```
Choose the answer to the question only from
options A, B, C, D.
Question: <Question>
Choices:
A) <Option1>
B) <Option2>
C) <Option3>
D) <Option4>
Answer: Let's think step by step
```

Input prompts have a generic structure, but behind "`Answer:`" we insert the formula "`Let's think step by step`" as done in Kojima et al. (2023); Wei et al. (2022), that is shown in Table 7.

Following the annotation process performed by LLMs, the answers generated by teachers models that are the annotations have been used to construct the Demonstrations (see Table 1).

2.2 Student Model

Several SLMs have been fine-tuned for instruction-following Taori et al. (2023) and reinforcement learning with human feedback Ouyang et al. (2022). However, whatever the techniques, the smaller Language Models[2] do not seem able to break down a given task and deliver a step-wise solution.

Recent work proposes techniques of knowledge distillation Li et al. (2023) and skill-refinement Huang et al. (2022); Ranaldi & Freitas (2024b) to teach SLMs to produce reasoned answers. Following this idea, we analysed the impact of synthetic *IT* from Demonstrations provided by different teachers. Hence, we chose two SLMs students, Llama-3-1b and Llama-3-8b Grattafiori et al. (2024). In both cases, we used "instruct" versions, i.e., Llama-3-1-chat and Llama-3-8-instruct (versions detailed in Table 8), which we will refer to in the paper as Llama-3-1 and -8.

Student models are evaluated before and after the IT phases. In particular, we propose the standard approach where the students are instructed via Demonstrations delivered arbitrarily and in the Curriculum Learning manner (introduced in Section 2.3). The structure of IT arises from Alpaca Taori et al. (2023), which operates via Demonstrations consisting of an instruction that, in our case, is fixed, i.e., `Choose the answer to the question only from options A, B, C, D.`, an input which is the question, and an expected output which, in our case, are the output generated by the LLMs teachers. Table 1 shows an example of input. Additional details about the IT steps are provided in Section 3.2.1.

---

[1] We use *GPT-4-turbo*, however in the rest of work we will use only GPT-4

[2] We consider Smaller models with less than 60B of parameters based on Wei et al. (2022).

Instruction

```
Choose the answer to the question only from options A, B, C, D.
```

Input

```
Question:  Which form can be made into a triangle?
Choices:
A) ice
B) steam
C) water
D) salt water
```

Output (that is the Answer produced by teacher)

```
Answer:
A) ice:  Ice is a solid form of water.  In solid form, substances can be
molded or shaped into various forms, including triangles.  If you had a
block of ice, you could carve or mold it into a triangular shape.
B) steam:  Steam is the gaseous form of water.  Being a gas, it doesn't
have a fixed shape.  You can't shape steam into a triangle without
containing it within some form of solid boundary, and even then, the
steam itself isn't triangular - the container is.
C) water:  Water is a liquid that takes the shape of its container.  You
can pour water into a triangular container, and while the volume inside
would be triangular, the water itself isn't inherently triangular.

D) salt water:  This is essentially the same as option C, but with salt
dissolved in it.  Being a liquid, it takes the shape of its container.

Given these considerations, the only substance among the options that can
be shaped into a triangle by itself is A) Ice.
```

Table 1: Example of Demonstration. The structure is composed by: `Instruction`, `Input` and `Output`. In this case, the last part is the `Answer` produced by GPT-4 CoT prompted.

2.3 Curriculum Learning

Aligning the teacher-student reasoning abilities via synthetic Demonstrations delivered by step-wise answers provided by teachers, CoT prompted is a promising technique. However, some aspects need clarification: what constitutes an answer containing a good reasoning chain and how to evaluate it to optimize the IT phase. Following the Curriculum Learning (CL), where training algorithms can achieve better results when training data are presented according to the model's current skills, Bengio et al. (2009). We propose a method for evaluating the reasoning chain present in the CoT Answers (that represent the outputs of CoT Demonstration) using two fundamental properties: (1) comprehensibility, that is, the comprehensibility of a text according to metrics proposed by Talburt (1986), and (2) informativeness, that is, every step of the chain provides new information that is useful and informative for deriving the generated answer. We apply this metric to the CoT Answers provided by the teachers; then, we reorder the demonstrations according to our measure.

Informativeness To quantify the effectiveness of each step contributing novel information beneficial for deriving the final Answer A, we propose an assessment based on the Entropy and Information Gain (IG). The Entropy, represented by $H(S)$, evaluates the unexpected within a given sequence S, where $S_i \in A$. The entropy is given by:

$$H(S) = -\sum_{w \in S} p(w) \log_2 p(w) \tag{1}$$

where $p(w)$ denotes the probability of token w occurring in the sequence. Hence, we compute the IG between a previous S_{prev} and a current sequence S_i as:

$$IG(S_{prev}, S_i) = H(S_{prev} + S_i) - H(S_{prev}) \tag{2}$$

This metric quantifies how much new information the current step adds relative to the cumulative content previously considered. To obtain a comprehensive measure, we calculate the average IG across the different steps as follows:

$$d_I(A_i) = \frac{1}{N} \sum_{i=1}^{N} IG(S_{prev}, S_i) \tag{3}$$

where N represents the total number of steps in the Answer or the sequences S_i. We calculate this value for each answer A_i and obtain the maximum $d_{I_{max}}$ and the minimum $d_{I_{min}}$ scores. Finally, we normalize these values:

$$\hat{d_I}(A_i) = \frac{d_I(A_i) - d_{I_{min}}}{d_{I_{max}} - d_{I_{min}}}, \forall i \in [0, |D|]. \tag{4}$$

where $|D|$ are all answers to a specific benchmark.

Comprehensibility Typical factors for measuring comprehensibility are Speed of perception, Perceivability in peripheral vision, Reflex blink technique, Eye movements, Cognitively motivated features, and Word difficulty. However, it is not always possible to capture all these features.

Hence, we used the Flesch-Kincaid metric Talburt (1986). This metric is used to assess the comprehensibility of a text. It is based on the length of sentences and words within a text and provides a score that indicates the text's difficulty level. The lower the score, the easier it is to read and comprehend the text. The formula for calculating the Flesch-Kincaid Grade Level score is as follows:

$$d_C(A_i) = 0.39 \frac{Avg(d_L(A_i))}{100} + 11.8 \frac{Avg(d_L(w_i))}{100} - 15.59 \tag{5}$$

where $Avg(d_L(A_i))$ average answer length is the number of words in a sentence divided by the number of sentences, and $Avg(d_L(w_i)$ is the average word length, i.e. is the number of syllables per word divided by the number of words. The value 0.39 is used to scale the effect of the average sentence length to compare it to the effect of the average word length, weighted by 11.8. The final score is then adjusted by subtracting the value of 15.59, which adjusts the score scale to match the grading levels used in education more closely. We calculate this value for each Answer A_i and obtain the maximum $d_{C_{max}}$ and the minimum $d_{C_{min}}$ scores. Finally, we normalize these values:

$$\hat{d_C}(A_i) = \frac{d_C(A_i) - d_{C_{min}}}{d_{C_{max}} - d_{C_{min}}}, \forall i \in [0, |D|]. \tag{6}$$

Constructing the CL-Demonstration We gather the annotations (answers) delivered by the CoT-prompted teachers (as explained in Section 2.1), and we estimate the informativeness $\hat{d_I}(A_i)$ and complexity $\hat{d_C}(A_i)$ for each answer $A_i, \forall i \in |D|$.

Then we merge the two values in:

$$d_{IC}(A_i) = \hat{d_I}(A_i) + \hat{d_C}(A_i) \tag{7}$$

We use $d_{IC}(A_i)$ as a pivot value to reorder the Answers provided by the teachers. The Answers (which form the output of the Demonstrations) will be delivered in the Instruction-tuning phase to the students in ascending order with respect to the value $d_{IC}(A_i)$. These heuristics are very lightweight: using only 16GB of memory, we can process up to 20k Responses per second to produce the informativeness and comprehensibility metrics.

3 Experimental Setup

To make the experiments comparable with state-of-the-art models, we use four benchmarks (introduced in Section 3.1) that are generally used to assess the abilities of Large Language Models (LLMs). Moreover, to conduct the Instruction-tuning phase on the Small Language Models (SMLs), we use two approaches: the first one is presented in Section 3.2, which we call Instruction-tuning on Demonstrations; the second is based on the Curriculum Learning (CL) approach where the students are exposed to CL-Demonstrations that are Demonstrations reordered in a CL way, as exemplified in Section 2.3. All code is available in the supplementary material, to be released if accepted.

3.1 Data

General Commonsense Reasoning We evaluate the models' ability to perform general reasoning on the CommonSenseQA Talmor et al. (2019) (CSQA) and OpenBookQA Mihaylov et al. (2018) (OBQA). CommonSenseQA is one of the best-known datasets of answers to multiple-choice questions dealing with different types of general commonsense knowledge. OpenBookQA is a resource that contains questions requiring multi-step reasoning, common knowledge, and rich text comprehension. It is inspired by high school-level open-book exams in physics and biology, aiming to assess human comprehension and application of foundational concepts.

Physical Interaction Reasoning We evaluate the models' ability to perform physical reasoning on the Interaction Question Answering (PIQA) Bisk et al. (2019). It is a resource consisting of everyday situations with typical and atypical solutions.

Social Interaction Reasoning We evaluate the models' ability to perform social reasoning on the Social Interaction Question Answering (SIQA) Sap et al. (2019). It is a benchmark focusing on reasoning about people's actions and social implications. The actions in Social IQa cover various social situations and candidates for plausible and not plausible answers.

Splitting Details Since a test split for all benchmarks is not always available open-source, we adopt the following strategy: we use 4000 examples with equally distributed target classes as training data and the validation versions found on huggingface as test data. We performed this split because we needed to observe the impact of the responses provided by the teacher models on different benchmarks. The same is true for validation since we needed open-source and reproducible data to conduct a detailed evaluation of the student models. In Table 10, we report the quantitative information, global, and splitting ratios, and in Table 9, we show one example for each benchmark. The data are fully accessible and open-source, as described in Table 11.

3.2 Teaching to Reason

We selected Llama-3-70 and GPT-4 as the teachers (introduced in Section 2.1). Consequently, the LLMs are prompted in the zero-shot scenarios, as shown in Table 6 and Table 7.

We selected Llama-3-1 and Llama-3-8 Grattafiori et al. (2024) as student models (as described in Section 2.2). Therefore, the students models are Instruction-tuned via Demonstrations, as introduced in Section 3.2, and via CL-Demonstrations, as explained in Section 2.3. Table 1 shows a Demonstration containing the Instruction, Input, and, as Output, the Answer-delivering CoT, an output generated by GPT-4 CoT-prompted.

3.2.1 Models Setup

We conduct the Instruction-tuning phases using QLoRA proposed by Dettmers et al. (2023). This approach allows tuning to be conducted while reducing memory usage and preserving the performances. We follow the training approach proposed in Alpaca Taori et al. (2023). Our models are trained for three epochs and set the learning rate as 0.00002 with 0.001 weight decay. We use the cosine learning rate scheduler with

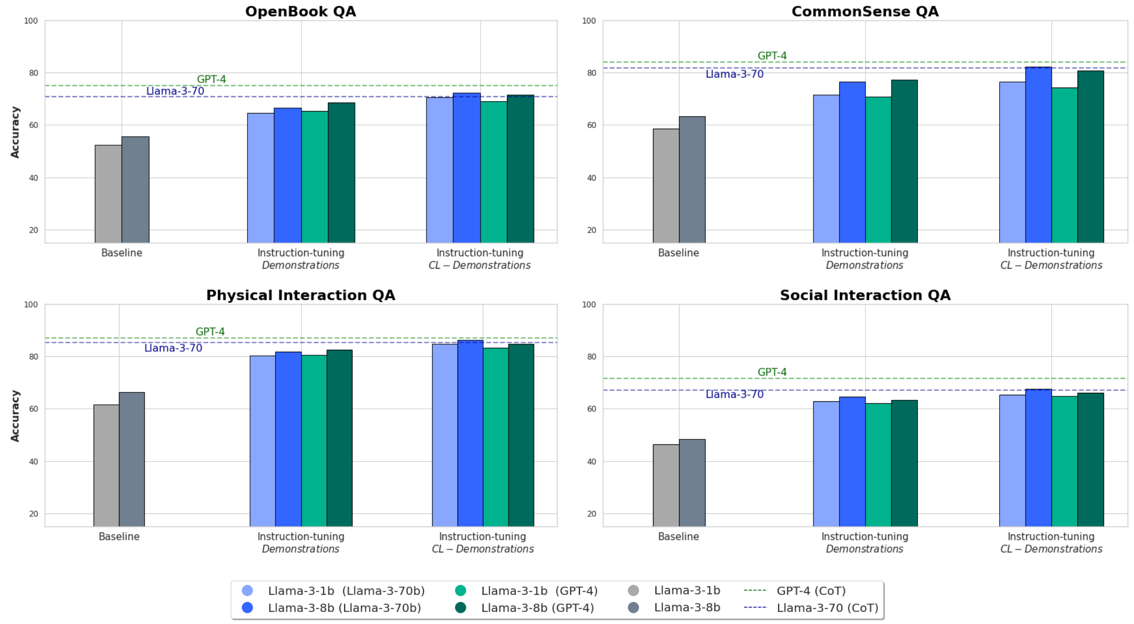

Figure 2: Accuracies (%) on benchmarks (Section 3.1) before Instruction-tuning (i.e., Baselines) and after on Demonstrations and CL-Demonstrations. Moreover, there are the teachers' performances also shown in Table 5

a warmup ratio of 0.03. We conducted our experiments on a workstation equipped with two Nvidia RTX A6000 with 48GB of VRAM.

3.3 Evaluation

The most commonly used evaluation methods for question-answering tasks are language-model probing, in which the option with the highest probability is chosen Brown et al. (2020), and multiple-choice probing, in which the models are asked to answer. The evaluation is performed with a function taking the maximum value and, in the second case, with string matching. The second method is widely used in recent evaluations because it applies to models such as GPT-4 and GPT-4 OpenAI (2024) where probability values cannot be accessed. In our experiments, we chose the latter to have a comparable and scalable pipeline. Hence, we performed a string matching between the generated outputs and the targets.

4 Results

Language Models that do not get it can be elicited to do it through the knowledge of teacher models. These conclusions can be observed in Figure 2, where we reported the downstream performances without the Instruction-tuning phase (see the Baseline) and the Instruction-tuning on Demonstrations. As discussed in Section 4.1, Small Language Models (SLMs) CoT prompted obtained weak results. In contrast, models that are instructed via Chain-of-Thought (CoT) Demonstrations, i.e., Demonstrations produced by CoT-prompted Large Language Models (LLMs), outperform non-instructed models (Section 4.2). However, although Demonstrations produced better students, the complete alignments between students and teachers are realized with the *Curriculum Learning* procedure, as discussed in Section 4.3. In particular, the students instructed via the CL (Instruction-tuning CL-Demonstration in Figure 2) outperformed the students instructed via standard Instruction-tuning.

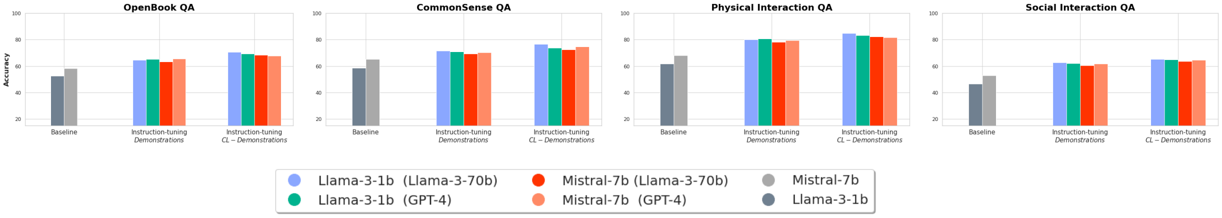

Figure 3: Accuracies of Llama-3-1 and Mistral-7 Instruction-tuned using setup proposed Section 3.

Finally, the CL approach delivers the teacher-student family-alignment. In Figure 2 (horizontal lines), it is possible to observe the phenomenon of family-alignment between Llama-3-70 and Llama-3-1 and -13 in more detail in Section 4.4.

4.1 CoT-abilities of Small Language Models

Chain-of-Thought (CoT) prompts do not consistently deliver downstream performance improvements. SLMs, i.e., with fewer parameters, have not benefitted the prompting with the CoT mechanism. In particular, we evaluated performance on four question-answering benchmarks, described in Section 3.1, using two versions of Llama. Using a baseline demonstrations produced via zero-shot prompt (which we call "Baseline") and a CoT prompt (Table 6 and Table 7), we obtained the performances in Table 2.

The results confirm what Wei et al. (2022) have claimed about the limitations of the emergent CoT prompting abilities that are not observable in SLMs. Moreover, using CoT prompting leads to model confusion with subsequent degradation of downstream results. It is possible to observe these phenomena in OpenBookQA (OBQA) and CommonSenseQA (CSQA) (down arrows in Table 2). In particular, there is a marked deterioration in Llama-3-1 (see ⇓), which has half the parameters of Llama-3-8 (see ↓).

The same behaviour was not observed for Physical- and Social-Interaction Question Answering (PIQA) and (SIQA). Not considering the nature of benchmarks, unlike the others, they are always question-answering multiple-choice-questions but have fewer possible choices, as shown in Table 10. In this regard, we hypothesize that the most controllable scenarios, where chain reasoning is limited to fewer options, are reasonable by SLMs elicited with CoT prompts.

Benchmarks	**Llama-3-1**		**Llama-3-8**	
	Baseline	**CoT**	**Baseline**	**CoT**
OBQA	**53.6**	51.0⇓	**58.7**	57.0↓
CSQA	**58.9**	51.2⇓	**63.8**	61.2↓
SIQA	46.8	*45.9*	48.7	47.8
PIQA	62.0	*63.9*	67.2	*69.0*

Table 2: Accuracies of Llama-3-1 and Llama-3-8, both without further tuning, on testing data with the standard prompt (Baseline) (see Table 6) and CoT prompt (CoT) (see Table 7).

4.2 The Instruction-tuning Method

Instruction-tuning led by Large Language Models (teachers models), able to reason elicit the Smaller Language Models (students models) to do the same. This is shown in Figure 2. The student models after Instruction-tuning on synthetic Demonstrations delivered by teacher models outperformed the baselines in the four proposed benchmarks. While performances are conspicuous improvements overall, they have sensible variations. The teacher models have different characteristics. They consequently achieve different performances in the proposed benchmarks. Table 5 shows the performances in the zero-shot scenario (CoT prompting and not) on the data used to conduct the Instruction-tuning phase and on the same test set used to evaluate the proposed models.

Although the performances on the "training set" are different (performances of GPT-4 and the same for Llama-3-70 in Table 5), this bias does not affect the students. The Llama-3-1 and -8 with GPT-4 as teacher outperform the Llama-3-1 and -8 with Llama-3-70 as teacher only on OBQA. As far as CSQA and PIQA are concerned, there is a balance that is not present in SIQA, where the students of Llama-3-70 outperform the others.

However, in the Instruction-tuning method, instruction is conducted using Demonstrations (composed of Answers provided by teachers) delivered arbitrarily. Therefore, we propose to study both the intrinsic complexity of the answers and their impact on the students' exposure. To this end, we introduce a CL-based instruction approach where demonstrations are delivered to students in a meaningful order (Section 4.3).

4.3 The Impact of Curriculum Learning

Instruction-tuning via Curriculum Learning Demonstrations elicits the reasoning abilities of students. The students gradually exposed to increasingly meaningful Demonstrations (CL-Demonstrations) learn better than those exposed to arbitrary Demonstrations. This is shown in Figure 2 (bars Instruction-tuned CL-Demonstrations), where Llama-3-1 and -13 consistently outperformed the other models.

The benchmarks where the most significant effects can be observed are CSQA and OBQA, with an increase in average accuracy scores of 6 and 5 points, respectively. The same effects are less evident in PIQA and SIQA. One possible reason for this phenomenon might be tied again to the nature of the benchmarks, as hypothesized in Section 4.1. To analyze this phenomenon, we studied the components of the complexity measure proposed in Section 4.5.

4.4 The role of CL in family-alignment

Instruction-tuning via CL-Demonstrations still aligns students' reasoning abilities with family teachers, even as instruction decreases. In fact, from Figure 2, we can observe that the performances of students instructed via CL-Demonstrations delivered by teachers from the same family outperform the others.

Moreover, to validate our hypothesis of family alignment, we introduced Mistral-7b MistralAI (2023), a new SLMs with 7 billion parameters that outperform the Llama-3-8 version on several benchmarks, as shown by MistralAI (2023). Specifically, we reproduced the experiments introduced in Section 4.2. In Figure 3, it can be seen that Llama-3-1 instructed on different types of Demonstrations delivered by Llama-3-70 almost consistently outperforms Mistral-7b. These results confirm that Demonstrations derived from in-family teachers have a more significant impact on student models than the others.

4.5 Ablation Study

The role of measures The informativeness and complexity exposed to students in a meaningful order instruct better students. We conducted an Ablation study to estimate the impact of our evaluation measures proposed in Section 2.3. Hence, we reproduced the same configurations proposed in Section 4.2, but removed one of the two components (informativeness and complexity) presented in Section 2.3. The results in Table 4 show that students instructed on CL-Demonstrations ordered by comprehensibility and informativeness consistently outperform students instructed via Demonstrations. The results show that students trained on the Demonstrations sorted by informativeness are more productive in QA tasks with more choices. In comparison, complexity proved helpful in cases where the number of options is minor. Phenomenon manifested in CSQA and OBQA with 5 and 4 choices and PIQA and SIQA with 2 and 3 choices, respectively (see Appendix, Tables 9 and 10).

The impact of quality In fact, the quality of demonstrations matters by filtering the demonstrations (using a heuristic based on string matching and GPT-4o annotations as described in Appendix **??**) we studied the impact of different types of demonstrations. Table 3 shows that models instructed by demonstrations ordered according to CL metrics outperform the others even in cases where they are incorrect (misleading). Thus, we can conclude that the impact of the measure introduced in Section **??** is positive beyond the outcome of the example used as a training demonstration.

Method	**OBQA**	**CSQA**	**SIQA**	**PIQA**
Instruction-tuning CL	71.2	**80.2**	84.3	66.4
Instruction-tuning CL (gold)	**72.7**	**80.2**	**85.0**	**66.8**
Instruction-tuning (gold)	70.3	78.5	82.6	65.4
Instruction-tuning CL (misleading)	60.0 ↑	63.0	48.9 ↑	68.4 ↑
Instruction-tuning (misleading)	59.2	62.6 ↓	46.5 ↓	67.0 ↓
baseline	58.3	63.1	47.9	67.3

Table 3: Accuracies of Llama-3-8, instructed using Instruction-tuning (NB without CL) and Instruction-tuning CL demonstrations as in Section 3, using misleading (incorrect) and gold (correct) demonstrations.

5 Related Work

5.1 Learning from Natural Language Explanation

Current methods for conditioning models on task instructions and provided explanations for individual data points replace the ancient intermediate structures Hase & Bansal (2022) that used rationales Zhang et al. (2016) or inputs Narang et al. (2020); Talmor et al. (2020) to learn the models. Reasoning via the CoT builds upon prior efforts wherein explanations are viewed as intermediary constructs produced during inference Rajani et al. (2019). Our research stems from the studies of Shridhar et al. (2023); Ho et al. (2023a). Particularly, we adopt the idea of an LLM teacher and a second smaller LLM that takes a student's position Magister et al. (2023). Learning uses teacher-generated explanations, demonstrating prompt CoT downstream Li et al. (2023); Ho et al. (2023a). Li et al. (2023) claims that massive demonstrations significantly improve performance over the single-sample approach Shridhar et al. (2023).

5.2 Large Language Models as a Teacher

Several papers have been published simultaneously, including those by Ranaldi & Freitas (2024a); Paul et al. (2024); Ranaldi & Freitas (2024b), and Saha et al. (2023) that prove the effect of transferring ability to produce CoT reasoning from larger to smaller models. Table 12 resumes all main points of these contributions.

Our work goes beyond the following ways: *i)* We propose a method for aligning CoT abilities via Instruction-tuning through Demonstrations produced by answers generated by GPT-4 and Llama-3-70. *ii)* We study how to provide Demonstrations to students by proposing a measure for evaluating the Answers provided by teachers. In particular, we analyse the alignment performance between in-family and out-family models. *iii)* Hence, we propose an approach for improving the alignment of reasoning abilities between teachers and students by employing our evaluations to expose the students meaningfully.

6 Conclusion

This paper proposes a method to enable step-wise reasoning over SLMs by introducing two mechanisms. First, we propose aligning CoT abilities via Instruction-tuning with the support of CoT Demonstrations delivered by LLMs teacher. Second, we use the Curriculum Learning approach to empower the Instruction-tuning phase. Hence, we analyse the impact on the downstream abilities of four question-answering benchmarks. Our results show that SLMs can be instructed to deliver robust step-wise reasoned answers via Demonstration produced by LLMs. We move a step further in research: conceiving SLMs as human learners, we expose them to a CL teaching-based approach, obtaining better results on downstream performances.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In Proceedings of the 26th annual international conference on machine learning, pp. 41–48, 2009.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.

Yao Fu, Litu Ou, Mingyu Chen, Yuhao Wan, Hao Peng, and Tushar Khot. Chain-of-thought hub: A continuous effort to measure large language models' reasoning performance, 2023.

Vedant Gaur and Nikunj Saunshi. Reasoning in large language models through symbolic math word problems. In Findings of the Association for Computational Linguistics: ACL 2023, pp. 5889–5903, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.364. URL `https://aclanthology.org/2023.findings-acl.364`.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang

Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Peter Hase and Mohit Bansal. When can models learn from explanations? a formal framework for understanding the roles of explanation data. In Proceedings of the First Workshop on Learning with Natural Language Supervision, pp. 29–39, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.lnls-1.4. URL `https://aclanthology.org/2022.lnls-1.4`.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 14852–14882, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.830. URL `https://aclanthology.org/2023.acl-long.830`.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers, 2023b.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve, 2022.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2665–2679, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.150. URL `https://aclanthology.org/2023.acl-long.150`.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. Evaluating the logical reasoning ability of chatgpt and gpt-4, 2023.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 1773–1781, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.151. URL `https://aclanthology.org/2023.acl-short.151`.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering, 2018.

MistralAI. Mistral-7b-instruct, 2023. URL `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1`.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. Wt5?! training text-to-text models to explain their predictions, 2020.

OpenAI. Gpt-4 technical report, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. Refiner: Reasoning feedback on intermediate representations, 2024.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4932–4942, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1487. URL `https://aclanthology.org/P19-1487`.

Leonardo Ranaldi and Andre Freitas. Aligning large and small language models via chain-of-thought reasoning. In Yvette Graham and Matthew Purver (eds.), Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1812–1827, St. Julian's, Malta, March 2024a. Association for Computational Linguistics. URL https://aclanthology.org/2024.eacl-long.109.

Leonardo Ranaldi and Andre Freitas. Self-refine instruction-tuning for aligning reasoning in language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 2325–2347, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.139. URL https://aclanthology.org/2024.emnlp-main.139.

Swarnadeep Saha, Peter Hase, and Mohit Bansal. Can language models teach weaker agents? teacher explanations improve students via personalization, 2023.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. Distilling reasoning capabilities into smaller language models. In Findings of the Association for Computational Linguistics: ACL 2023, pp. 7059–7073, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.441. URL https://aclanthology.org/2023.findings-acl.441.

John Talburt. The flesch index: An easily programmable readability analysis algorithm. New York, NY, USA, 1986. Association for Computing Machinery. ISBN 0897911865. URL https://doi.org/10.1145/10563.10583.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge, 2020.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

Ye Zhang, Iain Marshall, and Byron C. Wallace. Rationale-augmented convolutional neural networks for text classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 795–804, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1076. URL https://aclanthology.org/D16-1076.

Appendix A

	Benchmarks			
Students	**OBQA**	**CSQA**	**PIQA**	**SIQA**
Llama-3-1 (Llama-3-70)				
Arbitrary Teaching	64.7	71.6	80.2	62.8
Teaching via IC	**70.5**	76.5	**84.8**	65.3
Teaching via I	70.2⇑	**77.2**⇑	81.2	61.8⇓
Teaching via C	66.4	69.7⇓	84.3⇑	**66.2**
Llama-3-1 (GPT-4)				
Arbitrary Teaching	65.3	70.8	80.5	62.2
Teaching via IC	**69.2**	**74.2**	83.3	**64.8**
Teaching via I	68.5⇓	73.7⇑	79.6⇓	63.8
Teaching via C	66.3	69.8	**83.9**⇑	65.7⇑
Llama-3-8 (Llama-3-70)				
Arbitrary Teaching	66.5	76.5	81.9	64.5
Teaching via IC	72.3	**82.2**	**86.2**	67.7
Teaching via I	**73.4**⇑	81.9⇑	80.7⇓	63.8
Teaching via C	67.9	76.6	84.3⇑	**70.3**
Llama-3-1 (GPT-4)				
Arbitrary Teaching	68.5	77.3	82.6	63.3
Teaching via IC	**71.6**	80.5	**84.9**	**66.1**
Teaching via I	70.8⇑	**81.7**	81.9	62.7
Teaching via C	68.2⇓	78.5	82.3	65.9⇑

Table 4: Ablation study on our Instruction-tuning CL-Demonstrations approach.

Benchmarks	**Llama-3-70**		**GPT-4-o**	
	Baseline	**CoT**	**Baseline**	**CoT**
Training				
OBQA	65.8	71.3	67.1	**78.8**
CSQA	75.0	79.6	80.4	**85.6**
SIQA	66.2	67.5	69.2	**72.6**
PIQA	82.9	**86.0**	83.5	85.8
Testing				
OBQA	66.0	72.0	68.3	**75.6**
CSQA	73.4	81.8	81.4	**84.0**
SIQA	65.2	66.9	67.3	**71.8**
PIQA	83.8	85.6	85.2	**86.5**

Table 5: Accuracy (%) of Llama-3-70 and GPT-4 (teachers) on training and testing data with CoT prompt (CoT) and with the standard prompt (Baseline).

Appendix B

Zero-Shot

`Choose the answer to the question only from options A, B, C, D.`
`Question:` Which animal gives birth to live young?
A) Shark
B) Turtle
C) Giraffe
D) Spider
`Answer:`

Table 6: Example of Zero-Shot prompting.

Zero-Shot Chain-of-Thought

`Choose the answer to the question only from options A, B, C, D.`
`Question:` Which animal gives birth to live young?

A) Shark
B) Turtle
C) Giraffe
D) Spider
`Answer: Let’s think step by step`

Table 7: Example of Zero-Shot Chain-of-Thought prompting.

A Evaluation Metrics

We used a double-check to assess the accuracy of the responses delivered in the different experiments. In the first step, we used an exact-match heuristic (this was used for most of the evaluations, especially in cases of multiple-choice QA). However, since some experiments required a more accurate response check, we used GPT-4o as a judge.

GPT-4o Evaluation Prompt

Given the following "#Senteces", you are a decider that decides whether the "Generated Answer" is the same as the "Target Answer". If the output doesn’t align with the correct answer, respond with ’0’, whereas if it’s correct, then respond with ’1’. *Please, do not provide any other answer beyond ‘0’ or ‘1’.*
#Senteces:
Generated Answer: {model_result}
Target Answer: {correct_answer}.

Appendix C

Model	Version
Llama-3-1	meta-llama/Llama-3.2-1B-Instruct
Llama-3-8	meta-llama/Meta-Llama-3-8B-Instruct
Llama-3-70	meta-llama/Meta-Llama-3-70B
Mistral-7	mistralai/Mistral-7B-Instruct-v0.2
GPT-4o	OpenAI API (gpt-4o-2024-08-06)

Table 8: In this table, we list the versions of the models proposed in this work, which can be found on huggingface.co. We used all the default configurations proposed in the repositories for each model.

Appendix D

Dataset	Example
OBQA Mihaylov et al. (2018)	*When birds migrate south for the winter, they do it because* **A) they are genetically called to.** B) their children ask them to. C) it is important to their happiness. D) they decide to each.
CSQA Talmor et al. (2019)	*Aside from water and nourishment what does your dog need?* A) bone. B) charm. C) petted. **D) lots of attention.** E) walked.
PIQA Bisk et al. (2019)	*How do you attach toilet paper to a glass jar?* **A) Press a piece of double-sided tape to the glass jar and then press the toilet paper onto the tape.** B) Spread mayonnaise all over the jar with your palms and then roll the jar in toilet paper.
SIQA Sap et al. (2019)	*Taylor gave help to a friend who was having trouble keeping up with their bills. What will their friend want to do next?* A) Help the friend find a higher paying job. **B) Thank Taylor for the generosity.** C) pay some of their late employees.

Table 9: Examples of the benchmarks used in this paper.

	OBQA	**CSQA**	**PIQA**	**SIQA**
classes	4	5	2	3
Training # examples for each class	1000	800	2000	1330
Test # examples for each class	125* (± 8)	235* (± 11)	924* (± 18)	640* (± 19)

Table 10: Characteristics Training and Test set of benchmarks proposed in Section 3.1. The * indicates that the number of examples are not perfect balanced, but the difference from the average is marginal.

Name	Repository
CSQA Talmor et al. (2019)	`huggingface.co/datasets/commonsense_qa`
OBQA Mihaylov et al. (2018)	`huggingface.co/datasets/openbookqa`
PIQA Bisk et al. (2019)	`huggingface.co/datasets/piqa`
SIQA Sap et al. (2019)	`huggingface.co/datasets/social_i_qa`

Table 11: In this table, we list the versions of the benchmark proposed in this work, which can be found on huggingface.co.

Work	**Method**	**Teachers**	**Students**
Magister et al. (2023)	SFT	PaLM GPT-4	T5-small, -medium T5-large, -xxl
Li et al. (2023)	SFT	GPT-3 175B	OPT-1.3b
Shridhar et al. (2023)	SFT	GPT-3 175B	GPT-2
Ho et al. (2023a)	SFT	InstructGPT (text-davinci-002)	GPT-3 (ada,babbage,curie)
Ranaldi & Freitas (2024a)	IT	GPT-3.5 Llama-2-70	Llama-2-7, Llama-2-13
Ours	**IT**	**Llama-3-70b** **GPT-4**	**Llama-3-1b, -13b** **Mistral-7b**

Table 12: Summary of methods, teacher and student models of previous work, we indicate Supervised Fine-tuning as (SFT) employed in most previous work.