Large Language Models Are Reasoning Teachers

Namgyu Ho, Laura Schmid, Se-Young Yun
KAIST
{itsnamgyu, laura.schmid, yunseyoung}@kaist.ac.kr

Abstract

Recent works have shown that chain-of-thought (CoT) prompting can elicit language models to solve complex reasoning tasks, step-by-step. However, prompt-based CoT methods are dependent on very large models such as GPT-3 175B which are prohibitive to deploy at scale. In this paper, we use these large models as *reasoning teachers* to enable complex reasoning in smaller models and reduce model size requirements by several orders of magnitude. We propose *Fine-tune-CoT*, a method that generates reasoning samples from very large teacher models to fine-tune smaller models. We evaluate our method on a wide range of public models and complex tasks. We find that Fine-tune-CoT enables substantial reasoning capability in small models, far outperforming prompt-based baselines and even the teacher model in many tasks. Additionally, we extend our method by leveraging the teacher model's ability to generate multiple distinct rationales for each original sample. Enriching the fine-tuning data with such *diverse reasoning* results in a substantial performance boost across datasets, even for very small models. We conduct ablations and sample studies to understand the emergence of reasoning capabilities of student models.[1]

1 Introduction

Language models (LMs) have demonstrated remarkable performance in a wide range of downstream tasks. Recently, large language models (LLMs) have demonstrated in-context generalization capabilities: performing downstream tasks simply by conditioning on few in-context exemplars or plain natural language task descriptions (Brown et al., 2020; Sun et al., 2021). Despite these advancements, even the largest LLMs have been found to struggle with complex tasks which require multiple reasoning steps (Rae et al., 2021).

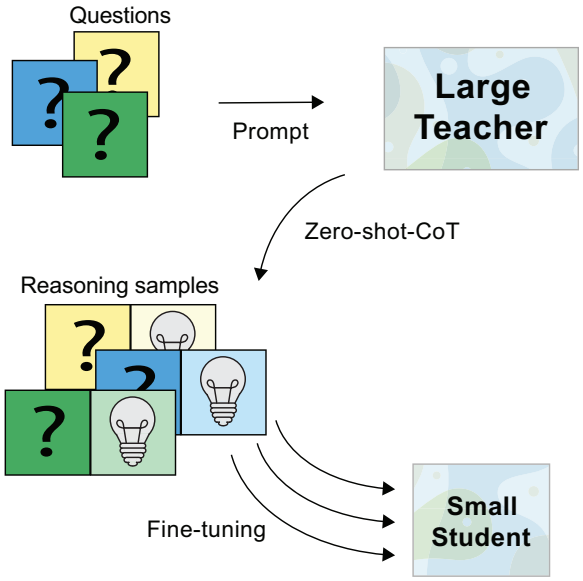


Figure 1: **Fine-tune-CoT uses teacher-generated reasoning to teach students**. We prompt a very large teacher model, such as GPT-3 175B, to solve complex questions via zero-shot chain-of-thought reasoning. We then use the reasoning samples to fine-tune a much smaller student model. See Figure 2 for details.

To solve complex tasks, recent works show that it is possible to elicit reasoning abilities by prompting LLMs to perform *chain-of-thought* (CoT) reasoning, i.e., generate a series of intermediate reasoning steps. This can be achieved by providing CoT demonstrations as exemplars in prompting (Wei et al., 2022b). More recently, Kojima et al. (2022) found that LLMs can be prompted to perform CoT reasoning simply by providing a natural language instruction to *think step-by-step*.

A major drawback of prompt-based CoT reasoning methods, however, is their reliance on extremely large models that span *hundreds of billions* of parameters (Wei et al., 2022b; Kojima et al., 2022). These models are prohibitive to deploy at scale due to overwhelming computational requirements and inference costs (Wei et al., 2022b).

[1]Our code implementation and data are available at https://github.com/itsnamgyu/reasoning-teacher.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 14852–14882
July 9-14, 2023 ©2023 Association for Computational Linguistics

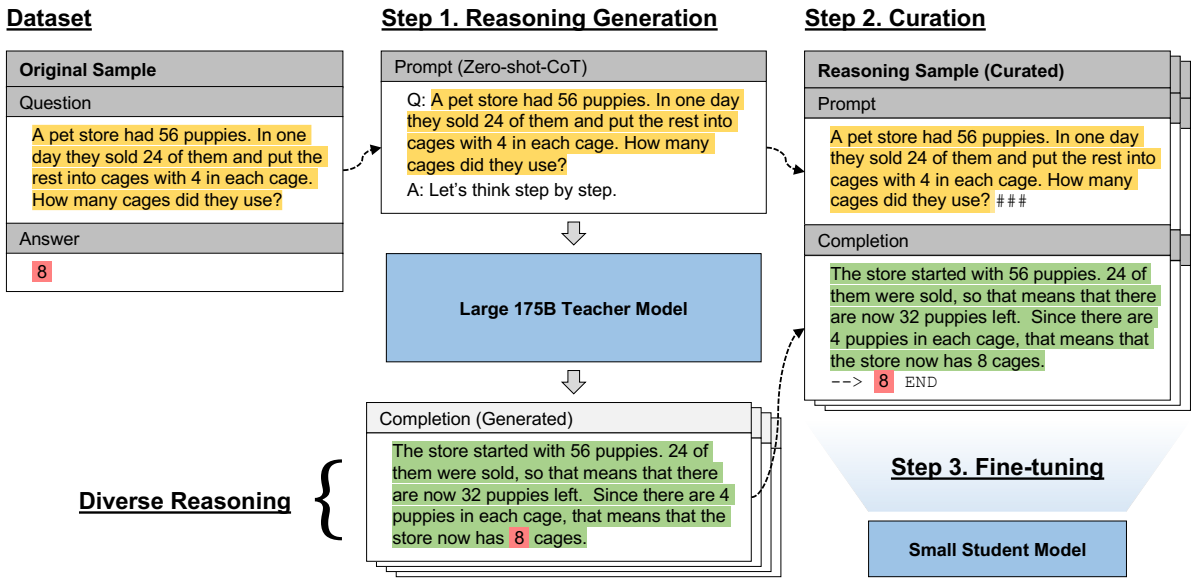


Figure 2: Detailed overview of our proposed Fine-tune-CoT method. **Step 1**: a very large teacher model is prompted to solve complex questions (yellow) by generating multi-step reasoning explanations (green). **Step 2**: completions are filtered based on the correctness of the final prediction (red). The question, rationale, and answer are used to compose a *reasoning sample* comprised of the prompt and a multi-step solution. **Step 3**: the curated reasoning samples are used to fine-tune a small, lightweight student to exhibit reasoning capabilities. The application of an *LM-based* teacher enables **diverse reasoning**—generating multiple distinct rationales for each original sample to enrich the fine-tuning data. This boosts the performance of student models without any additional human annotation.

Therefore, we strive to enable complex reasoning in small models which are more feasible for large-scale deployment.

In this light, we propose an approach named *Fine-tune-CoT*, which utilizes the reasoning capabilities of very large LMs to teach small models how to solve complex tasks. We apply existing zero-shot CoT prompting (Kojima et al., 2022) to generate rationales from very large *teacher* models, and use them to fine-tune smaller *student* models[2]. We illustrate this in Figure 2. We note that standard fine-tuning *without rationales* has been shown to be inadequate for solving reasoning tasks with small models (Talmor et al., 2018). While there have been attempts to fine-tune small models with *hand-annotated* reasoning steps (Nye et al., 2021; Cobbe et al., 2021), they often require task-specific training setups and high-quality rationales which are costly to annotate (Wei et al., 2022b). In contrast, our approach can be readily applied to novel downstream tasks without hand-crafted reasoning or task engineering.

We also propose a novel extension to our method, termed *diverse reasoning*, to maximize the teaching effects of Fine-tune-CoT. Inspired by the intuition that complex tasks can have multiple solutions with distinct reasoning paths (Evans, 2010), we generate multiple reasoning solutions from teacher models using stochastic sampling to augment the training data for student models[3]. We find that this is a simple yet highly effective approach to maximizing student performance, which has not been explicitly recognized in concurrent works on fine-tuning with CoT reasoning (Huang et al., 2022; Li et al., 2022b; Magister et al., 2022; Fu et al., 2023).

We evaluate our method on 12 tasks using a wide range of publicly available models. We find that Fine-tune-CoT can elicit notable reasoning performance in small models while preserving much of the versatility of prompt-based CoT reasoning, which previously required >100B parameter models (Wei et al., 2022b). Diverse reasoning enables remarkable gains in performance at the minor cost of additional teacher inference at development time, by exploiting our unique learning setup. This enables models as small as 0.3B to outperform larger students, and even the 175B teacher model in some tasks. Our ablations show that performance is con-

[2]This can be interpreted as a variant of knowledge distillation (Hinton et al., 2015).

[3]Diverse reasoning is orthogonal to existing data augmentation techniques (Yoo et al., 2021; Meng et al., 2022) which aim to augment new *question-answer* pairs rather than diverse reasoning *solutions* for complex questions.

sistently scalable across all axes considered: diverse reasoning, dataset size, teacher performance, and student model size. This shows the potential of our method to enable reliable performance in small models that are feasible for use in real-world applications. Lastly, we conduct thorough sample studies and analyses which shed light on crucial details previous overlooked in fine-tuning for CoT and provide intuition on the emergence of reasoning abilities in small models.

2 Related Work

Downstream transfer in language models Much previous work established a "pre-train and fine-tune" paradigm for enhancing LLM performance on downstream tasks (Radford et al., 2018; Dong et al., 2019; Vaswani et al., 2017; Devlin et al., 2018). However, fine-tuning is not always easily applicable (Hendrycks et al., 2020). More recent literature exhibits a paradigm shift towards "prompting" the model to predict the desired output (Liu et al., 2021; Raffel et al., 2020). Large LMs can exhibit strong performance in this setting (Brown et al., 2020). For smaller models to be able to perform similarly, additional engineering is usually required (Gao et al., 2021; Schick and Schütze, 2021b; Schick et al., 2020). For more complex tasks, the idea of using samples with explicit reasoning steps for fine-tuning a model (Nye et al., 2021; Cobbe et al., 2021) preceded the approach of chain-of-thought (CoT) prompting (Wei et al., 2022b), which enables very large LMs to perform well.

Chain-of-thought reasoning In few-shot CoT prompting, the model learns to generate intermediate reasoning steps that lead to a problem solution, after being fed examples of step-by-step reasoning. This enables very good performance on a wide range of tasks. (Wang et al., 2022). Additionally, LLMs can perform well in an unsupervised task-agnostic setting, using Zero-shot-CoT (Kojima et al., 2022). This requires no fine-tuning or task specific conditioning, and substantially outperforms standard zero-shot learning and sometimes even few-shot learning on a wide number of tasks.

Yet, prior work has shown that CoT requires extremely large models for optimal performance (Hoffmann et al., 2022; Chowdhery et al., 2022). In our work, we contrast this by showing how to utilize CoT reasoning methods for smaller models by fine-tuning them on rationales generated by a very large model. Using various LLM-generated explanations for fine-tuning smaller models has been successfully used in prior work (Li et al., 2022a), with a focus on specific single tasks. Also, a similar approach to ours is mentioned in (Huang et al., 2022); however we note that this concurrent work focuses on using Few-shot-CoT to self-generate fine-tuning examples by and for very large proprietary models. There is a brief glimpse into fine-tuning on smaller distilled models, but the results are limited to one dataset and very large teacher models that are inaccessible to the general community. In contrast, we provide a rich set of results and qualitative/quantitative analysis on a wide range of datasets, using open-source models that are small and accessible to everyone.

Knowledge distillation Typically, knowledge distillation (KD) refers to training small models derived from large models in order to reduce model size and latency, while still preserving accuracy and capacity to generalize (Hinton et al., 2015; Sanh et al., 2019). Essentially, KD is a form of model compression, making efficient deployment to capacity-limited devices possible (Bucilua et al., 2006). We note that our work could also be considered a distant variant of KD (Gou et al., 2021), similar to works on improving prompt-based methods such as Yoo et al. (2021); Schick and Schütze (2021b,a); Zelikman et al. (2022), or works on data-free distillation (Micaelli and Storkey, 2019; Nayak et al., 2019; Shen et al., 2021), where the transfer data is synthetically generated from a large teacher model. Similarly, sequence-level distillation, i.e. training a student model on sequence distributions of a larger teacher, can make neural machine translation more efficient (Kim and Rush, 2016). Despite being similar in spirit, our method still distinguishes itself from such previous work. The role of the teacher model in our method is to teach the notion of intermediate reasoning. It is not the specific output that is the main supervising signal for reasoning, but rather the generation's structure. Hence, we do not use a standard KD loss function that reflects trying to match the teacher output. Adding to this, we note that our diverse reasoning is also unusual in the context of KD, where it is e.g. sufficient in practice to only generate one teacher sequence for sequence level distillation.

3 Chain-of-Thought Fine-Tuning

We propose Fine-tune-CoT, a task-agnostic approach to enable chain-of-thought reasoning in small LMs. The core idea is to generate reasoning samples from very large teacher models using CoT prompting and subsequently fine-tune small student models using the generated samples. This approach preserves the versatility of prompt-based CoT methods while overcoming their reliance on prohibitively large models. To maximize versatility and minimize teacher inference costs, we use the task-agnostic Zero-shot-CoT prompting method (Kojima et al., 2022) on teacher models, as it does not require *any* reasoning examples or long inference context. We discuss our choice of teacher CoT prompting method in Section 7.3. In the following, we characterize Fine-tune-CoT in three distinct steps. We also provide a visual overview in Figure 2.

Step 1. Reasoning generation First, we utilize a large teacher model to generate CoT reasoning explanations for a given task. Consider a standard sample S_i consisting of a question q_i and its true answer a_i. Using Zero-shot-CoT[4]. we prompt the teacher model to generate a reasoning explanation, or rationale, $\hat{r}_i$ to solve question q_i and make a final answer prediction $\hat{a}_i$. The resulting text sequence, including the prompt and generations, takes the following form: "Q: <q_i>. A: Let's think step by step. <$\hat{r}_i$> Therefore, the answer is <$\hat{a}_i$>".

Step 2. Curation Next, we filter the generated samples and reformat them into prompt-completion pairs. For filtering, we simply compare the final prediction of the teacher model $\hat{a}_i$ with the ground-truth answer a_i, following previous works (Zelikman et al., 2022; Huang et al., 2022). Note that this filtering incurs some loss of training samples. For all instances i where $\hat{a}_i = a_i$, we repackage $(S_i, \hat{r}_i, \hat{a}_i)$ into a *reasoning sample* $S'_i = (p_i, c_i)$, a prompt-completion pair. To maximize inference-time efficiency, we use special-character based delimiters to minimize token usage. Specifically, p_i and c_i each take the form of "<q_i> ###" and "<$\hat{r}_i$> --> <a_i> END". We note that answer-based filtering does not ensure the correctness of the rationales, especially for multi-choice questions. We provide an analysis in Appendix E.1 regarding this important detail which has not been addressed in concurrent work.

Step 3. Fine-tune Finally, we fine-tune a small pre-trained student model on the assembled reasoning samples. We use the same training objective of that used during pre-training, i.e., autoregressive language modeling objective, or next-token prediction (Radford et al., 2018).

Diverse reasoning To maximize the teaching effects of Fine-tune-CoT, we can generate multiple reasoning explanations for each training sample. This approach is motivated by the intuition that multiple reasoning paths can be used to solve complex tasks, i.e., *type-2* tasks (Evans, 2010). We posit that this unique feature of complex tasks, in tandem with the stochastic generation abilities of the teacher model, can enable diverse reasoning to significantly boost reasoning supervision simply through additional teacher inference. In detail, for a given sample S_i, instead of applying Zero-shot-CoT using greedy decoding to obtain a single explanation-answer pair $(\hat{e}_i, \hat{a}_i)$, we use a stochastic sampling strategy, i.e., temperature sampling with large T, to obtain D distinct generations $\{(\hat{r}_{ij}, \hat{a}_{ij})\}_j^D$. Subsequent reasoning sample curation and fine-tuning then proceed as before. We refer to D as the *degree of reasoning diversity*. A similar approach is used in Wang et al. (2022); Huang et al. (2022), where multiple CoT outputs are generated and marginalized to find the optimal answer. However, the effects of such diverse reasoning on teaching student models has not been acknowledged or thoroughly investigated in concurrent work (Huang et al., 2022; Li et al., 2022a; Magister et al., 2022; Fu et al., 2023). We note that diverse reasoning imposes an important trade-off between the development cost and inference cost/quality of student models which we discuss in Section 5.3.

4 Experiments

Tasks and datasets We evaluate our method on 12 datasets pertaining to four categories of complex reasoning, following Kojima et al. (2022). These include arithmetic (SingleEq, AddSub, MultiArith, GSM8K, SVAMP), other (Date Understanding, Tracking Shuffled Objects), symbolic (Last Letter Concatenation, Coin Flip), and common sense (CommonSenseQA, StrategyQA) reasoning. We provide details and references in Appendix B.

[4]Note that Zero-shot-CoT is itself a two-step prompting method. The reasoning (blue) is generated in the first step and answer prediction (red) is generated in the second step.

Method	Params	Single Eq	Add Sub	Multi Arith	GSM8K	Aqua	SVAMP	Date Understanding	Shuffled Objects	Last Letter	Coin Flip	Common SenseQA	Strategy QA
Random		0.00	0.00	0.00	0.00	20.00	0.00	17.12	33.33	0.00	50.00	20.00	50.00
Teacher: InstructGPT (`text-davinci-002`)													
Zero-shot-CoT	175B	82.24	78.99	78.89	40.26	34.25	64.67	73.87	50.22	56.00	92.67	61.75	53.57
Student: GPT-3 (`ada`, `babbage`, `curie`)													
Zero-shot	6.7B	0.66	0.84	3.33	1.74	16.54	2.67	9.91	32.89	0.00	56.67	20.23	52.98
Zero-shot-CoT	6.7B	1.32	2.52	5.00	2.35	21.26	1.33	15.32	31.11	0.00	46.67	19.98	51.09
Few-shot-CoT	6.7B	22.37	**31.93**	10.00	2.50	15.75	11.33	12.84	-	0.67	40.00	24.73	54.68
Fine-tune	6.7B	**24.34**	25.21	15.00	6.14	15.35	20.67	14.41	33.78	32.67	72.00	**76.17**	**65.21**
Fine-tune-CoT	0.3B	7.24	6.72	6.11	3.11	23.62	5.00	17.12	49.33	50.67	99.33	32.68	52.55
	1.3B	11.18	11.76	13.33	4.70	19.69	8.00	38.74	52.44	50.67	100.00	43.08	52.69
	6.7B	20.39	21.01	33.33	**6.75**	**24.02**	12.67	60.36	64.44	52.67	98.67	56.76	55.02
Fine-tune-CoT	0.3B	9.21	10.08	23.89	-	-	14.33	58.56	61.78	59.33	99.33	-	57.21
w/ diverse reasoning	1.3B	18.42	19.33	27.78	-	-	16.33	70.27	72.00	60.67	100.00	-	57.06
	6.7B	**24.34**	31.09	**53.33**	-	-	**30.33**	**83.78**	**73.33**	**62.00**	**100.00**	-	58.22

Table 1: **Fine-tune-CoT Performance.** Accuracy (%) of OpenAI models on 12 tasks under Fine-tune-CoT (with diverse reasoning) and baseline methods. 'Random' refers to random-guess performance derived based on the number of choices in multi-choice tasks. For diverse reasoning, we report results for maximum degree D considered: $D = 64$ for MultiArith and SVAMP; $D = 8$ for other datasets. We omit diverse reasoning for large datasets due to resource constraints and Few-shot-CoT for Tracking Shuffled Objects due to absence of prompts.

Models For teacher models, we use four variants of GPT-3 175B (Brown et al., 2020), provided by the OpenAI API. Unless otherwise stated, we use `text-davinci-002` based on InstructGPT 175B (Ouyang et al., 2022) as the teacher for Fine-tune-CoT. For student models, we consider four popular model families. For our main experiments, we use GPT-3 {`ada`, `babbage`, `curie`} as they are readily available for fine-tuning via the OpenAI API. Due to the blackbox nature of the API, we also consider various open-source models under controlled settings. We use GPT-2 {Small, Medium, Large} (Radford et al., 2019) and T5-{Small, Base, Large} (Raffel et al., 2020) as representative model families for decoder-only and encoder-decoder architectures, respectively. We also use the instruction-tuned version of T5, Flan-T5-{Small, Base, Large} (Chung et al., 2022), to investigate the effects of instruction tuning on student models, prior to applying Fine-tune-CoT. These student models are 25–2500x smaller than the teacher model, thus considerably more feasible for real-world deployment. We provide details on models and API usage in Appendix C.

Baseline methods We provide a comparison of Fine-tune-CoT (ours) with four baseline methods: standard zero-shot prompting, vanilla fine-tuning, Zero-shot-CoT (Kojima et al., 2022), and Few-shot-CoT (Wei et al., 2022b). Given a training sample $\{(q_i, a_i)\}_i$, we use a simple format "Q: <q_i>" for zero-shot prompting. For vanilla fine-tuning, we format the prompt and completion as "<q_i> ###" and "<a_i> `END`", respectively. We clarify the taxonomy of methods in Table 2. For text generation, we use greedy decoding following Wei et al. (2022b); Kojima et al. (2022) throughout our experiments, except for diverse reasoning. For diverse reasoning on the teacher, we use temperature sampling with $T = 0.7$, following Wang et al. (2022). We provide experimental details in Appendix A.

Method	Model Updates	CoT Output	Sample Utilization	Teacher Usage	Reference
Zero-shot	✗	✗	✗	✗	(Radford et al., 2019)
Zero-shot-CoT	✗	✓	✗	✗	(Kojima et al., 2022)
Few-shot-CoT	✗	✓	△	✗	(Wei et al., 2022b)
Fine-tune	✓	✗	✓	✗	(Radford et al., 2018)
Fine-tune-CoT	✓	✓	✓	✓	Ours

Table 2: **Taxonomy of methods**. CoT methods are more interpretable due to reasoning output. While Few-shot-CoT can utilize few in-context samples, fine-tuning can utilize any number of training samples via model updates. Fine-tune-CoT benefits from the reasoning capabilities of teacher models.

4.1 Results

In this section, we present the reasoning performance of models using Fine-tune-CoT and diverse reasoning. We compare with various baselines and demonstrate the scalability of our method across four axes: degree of diverse reasoning (Figure 3), dataset size (Figure 4), performance of the teacher (Figure 5), and size of the student model (Figure 6).

We present our findings on GPT-3 models in the main text and defer results on open-source models to Appendix G, with a brief summary at the end of this section.

Fine-tune-CoT elicits complex reasoning in small models Table 1 summarizes the accuracy of student models using the proposed Fine-tune-CoT, compared to prompt-based CoT baselines as well as standard fine-tuning. While Zero-shot-CoT exhibits remarkable performance on the very large 175B model (Kojima et al., 2022), it fails to enable complex reasoning in all three smaller models, showing near-negligible performance across *all* tasks. We also find that small models are unable to approach these tasks under standard zero-shot prompting. On the other hand, Fine-tune-CoT elicits notable reasoning performance, demonstrating significant gains over Zero-shot-CoT when using smaller models and outperforming both fine-tuning and Few-shot-CoT in more than half of the tasks. For complex arithmetic, Fine-tune-CoT achieves a notable 33% accuracy on MultiArith while Zero-shot-CoT only reaches 5%. Few-shot-CoT and fine-tuning only achieve 10% and 15%, respectively. For two commonsense reasoning tasks, our method outperforms the near-random performance of Zero-shot-CoT by 37% and 5%, respectively. Furthermore, it surpasses Few-shot-CoT on CommonSenseQA by 32% and performs similarly on StrategyQA. We observe that Fine-tune-CoT performance is most notable for tasks that are not overly complex, which include other reasoning tasks (Date Understanding, Shuffled Objects) and symbolic reasoning (Last Letter, Coin Flip), significantly outperforming other baselines. See Appendix Table 9 for performance of all students.

Small models can outperform very large teachers in reasoning Table 1 also shows that Fine-tune-CoT is highly effective on small models compared to the large 175B teacher model. For the tasks Shuffled Objects and Coin Flip, Fine-tune-CoT is shown to outperform the teacher model using either 1.3B or 6.7B parameters, i.e., reducing the number of required parameters by approx. 25–100x. We also find that Fine-tune-CoT with the very small 0.3B model consistently outperforms the 6.7B model under Zero-shot-CoT, demonstrating that our method is able to unlock a wider range of capabilities compared to the baseline, even when model size is vastly reduced.

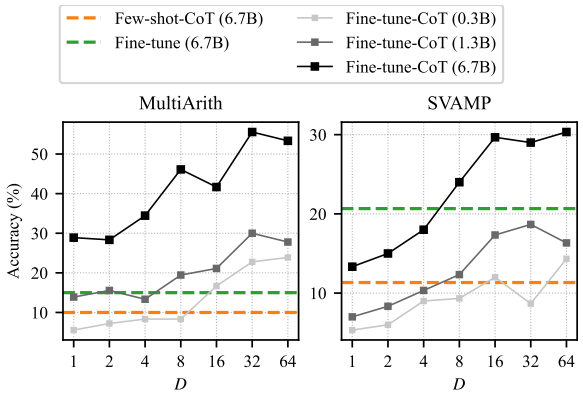


Figure 3: **Diverse reasoning performance.** Accuracy (%) of GPT-3 student models under Fine-tune-CoT with varying degrees of diverse reasoning D. Baseline performance of the *largest model* under vanilla fine-tuning and Few-shot-CoT are shown for comparison. Diverse reasoning is not applicable to the baselines.

Diverse reasoning substantially improves Fine-tune-CoT performance. To examine the learning effects of diverse reasoning and compare it with two baselines given by fine-tuning and Few-shot-CoT, we apply Fine-tune-CoT using 1–64 reasoning explanations per sample across three model scales on MultiArith and SVAMP[5]. Figure 3 shows that diverse reasoning can significantly improve the performance of student models using Fine-tune-CoT. For the 6.7B student model, we find a boost of around 26% on MultiArith, and around 17% on SVAMP. We also note that using diverse reasoning always leads to outperforming the baseline within the respective model size, and can even boost performance of our method beyond that of a larger model that does not use diverse reasoning. This even includes the teacher in two cases (Date Understanding, Last Letter). Moreover, we find that diverse reasoning can boost the performance of Fine-tune-CoT to surpass that of both Few-shot-CoT and vanilla fine-tuning across *all* model sizes. We posit that due to our focus on *complex tasks*, the diversity of reasoning paths and linguistic templates can substantially aid in teaching student models to reason.

Fine-tune-CoT consistently benefits from more data. We perform an ablation on dataset size to study the performance scalability of our method with dataset size. We see that the performance of

[5]For diverse reasoning, we generate teacher rationales stochastically with $T = 0.7$ instead of greedy decoding, which accounts for small differences in absolute performance numbers between Table 1 and diverse reasoning with $D = 1$.

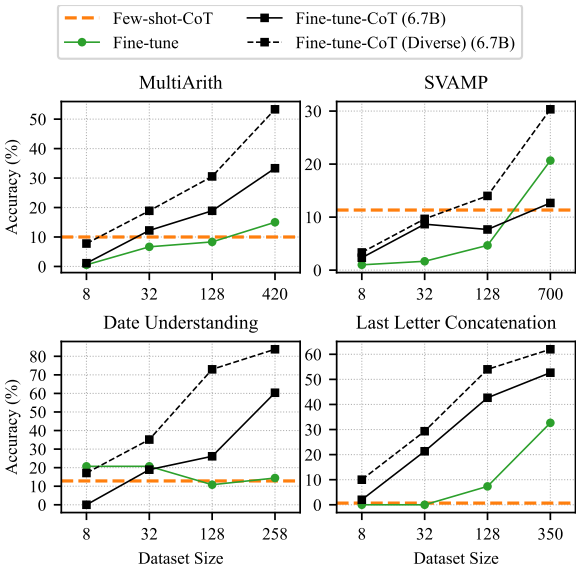


Figure 4: **Effects of dataset size.** Accuracy (%) of the GPT-3 6.7B student model by dataset size under vanilla fine-tuning vs Fine-tune-CoT (with diverse reasoning). Baseline performance under Few-shot-CoT is shown for comparison. Diverse reasoning is not applicable to standard fine-tuning. We show diverse reasoning performance with $D = 64$ for MultiArith and SVAMP; $D = 8$ for others.

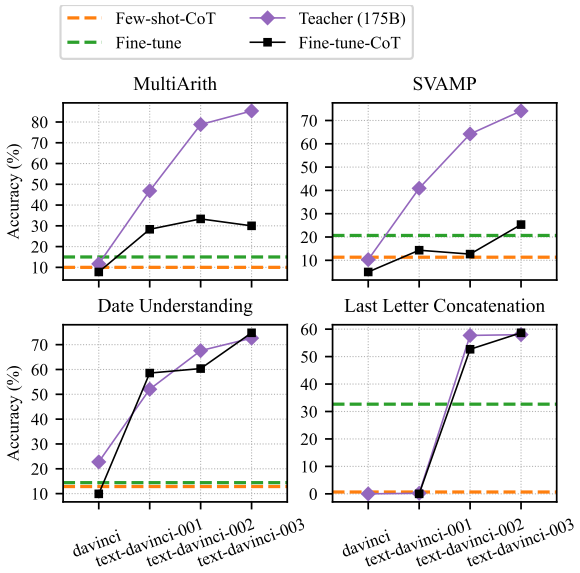


Figure 5: **Effects of teacher performance on students.** Accuracy (%) of teacher models (Zero-shot-CoT) and their corresponding GPT-3 6.7B student models (Fine-tune-CoT). Baseline performance under vanilla fine-tuning and Few-shot-CoT are shown for comparison. Teacher models are not applicable to Few-shot-CoT which uses few human-annotated examples.

the 6.7B model clearly scales with the size of the dataset, independent of the task. In comparison, vanilla fine-tuning does not always exhibit this behavior. In fact, for Date Understanding, we find that an increase in dataset size harms the performance of fine-tuning. Furthermore, Fine-tune-CoT sees additional benefits from diverse reasoning, which is not applicable in standard fine-tuning.

Better reasoners are better teachers Next, we can ask the question of whether the performance of the teacher is correlated with that of their student when using Fine-tune-CoT. To test this, we use different versions of GPT-3 as teacher models, keeping the size of the student model constant at 6.7B parameters (Figure 5). We find that student performance indeed scales with teacher performance, particularly in the less complex tasks Date Understanding and Last Letter. There, the performance of the student matches the performance of the teacher very closely. This also fits with our observations in Appendix D, which show that the successes and failures of teachers are correlated with those of the students. We note that this scaling effect is in contrast not a given in knowledge distillation, where more accurate teachers do not always result in better students (Menon et al., 2021).

Fine-tune-CoT performance scales with model size for small LMs Finally, we explore the effect of scaling up student model size on our method, and compare it with the effects of increasingly larger student models in Few-shot-CoT as well as vanilla fine-tuning. We can observe that the performance of Fine-tune-CoT is consistently scalable with student size (Figure 6). In contrast, the two baselines do not always exhibit the same behavior: in Date Understanding, neither Few-shot-CoT nor vanilla fine-tuning results in scalable performance.

Results on open-source student models Overall, our findings on T5, Flan-T5, and GPT-2 show similar trends to those observed on GPT-3. Small models exhibit near-random performance under standard zero-shot or CoT prompting in nearly all cases. Notable, we find that encoder-decoder models, i.e., T5 and Flan-T5, show noteworthy performance under standard fine-tuning, suggesting that causal masking may be a bottleneck to reasoning in decoder-based language models in the absence of CoT output. Fine-tune-CoT consistently outperforms prompt-based baselines and is comparable or superior to vanilla fine-tuning. Diverse reasoning improves performance even further, often exhibiting significant gains. We report our full findings on open-source models in Appendix G.

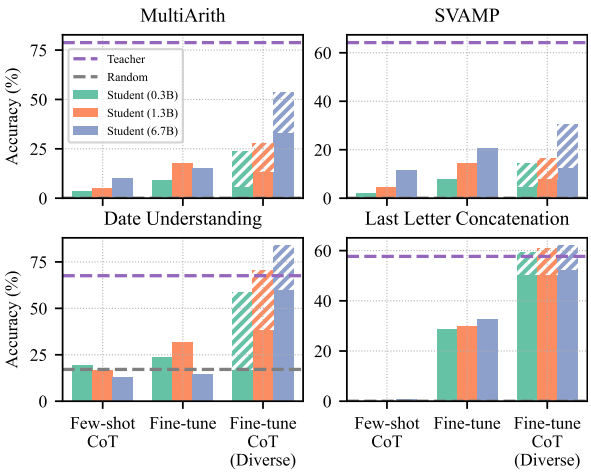


Figure 6: **Effects of student model scale.** Accuracy (%) of GPT-3 student models of various sizes under Few-shot-CoT, vanilla fine-tuning, and Fine-tune-CoT (with diverse reasoning). The hatched portion indicates the performance boost of Fine-tune-CoT when using diverse reasoning with $D = 64$ on MultiArith and SVAMP; $D = 8$ on others.

4.2 Analysis

Sample study To identify the strengths and weaknesses of our method, we perform a thorough sample study across all datasets and methods. Across all arithmetic tasks, we find that a large portion of errors arises from calculations. MultiArith and SVAMP also show many semantic errors, but these are significantly reduced with diverse reasoning. For difficult tasks such as GSM8K and AQUA, we found that all methods tend to struggle. We found that our method is highly effective in text-based tasks, excluding commonsense reasoning, as well as tasks that contain common linguistic patterns. On the other hand, we find that students under Zero-shot-CoT often repeat questions or produce incoherent repetitive statements. While Few-shot-CoT elicits step-by-step sentences, the student models rarely seem to understand the semantics of the question, and generations often contain logical or commonsense errors. For details on our sample study, see Appendix D.

Nuances of fine-tuning on CoT reasoning We shed light on nuances that have often been overlooked in previous or concurrent work (Wei et al., 2022b; Li et al., 2022a; Magister et al., 2022). First, we acknowledge the possibility that *correct* samples may contain incorrect reasoning. In fact, we find that 27.6% of *correct* teacher completions for Date Understanding contained reasoning errors. However, ablations on rationale filtering suggest that these incorrect rationales can aid in student supervision (Appendix E.1). Secondly, we find that common maximum sequence lengths used for CoT generations often lead to incomplete answers. We observe that reasoning length differs among datasets, and longer generations typically improve accuracy, but may not be beneficial for fine-tuning (Appendix E.2). Lastly, we find that many datasets are comprised of samples that share common templates, potentially compromising the validity of our random train-test splits. To address this, we evaluate our method on manual template-wise data splits, and confirm that students retain meaningful reasoning capabilities (Appendix E.3).

5 Discussion

5.1 Accessibility of Fine-tune-CoT

Owing to the versatility of the teacher generation method, i.e., Zero-shot-CoT, our method can be readily applied to any complex task without task-specific engineering. Rationales can be readily generated using publicly available APIs such as those provided by OpenAI or Anthropic. This makes it viable to obtain CoT training data in low-resource scenarios, which not only outperforms standard fine-tuning, but elicits the student to output interpretable explanations. Fine-tuning and inference on student models can also be performed on much more accessible hardware, in contrast to very large models. This can reduce long-term inference costs and minimize environmental impact while making our method fully accessible to a wide community.

5.2 Viability of Fine-tune-CoT

While Fine-tune-CoT elicits notable complex reasoning capabilities in small models, performance on some difficult datasets would not be considered viable for real-world use, such as 30.33% on SVAMP. However, our findings in Section 4.1 indicates significant potential for improvement, as our method is shown to be uniquely scalable with (1) diverse reasoning, (2) dataset size, (3) teacher model performance, and (4) student model size. The use of diverse reasoning and better teacher models is especially promising, as these can benefit from improved teacher LLM performance and inference costs in the future. In addition, it is possible to incorporate recent CoT methods, which lead to significant performance improvements, in student models, which we discuss in Section 7.3.

5.3 Tradeoffs of Fine-tune-CoT

The aforementioned opportunities to enhance Fine-tune-CoT also pose many important tradeoffs. We leave further analysis to future work.

Degree of diverse reasoning The performance benefits of diverse reasoning come at the cost of additional teacher inference. Therefore, diverse reasoning poses a tradeoff between development cost vs inference cost/quality. In other words, performance gains from diverse reasoning may be utilized to enhance student performance or alleviate the need for larger student models. This must also be taken into account for fair evaluation of similar distillation methods in the future.

Data acquisition Data annotation and diverse reasoning can both be used to enlarge fine-tuning data, but each have their associated costs. We note that the cost of diverse reasoning is linear to the number of generated rationale *and* the number of original samples. Despite this, it can still be a cost-effective alternative to hand-annotating additional data. A preliminary cost analysis in Appendix F shows that the pareto front of data-acquisition-cost to performance always incorporates diverse reasoning. We expect that the cost benefits of diverse reasoning will continue to improve with improvements in teacher model performance and efficiency.

5.4 Emergence of CoT reasoning

The emergence of abilities such as CoT reasoning has become a point of interest in recent works (Wei et al., 2022b,a; Schaeffer et al., 2023). We note that the efficacy of Fine-tune-CoT on small models does not disprove this emergence, as our method is based on fine-tuning. However, we believe our results can provide some insight into this phenomena.

Why does Fine-tune-CoT work in small models? In a seminal work, Wei et al. (2022b) suggests that CoT reasoning is an emergent ability of scale—more specifically, a complicated phenomena involving a variety of emergent abilities, such as semantic understanding, symbol mapping, arithmetic ability. However, our sample studies suggest that Fine-tune-CoT elicits these *emergent* abilities even in relatively small models (see Appendix D). We explain this from two perspectives. First, Wei et al. (2022b) demonstrated the emergence of reasoning abilties by identifying a reduction in the frequency of reasoning errors with larger model scale. Similarly, we find that more potent forms of supervision also lead to a *gradual* reduction in reasoning errors. For example, we found a clear distinction between Zero-, Few-shot-CoT and Fine-tune-CoT (with diverse reasoning) in the frequency and severity of semantic errors, i.e., understanding complex questions, and calculation errors. This suggests that explicit supervision on reasoning can also lead to the emergence of reasoning abilities. Second, we qualitatively find that students show capabilities that are reminiscent of the larger teacher model. We found that students can recognize common semantics and reasoning cues of the given task, and is able to imitate the process of splitting large tasks into subtasks. This suggests that it is possible to learn reasoning abilities pertaining to a particular domain. We posit that this is possible in small models due to the limited domain of reasoning, and may not be applicable in reasoning tasks that require large domains of knowledge.

Distillation of emergent abilities Chain-of-thought reasoning has been recognized as a prime example of emergent abilities in very large language models (Wei et al., 2022a). Our findings show that it is possible to distill this ability, under certain domains, to much smaller models simply through fine-tuning. The potential for distillation implies that future advancements in language models may lead to emergent abilities that are not only pertinent to those larger models, but could also have a broader impact, cascading benefits to smaller models.

6 Conclusion

We have proposed Fine-tune-CoT, a method that uses LLMs as *reasoning teachers* to transfer the broad reasoning capabilities previously found in $>$100B models to student models as small as 0.3B. We propose diverse reasoning as a novel approach to maximize these teaching effects, exploiting the unique characteristics of this new learning setup to *vastly* improve performance. Our extensive experiments show that Fine-tune-CoT elicits significant reasoning performance in small models, thus demonstrating the distillation of CoT reasoning which has been considered an *emergent* ability of scale. By leveraging publicly available models with zero-shot prompting, we demonstrate a task-agnostic approach to elicit reasoning performance in small models, making complex reasoning feasible for real-world deployment and accessible to the broader community.

7 Limitations

7.1 Towards concise answers

Sample studies show that rationales output from student models may occasionally be repetitive and digressive. This is undesirable in terms of inference-time efficiency as well as interpretability. As a minor optimization to inference computation, we construct our fine-tuning sample templates using special-character based delimiters instead of natural language used in concurrent work (Huang et al., 2022) to minimize sequence length. Preliminary findings showed this had no significant impact on reasoning performance. More importantly, it is desirable to train student models to generate concise answers in terms of substance. Appendix E.2 hints at the possibility for this, showing that fine-tuning on shorter reasoning samples causes the student model to also produce shorter rationales.

7.2 Exploring a wider array of models

We note that the performance of our method is currently not state-of-the-art. However, it can benefit from advances in teacher models as well as other prompting methods. For example, future work should include a wider array of teachers, such as the highly versatile ChatGPT, which typically generates detailed long responses that may be able to impart more knowledge to the student. More recent models such as GPT-4 have demonstrated significant advances in complex reasoning abilities, which may improve the efficacy of Fine-tune-CoT on very difficult datasets, such as GSM8K. Conversely, our method could prove even more advantageous when applied to recent models with improved efficiency, such as those based on the recent LLaMA model (Touvron et al., 2023), which has sparked a proliferation of work focused on compact language models. Both of these avenues are promising for future work.

7.3 Better CoT inference methods

The use of diverse reasoning and better teacher or student models is especially promising, as it is possible to leverage future improvements in model performance and decreased inference costs. However, we can also consider other ways to boost performance, such as using different prompting methods. For example, previous work shows that Few-shot-CoT (Wei et al., 2022b) can improve accuracy over Zero-shot-CoT by a wide margin, e.g., going from 78.7% to 93.0% on MultiArith (Kojima et al., 2022). However, our choice to use Zero-shot-CoT to generate reasoning samples from the teacher model is motivated by the fact that Few-shot-CoT requires a significantly larger inference context. With the current pricing models based on token usage, the typical setup of 8-shot CoT would cost approximately 8 times more compared to Zero-shot-CoT. Therefore, we see a tradeoff between using the inference budget for Few-shot-CoT and using it for diverse reasoning with Zero-shot-CoT. On the other hand, we also note that recent works introduce various ways to improve CoT reasoning performance substantially (often to near-perfect levels), which can be applied to our student models. These include refinement over repeated inferences (Wang et al., 2022; Li et al., 2022b) and self-improvement (Zelikman et al., 2022; Huang et al., 2022). In particular, self-consistency (Wang et al., 2022) can be utilizied on unlabeled samples to maximize the teaching signal. In contrast, we aim to achieve CoT reasoning without the inference time cost incurred by very large LMs. Future work is needed to incorporate these methods into Fine-tune-CoT while minimizing development and inference costs.

7.4 Connection with knowledge distillation

We assume that there is a lot of potential in strengthening the connections between knowledge distillation and our method. We have already seen in this work that our method shares some characteristics with KD, such as the fact that the knowledge of intermediate reasoning imparted by using also incorrect samples can have positive effects on student accuracy, akin to "dark knowledge" (Menon et al., 2021) that is transferred by training on teacher output logits and not one-hot labels. We have seen that this leads to a quantity-quality tradeoff when it comes to the ability of the student model to generalize: having fewer but perfectly curated reasoning samples is not necessarily as helpful as having a larger amount of reasoning samples that might not always be fully correct. On the other hand, we have also found that more accurate teachers do lead to more accurate students, which is not always the case in KD (Müller et al., 2019). It would therefore be of interest for future work to formalize the connection of Fine-tune-CoT with classic KD methods, and potentially test the use of a different distillation loss function that takes the teacher's actual output into account.

8 Ethics Statement

Our work presents various challenges and opportunities in terms of bias and toxicity in language models. It is widely known that LLMs trained on large corpora have been shown to capture biases found in the training data (Brown et al., 2020; Chowdhery et al., 2022). Since our student models are trained on reasoning samples generated by these LLMs, it is possible that such characteristics of the teacher model can get passed along to the student. This is an important point to consider when selecting the teacher model for our method.

Our training setup, however, does offer a unique opportunity to minimize bias and toxicity in student models, by influencing the samples used for fine-tuning. One approach would be to augment the curating step of Fine-tune-CoT to filter out biased or toxic samples. It is possible to automate this via neural network-based verifiers, previously used to filter correct output (Cobbe et al., 2021; Li et al., 2022b). Alternatively, one may consider optimizing the CoT prompts to minimize bias and toxicity in teacher-generated rationales.

We note that bad actors can also potentially take advantage of our method to utilize complex reasoning for malicious purposes and deploy it at scale, using small models. This highlights the importance of safeguarding the potential capabilities of LLMs by major providers. To prevent the distillation of malicious reasoning abilities in small (or large) students, future work in identifying usage patterns involved in these distillation schemes may help providers apply more stringent precautions to these use cases.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by Korea government (MSIT) [No. 2021-0-00907, Development of Adaptive and Lightweight Edge-Collaborative Analysis Technology for Enabling Proactively Immediate Response and Rapid Learning, 90%], [No. 2019-0-00075, Artificial Intelligence Graduate School Program (KAIST), 5%], and the Stochastic Analysis and Application Research Center (SAARC) under the National Research Foundation of Korea grant (NRF-2019R1A5A1028324, 5%).

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 535–541, New York, NY, USA. Association for Computing Machinery.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ishita Dasgupta, Andrew K. Lampinen, Stephanie C. Y. Chan, Antonia Creswell, Dharshan Kumaran, James L. McClelland, and Felix Hill. 2022. Language models show human-like content effects on reasoning. *arXiv preprint, arXiv:2207.07051*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jonathan St BT Evans. 2010. Intuition and reasoning: A dual-process perspective. *Psychological Inquiry*, 21(4):313–326.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. *arXiv preprint arXiv:2301.12726*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*

and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3816–3830, Online. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *EMNLP*, pages 523–533. Citeseer.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint, arXiv: 1606.07947*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. *arXiv preprint, arXiv: 2206.14858*.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. 2022a. Explanations from large language models make small reasoners better. *arXiv preprint, arXiv: 2210.06726*.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2022b. On the advance of making language models better reasoners. *arXiv preprint arXiv:2206.02336*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410*.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *arXiv preprint, arXiv: 2202.04538*.

Aditya K Menon, Ankit Singh Rawat, Sashank Reddi, Seungyeon Kim, and Sanjiv Kumar. 2021. A statistical perspective on distillation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7632–7642. PMLR.

Paul Micaelli and Amos Storkey. 2019. *Zero-Shot Knowledge Transfer via Adversarial Belief Matching*, chapter -. Curran Associates Inc., Red Hook, NY, USA.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. 2019. *When Does Label Smoothing Help?* Curran Associates Inc., Red Hook, NY, USA.

Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. 2019. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, pages 4743–4751. PMLR.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma,

David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

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. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. -.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Yasaman Razeghi, Robert L. Logan, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint, arXiv:2202.07206*.

Subhro Roy and Dan Roth. 2016. Solving general arithmetic word problems. *arXiv preprint arXiv:1608.01413*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Chengchao Shen, Xinchao Wang, Youtan Yin, Jie Song, Sihui Luo, and Mingli Song. 2021. Progressive network grafting for few-shot knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2541–2549.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

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. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*.

Eric Zelikman, Yuhuai Wu, and Noah D Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *arXiv preprint arXiv:2203.14465*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

A Experimental Details

A.1 Generation

Maximum sequence length For the maximum sequence length of teacher-generated rationales, $\hat{r}_i$, we use $L_r = 128$, following Kojima et al. (2022), unless stated otherwise. For the maximum sequence length of the student model predictions, we use $L_p = 1024$, unless stated otherwise. We retroactively applied $L_p = 1024$ as the default, after discovering that $L_p = 128$ is insufficient for many tasks, as discussed in Appendix E.2.

Sampling temperature We apply greedy decoding for all generations, except diverse reasoning, to obtain deterministic results following (Wei et al., 2022b; Kojima et al., 2022). For diverse reasoning, we use temperature sampling with $T = 0.7$ to obtain diverse samples, following a similar approach from Wang et al. (2022).

A.2 Answer cleansing

We follow the method used in Kojima et al. (2022) to cleanse answers generated by models to assess their correctness.

A.3 Few-shot-CoT exemplars

For Few-shot-CoT prompting, we use exemplars provided by Wei et al. (2022b), with some minor formatting adaptations for consistency with our other experiments. For Last Letter Concatenation and Coin Flip, for which Few-shot-CoT prompts are not provided, we use 8 training samples used in our 8-shot data experiments shown in Figure 4 and adapt them for Few-shot-CoT using the format of Wei et al. (2022b). This was not applicable to Tracking Shuffled Objects, therefore it was omitted from Few-shot-CoT experiments.

A.4 Fine-tuning OpenAI models

We use default hyperparameters set by the OpenAI API for both vanilla fine-tuning and Fine-tune-CoT. While the specifics of the fine-tuning API is not publicly known, some details on hyperparameters are documented in the API reference [6]. According to the default settings, our models are trained for 4 epochs. The batch size and learning rate determined based on the number of examples used for training. The batch size is set to 0.2% of the number of training examples capped at 256. The learning rate is set to 0.05, 0.1, or 0.2 times that of the learning rate used to pre-train the base model, depending on the batch size. Training loss is also applied to the prompt portion of the training examples, i.e., the question, with a small weight of 0.01. Based on API pricing , we posit that OpenAI employs a form of parameter efficient fine-tuning such as LoRA (Hu et al., 2021) for their fine-tuning API instead of updating all model parameters.

[6] https://platform.openai.com/docs/api-reference/fine-tunes/create

A.5 Fine-tuning open source models

For vanilla fine-tuning and Fine-tune-CoT on open source models, we strictly control for hyperparameters. Across all experiments, we fine-tune the entire model with a fixed learning rate of 3e-4 and batch size of 8. Upon inspection of model performance under various learning rates and batch sizes, we found that optimal parameters varies among datasets, even between those with similar number of reasoning samples. We train all models for a maximum of 20 epochs, which we found to be sufficient for test accuracy to plateau. We report the best test accuracy from 20 epochs, but found that performance varies significantly between epochs. Overall, we found that performance by epoch is stable for larger models, and that instruction-tuned Flan-T5 is more stable compared to T5. Similar to learning rate and batch size, the optimal number of epochs also varies between datasets, even those with similar number of reasoning samples. Based on the above, we note that our reported performances of fine-tuned open-source models may be significantly under-estimated compared to those with optimal hyperparameters, and recommend practitioners to optimize hyperparameters using a separate validation set, per each training setting.

B Datasets

We provide a summary of datasets used in our experiments, including their original licenses, in Appendix Table 3. We consider the 12 datasets used in Kojima et al. (2022) to measure reasoning performance. For Last Letter Concatenation and Coin Flip, we use the publicly available data provided by Kojima et al. (2022).

Train-test split Contrary to previous works on prompt-based CoT such as Wei et al. (2022b); Kojima et al. (2022), our fine-tuning approach requires distinct sets of samples for training and testing. If

Dataset	Choices	Training Samples	Test Samples	Data Split	License	References
SingleEq	-	356	152	70:30	None	Koncel-Kedziorski et al. (2015)
AddSub	-	276	119	70:30	Unspecified	Hosseini et al. (2014)
MultiArith	-	420	180	70:30	Unspecified	Roy and Roth (2016)
GSM8K	-	7473	1319	Original	MIT	Cobbe et al. (2021)
AQUA-RAT	5	10000	254	Custom	Apache-2.0	Ling et al. (2017)
SVAMP	-	700	300	70:30	MIT	Patel et al. (2021)
Date Understanding	5–6	258	111	70:30	Apache-2.0	Srivastava et al. (2022)
Tracking Shuffled Objects	3	525	225	70:30	Apache-2.0	Srivastava et al. (2022)
Last Letter Concatenation	-	350	150	70:30	Unspecified	Wei et al. (2022b); Kojima et al. (2022)
Coin Flip	2	350	150	70:30	Unspecified	Wei et al. (2022b); Kojima et al. (2022)
CommonSenseQA	5	9741	1221	Original	Unspecified	Talmor et al. (2018)
StrategyQA	2	1603	687	70:30	Apache2.0	Geva et al. (2021)

Table 3: Description of datasets used in our study.

separate subsets for training and testing (or development) are provided, we use those. Otherwise, we perform a samplewise random split with a train-test ratio of 70:30. For AQUA, due to the disproportionately large size of the original training set, we randomly sample 10,000 instances for training in our experiments. Note that due to the highly templated nature of many datasets, this naive data split may not be appropriate for evaluating reasoning capabilities. This is an important nuance of fine-tuning on CoT reasoning, which we address in Appendix E.3.

Model Family	Params	Role	Variant / Name in API
GPT-3	175B	Teacher	`davinci`
InstructGPT	175B	Teacher	`text-davinci-001`
InstructGPT	175B	Teacher	`text-davinci-002`
InstructGPT	175B	Teacher	`text-davinci-003`
GPT-3	6.7B	Student	`curie`
GPT-3	1.3B	Student	`babbage`
GPT-3	0.3B	Student	`ada`
T5	60M	Student	Small
T5	220M	Student	Base
T5	700M	Student	Large
Flan-T5	60M	Student	Small
Flan-T5	220M	Student	Base
Flan-T5	700M	Student	Large
GPT-2	124M	Student	(Small)
GPT-2	355M	Student	Medium
GPT-2	774M	Student	Large

Table 4: Description of models used in our study.

C Models and API Usage

Appendix Table 4 describes all teacher and student models used in our study. We use InstructGPT (Ouyang et al., 2022) as the default teacher model in our experiments, due to its superior zero-shot reasoning performance, compared to GPT-3 (Brown et al., 2020) of the same size (Kojima et al., 2022). Specifically, we use `text-davinci-002` at the default, as it was the best available model at the start of our experiments. We were unable to consider small InstructGPT models for fine-tuning, as it is not offered by the OpenAI API. We attach model size information based on (https://blog.eleuther.ai/gpt3-model-sizes/), following Kojima et al. (2022).

Our total expenditure for API usage, including all preliminary experiments, was $1,981 USD. The majority of this expenditure occurred after September 1st, 2022, from which the pricing for inference on `davinci` models was $0.02/1K tokens, among others. Between teacher model inference, student model {fine-tuning, inference}, the majority of API usage in terms of cost was focused on teacher model inference.

D Sample Study

To understand where our method makes mistakes, where diverse reasoning can improve performance, and where our method always performs well, we observe randomly sampled instances and analyze the reasoning performance on them. To do so, we compare its generations for these samples with (1) the output of the large teacher model, (2) a student model using Zero-shot-CoT (3) a student model using Few-shot-CoT, and (4) a student model using fine-tuning without chain of thought reasoning. Our analysis reflects our overall findings, which we exemplify with representative examples in Tables 10–13.

D.1 Error analysis

For our analysis of the most common types of errors, we take a look at datasets where we find particularly bad performance of our vanilla method, also in comparison to other students. We also dis-

cuss the benefits of using diverse reasoning in D.2. We summarize our observations below.

Difficult datasets First, we observe that the sets GSM8K and AQUA appear to be too difficult for any small student model, in particular, given that the teacher model gets below 50% accuracy on both. In fact, even correct answers are usually correct only by chance, due to the high complexity of the tasks (Appendix Tables 10a,b). For AQUA in particular, we note that while we occasionally find meaningful reasoning in the 6.7B student model, students clearly cannot sufficiently learn to solve the tasks. We do note however that of all the student methods, Fine-tune-CoT still gets the best performance in these two datasets. A similar, if less salient, issue arises for StrategyQA. Here, the teacher also performs only 3% above the random guess accuracy of 50%. The smaller student models actually manage to improve on this performance as long as they do not use Zero-shot-CoT, in particular vanilla fine-tuning, but the errors arising in Fine-tune-CoT often look very similar to the ones in the large teacher model. We see that all models usually merely retrieve information related to the question, but cannot synthesize an answer from it (Appendix Tables 10c,11a).

Arithmetic mistakes Next, we note that small models overall exhibit weak arithmetic skills. This has already been discussed in previous literature, where calculation capability has been found to scale with model size (Wei et al., 2022a). Especially in SingleEq (Appendix Table 11b) and AddSub (Appendix Table 11c), a majority of errors in the output of student models using Fine-tune-CoT simply arise from wrong calculations, less so bad reasoning. This is also a major factor in the bad performance our method exhibits on SVAMP as well as GSM8K; even correct multi-step reasoning cannot compensate for the fact that the model's arithmetic tends to be wrong on intermediate steps (Appendix Tables 11d, 12a). Only the teacher model then does better on these tasks, given its much larger size, even though it does not get perfect accuracy either. However, we note here that very large language models, such as PaLM 540B, can be trained on arithmetic and scientific data to be able to reason correctly about a wide range of mathematical tasks in a step-by-step fashion (Lewkowycz et al., 2022).

Problematic benchmarks, impact of commonsense reasoning errors Meanwhile, when looking at our method's performance in CommonsenseQA, we note that producing consistent multi-step reasoning is not always the issue. We find that the student model utilizing Fine-tune-CoT can often generate logical reasoning paths for many of the samples that are marked as false (Appendix Table 13b). Rather, the exact answer is often subjective, making it difficult to guess the correct output from logical reasoning alone (Appendix Table 13c). CommonsenseQA thus is not always an ideal benchmark when judged on accuracy, but gives insight into how well the model can produce reasoning. We also note a difference compared to Few-shot-CoT in terms of the impact of reasoning errors: the latter only performs around 5% above random, lacks understanding of the question in many cases, and makes more severe logical and commonsense mistakes compared to our method. In fact, Fine-tune-CoT comes close to the teacher due to the relatively lower impact of errors that do arise (Appendix Table 13d). This suggests that Fine-tune-CoT enables stronger task-solving capabilities and avoids making serious commonsense mistakes that prevent it from arriving at a reasonable conclusion.

Aligned failures Importantly, we note that for each dataset, there seems to be a difference between "easy" and "hard" instances. When we consider the accuracy of the teacher and other student models (using fine-tuning, Zero-shot- or Few-shot-CoT) on tasks where our method fails, we find that it is always lower than on tasks where our method is successful. That is, successes and failures tend to be aligned across the different methods. We can hypothesize that factors such as content bias may play a role here; language models have been found to fail depending on context and content of the task, in a way similar to human reasoners (Dasgupta et al., 2022). We can identify samples that hint at this issue when we look at questions that include phrasing that seems contradictory or counterintuitive to the context that the model expects (see Appendix Table 13d, where the number of movies watched is larger than the number of available movies). Additionally, previous work shows that GPT-3 exhibits a performance gap between instances including terms that are frequent in the pretraining corpus, and instances including less frequent terms (Razeghi et al., 2022). This can

contribute to uneven performance on a multitude of (especially numerical) tasks across different methods and model sizes. We can then surmise the observed absolute differences in accuracy to stem from the various sources of errors for each method. For example, fine-tuning has much less room for error than Fine-tune-CoT, which can additionally make mistakes on intermediate reasonings such that errors compound.

D.2 Improvements from diverse reasoning

Semantic issues We find that models seem sensitive to how a question is formulated. This is noticeable in all datasets, in particular in SVAMP and to a certain degree in MultiArith. Besides arithmetic mistakes, we observe that such semantic issues are one of the main factors for uneven performance of vanilla Fine-tune-CoT on these two datasets.

In particular, we observe this issue when there is redundant information present in the question (Appendix Table 12b). Such cases elicit wrong reasoning, or lead the model to become stuck on the question, similarly to what usually happens with Zero-shot-CoT in the student model (i.e. repeating the question, or coming up with information that only vaguely pertains to the question). Other common sources of errors are when hidden variables make up the first part of the task (i.e. those tasks that force the model to calculate a previously unknown value that is described in the first sentence (Appendix Table 12c), or when the model encounters overloaded words (e.g., "landing" in Appendix Table 12d). We also observe samples where the model gets stuck on an intermediate result (Appendix Table 13a). This observation agrees with previous findings that language models have recency bias (Zhao et al., 2021).

However, this source of errors can be compensated for by using diverse reasoning. When comparing the generations from Few-shot-CoT, vanilla Fine-tune-CoT and Fine-tune-CoT with diverse reasoning on MultiArith, we find that diverse reasoning enables the model to understand the question better. While calculation errors are still relatively frequent, the generations show clear advantages in terms of semantic understanding and being able to reason logically as a consequence. This is especially clear when compared to Few-shot-CoT, which exhibits problems both in understanding the question and formulating coherent expressions, especially when three or more terms are involved in the calculation, as mentioned in Kojima et al. (2022). By contrast, Fine-tune-CoT with diverse reasoning makes for a significantly smoother reasoning performance than using Few-shot-CoT or even vanilla Fine-tune-CoT. This results in vastly improved accuracy on both MultiArith and SVAMP.

D.3 Strengths

Having analyzed the main sources of errors, we can now focus on the datasets that elicit good performance from our method, regardless of whether we use diverse reasoning.

Text-based datasets As arithmetic errors are one of the main reasons for the decrease in performance of small student models, it comes as little surprise that our vanilla method without diverse reasoning performs well on datasets that are mainly text-based and do not require actual calculation skills. This includes Date Understanding (60.4%) (Appendix Table 14a), Last Letter Concatenation (52.67%) (Appendix Table 14b), Coin Flip (98.7%) (Appendix Table 14c), and Shuffled Objects (64.4%) (Appendix Table 14d). Our methods performs significantly above random choice on these sets, and additionally beats the teacher on Shuffled Objects and Coin Flip. We find that accuracy metrics for these sets are mostly faithful: while the elicited reasoning is not always very detailed, and occasionally misses some reasoning steps (Appendix Table 14e) , the model draws correct conclusions from mostly correct steps. We also note that similar to MultiArith and SVAMP, performance on these four datasets can be even further boosted with diverse reasoning, outperforming the teacher model across all four.

Patterns These datasets also have very clear patterns in their tasks, which helps Fine-tune-CoT to perform well by providing cues on how to solve a specific task. We note that in contrast, classic fine-tuning does not have an advantage in these datasets, and it gets significantly lower accuracy than Fine-tune-CoT on all four. The same is also true for MultiArith, which we have used as a benchmark in the main text. While arithmetic errors cause the absolute accuracy of our method to be lower than the teacher, it significantly outperforms fine-tuning on MultiArith even without using diverse reasoning. Indeed, we find that also in the presence of arithmetic errors, our model reasons correctly in many cases. We can surmise that the

patterned nature of the tasks in MultiArith helps the student model to understand what is asked of it, eliciting the correct reasoning. Additionally, we note that the presence of such patterns in successful datasets does not mean that our method overfits to existing templates. In our template-split analysis (Appendix E.3), we in fact show that while tasks look similar to one another in certain datasets such as Date Understanding, the student model's reasoning does not rely on simply matching templates or memorizing particular solutions. This implies that our method can generalize to previously unseen tasks; the patterns in the datasets do not produce overfitting, but can be surmised to act as cues for the model's understanding of its current task. Thus, we observe that the reasoning skills of a student using Fine-tune-CoT can overcome the smaller model capacity (which proves to be completely prohibitive, e.g., for Zero-shot-CoT to have any success on the various tasks).

E Nuances of Fine-tune-CoT

E.1 Rationale filtering

We investigate whether answer-based filtering is sufficient for selecting *good* teacher-generated reasoning samples. It is possible for the teacher model to answer correctly despite incorrect reasoning, especially in multi-choice questions where the random-guess probability is significant. To investigate the potential impact of a better filtering scheme (as opposed to our baseline answer-based filtering) we manually annotate the correctness of rationales from the teacher model and evaluate student performance when fine-tuning on *correctly reasoned* samples. We use the Date Understanding dataset for this ablation, as it is comprised of well-grounded multi-choice questions for which Fine-tune-CoT achieves adequate performance. Appendix Table 6 compares the Fine-tune-CoT performance of student models on Date Understanding when using *correct* samples filtered based on answer predictions vs *golden* samples, hand-picked based on the correctness of rationales. For golden samples, we exclude samples that contain incorrect reasoning steps or irrelevant steps which are misleading. We find that 28% of correct samples have incorrect rationales–significantly more than the random-guess performance of 17.12%, indicating the importance of filtering. Surprisingly, we however find that answer-based filtering outperforms the more stringent human filtering by 5-11%, given the same initial samples. When we match the number of samples post-filtering (via undersampling), we do find that fine-tuning on golden samples outperforms that on correct samples by 5-8%. These results suggest that there is a tradeoff between the quality and quantity of reasoning samples which must be addressed when considering sample-filtering methods. We also note that this must be considered in tandem with diverse reasoning, which can drastically increase the quantity of reasoning samples.

E.2 Maximum sequence length

Following the original setting for Zero-shot-CoT (Kojima et al., 2022), we limit the max sequence length, or max tokens, allowed for the teacher-generated rationale and student reasoning predictions, denoted L_r, L_p, to 128 initially. However, we find that this can be insufficient in many datasets. Allowing for longer inference, we observe that model performance improves significantly on AQUA and commonsense reasoning tasks (Appendix Table 5). Sample inspection shows that rationales with over $\sim$500 tokens are typically repetitive or too digressive. To investigate the effect of the max length L_r of the teacher rationale on fine-tuning, we compare student performance using $L_r = \{128, 512\}$ (Appendix Table 7). The effect of L_r on student performance varies across datasets, and increased L_r does not necessarily improve student performance on tasks that require longer rationales, such as AQUA. Finally, we examine the length distribution of the generated rationales from the teacher model and student trained on short ($L_r = 128$) and long ($L_r = 512$) reasoning samples, respectively (Appendix Figure 7). We find that the distribution is different for each dataset. Notably, we find that while the distributions from the *long* students were similar to that of the teacher, the generated rationale from the *short* students were typically limited to less than $\sim$128 tokens. These findings are in line with the intuition that different tasks require different lengths of rationales, and suggest that careful consideration is needed in determining parameters related to sequence length.

E.3 Templated datasets

Upon inspection, we found that many datasets contain groups of samples which share common templates. Therefore, naive samplewise data split has the potential to leak the same templates into the train and test sets, essentially demoting the learn-

Model Params	Max Tokens	Single Eq	Add Sub	Multi Arith	GSM8K	Aqua	SVAMP	Date Understanding	Shuffled Objects	Last Letter	Coin Flip	Common SenseQA	Strategy QA
Teacher: InstructGPT (`text-davinci-002`)													
175B	128	81.18 **(84.83)**	75.72 (90.22)	76.90 (95.24)	42.42 **(69.85)**	29.63 **(44.04)**	64.00 **(86.57)**	65.89 (98.06)	54.10 (97.14)	57.43 (99.71)	89.71 (97.14)	59.86 **(82.55)**	53.40 **(71.55)**
	2048	81.18 **(84.83)**	75.72 (90.22)	76.48 (94.29)	47.73 (99.34)	34.77 (96.42)	66.00 (99.00)	63.28 (97.14)	54.10 (97.14)	57.43 (99.71)	89.71 (97.14)	59.40 (99.92)	53.03 (99.69)
Student: GPT-3 (`ada`, `babbage`, `curie`)													
0.3B	128	7.24 (96.05)	6.72 (99.16)	5.56 (96.11)	3.11 **(74.75)**	16.54 **(45.67)**	4.33 (91.33)	17.12 (100.00)	48.89 (100.00)	50.67 (100.00)	99.33 (100.00)	30.30 **(86.73)**	47.16 **(87.63)**
	1024	7.24 (98.68)	6.72 (99.16)	6.11 (97.22)	3.11 (99.77)	23.62 (100.00)	5.00 (97.33)	17.12 (100.00)	49.33 (100.00)	50.67 (100.00)	99.33 (100.00)	32.68 (100.00)	52.55 (99.71)
1.3B	128	11.18 (92.76)	11.76 (96.64)	13.89 (98.89)	4.02 **(75.36)**	15.35 **(48.03)**	7.33 (90.33)	38.74 (100.00)	53.78 (99.56)	50.67 (100.00)	100.00 (100.00)	40.95 **(86.57)**	47.02 **(83.99)**
	1024	11.18 (98.68)	11.76 (98.32)	13.33 (99.44)	4.70 (99.92)	19.69 (99.61)	8.00 (99.00)	38.74 (100.00)	52.44 (100.00)	50.67 (100.00)	100.00 (100.00)	43.08 (99.92)	52.69 (98.98)
6.7B	128	21.05 (92.76)	20.17 (97.48)	34.44 (99.44)	7.20 **(76.19)**	16.93 **(55.91)**	12.67 (93.67)	60.36 (99.10)	64.00 (100.00)	52.00 (100.00)	98.00 (100.00)	51.27 **(85.26)**	47.16 **(84.28)**
	1024	20.39 (98.68)	21.01 (100.00)	33.33 (100.00)	6.75 (99.92)	24.02 (100.00)	12.67 (99.00)	60.36 (100.00)	64.44 (100.00)	52.67 (100.00)	98.67 (100.00)	56.76 (100.00)	55.02 (99.71)
Random		0.00	0.00	0.00	0.00	20.00	0.00	17.12	33.33	0.00	50.00	20.00	50.00

Table 5: **Ablation on maximum sequence length.** Accuracy (%) of Zero-shot-CoT on the teacher model and Fine-tune-CoT on GPT-3 student models, based on maximum sequence length. Values in parentheses refer to the percentage of generated rationales that were completed within the allotted maximum sequence length. Percentages lower than 90% are marked in bold. Note that the maximum sequence length is applied to the reasoning portion, i.e., step 1, of Zero-shot-CoT and to the entire output of Fine-tune-CoT.

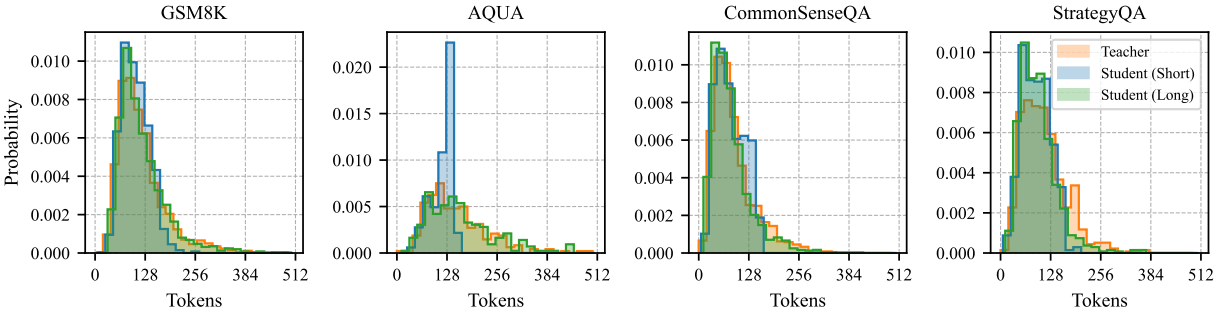


Figure 7: **Effects of teacher reasoning length on student reasoning length.** Distribution of the length of generated reasoning sequences from the 175B teacher model and fine-tuned 6.7B student models on four datasets. Student (Short) refers to students that were fine-tuned on reasoning samples with maximum rationale sequence length of $L_r = 128$, and Student (Long) refers to students that were fine-tuned on longer reasoning samples with $L_r = 512$.

Method	Filter	Samples	Model Params 0.3B	1.3B	6.7B
Zero-shot-CoT		0	10.81	14.41	15.32
Fine-tune-CoT	Answer	170	17.12	38.74	60.36
Fine-tune-CoT	Golden	123	17.12	28.83	54.95
Fine-tune-CoT	Answer†	123	17.12	18.92	50.45
Random			16.09		

Table 6: **Effects of rationale filtering.** Accuracy (%) of GPT-3 student models under Fine-tune-CoT when using samples filtered using answer predictions (Answer), or filtered by humans based on the correctness of the rationale (Golden). Answer† refers to using a randomly sampled subset of the correct samples to match the number of golden samples.

Model Params	Max Tokens	GSM8K	AQUA	Common SenseQA	Strategy QA
0.3B	128	3.11	23.62	32.68	52.55
	512	3.41	15.35	32.10	52.98
1.3B	128	4.70	19.69	43.08	52.69
	512	3.79	18.90	43.65	53.42
6.7B	128	6.75	24.02	56.76	55.02
	512	7.96	18.90	58.15	54.15
Random		1.01	20.44	20.01	50.18

Table 7: **Effects of teacher reasoning length on student performance.** Accuracy (%) of GPT-3 students models under Fine-tune-CoT on four datasets which require longer rationales, when trained on reasoning samples with maximum rationale sequence lengths of $L_r = 128, 512$.

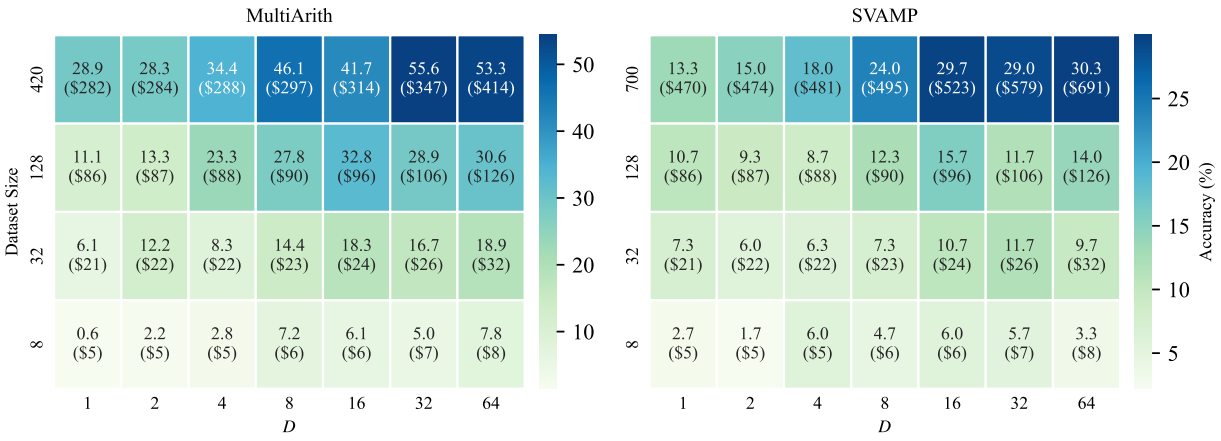


Figure 8: **Cost analysis of data acquisition methods**. Accuracy (%) of GPT-3 (6.7B) student model under Fine-tune-CoT with varying degrees of diverse reasoning and dataset sizes. Values in parentheses indicate estimated total cost of data acquisition, i.e., data annotation and diverse reasoning inference.

Params	Split	MultiArith	Date Understanding
0.3B	Sample-wise	5.56	17.12
	Template-wise	5.35	22.22
1.3B	Sample-wise	13.89	38.74
	Template-wise	7.49	35.19
6.7B	Sample-wise	34.44	60.36
	Template-wise	21.39	49.07

Table 8: **Sample-wise vs template-wise split.** Accuracy (%) of GPT-3 student models under Fine-tune-CoT on two moderately templated datasets when using a sample-wise vs template-wise train-test split.

ing problem into simple pattern matching, rather than complex reasoning. This brings into question the validity of naive samplewise data split, as it has the potential to leak the same templates into the train and test sets. To investigate whether the student models are truly learning to reason rather than matching simple patterns, we manually group samples by template and evaluate Fine-tune-CoT using a template-wise data split. We consider MultiArith and Date Understanding as they contain a moderate number of templates. Note that all datasets excluding GSM8K, CommonsenseQA, and StrategyQA contain templates to varying degrees. Appendix Table 8 shows the performance of Fine-tune-CoT when using sample-wise vs template-wise split, using the same train-test ratio of 70:30. While student performance is typically lower with a template-wise split, it still significantly outperforms random guess performance, as well as prompt-based baselines shown in Appendix Table 1. This reaffirms that Fine-tune-CoT is able to elicit complex reasoning capabilities in small language models.

F Data Annotation vs Diverse Reasoning

In Appendix Figure 8, we analyze the cost of data annotation and diverse reasoning, based on current OpenAI API pricing and a low estimate of annotation cost at 30 annotations per hour at an hourly rate of $20, i.e., $0.67 per question-answer sample. When comparing the cost and student performance of models trained with $D = 1$ and $D = 64$, we can clearly see that using diverse reasoning can enhance the cost-effectiveness of data acquisition. However, as the cost of diverse reasoning correlates with the size of the dataset, it is important to consider the cost-performance tradeoffs.

G Experiments on Open Source Models

To validate the generality of our method, we apply our method to a wide range of student models beyond variants of GPT-3. While the OpenAI API for GPT-3 inference and fine-tuning is accessible and does not require high-end GPUs, the model weights and implementation are not publicly available and may involve black-box processing. We therefore conduct experiments from Section 4 on open-source models under a standard setting with fixed hyperparameters, as explained in Appendix A and report our results in the following. Tables and figures include results from Section 4 on GPT-3 for reference.

Prompt-based baselines A comprehensive performance evaluation of student models across multiple tasks is encapsulated in Table 9, comparing Fine-tune-CoT against baseline methods. Performance of standard zero-shot prompting, predominantly insignificant, is omitted when negligible but does exhibit unexpected spikes on Flan-T5, such as 94.22% on Tracking Shuffled Objects on the smallest model. Few-shot-CoT likewise demonstrates inconsequential performance across most student models, yet the Flan-T5 models reveal significant performance on some tasks such as 7.51% on GSM8K and 83.87% on CommonSenseQA. This hints at the possibility that instruction tuning may empower models to comprehend and execute CoT prompts, unveiling a latent reasoning capacity within smaller language models.

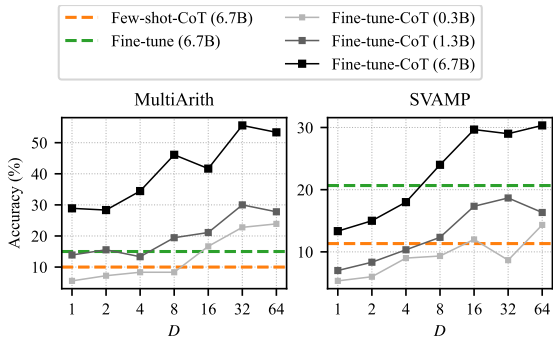
Fine-tune-CoT vs vanilla fine-tuning Further examining Table 9, we note that vanilla fine-tuning achieves notable performance in encoder-decoder architectures, namely T5 and Flan-T5, achieving more than 80% on Date Understanding and 100% on Coin Flip, significantly outperforming vanilla fine-tuning on GPT-2 and GPT-3 student models. This leads us to believe that the causal attention masking present in decoder-only models could impede complex inter-token reasoning. CoT reasoning, in this regard, may serve to mitigate this limitation by repeating key information within the decoding context. Other the other hand, Fine-tune-CoT either surpasses or matches the performance of vanilla fine-tuning across a variety of tasks. Our method also displays consistent scalability with model size, in contrast to the fluctuating performance between model sizes for baseline methods. The incorporation of diverse reasoning enhances this scalability. Particularly, we find that the Flan-T5 models benefit more from Fine-tune-CoT compared to T5 models, implying a favorable role of instruction tuning. When enhanced with diverse reasoning, Fine-tune-CoT excels over vanilla fine-tuning across several complex reasoning tasks, notably observed in the performance of Flan-T5 on Tracking Shuffled Objects (44.00%→89.33%) and GPT-2 on MultiArith (11.67%→19.44%).

Effects of diverse reasoning Figure 9 shows the performance of all student models on MultiArith and SVAMP under varying degrees of diverse reasoning. We observe that performance scales with diverse reasoning in all student models, with the exception of T5-Small. It is shown that diverse reasoning enables Fine-tune-CoT to outperform standard fine-tuning in all cases.

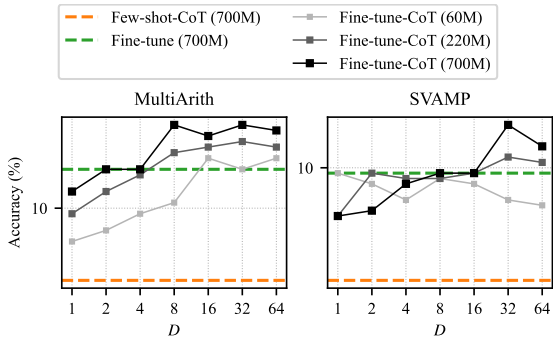
Effects of student model scale Figure 10 shows the performance of all student model families according to model size. While we observe performance scaling for Fine-tune-CoT on GPT-3 models, this is not apparent in other open-source models. We posit that this may be due to under-tuned hyperparameters, as we used fixed hyperparameters for all open-source models, in contrast to default suggested settings for GPT-3.

Method	Params	Single Eq	Add Sub	Multi Arith	GSM8K	Aqua	SVAMP	Date Understanding	Shuffled Objects	Last Letter	Coin Flip	Common SenseQA	Strategy QA
Random		0.00	0.00	0.00	0.00	20.00	0.00	17.12	33.33	0.00	50.00	20.00	50.00
Teacher: InstructGPT 175B (`text-davinci-002`)													
Zero-shot-CoT	175B	81.50	76.71	78.79	42.17	29.74	64.20	67.58	53.20	57.71	90.04	60.07	53.45
Student: GPT-3 (`ada`, `babbage`, `curie`)													
Few-shot-CoT	0.3B	0.66	0.84	3.33	1.74	15.75	2.00	19.27	-	0.00	44.67	18.43	42.98
	1.3B	3.29	5.88	5.00	1.59	13.78	4.33	16.51	-	0.00	46.00	18.67	46.05
	6.7B	22.37	31.93	10.00	2.50	15.75	11.33	12.84	-	0.67	40.00	24.73	54.68
Fine-tune	0.3B	9.87	8.40	8.89	5.08	24.41	7.67	23.42	32.44	28.67	100.00	51.68	60.41
	1.3B	11.84	17.65	17.78	5.38	21.26	14.33	31.53	30.22	30.00	100.00	70.93	60.70
	6.7B	24.34	25.21	15.00	6.14	15.35	20.67	14.41	33.78	32.67	72.00	76.17	65.21
Fine-tune-CoT	0.3B	7.24	6.72	6.11	3.11	23.62	5.00	17.12	49.33	50.67	99.33	32.68	52.55
	1.3B	11.18	11.76	13.33	4.70	19.69	8.00	38.74	52.44	50.67	100.00	43.08	52.69
	6.7B	20.39	21.01	33.33	6.75	24.02	12.67	60.36	64.44	52.67	98.67	56.76	55.02
Fine-tune-CoT	0.3B	9.21	10.08	23.89	-	-	14.33	58.56	61.78	59.33	99.33	-	57.21
w/ diverse reasoning	1.3B	18.42	19.33	27.78	-	-	16.33	70.27	72.00	60.67	100.00	-	57.06
	6.7B	24.34	31.09	53.33	-	-	30.33	83.78	73.33	62.00	100.00	-	58.22
Student: T5-{Small, Base, Large}													
Few-shot-CoT	60M	1.32	3.36	3.33	1.97	24.80	1.33	20.72	-	0.00	44.67	19.25	46.00
	220M	1.97	2.52	1.11	1.74	23.23	0.33	9.91	-	0.00	55.33	13.35	52.55
	700M	1.32	1.68	2.78	2.43	19.69	3.00	9.91	-	0.00	55.33	18.92	53.13
Fine-tune	60M	5.92	8.40	13.89	4.02	29.92	11.33	80.18	94.22	24.67	100.00	22.11	58.81
	220M	5.92	11.76	15.00	5.00	24.80	8.67	78.38	37.78	44.00	100.00	51.60	59.24
	700M	6.58	9.24	13.89	4.25	26.77	9.67	79.28	33.78	50.67	100.00	20.88	61.72
Fine-tune-CoT	60M	2.63	5.04	5.56	2.58	24.02	9.33	77.48	40.00	29.33	100.00	29.48	54.73
	220M	4.61	7.56	10.56	3.18	26.77	7.00	80.18	42.67	47.33	98.67	45.37	55.90
	700M	5.26	10.92	10.56	4.55	29.92	9.00	80.18	46.22	52.00	100.00	54.22	56.33
Fine-tune-CoT	60M	7.24	7.56	15.00	-	-	7.67	81.08	59.11	46.67	100.00	-	56.04
w/ diverse reasoning	220M	5.26	10.08	16.11	-	-	10.33	82.88	65.33	60.67	100.00	-	59.68
	700M	7.89	11.76	17.78	-	-	11.33	81.98	81.78	63.33	100.00	-	62.15
Student: Flan-T5-{Small, Base, Large}													
Zero-shot	60M	0.00	0.00	1.67	2.12	23.62	2.00	32.43	33.78	0.00	54.00	39.07	48.47
	220M	1.32	0.00	5.00	2.50	27.95	2.00	30.63	31.11	0.00	7.33	72.24	53.42
	700M	1.32	4.20	3.89	2.05	24.41	2.67	9.91	28.89	0.00	54.00	84.03	49.34
Few-shot-CoT	60M	1.32	0.84	1.67	2.81	20.87	1.67	27.93	-	0.00	44.67	11.79	51.97
	220M	2.63	0.84	3.89	3.64	24.80	3.67	12.61	-	0.00	44.67	70.27	53.86
	700M	12.50	10.08	10.00	7.51	23.23	8.33	20.72	-	0.00	44.67	83.87	65.21
Fine-tune	60M	7.24	9.24	16.67	4.93	28.74	10.33	81.08	33.78	39.33	100.00	45.95	58.95
	220M	5.26	10.08	16.11	5.08	29.53	10.67	83.78	44.00	45.33	100.00	63.55	61.14
	700M	7.24	12.61	18.89	5.53	24.80	11.00	82.88	33.78	53.33	100.00	66.75	63.90
Fine-tune-CoT	60M	6.58	5.88	8.33	2.96	23.23	5.67	80.18	36.00	35.33	100.00	42.01	54.15
	220M	4.61	9.24	12.22	4.40	29.13	6.00	83.78	48.89	50.00	100.00	59.05	59.97
	700M	11.84	10.92	14.44	5.38	28.35	10.67	84.68	55.11	64.00	100.00	66.83	59.83
Fine-tune-CoT	60M	7.24	10.92	17.22	-	-	10.67	84.68	62.22	46.00	100.00	-	56.04
w/ diverse reasoning	220M	9.21	10.92	21.11	-	-	12.33	84.68	67.11	56.67	100.00	-	60.84
	700M	10.53	15.13	20.00	-	-	13.67	87.39	89.33	65.33	100.00	-	61.72
GPT-2 {Small, Medium, Large}													
Few-shot-CoT	124M	1.32	0.00	0.00	0.45	17.32	0.33	13.51	-	0.00	44.67	20.15	0.00
	355M	0.00	0.00	0.56	0.00	3.94	0.00	9.91	-	0.00	55.33	0.00	0.15
	774M	0.00	0.00	0.00	0.00	0.39	0.00	13.51	-	0.00	55.33	0.16	35.08
Fine-tune	124M	2.63	3.36	11.67	2.88	25.59	7.67	7.21	33.78	0.67	60.00	20.80	54.00
	355M	0.66	0.84	5.00	0.38	18.90	0.00	23.42	36.89	1.33	57.33	19.82	50.22
	774M	1.32	5.04	8.33	2.58	24.80	7.67	13.51	32.44	0.67	1.33	20.88	53.57
Fine-tune-CoT	124M	4.61	4.20	10.00	3.03	24.02	5.67	17.12	38.67	4.67	88.00	22.19	53.57
	355M	3.29	5.88	7.22	2.73	23.62	7.33	28.83	35.56	10.67	80.00	22.03	55.02
	774M	3.95	5.88	10.56	2.58	22.05	6.33	15.32	39.11	4.00	89.33	25.80	53.13
Fine-tune-CoT	124M	7.24	9.24	19.44	-	-	10.67	21.62	57.33	10.67	93.33	-	56.62
w/ diverse reasoning	355M	5.92	9.24	17.22	-	-	9.67	20.72	56.00	20.00	95.33	-	55.60
	774M	8.55	12.61	17.22	-	-	8.67	18.02	52.44	7.33	84.67	-	57.06

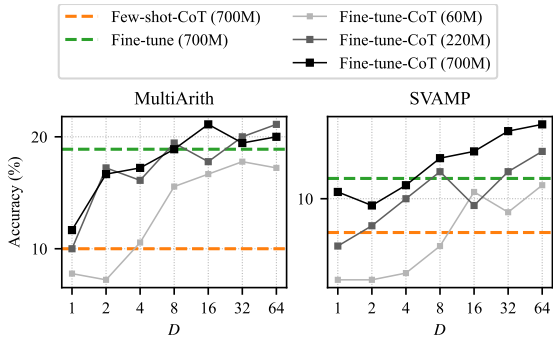
Table 9: **Fine-tune-CoT performance on all models.** Accuracy (%) of all models on 12 tasks under Fine-tune-CoT (with diverse reasoning) and baseline methods. 'Random' refers to random-guess performance derived based on the number of choices in multi-choice tasks. For diverse reasoning, we report results for maximum degree D considered: $D = 64$ for MultiArith and SVAMP; $D = 8$ for other datasets. We omit diverse reasoning for large datasets due to resource constraints and Few-shot-CoT for Tracking Shuffled Objects due to absence of prompts. Zero-shot baseline performance is omitted due to negligible performance, except for Flan-T5 models.



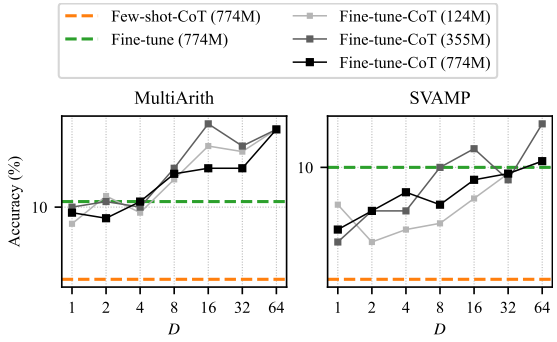
(a) GPT-3



(b) T5

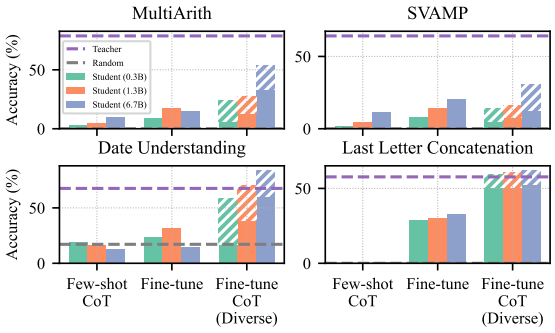


(c) Flan-T5

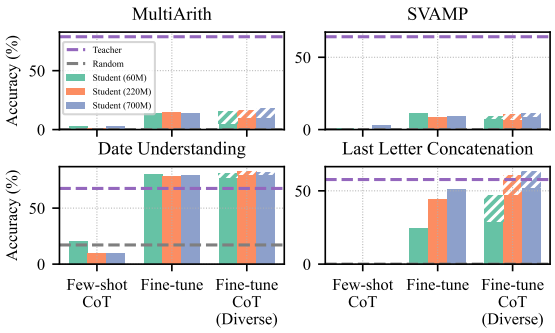


(d) GPT-2

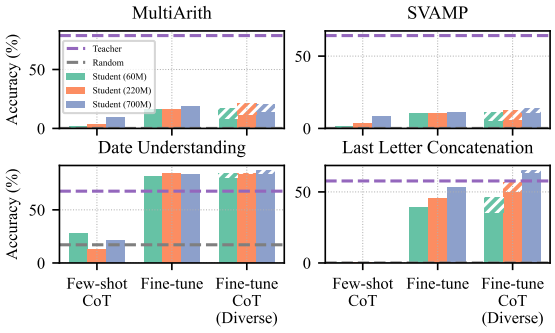
Figure 9: **Diverse reasoning performance on all models.** Accuracy (%) of all student models under Fine-tune-CoT with varying degrees of diverse reasoning D. Baseline performance of the *largest model* under vanilla fine-tuning and Few-shot-CoT are shown for comparison. Diverse reasoning is not applicable to the baselines.



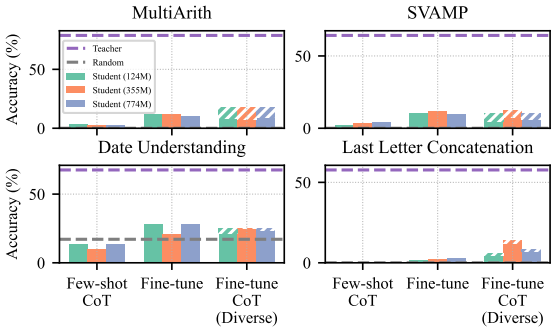
(a) GPT-3



(b) T5



(c) Flan-T5



(d) GPT-2

Figure 10: **Effects of student model scale on all models.** Accuracy (%) of all student models of various sizes under Few-shot-CoT, vanilla fine-tuning, and Fine-tune-CoT (with diverse reasoning). The hatched portion indicates the performance boost of Fine-tune-CoT when using diverse reasoning with $D = 64$ on MultiArith and SVAMP; $D = 8$ on others.

Table 10: Examples of incorrect output.

(a) GSM8K	
Question	Zaid spends 1/4 of his salary on rent, 1/3 on car fuel and donates half of the remaining amount to his favorite charity. He gives his daughter 200$ to use for her weekly expenses and 700$ to his wife to budget for groceries and other household goods. If Zaid earns 6000$ per month, how much money will he still have after all these expenses and donations? ###
Answer	350
Completion	Zaid spends 1/4 of his salary on rent, so that's 300$ per month. Zaid spends 1/3 of his salary on car fuel, so that's 150$ per month. That leaves him with 3000$ per month. He donates half of the remaining amount to his favorite charity, so that's 1500$ per month. That leaves him with 1500$ per month. He gives his daughter 200$ per month, so that's 400$ per month. He gives his wife 700$ per month, so that's 350$ per month.
Prediction	350
(b) AQUA	
Question	20 marbles were pulled out of a bag of only white marbles, painted black, and then put back in. Then, another 20 marbles were pulled out, of which 1 was black, after which they were all returned to the bag. If the percentage of black marbles pulled out the second time represents their percentage in the bag, how many marbles in total Q does the bag currently hold? Answer choices: (A) 40, (B) 200, (C) 380, (D) 400, (E) 3200. ###
Answer	D
Completion	There are 20 white marbles, 20 black marbles, and 1 black marble. The first time, 10 white marbles and 10 black marbles are pulled out. The second time, 5 white marbles and 5 black marbles are pulled out. Now, we can calculate the total number of marbles that are in the bag. There are 20 white marbles and 20 black marbles. The first time, 10 white marbles and 10 black marbles are pulled out. The second time, 5 white marbles and 5 black −−> D
Prediction	D
(c) StrategyQA	
Question	Is it common to see frost during some college commencements? ###
Answer	Yes
Completion	First, what is a college commencement? A college commencement is the ceremony at which a student graduates from college. Second, what is frost? Frost is a type of weather that occurs when the temperature is below freezing and the ground is frozen. So, is it common to see frost during some college commencements? No, it is not common to see frost during some college commencements. −−> No
Prediction	No

Table 11: Examples of incorrect output.

(a) StrategyQA	
Question	Is sternum connected to any of the bones mentioned in James Weldon Johnson's Dem Bones? ###
Answer	Yes
Completion	The sternum is a bone in the chest. The bones in the chest are the sternum, the clavicle, the two collarbones, and the two breastbones. The breastbones are the two right and left breastbones. So, no, the sternum is not connected to any of the bones mentioned in James Weldon Johnson's Dem Bones. −−> No
Prediction	No
(b) SingleEq	
Question	There are 33 walnut trees currently in the park. Park workers will plant 44 more walnut trees today. How many walnut trees will the park have when the workers are finished ? ###
Answer	77.0
Completion	There are 33 walnut trees currently in the park. Park workers will plant 44 more walnut trees today. This means that when the workers are finished, there will be a total of 83 walnut trees in the park. −−> 83.0
Prediction	83.0
(c) AddSub 25	
Question	Jason has 43 blue and 16 red marbles . Tom has 24 blue marbles . How many blue marbles do they have in all ?###
Answer	67.0
Completion	Jason has 43 blue marbles and 16 red marbles. Tom has 24 blue marbles. To find out how many blue marbles they have in all, we need to add 43 and 24. 43 + 24 = 77 Therefore, they have 77 blue marbles in all. −−> 77
Prediction	77
(d) GSM8K	
Question	Four dogs sat in a line within the veterinarian's waiting room. The first dog was the Affenpinscher, who weighed only 10 pounds. Next to him sat a Dachshund, who weighed twice as much as the Affenpinscher. Next to the Dachshund sat a Papillon, who weighed one-fourth as much as the Dachshund. And at the end of the line sat a Mastiff, who weighed 44 times the weight of the Papillon. How much did the Mastiff weigh, in pounds? Let's think step by step ###
Answer	220
Completion	The Affenpinscher weighs 10 pounds. The Dachshund weighs twice as much as the Affenpinscher, so it weighs 20 pounds. The Papillon weighs one-fourth as much as the Dachshund, so it weighs 2 pounds. The Mastiff weighs 44 times as much as the Papillon, so it weighs 44 pounds. −−> 44
Prediction	44

Table 12: Examples of incorrect output.

(a) SVAMP	
Question	If you had 33 cookies and each bag has 11 cookies How many bags of cookies do you have? ###
Answer	3.0
Completion	There are 33 cookies. Each bag has 11 cookies. That means that if we divide 33 by 11, we will know how many bags of cookies there are. 33 divided by 11 equals 2. There are 2 bags of cookies. −−> 2.0
Prediction	2.0

(b) CommonsenseQA	
Question	What is someone who isn't clever, bright, or competent called? Answer choices: (A) clumsy, (B) ineffectual, (C) dull, (D) clumsy, (E) stupid. ###
Answer	E
Completion	If someone isn't clever, bright, or competent, then they're not effective. If someone's not effective, then they're not useful. If someone's not useful, then they're not needed. If someone's not needed, then they're not important. Therefore, the answer is (B) ineffectual. −−> B
Prediction	B

(c) CommonsenseQA	
Question	Billy set aside a block of time for having fun after work. Why might he do this? Answer choices: (A) happiness, (B) stress relief, (C) pleasure, (D) ocean, (E) may laugh. ###
Answer	B
Completion	After work, Billy might be tired and want to relax. He could set aside a block of time and say, "OK, I'm going to spend the next X minutes doing something I enjoy." This might help him relax and feel happier. −−> A
Prediction	A

(d) CommonsenseQA	
Question	What island country is ferret popular? Answer choices: (A) own home, (B) north carolina, (C) great britain, (D) hutch, (E) outdoors.###
Answer	C
Completion	A ferret is a small mammal, so (D) hutch is probably not the best answer. It's also not a very common animal, so (A) own home and (B) north carolina are probably not the best answers. That leaves us with (C) great britain and (E) outdoors. Great britain is a larger island than outdoors, so the ferret is probably more popular in that country. −−> C
Prediction	C

(e) SVAMP	
Question	There are 17 different movies and 11 different books in the ' crazy silly school ' series. If you read 13 of the books and watched 63 of the movies How many more movies than books are there in the ' crazy silly school ' series? ###
Answer	6.0
Completion	There are 17 different movies and 11 different books in the 'crazy silly school' series. If you read 13 of the books, that means you have read 7 books. If you watched 63 of the movies, that means you have watched 63 movies. So there are 17 movies and 11 books, and you have read 7 books and watched 63 movies. That means there are 8 more movies than books in the 'crazy silly school' series. −−> 8.0
Prediction	8.0

Table 13: Examples of incorrect output.

(a) SVAMP	
Question	Danny collects bottle caps and wrappers. He found 65 wrappers and 5 bottle caps at the park. Now he has 31 bottle caps and 9 wrappers in his collection. How many more wrappers than bottle caps did danny find at the park? ###
Answer	60.0
Completion	First, Danny finds 65 wrappers and 5 bottle caps at the park. Then, he has 31 bottle caps and 9 wrappers in his collection. Now, we want to know how many more wrappers than bottle caps Danny found at the park. We can find this out by subtracting the number of bottle caps from the number of wrappers: 9 - 5 = 4 So, Danny found 4 more wrappers than bottle caps at the park. −−> 4.0
Prediction	4.0
(b) SVAMP	
Question	Next on his checklist is wax to stick the feathers together and colors to paint them. He needs a total of 57 g of colors to paint them. He needs 22 g of wax more. If the feathers require a total of 353 g of wax How many grams of wax does he already have? ###
Answer	331.0
Completion	First, he needs 57 grams of colors to paint the feathers. Second, he needs 22 grams of wax more. Third, the feathers require a total of 353 grams of wax. So, he already has 353 grams of wax. −−> 353.0
Prediction	353.0
(c) CommonsenseQA	
Question	You'll find a landing at the top of what? Answer choices: (A) ocean, (B) apartment building, (C) stairwell, (D) airport, (E) room. ###
Answer	C
Completion	A landing is a place where something can touch the ground. A landing at the top of an ocean would be a very strange place to find, since there would be no ground to touch. A landing at the top of an apartment building would be a very strange place to find, since there would be no ground to touch. A landing at the top of a stairwell would be a very strange place to find, since there would be no ground to touch. A landing at the top of an airport would be a very strange place to find, since there would be no ground to
Prediction	A
(d) SVAMP	
Question	18 red peaches, 14 yellow peaches and 17 green peaches are in the basket. How many red and yellow peaches are in the basket?###
Answer	32.0
Completion	There are a total of 48 peaches in the basket. There are 18 red peaches, 14 yellow peaches, and 17 green peaches. This means that there are a total of 33 peaches in the basket. There are 33 peaches in the basket. This means that there are 17 red peaches and 14 yellow peaches in the basket. −−> 17.0
Prediction	17.0

Table 14: Examples of correct output.

(a) Date Understanding	
Question	Jane was born on the last day of Feburary in 2000. Today is her 16-year-old birthday. What is the date a month ago in MM/DD/YYYY? Which choice is true? Answer choices: (A) 01/30/2016, (B) 11/08/2015, (C) 01/29/2016, (D) 07/29/2015, (E) 05/04/2016, (F) 02/02/2016. ###
Answer	C
Completion	Jane's 16th birthday is on 02/28/2016. A month ago from that would be 01/28/2016. Therefore, the answer is (C). −−> C
Prediction	C
(b) Last Letter Concatenation	
Question	Take the last letters of each words in "Lino Mariel Aditya Elisabeth" and concatenate them. ###
Answer	olah
Completion	The last letter of "Lino" is "o". The last letter of "Mariel" is "l". The last letter of "Aditya" is "a". The last letter of "Elisabeth" is "h". So the final answer is "olah". –> olah
Prediction	olah
(c) Coin Flip	
Question	A coin is heads up. Cinthia does not flip the coin. Lloyd does not flip the coin. Jacqueline flips the coin. Jc flips the coin. Is the coin still heads up? Note that "flip" here means "reverse". ###
Answer	Yes
Completion	Cinthia does not flip the coin, so the coin remains heads up. Lloyd does not flip the coin, so the coin remains heads up. Jacqueline flips the coin, so the coin is now tails up. Jc flips the coin, so the coin is now heads up again. –> yes
Prediction	Yes
(d) Tracking Shuffled Objects	
Question	Alice, Bob, and Claire are playing a game. At the start of the game, they are each holding a ball: Alice has a blue ball, Bob has a yellow ball, and Claire has a green ball. As the game progresses, pairs of players trade balls. First, Bob and Claire swap balls. Then, Bob and Alice swap balls. Finally, Bob and Claire swap balls. At the end of the game, Bob has the Which choice is true? Answer choices: (A) blue ball, (B) yellow ball, (C) green ball. ###
Answer	B
Completion	First, Bob and Claire swap balls. This means that Bob now has the green ball and Claire has the yellow ball. Then, Bob and Alice swap balls. This means that Bob now has the blue ball and Alice has the green ball. Finally, Bob and Claire swap balls again. This means that Bob now has the yellow ball and Claire has the green ball. At the end of the game, Bob has the yellow ball. –> B
Prediction	B
(e) Date Understanding	
Question	Tomorrow is 11/12/2019. What is the date one year ago from today in MM/DD/YYYY? Which choice is true? Answer choices: (A) 11/02/2018, (B) 11/04/2018, (C) 09/04/2018, (D) 08/25/2018, (E) 11/11/2018. ###
Answer	E
Completion	One year ago from today would be 11/11/2018. −−> E
Prediction	E

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7

☑ A2. Did you discuss any potential risks of your work?
Section 8

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☑ A4. Have you used AI writing assistants when working on this paper?
We used ChatGPT sparingly for paraphrasing.

B ☑ Did you use or create scientific artifacts?

We create code for our experiments and provide a link to the anonymized code in a footnote within the abstract. We use 12 existing datasets which are listed in Section 4 and Appendix B.

☑ B1. Did you cite the creators of artifacts you used?
Appendix B

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We include an MIT license notice in our code repository.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Appendix B

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3, Appendix B

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix C

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4. We use default hyperparameters.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4, Appendix A

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.