Improving the Out-of-Distribution Performance of Distilled Chain of Thought Language Models

Thomas Lai
Student
McGill University
Electrical Engineering

Alex Koran
Student
McGill University
Electrical Engineering

Jonathan Lebensold
Mentor, Reliant AI
McGill University
Computer Science

Abstract

Recent work has demonstrated that chain of thought (CoT) reasoning has improved the quality of search results for a number of benchmark datasets (e.g., GSM8K, ScienceQA, MATH), but these methods tend to be computationally demanding, or are known to require large-parameter models. Several studies have shown that knowledge distillation from a large parameter model to small parameter model through supervised fine-tuning can be used to endow a small model (millions of parameters) with apparent reasoning abilities. However, it is unclear whether or not these small models have truly acquired semantic understanding and reasoning abilities thought to be correlated with number of parameters, due to the lack of out-of-distribution (OOD) results. In this project, we first perform an OOD study with the state-of-the-art "Distilling step-by-step" model on three domains: symbolic reasoning, math, and natural language inference. Secondly, we found that applying Fishr regularization can improve the OOD performance of "Distilling step-by-step" on the coin flip task. Due to GPU constraints, we only tested the method on one task, thus future experiments should be conducted on multiple other domains.[1]

1 Introduction

We have witnessed strikingly better performance of large language models in reasoning tasks such as solving math or common sense reasoning problems when chain of thought (CoT) techniques are employed. CoT reasoning studies generally fall under either fine-tuning a pretrained model (Liu et al., 2023) or in-context few-shot learning via prompt engineering (Wei et al., 2022). Both types of CoT reasoning were found to be emergent abilities of increased model scale, meaning that there tends to be a number of parameters required for a model to perform better with chain-of-thought compared to standard fine-tuning or standard prompting. In fact, CoT may actually hurt the performance for many models under 10B parameters (Wei et al., 2022).

The main hypothesis for this phenomenon is that semantic understanding and logical reasoning skills of large language models is a function of model scale (Wei et al., 2022). Small language models are observed to have weaker arithmetic abilities or relatively easy symbol mapping tasks (Wei et al., 2022). Symbolic reasoning tasks include toy examples such as last letter concatenation and coin flip. In last letter concatenation, the language model is asked to isolate the last letter of each word in a name, and concatenate the isolated letters. In coin flip, the language model is asked whether a coin still remains heads after people flip or don't flip the coin. These two tasks are almost trivial for humans to perform, but were observed to be challenging for language models.

While CoT has been shown to be able to endow a large language model the ability to perform symbol mapping tasks, models under 10B parameters with CoT still do not perform much better than small models without CoT. As a result, it is costly to apply CoT reasoning to real-world applications if large models are required, and several studies have investigated how CoT reasoning abilities can be distilled from a large language model to a small language model. While it has now been shown that small models can also produce correct CoTs after distilling from large models, there is no substantial proof that the small models are not simply just memorizing aspects about the large models' generated rationales, since the current studies of CoT distillation have not shown any robust out-of-distribution (OOD) studies.

Therefore, in our project, we first conduct an OOD study with the "Distilling Step-by-Step" Model (Hsieh et al., 2023). Following (Wei et al., 2022), we group the diverse range of CoT datasets

[1] Code is available at: https://github.com/thomaslai414/Comp767Project

into domains. Then, within each domain, we conduct OOD studies by fine-tuning the model on one dataset, and evaluating the fine-tuned model on another dataset from the same domain that share some similarities with the first dataset and is sufficiently distinct.

Secondly, we apply Fishr regularization (Rame et al., 2022) to "Distilling Step-by-Step" and observed improved OOD performance on the coin flip task. Fishr regularization enforces the similarity between the variances of the gradients of different groups of samples during fine-tuning. In doing so, Fishr regularization guides the gradient descent to a more generalized local minima.

2 Related Work

In order to explore the related work of our project, we must position ourselves in the fields of chain of thought reasoning (CoT) and knowledge distillation of LLMs. We must also explore further the intersection field of distilling of CoT LLMs.

2.1 Chain of Thought Reasoning

Basic approaches of CoT reasoning in LLMs are "Let's think step by step" (Wei et al., 2022) (prompting) and "Let's verify step by step" (Liu et al., 2023) (fine-tuning). "Let's think step by step" is a prompt that is used to activate an LLM's ability as a zero-shot CoT reasoner. Another standard CoT prompting technique is to include a CoT example, known as chain-of-thought prompting (Wei et al., 2022), before asking an LLM to solve a new problem. For fine-tuning, the authors of "Let's verity step by step" demonstrated the performance boost of adding a loss for each step of the CoT reasoning instead of only having a loss for the final answer. An important baseline for this is notably generating multiple CoTs before majority voting. In addition, verifiers can also be used for the LLM to self-improve on reasoning (Hosseini et al., 2024), and encourage the LLM to generate rationales before predicting each future token (Zelikman et al., 2024), attempting to simulate thinking in humans.

2.2 Knowledge Distillation of LLMs

Knowledge Distillation is a technique used to transfer knowledge from a large, complex model ("teacher") to a smaller, more efficient model ("student"), while aiming to retain the teacher's performance levels (Hinton et al., 2015). This method has become crucial in the development of LLMs, enabling the deployment of these models in resource constrained environments, by reducing computational requirements without drastically compromising their capabilities.

Within the context of LLMs, knowledge distillation has been used to create smaller models that approximate the performance of larger ones. For example, DistilBERT reduces the size of BERT by 40% while retaining 97% of its language understanding capabilities and running 60% faster (Sanh et al., 2019). Similarly, TinyBERT applies a two-stage learning framework involving transformer distillation at both the pre-training and task-specific learning stages, achieving competitive performance on natural language understanding tasks (Jiao et al., 2020).

Recent advances have focused on distilling even larger and more powerful models like GPT and LLaMA into smaller, more efficient models without significant loss in performance. Alpaca, developed by Stanford researchers, demonstrates that instruction-following capabilities of large models can be distilled into smaller models based on LLaMA (Taori et al., 2023). LLaMA-Adapter provides an efficient fine-tuning approach for LLaMA models, enabling them to perform well on downstream tasks with less computational resources (Gao et al., 2023). Additionally, QLoRA introduces a method for efficient fine-tuning of quantized LLMs, making it feasible to adapt large models like LLaMA-13B on a single GPU (Dettmers et al., 2023).

Common techniques in LLM knowledge distillation include soft label distillation, where the student model is trained to match the softened probability outputs of the teacher model, providing richer information than hard labels (Hinton et al., 2015). Intermediate layer distillation is also employed, aligning the student's intermediate representations with those of the teacher to capture more nuanced features (Jiao et al., 2020). Attention distillation transfers the teacher's attention patterns to the student, preserving contextual relationships critical for language understanding (Wang et al., 2020). Recent methods have also explored distilling reasoning capabilities into smaller models to enhance their problem-solving abilities (Sun et al., 2022).

Our proposal builds upon these advancements by focusing on distilling the chain-of-thought reasoning abilities of large LLMs into smaller models, with the specific aim of evaluating out-of-

distribution (OOD) performance. By training the smaller model to predict the next step in the rationale provided by the larger LLM, we aim to transfer not just the final outputs but the reasoning processes themselves. This approach leverages knowledge distillation to reduce model size while emphasizing the preservation of complex reasoning skills, which are crucial for robust performance in OOD settings. Such an endeavor aligns with recent research that seeks to imbue smaller models with the step-by-step reasoning capabilities of larger models (Shridhar et al., 2022; Wei et al., 2022), and contributes to a better understanding of how distilled models perform on tasks beyond their original training distributions.

2.3 Distillation of Chain of Thought LLMs

There have been studies where CoT capabilities are distilled from LLMs. To begin, standard task-specific distillation can be used to train a small model, using labels generated by an LLM. Albeit, there is no absolute guarantee for correctness for the generated labels, so these labels are considered noisy. In "Distilling Step By Step," an LLM with CoT is used to generate rationales as labels for training a small language model in a multi-task fashion (Hsieh et al., 2023). Under the multi-task framework, one task is to pass a (label) token along with the logical problem and the small model is expected to predict the correct final answer, while the other task is to pass the same problem with a (rationale) label and the small model is taught to return the same rationale as the LLM. Further, this work also separates the intermediate steps, hence the term "distilling step-by-step." On the other hand, Socratic CoT proposes a different framework where the CoT LLM would train one small model to break down the initial problem into smaller sub-problems, hence the term "Socratic." Then, the second small model is trained specifically to provide an answer to the sub-problems. Other related work focuses on specific tasks such as symbolic reasoning or mathematical reasoning. In "Symbolic Chain of Thought Distillation" (SCoTD), the authors highlighted the importance of sampling multiple generated answers from the LLM for each problem and discarding incorrect answers before training the small model (Xu et al., 2023). In "Equation of thought distillation" (EoTD) the small model is fine-tuned on mathematical reasoning using LLM-generated rationales by asking it to break down the problem into intermediate equations (Zhu et al., 2024). This paper also provides the insight that using multiple datasets during training improves reasoning ability. While these studies have shown that small models are capable of CoT, which was previously only observed in large models, there is insufficient proof to show that the small models actually obtained proper semantic understanding and logical understanding skills via robust out-of-distribution task studies.

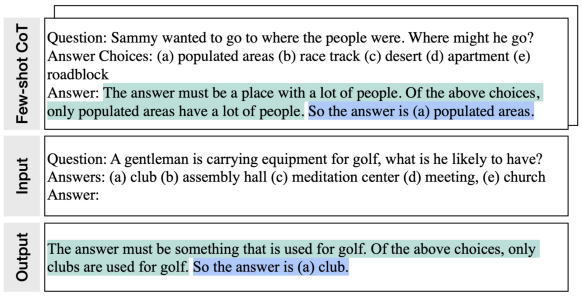


Figure 1: Few-shot CoT prompting (Hsieh et al., 2023). The prompt for the LLM consists of the "Few-shot CoT" and the "Input" blocks, while the LLM outputs the "Output" block. Example and output rationales are highlighted in green while example and output labels are highlighted in blue.

3 Models and Approach

Our models and approach will be heavily based on the "Distilling step-by-step" model, since we wish to investigate the generalizability of distilled models and "Distilling step-by-step" would be the state-of-the-art representative approach. We then propose to apply Fishr regularization - a technique novel to the natural language processing field - to "Distilling step-by-step" to improve its generalizability. In this part of the project, the rest of our approach is set to be consistent with "Distilling step-by-step" to ensure a fair comparison.

3.1 Baseline

We compare our approach to the state-of-the-art CoT distillation model. The approach "Distilling step-by-step" (Hsieh et al., 2023) distills CoT from PaLM (Anil et al., 2023) to small language models of the T5 architecture (Raffel et al., 2020). Adapting from their notation, given a dataset of inputs $x_i \in D$, a prompt tuple p is constructed as $p = (x_i, y_i, r_i)$, where x_i is an input question, y_i is the corresponding label, or the expected final answer of that question, and r^i is the user-given rationale. In our case, the datasets used contains

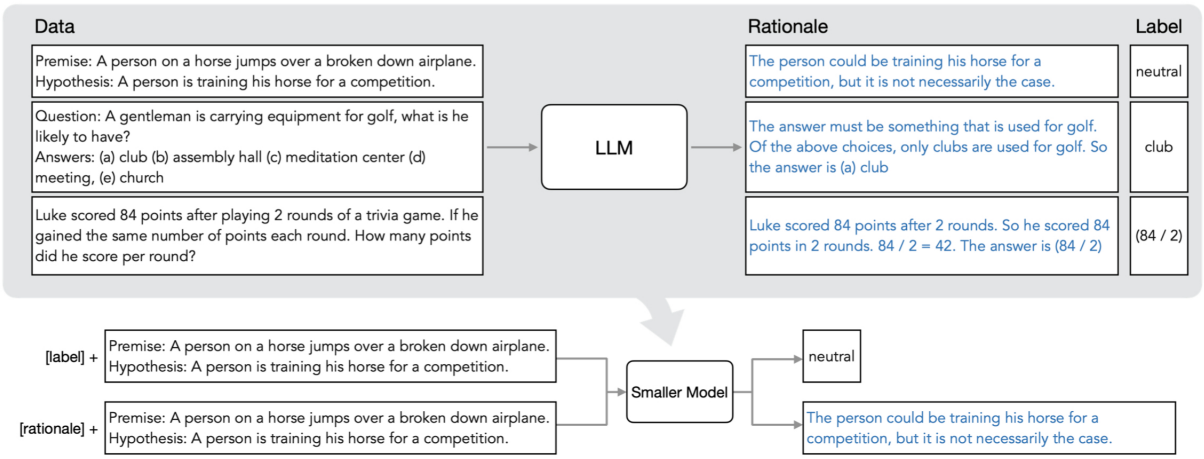


Figure 2: Framework of "Distilling Step-by-Step" (Hsieh et al., 2023), which is both the baseline and the backbone of our approach.

ground truth labels, which can be directly used for y_i. In the case where ground truth labels are not available, y_i can be obtained from the LLM through CoT prompting and/or few-shot learning, along with the rationale r_i. More specifically, a CoT few-shot prompt is constructed as the following concatenation: an example input + an example rationale + an example label + additional input, rationales, and labels if more than one shot + new input. This prompt then illicits a new output rationale and a new output label from PaLM. The few-shot CoT approach is illustrated in Figure 1.

Once the full tuple $p = (x_i, y_i, r_i)$ is obtained, supervised training is used to train the small model, denoted as f (a T5 model), in a multitask fashion. For each training example, the following loss objective is minimized:

$$\mathcal{L} = (1-\alpha)\mathcal{L}_{label} + \alpha\mathcal{L}_{rationale},$$

where α is a weight hyperparameter and $\mathcal{L}_{label}$ and $\mathcal{L}_{rationale}$ are the respective losses of each task in the multi-task framework. Specifically,

$$\mathcal{L}_{label_i} = l(f(\texttt{[[label]]}+x_i), y_i),$$

where l is cross entropy loss. In other words, for this task, the prompt is a `[label]` token concatenated with the input x_i, and the model f is trained to predict the label y_i. On the other hand,

$$\mathcal{L}_{rationale_i} = l(f(\texttt{[[rationale]]}+x_i), r_i),$$

which is analogous to the other task, but with the prefix `[rationale]`, the model f is trained to reproduce CoT rationales r_i, which are generated by the LLM. An illustration of this framework is provided in Figure 2.

In addition to "Distilling step-by-step", the LLM PaLM also serves as a baseline for an LLM's baseline performance on each datasets tested.

3.1.1 Our Proposed Approach

In order to improve out-of-distribution performance of distilled CoT models, we propose to add a penalty to the objective loss function as follows:

$$\mathcal{L} = (1-\alpha)\mathcal{L}_{label} + \alpha\mathcal{L}_{rationale} + \lambda\mathcal{L}_{fishr}.$$

In addition, our project provides novel insights about the out-of-distribution performance of "Distilling step-by-step" across multiple domains, which is described in the Experimental Details section.

In order to improve the generalizability of our baseline, we impose a penalty to align domain-level properties based on the gradient. First, each batch of data is divided based on groups $\mathcal{E} = A, B, C, ...$ In the coin flip example, since we can define multiple coin flip datasets that are distributionally distinct in a straightforward manner, each in distribution dataset is assigned a group. In other examples where this does not apply, there are other ways to assign groups, such as using the labels, which is a practice in invariant risk minimization (IRM) (Dranker et al., 2021).

In line with IRM, we choose a penalty that aims to extract domain-invariant features from the multiple groups present in our training set, by matching the group-level gradient variances during training. We choose to impose our penalty on the gradients

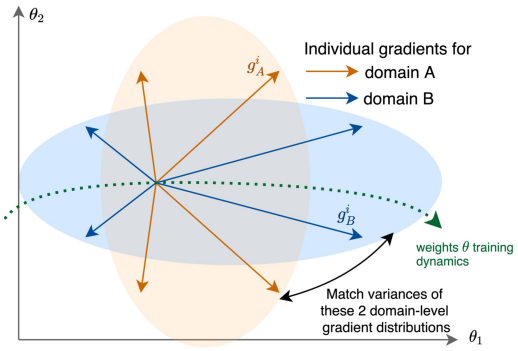


Figure 3: The Fishr principle (Rame et al., 2022) where the variances of the gradients of each group is matched.

of the last layer as opposed to directly on the representation space to minimize any restrictions on feature learning. For similar reasons, we choose to match the variances of the gradients between each group as opposed to the mean, which also allows the gradients of individual samples of the batch to maintain their diversity more. An illustration is provided in Figure 3.

More specifically, we use the Fishr loss described by (Rame et al., 2022):

$$\mathcal{L}_{fishr} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} ||\mathbf{v}_e - \mathbf{v}||_2^2,$$

where $\mathbf{v} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \mathbf{v}_e$ and $\mathbf{v}_e = \text{Var}(G_e)$, where G_e is all the gradients of the group e in the batch. In other words, for each batch, the gradient of the last layer for each sample within the batch is first calculated. Then, the gradients of the batch are separated based on their assigned groups and the variance of each group is calculated. Then, the mean of the variances are calculated, and the variances of each group is aligned by minimizing the Euclidean distance between each group variance and the mean.

4 Datasets and Evaluation Metrics

4.1 Datasets

Chain-of-thought is applicable to a wide range of domains, as established by (Wei et al., 2022). Thus, we investigate the OOD performance of "Distill step-by-step" on three domains: symbolic reasoning, math, and natural language inference.

4.1.1 Symbolic Reasoning

For this domain, we select the toy-example of coin flip (Wei et al., 2022). In a typical coin flip example, the coin starts heads up, then a person either flips or does not flip the coin, then a second person flips or does not flip the coin, then the model is asked whether or not the coin is still heads up. An example is provided in Figure 4. Even though this seems to be a trivial problem for humans, even large language models have been shown to struggle with this problem (for models less than ∼50B parameters) (Wei et al., 2022).

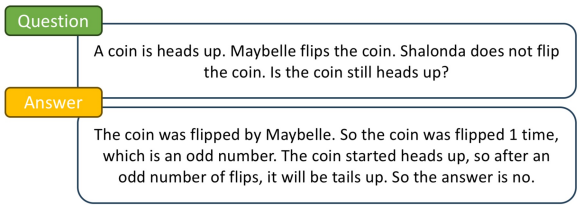


Figure 4: An example of a Coin Flip question and its corresponding chain-of-thought answer.

Following (Wei et al., 2022), to setup OOD datasets with coin flip, we assign coin flip examples with two people either flipping or not flipping the coin (like the default problem described) as one dataset, denoted as CF2, and coin flip examples that are extended to 4 people either flipping or not flipping the coin as another dataset, denoted as CF4. When a model is only trained on CF2, its performance on CF4 is considered OOD performance (Wei et al., 2022). Extending this idea, we include CF5, CF7, and CF8, and two or three CF datasets can be chosen at a time to be in distribution while the remaining CF datasets are considered OOD.

Each CF datasets contain 1000 training examples and 1000 testing examples. The average length of inputs and outputs are consistent as they all follow the same template. The labels are either "yes" or "no" and they are uniformly distributed. In coin flip, while the training rationales r_i can be obtained from an LLM via few-shot CoT prompting, it is not necessary. In this project, we generate the rationales with a straightforward template.

4.1.2 Math

We investigate the OOD performance in the Math domain by using two publicly available datasets for this work: SVAMP (Patel et al., 2021) and ASDiv (Miao et al., 2020). These datasets are well-suited for evaluating models on elementary math word problems due to their design, diversity, and relevance to real-world generalization. We use SVAMP for finetuning the models and ASDiv for evaluating them, thus constructing our OOD experiment.

SVAMP consists of single-variable arithmetic

problems where the input is presented as a natural language description (word problem) and the output is a numerical answer. The dataset showcases examples designed to evaluate a model's question sensitivity, robust reasoning ability, and invariance to structural alterations (Patel et al., 2021). SVAMP is particularly appropriate for evaluating in-domain performance since it introduces linguistic variations that require reasoning rather than just pattern matching.

The ASDiv dataset was selected for OOD evaluation to test the model's generalization to more linguistically diverse and structurally complex problems. ASDiv similarly incorporates word problems with varied phrasing, numerical values, and formats. This provides a challenging benchmark for OOD generalization.

During our experiments, SVAMP had 800 training examples and 200 testing examples. ASDiv had 1727 training examples and 432 testing examples. When preprocessing ASDiv, we removed some questions (146 of them) whose answers were non-numerical (e.g. "Which student has more apples"). This was needed since we finetuned on SVAMP, which only has numerical answers, so we kept the ASDiv answers in the same format.

The average length of SVAMP inputs is around 25 words, and for ASDiv 30 words. The output of both is 1 token since it is a numerical answer. There are no class labels as these are regression type tasks (continuous space of possible of answers, not discrete).

4.1.3 Natural Language Inference

For the NLI domain, each question consists of a Premise and a Hypothesis and the language model is expected to determine if the relationship between the two statements are entailment, neutral, or negative, which are the three possible labels for this task. For in distribution, we select the ANLI dataset (Williams et al., 2020) and for OOD, we select the e-SNLI dataset (Camburu et al., 2018). Visual inspection is used to verify the distinction between the two datasets, which is often in the subject matter. Examples of the ANLI and e-SNLI datasets are provided in Appendix A.

The ANLI dataset contains 16946 training examples and 1000 test examples. The e-SNLI dataset contains 549267 training examples and 9824 test examples. Both datasets are annotated by humans. Validation sets are ignored for both datasets.

4.2 Evaluation Metrics

The evaluation metric used is accuracy. We compute the accuracy of the getting the labels y_i correct. In our results we report both in distribution and out of distribution performance. When a result is marked as OOD, then the model has not been trained on the same dataset as the dataset that it is evaluated on. Otherwise, unmarked accuracy is in distribution. Note that the rationales are not scored.

5 Experimental Details

Existing CoT distillation studies do not report robust OOD experiments, while CoT in LLM studies do. Therefore, for each of the three domains, we conduct OOD experiments by fine-tuning a distilled CoT model on one of the datasets of that domain, and then reporting the evaluation results of the other datasets of that domain. Further, we vary the model size of the distilled model (220M, 770M, 3B) and observe its effects on the OOD performance in the natural language inference domain. Finally, we conduct a study to observe the effects of our proposed penalty term on the OOD performance on the coin flip task. In this experiment, since groups are assigned to datasets, multiple datasets are concatenated to form the training set, e.g., CF2+CF4, while the rest of the datasets, i.e., CF5, CF7, and CF8, are considered OOD.

5.0.1 Implementation Details

Following (Hsieh et al., 2023), pretrained T5 models from hugging face are used for the distilled model, with CoTs generated with PaLM. We use CoT generated by the Distilling step-by-step people. Unless noted otherwise, the base T5 model with 220M parameters are used. The hyperparameter α is set to 0.5, while two values for λ are experimented with: 10e3 and 10e4. The models are fine-tuned with the AdamW optimizer (Loshchilov, 2017) with an initial learning rate of 5e-5, to 2000 steps with no early stopping. The batch size used is 64 at first for all models. However, due to GPU constraints, for some models, the batch size is lowered until there is no memory issue. Also due to resource constraints, only one run is reported per experiment. The code is implemented in PyTorch and Hugging Face based on the Distill step-by-step repository https://github.com/google-research/distilling-step-by-step, while the Fishr loss is adapted from https://github.com/alexrame/fishr.

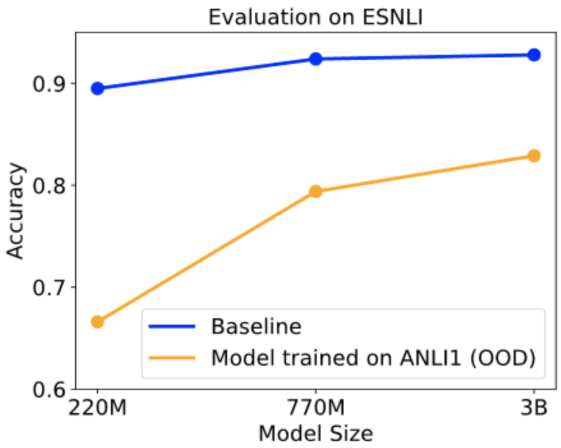


Figure 5: Caption ToDo

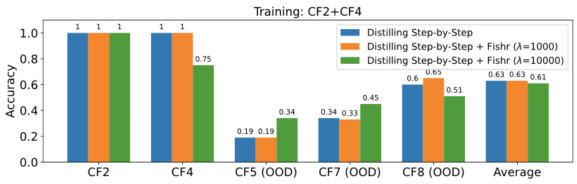


Figure 6: In-Distribution and Out-of-Distribution evaluation results with Fishr regularization after fine-tuning on the Coin Flip 2 and Coin Flip 4 datasets.

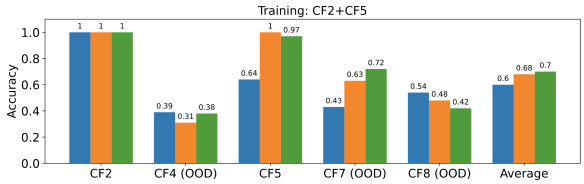


Figure 7: In-Distribution and Out-of-Distribution evaluation results with Fishr regularization after fine-tuning on the Coin Flip 2 and Coin Flip 5 datasets. The legend follows previous figures: blue, orange, and green correspond to $\lambda = 0$, 1000, and 10000, respectively.

6 Results and Discussion

6.1 OOD Experiments

Table 1 presents the results of various OOD experiments across different domains. The first row displays the performance of the T5-base model (220M parameters) fine-tuned on the leftmost dataset and evaluated on both the same dataset (in-distribution) and a different dataset (out-of-distribution). This process is repeated across the Symbolic, Math, and Natural Language Inference (NLI) domains. The second row provides baseline results, where the Distilling Step-by-Step approach is applied, but the datasets are no longer OOD (i.e., both fine-tuning and evaluation are performed on the same dataset). Finally, the third row reports the performance of a few-shot chain-of-thought (CoT) LLM, specifically PaLM (540B), on each dataset.

Overall, these OOD experiments highlight the significant challenges that fine-tuned smaller language models face when performing on OOD tasks. By comparing the OOD results in row 1 with those in row 2, we observe the impact of fine-tuning on a different dataset versus fine-tuning on the same dataset. Notably, some prior works, such as Distilling Step-by-Step (Hsieh et al., 2023), do not report OOD performance for their distilled models, instead focusing only on their larger models. This gap motivated us to conduct these experiments specifically on smaller models to better understand their OOD performance.

6.2 Model Size Experiments

Figure 5 shows the comparison of in-distribution (ESNLI) and out-of-distribution (ANLI1) performance in the Natural Language Inference domain for different model sizes. We see that as the number of parameters in the model increases, the performance gap between OOD and baseline results decreases. This suggests that smaller language models struggle with reasoning and generalization, often relying on pattern matching and other suboptimal strategies.

6.3 Distill Step-by-Step + Fishr

We now use Fishr Regularization when finetuning our model, within the Symbolic Reasoning tasks. Figures 6, 7, and 8 show the in-distribution and out-of-distribution evaluation results when finetuning on various combinations of coin-flip datasets. We plot the accuracies without any regularization (blue) and with Fishr Regularization of $\lambda = 1000$ (orange) and $\lambda = 10000$ (green).

From these experiments we notice that increas-

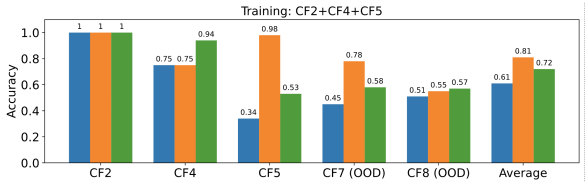


Figure 8: In-Distribution and Out-of-Distribution evaluation results with Fishr regularization after fine-tuning on the Coin Flip 2, Coin Flip 4, and Coin Flip 5 datasets. The legend follows previous figures: blue, orange, and green correspond to $\lambda = 0$, 1000, and 10000, respectively.

	Symbolic		Math		NLI	
	CF2	**CF4**	**SVAMP**	**ASDiv**	**ANLI1**	**e-SNLI**
Distilling Step-by-Step	1.00	0.75 (OOD)	0.64	0.37 (OOD)	0.50	0.67 (OOD)
Baseline	-	1.00	-	0.73	-	0.90
Few-Shot CoT LLM	1.00	0.90	0.79	0.74	0.66	0.73

Table 1: Performance comparison across datasets with Distilling Step-by-Step, Baseline, and Few-Shot CoT LLM. Symbolic, Math, and NLI domains are evaluated.

ing the amount of Fishr Regularization appears to lead to better generalization to OOD datasets. For instance, in Figure 6 we see that when $\lambda = 10000$, CF5 and CF7 (both OOD) have an increase in performance. In addition, the performance of the CF2 and CF4 in-distribution sets actually decreases or stays the same. This follows the trend of generalization, where the model gets better at evaluating OOD data, and perhaps some implicit tradeoff causes the in-distribution performance to slightly decrease. CF8 is OOD and does not follow this trend, however, it could be argued that 8-person coin flipping closely resembles 2- or 4-person coin flipping, making CF8 significantly more "in-distribution" compared to CF5 or CF7.

In Figure 6 we notice the performances of CF5 are far below 50% accuracy, which is strange given that random guessing a coin flip should give around 50% accuracy. The model seems to deliberately miscount the number of flips, and we hypothesized that this was due to the parity of both in-distribution datasets being even.

For example, it is possible that the model is counting the number of people not flipping the coin (hence counting the number of 'not' tokens). This would yield the correct answer in an even-numbered coin-flip dataset but the exact wrong answer in an odd-numbered coin-flip dataset. Another possibility is that the model determined the flip state of the coin in some way before counting the people, and then picked a number of people whose parity matches the state of the coin. When analyzing specific examples of coin flip, we saw that it would correctly identify who flipped it and the flip parity from the number of coin flippers, but would sometimes count the number of coin flippers incorrectly. This would suggest that the model relies primarily on pattern matching to generate the answer itself and the rationale.

This motivated additional experiments with distribution sets of CF2 + CF5 and CF2 + CF4 + CF5. The results are shown in Figures 7 and 8. Using both even and odd person coin flip datasets for fine tuning appears to ameliorate this issue but does not fully fix it. Figure 7 shows OOD performance on CF4 is better but still below 50%. Figure 8 does not have many accuracies below 50%, but an argument could be made that the Law of Large Numbers comes into play, and the model is mostly making guesses at random. That being said, we do see glimpses of the generalization trend with increased lambda as observed earlier, although not all of the performance data follow this trend. Thus the baseline models (no Fishr) tend to succeed more on in-distribution datasets, whereas with Fishr regularization the models tend to succeed more with out-of-distribution datasets.

7 Conclusion

In this work, we conducted comprehensive out-of-distribution (OOD) experiments on smaller language models across three domains: symbolic reasoning, mathematics, and natural language inference. Our findings reveal that these models struggle significantly with OOD evaluation, suggesting a reliance on behaviors such as pattern matching rather than true chain-of-thought (CoT) reasoning. Furthermore, we demonstrated that incorporating Fishr Regularization can, in certain scenarios, enhance OOD performance in these smaller distilled models.

We plan to extend our analysis of Fishr Regularization to additional domains to assess its broader applicability. Moreover, we aim to explore other regularization techniques and model modifications that claim to enhance generalization across diverse domains. Through these efforts, we seek to advance the understanding of how language models can be improved to reason, especially in OOD scenarios, ultimately contributing to the development of more robust language models.

8 Contributions

Thomas worked on the Symbolic and NLI OOD experiments. He implemented Fishr Regularization and ran those experiments. He created some sections of the poster, and worked on the report.

Alex worked on the Math OOD experiments. He helped Thomas with the Fishr experiments. He created the poster template and some sections of the poster, and worked on the report.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Yana Dranker, He He, and Yonatan Belinkov. 2021. Irm—when it works and when it doesn't: A test case of natural language inference. *Advances in Neural Information Processing Systems*, 34:18212–18224.

Shuohang Gao, Siyuan Zhuang, Hongyin Sun, and Jidong Ma. 2023. Llama-adapter: Efficient fine-tuning of llama models. *arXiv preprint arXiv:2303.16199*.

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

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xiao Jiang, Xusen Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4163–4174.

Jiacheng Liu, Chunting Zhou, Yelong Shen, Xuedong Huang, and Michael Zeng. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

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

Alexandre Rame, Corentin Dancette, and Matthieu Cord. 2022. Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 18347–18377. PMLR.

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*.

Kumar Shridhar, Tony Qiu, David Danks, and Yao Wu. 2022. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2205.10625*.

Kevin Sun, Ziyi Lin, Chenguang Zhu, and Xiang Ren. 2022. Distilling reasoning capabilities into smaller language models. *arXiv preprint arXiv:2204.02155*.

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

Wenhui Wang, Hangbo Bao, Shaohan Huang, and Jianfeng Gao. 2020. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Adina Williams, Tristan Thrush, and Douwe Kiela. 2020. Anlizing the adversarial natural language inference dataset. *arXiv preprint arXiv:2010.12729*.

Context	Hypothesis	Rationale
Eduard Schulte (4 January 1891 in Düsseldorf – 6 January 1966 in Zürich) was a prominent German industrialist. He was one of the first to warn the Allies and tell the world of the Holocaust and systematic exterminations of Jews in Nazi Germany occupied Europe.	Eduard Schulte is the only person to warn the Allies of the atrocities of the Nazis.	The context states that he is not the only person to warn the Allies about the atrocities committed by the Nazis.
Kota Ramakrishna Karanth (born May 1, 1894) was an Indian lawyer and politician who served as the Minister of Land Revenue for the Madras Presidency from March 1, 1946 to March 23, 1947. He was the elder brother of noted Kannada novelist K. Shivarama Karanth.	Kota Ramakrishna Karanth has a brother who was a novelist and a politician	Although Kota Ramakrishna Karanth's brother is a novelist, we do not know if the brother is also a politician
Toolbox Murders is a 2004 horror film directed by Tobe Hooper, and written by Jace Anderson and Adam Gierasch. It is a remake of the 1978 film of the same name and was produced by the same people behind the original. The film centralizes on the occupants of an apartment who are stalked and murdered by a masked killer.	Toolbox Murders is both 41 years old and 15 years old.	Both films are named Toolbox Murders one was made in 1978, one in 2004. Since it is 2019 that would make the first 41 years old and the remake 15 years old.

Figure 9: Examples from the ANLI dataset.

Premise: An adult dressed in black holds a stick. Hypothesis: An adult is walking away, empty-handed. Label: contradiction Explanation: Holds a stick implies using hands so it is not empty-handed.
Premise: A child in a yellow plastic safety swing is laughing as a dark-haired woman in pink and coral pants stands behind her. Hypothesis: A young mother is playing with her daughter in a swing. Label: neutral Explanation: Child does not imply daughter and woman does not imply mother.
Premise: A man in an orange vest leans over a pickup truck. Hypothesis: A man is touching a truck. Label: entailment Explanation: Man leans over a pickup truck implies that he is touching it.

Figure 10: Examples from the e-SNLI dataset.

Canwen Xu, Xiang Li, Yutao Sun, Dian Yu, Kai-Wei Chang Wang, and Ying Nian Wu. 2023. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. *arXiv preprint arXiv:2305.14327*.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. 2024. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. Distilling mathematical reasoning capabilities into small language models.

A Datasets