Towards Understanding the Relationship between In-context Learning and Compositional Generalization

Preprint

Abstract

According to the principle of *compositional generalization*, the meaning of a complex expression can be understood as a function of the meaning of its parts and of how they are combined. This principle is crucial for human language processing and also, arguably, for NLP models in the face of out-of-distribution data. However, many neural network models, including Transformers, have shown to struggle with compositional generalization. In this paper, we hypothesize that *learning to in-context learn* can provide the right inductive bias to promote compositional generalization. We do this by implementing a meta-learning approach that teaches a causal Transformer to utilize earlier examples to generalize to later ones: We construct a task distribution using different orderings of the training dataset and possibly shuffling the labels, which corresponds to training the model on all possible few-shot learning problems attainable from the dataset. At evaluation, we retain the zero-shot prediction setting by providing randomly sampled training examples for the model to in-context learn. Experiments on the SCAN and COGS datasets show that our method improves compositional generalization, indicating the usefulness of in-context learning problems as inductive bias for generalization.

1 Introduction

As humans, we have the ability to combine atomic parts in reoccurring structures in novel manners (Fodor and Pylyshyn, 1988). This ability, known as *compositional generalization*, is an important aspect of human language processing, affording us with an "infinite use of finite means" (Chomsky, 1965). For example, when we understand the meaning of a predicate *dax* from sentences such as “I dax” and “dax twice”, we can also understand novel sentences as “dax voluntarily” or “must dax”.

In contrast, many modern deep neural architectures struggle with compositional generalization (Baroni, 2020; Lake and Baroni, 2017; Hupkes et al., 2020; Kim and Linzen, 2020; Keysers et al., 2020). While they excel at making predictions for test sets similarly distributed to the train (i.e., *in-distribution*), their performance significantly decreases when generalizing to test distributions that are differently structured (i.e. *out-of-distribution*) even if they contain the same set of atoms.

We believe that standard models lack an inductive bias towards acquiring compositional representation, which arises from the independent parallel processing of examples in mini-batches. In most mini-batches, the models do not have *explicit* access to a sufficient number of instances of the atoms to make it worthwhile to learn compositionally generalizable representations for the atoms. Contrast this with symbolic accounts of compositional generalization, e.g., in the shape of *case-based reasoning* (Leake, 1996), where prediction can always rely on the availability of a sufficient number of relevant examples in memory. Along these lines, the ability to understand “dax thrice” from "dax twice" can be thought of as a generalization of relevant past uses of "thrice" in memory, such as “eat thrice”, combined with the use of "dax" in "dax twice".

Our hypothesis is that compositional generalization can be induced in models by forcing them to *in-context learn* (Brown et al., 2020; Chowdhery et al., 2022). *In-context learning* (ICL) refers to the ability to generalize to new examples conditioned on a few demonstrations of input-output mappings provided in the model’s context (or memory) without parameter updates. Hence, learning to in-context learn forces the model to compute in the forward pass how the past examples provided the context can be utilized in a novel manner for the later examples. We observe that it is the same mechanism that supports the learning of compositionally generalizable input-output mappings.

The intuition for our hypothesis is aligned with theoretical studies that explain in-context learning (Ortega et al., 2019; Xie et al., 2022) as an implicit Bayesian inference, where the model learns to approximate the latent parameters. However, it is yet unclear empirically how compositional generalization and in-context learning are related. On one hand, the reported improvement in compositional generalization for the large Transformer-based language models (LLMs) (Zhou et al., 2023; Hosseini et al., 2022) with *emergent* in-context learning ability seem to point to an underlying relationship. On the other, the uncontrolled training data in these studies and uncertainty regarding how much of the inductive biases implicit in the prompting methods contribute to the improvement clouds our understanding. Indeed, Hosseini et al. (2022) reported that only some in-context learning LLMs can compositionally generalize and only as they scale up.

As implementation, we propose a novel *meta-learning* (Schmidhuber et al., 1996; Bengio et al., 1991; Hochreiter et al., 2001; Duan et al., 2017; Ortega et al., 2019) regime to explicitly incentivize in-context learning for a causal Transformer (Vaswani et al., 2017; Radford et al., 2019) with the language modelling objective, training from scratch. Each task of our *meta*-task distribution is one possible linear ordering of input-output pairs of the training dataset formed into a single sequence via concatenation. This trains the model on all possible few-shot in-context learning problems attainable from the dataset. In order to discourage the model from relying on memorization, we also shuffle the labels. At prediction time, we condition the inference on the test examples on randomly sampled training mappings, maintaining the *zero-shot* prediction setting. We evaluate our approach on two widely used datasets targeting specifically compositional generalization, namely SCAN (Lake and Baroni, 2017) and COGS (Kim and Linzen, 2020).

Our main contributions are as follows:

1. We empirically study the relationship between in-context learning and compositional generalization through a novel meta-learning training regime for sequence to sequence datasets that incentivizes in-context learning and a corresponding evaluation regime that maintains a zero-shot prediction setting.
2. We show that our Transformer trained through meta-in-context learning shows a significant improvement in performance on compositional generalization across the datasets compared to the baseline without meta-learning.
3. We demonstrate how exactly the ability to in-context learn is related to compositional generalization through ablations. We show that 1) the performance of the model increases when it is trained on more diverse range of few-shot in-context learning problems 2) the effect of the providing more demonstrations leads to improved compositional generalization 3) they are able to learn from new distributions.

The paper is organized as follows. §2 introduces important background concepts and reviews notable related works. §3 presents our meta-learning regime in detail. §4 provides information on experimental setup, followed by the results in §5. §6 concludes the paper along with future directions.

2 Related Works

2.1 Compositional Generalization

Difficulties of neural networks to compositionally generalize have been identified by many studies.[1]. Notable text-to-text benchmarks include SCAN (Lake and Baroni, 2017), PCFG (Hupkes et al., 2020), COGS (Kim and Linzen, 2020), and CFQ (Keysers et al., 2020). These datasets are split into train and test systematically that requires a compositional solution to be successful.

Many studies have proposed different inductive biases to promote compositionality. They include new deep learning architectures structurally constraining how the inputs are processed and represented (Li et al., 2019; Russin et al., 2019; Gordon et al., 2020; Bergen et al., 2021), providing additional supervisory signals (Jiang and Bansal, 2021), data augmentation (Andreas, 2020; Guo et al., 2020b; Akyürek et al., 2021; Qiu et al., 2022), and hybrid symbolic reasoning approaches (Nye et al., 2020; Liu et al., 2020b; Guo et al., 2020a). These approaches have shown to improve compositional generalization. However, they often require prior knowledge of the dataset, and their scalability to bigger and more general datasets is uncertain.

Following these concerns, some studies have constrained their investigations to the most popular neural sequences model such as the Transformer (Ontanon et al., 2022; Csordás et al., 2021), finding that their compositional generalization capacity can be improved with the available variants (e.g. relative positional encoding (Dai et al., 2019) or tying the layers (Dehghani et al., 2019). Patel et al.

[1] We focus on studies on unimodal language data.

(2022) showed that popular architectures including the Transformer can be improved by increasing diversity in the data distribution. Finally, Herzig et al. (2021) showed that better formatting of tasks using bespoke representation can lead to improvement.

2.2 Meta-learning

Meta-learning (Bengio et al., 1991; Schmidhuber et al., 1996) aims at enabling machine learning models to learn how to learn by exposing them to a *distribution of tasks* where one can improve from past experience. The tasks are selected to be similarly structured but differ in details such that it is profitable for the model to find a generalizable solution rather than memorize examples. Our work follows the line of work known as *memory-based meta-learning* or *meta-in-context learning* (Hochreiter et al., 2001; Santoro et al., 2016; Duan et al., 2017; Wang et al., 2017; Ortega et al., 2019), which incentivizes the model to learn to in-context learn by training on a task distribution of *sequences* of input-output mappings.

Meta-learning was applied to various application tasks in language processing such as cross-lingual transfer (Gu et al., 2018), question answering (Nooralahzadeh et al., 2020), and domain adaption (Qian and Yu, 2019). However, it has rarely found application in semantic processing. The challenge arises from the difficulty of not knowing beforehand the relevance of specific examples, which makes it difficult to construct the task distribution with the right inductive bias for compositional generalization. Lake (2019) evaded this problem by using the ground truth grammar of the data distribution. This allowed them to permute only the input-output mappings of the primitives, which was shown to improve compositional generalization. Conklin et al. (2021) used MAML (Finn et al., 2016) as an auxiliary loss for supervised learning which alleviated the problem of selecting support examples, but still relied on ground truth structural knowledge. We discuss how we overcome these challenges in the next section.

2.3 In-context Learning

There is a long line of work attempting to understand the property of in-context learning, especially related to their ability to generalize to out-of-distribution. A number of studies has shown that in-context learning in LLMs can be utilized for compositional generalization using specific prompting methods (Zhou et al., 2023; Wei et al., 2022; Fu et al., 2023), especially when the model is scaled up (Hosseini et al., 2022). As explained above, the in-context learning ability in these models was also analyzed theoretically, and the driving force was found to be latent text properties that heavily affect token distributions (Xie et al., 2022).

Our work is closest to previous studies that train Transformers from scratch using meta-learning instead of looking at LLMs. Chan et al. (2022) showed that the emergence of in-context learning to depend on the informativeness of the contexts. Garg et al. (2022) showed that Transformers are able to in-context learn simple functions and generalize to out-of-distribution samples, and Kirsch et al. (2022) extended for learning to in-context learn arbitrary image-label mappings.

3 Methods

We now introduce a meta-learning regime that can be generally applied to a sequence to sequence dataset consisting of input-output sequence pairs. The main goal is how to construct a meta task-distribution with the right inductive bias for compositional generalization. The key idea is the inductive bias created by the *online* learning of the entire dataset: The model observes each example in the dataset only once and sequentially one after the other. This means that the model cannot memorize when learning on such a linear ordering of examples (i.e., trajectory) and needs to successfully store and represent the past examples for generalization for the future examples.

Since there is no inherent order between the examples in a sequence to sequence dataset, a different linear ordering of the dataset poses a different generalization problem for the model. However, no matter which ordering we choose, the structure behind each trajectory remains invariant as it is governed by the same latent parameters. Hence, when meta-learning on such a distribution, the model has a chance of approximating the underlying structure of the dataset. Note that this way of constructing the task distribution do not require any prior knowledge of the dataset, in contrast to earlier approaches (Lake and Baroni, 2017; Conklin et al., 2021).

3.1 Meta-training

Given a sequence to sequence dataset $D = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$ with a vocabulary V, we form the task distribution $P(\tau)$ for meta-learning, where each task τ is one possible linear ordering of

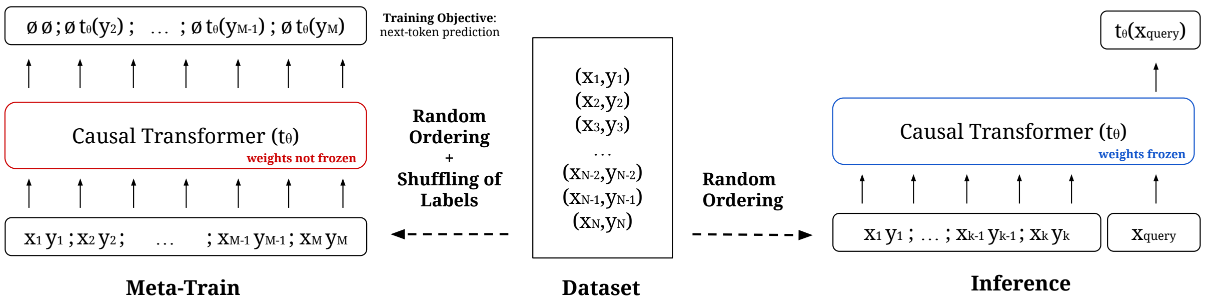
ø ø ; ø tθ(y2) ; … ; ø tθ(yM-1) ; ø tθ(yM)
Training Objective: next-token prediction
Causal Transformer (tθ)
weights not frozen
x1 y1 ; x2 y2 ; … ; xM-1 yM-1 ; xM yM
Random Ordering + Shuffling of Labels
(x1,y1)
(x2,y2)
(x3,y3)
…
(xN-2,yN-2)
(xN-1,yN-1)
(xN,yN)
Random Ordering
tθ(xquery)
Causal Transformer (tθ)
weights frozen
x1 y1 ; … ; xk-1 yk-1 ; xk yk
xquery
Meta-Train
Dataset
Inference

Figure 1: Illustration of our meta-in-context learning framework. (Left) We build our meta-task distribution by sampling a random linear ordering of a sequence to sequence dataset and concatenating the input-output mappings (i.e. (x_i, y_i)). We possibly shuffle the labels to eliminate memorization and keep only M examples. Then a causal Transformer (t_θ) is trained with these concatenated results for next-token prediction, only predicting for the outputs. ϕ refers to the pad-token. (Right) At inference, we freeze the weights and randomly sample $k < M$ train examples to condition the prediction of a test query example x_{query}.

the dataset $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})$. We feed this to the model as a concatenation $\tau = [\mathbf{x}^{(1)}; \mathbf{y}^{(1)}; \ldots; \mathbf{x}^{(N)}; \mathbf{y}^{(N)}]$ using two delimiter tokens, one to distinguish the inputs from the outputs and another to separate the sequence elements. We assume a uniform distribution for $P(\tau)$.

However, one limitation of this approach is the possibility of memorization as each example occurs only once *within* each trajectory, but many times *across* different trajectories. Hence, a model might learn to ignore the context and memorize the examples, which is especially true for small datasets. To counteract this danger, we randomly shuffle the labels of the vocabulary V. For example, given a dataset $\{(a\ b, A\ B), (a\ d, A\ D)\}$, we can create an alternate version of the dataset $\{(a\ b, B\ A), (a\ d, B\ D)\}$ by the shuffling.

Formally, we train the model M_θ given a possible linear ordering sequence τ to predict the next token $f(\tau[i]) = \tau[i+1]$ if the i-th token belongs to the output and a pad token $f(\tau[i]) = \phi$ if it belongs to the input or the first output of the sequence. Hence, the model is trained to minimize the expected loss over all possible orderings of the dataset, possibly extended with label shuffling:

$$\min_\theta \mathrm{E}_{\tau \sim P(\tau)}[\sum_{i=1}^{|\tau|} \frac{1}{|\tau|} \ell(M_\theta(\tau[i]), f(\tau[i]))] \quad (1)$$

where $\ell(\cdot, \cdot)$ is the cross-entropy loss function. Figure 1 (left half) illustrates the training procedure.

This objective can be interpreted as training the model on all possible few-shot learning problems attainable from the dataset. Hence, the strength of the inductive bias for compositional generalization is limited to the kinds of generalization problem inherent in each dataset. A final practical problem is that most datasets do not fit entirely into memory. Hence, we fix a certain roll-out length $M < N$ to limit each sequence τ to consist of M input-output pairs. Note that as we make M smaller, the number of distinct tasks in the task distribution decreases. We investigate the choice of M in Exp. 2 below.

Underlying Neural Network. This meta-training can be applied to any neural network model with memory. However, the use of an autoregressive model is very advantageous: In such a model, a single trajectory consisting of N concatenated input-output mappings can be provided with a causal masking to encompass $k-1$ few-shot learning problems. In a bidirectional model, in contrast, one needs to provide $k-1$ different problems separately to the model. Hence, we adopt the causal Transformer (Vaswani et al., 2017; Radford et al., 2019) in our work.

3.2 Inference

Compositional generalization datasets are designed to be a zero-shot generalization task. This means that the model is required to generalize to the test examples only by using the train examples. Since we do not assume any prior knowledge to determine the possible relevance between examples, we randomly sample a *training* trajectory of length $k < M$ to condition the inference for each test input $\mathbf{x}_q$. Note that though we cannot guarantee the relevance of all the samples, the model can still *choose* among these samples through the attention mechanism, analogously to case-based reasoning (Leake, 1996). See Figure 1 (right half) for the illustration of our inference method. We investigate the implications of the choice of k in Exp. 3 below.

4 Experimental Setup

4.1 Datasets

SCAN (Lake and Baroni, 2017) consists of natural language commands that needs to be mapped to a sequence of actions (e.g. jump twice → JUMP JUMP). The commands were generated using a phrase structure grammar without recursion and mapped to the actions using semantic interpretation rules. Among various compositional generalization *splits* of SCAN, we use the Maximum Compound Divergence (MCD) splits introduced by Keysers et al. (2020). These splits capture the notion of compositional generalization by maximizing the divergence between the compounds while maintaining the closeness of the atom frequency distribution. There are three SCAN-MCD splits with increasing difficulty (i.e. MCD1 being the easiest), each with 8365 train and 1045 test examples.

COGS (Kim and Linzen, 2020) is a semantic parsing dataset with diverse natural language sentences. The compositional generalization test set was constructed based on different kinds of linguistic generalizations that humans are able to make (e.g. generalizing subject role → object role). The training set consists of 24155 examples while 21000 examples make up the test.

Preprocessing. For both datasets, we preprocess the output sequences to reduce their lengths in order to be able to fit longer trajectories (i.e. higher M). For SCAN, we represent the action sequences in Python syntax as it was done in (Zhou et al., 2023) for evaluating LLMs. For example, "LOOK LOOK" is represented as LOOK * 2. For COGS, we simply omit the brackets and represent the variables x_n as n. Both are intermediate representations that can be mapped fully back to the original form. See Appendix C for more details.

4.2 Model Configuration and Training Details

We use an 8-layer 8-head causal Transformer with model dimension of 512 and feedforward dimension of 2048 with absolute sinusoidal positional encoding (Vaswani et al., 2017), using the basic implementation available in PyTorch (Paszke et al., 2019). We do not use any pre-trained weights, and initialize the model and the word embeddings from scratch. Appendices A and B provide details on hyperparameters and checkpoint selection.

For SCAN, we apply shuffling of output labels to completely eliminate the effect of memorization, as its relatively small vocabulary ($|\mathrm{V}| = 30$) affords us the full coverage of all the words with a few samples. Since COGS has a much bigger vocabulary ($|\mathrm{V}| = 871$), we do not apply shuffling here.

4.3 Evaluation

For evaluation, we report on *sequence-level* accuracy, where a sequence is only deemed correct if it is predicted entirely correctly. For each accuracy result, we also report the number of randomly sampled training examples k used for testing. If not mentioned, we set k to be one less than the maximum roll-out length M. All results report averages over five training runs for SCAN and three for COGS (for computational reasons).

4.4 Baselines and Points of Comparison

Herzig et al. (2021) showed that intermediate representations can lead to an improved compositional generalization. This especially applies to SCAN. Hence, we additionally train a 3-layer encoder-decoder Transformer (Vaswani et al., 2017) and Universal Transformer (Dehghani et al., 2019) with absolute positional encoding for SCAN. On the other hand, the impact of preprocessing on COGS is minimal and is different from the format found to be useful, which converted the task to sequence tagging (Ontañón et al., 2022). Hence, we use Transformer and Universal-Transformer results reported in literature for COGS.

For both datasets, we also report on a causal Transformer baseline trained using standard supervised learning, which is equivalent to training with the roll-out length of 1 (i.e. $M = 1, k = 0$). Finally, we compare our approach with the prior meta-learning work: the MAML-augmented Transformer with Tree-based search for COGS and string-based for SCAN (Conklin et al., 2021). See Appendices A and B for all further details.

5 Experiments and Results

We now present the results of four experiments to evaluate our approach and to better understand the relationship between in-context learning and compositional generalization: (1) We compare the performance of our models with the baselines using $k = M - 1$ for evaluation. (2) We test the effect of training the model with longer trajectories (i.e.

Method	Bidir	IntRep	SCAN			COGS
			MCD1	MCD2	MCD3	
Transformer (lit.)	+	-	0.4± 0.4 [1]	1.8± 0.4 [1]	0.5± 0.1 [1]	35 ± 6 [2]
Transformer (ours)	+	+	41.7 ± 4.7	20.3 ± 5.6	17.1 ± 6.1	**80** ± 0.0 [3]
Universal Transformer (ours)	+	+	36.4 ± 9.0	34.1 ± 6.6	25.5 ± 10.4	78 ± 0.0 [3]
Transformer + MAML (lit.)	+	-	2.6 ± 0.6 [4]	5.6 ± 1.6 [4]	6.7 ± 1.8 [4]	66.7 ± 4.4 [4]
Causal Transformer (ours)	-	+	21.8 ± 3.7	25.6 ± 2.6	19.7 ± 2.1	51.9 ± 4.3
Causal Transformer + meta-ICL	-	+	**71.2** ± 7.3	**74.8** ± 9.7	**38.7** ± 8.3	75.7 ± 1.9

Table 1: Exp. 1: Mean sequence-level accuracies and standard deviations across runs. For our meta-ICL causal Transformer, we use $M = 25$, $k = M - 1$. "Bidir" stands for bidirectional (vs. causal). "IntRep" indicates the use of the optimized intermediate representation for SCAN. Best model on each dataset boldfaced. References: [1] Furrer et al. (2021), [2] Kim and Linzen (2020), [3] Csordás et al. (2021), [4] Conklin et al. (2021).

bigger M) which results in training with a bigger and more diverse task distribution. (3) We test the effect of varying the number of support *training* examples used during evaluation (i.e. varying k). (4) We test the ability of the resulting models to learn from a new distribution by providing the model with $test$ examples.

5.1 Exp. 1: Main results

Table 1 summarizes the main results of a causal Transformer trained from scratch using our meta-training method, along with the baselines.

We first make the observation that our intermediate representation for SCAN leads to an improvement in compositional generalization (compare rows 1 and 2). Although the improvement is substantial, the datasets still remain difficult for the models and the relative difference of difficulty between the MCD splits are retained (see the decreasing performance for SCAN in columns 3-5). We also note that our causal Transformer baseline (row 5) performs worse than the encoder-decoder counterparts, probably due to its unidirectionality.

For all three MCD splits, our causal Transformer trained with meta-in-context learning (row 6) substantially outperforms all other approaches. For COGS, it is not able to beat the Transformer model of Csordás et al. (2021), but it is able to outperform others. This result provides a positive evidence for our hypothesis that the in-context learning ability of Transformers encompasses compositional generalization. This is especially interesting for COGS as no shuffling of labels was used, hence the improvement gain compared to its causal baseline came simply from how the data was being presented to the model. We believe that training on linear orderings of examples resulted in a form of regularization, where the pressure to learn representations not only for predictions but also for their use in the future contributed to the improvement.

5.2 Exp. 2: Training on Longer Trajectories

Setup. Next, we investigate how the performance of the model changes when training on trajectories of different lengths. This is interesting because constructing the task distribution with longer trajectories affords us with more unique few-shot learning problems, which might lead to better compositional generalization. However, longer trajectories could also lead to a higher chance of overfitting, as the model needs to extrapolate less when given more support samples. Hence, we train three different values of $M = \{10, 25, 50\}$ for SCAN and $\{5, 10, 25\}$ for COGS, evaluating with $k = M - 1$.
Result. The results in Figure 2 show an improvement for MCD1 and MCD2 of SCAN from increasing the trajectory length from 10 to 25, but we see signs of overfitting as we increase further. For MCD3, we see a monotonic improvement as we increase the trajectory length. For COGS, we see a similar trend with increasing performance as the length of the trajectories are increased.

This confirms our hypothesis that training on bigger and more diverse meta task distribution can lead to an improvement. The gain diminishes after a certain point because the task becomes easier for the model. Where this turning point occurs is dependent on the available few-shot learning problems implicit in each dataset, but this could be tuned using a standard hyperparameter search. Thus, the in-context learning ability of Transformers is sensitive to the kinds of few-shot generalization problems

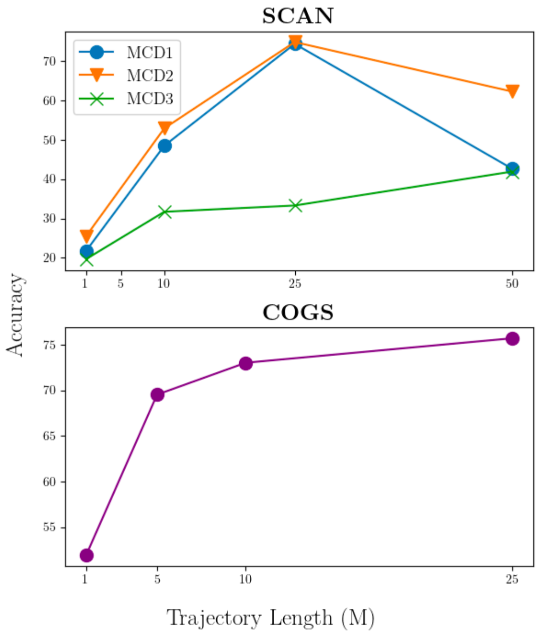

Figure 2: Exp. 2: Results for models trained on different lengths of trajectories (i.e. M), with $k = M - 1$. $M = 1$ is equivalent to the causal Transformer baseline.

that it is exposed to during training.

5.3 Exp. 3: Fewer vs. More Support Examples

Setup. So far, we have only used $k = M - 1$ number of support examples for evaluation. We also need to investigate how the model generalizes for different number of support examples – e.g., to rule out that the improvement that we saw in the previous experiment was simply due to the models with higher M having access to more support examples during evaluation. A setup that avoids this confound compares generalization performance of the models trained with different trajectory lengths using the same k. To do so, we evaluate the SCAN-MCD models using different values of $k = \{1, 3, 5, 9, 12, 16, 24, 49\}$ using the full test set. For COGS, we take 20% of the test set and vary $k = \{1, 4, 9, 24\}$[2]. Note that we only evaluate when k does not exceed the maximum roll-out length for each given model (i.e. $k < M$).

Result. Figure 3 shows the result. For SCAN, all models improve as it receives more and more support examples, though it starts to plateau around nine examples. The result also confirms the conclusion drawn in Exp. 2: For all MCD splits of SCAN, even when the model is given the same number of support examples, the best model performs better than the others (i.e. 25 for MCD1 and MCD2 and 50 for MCD3), suggesting that these models did learn more generalizable in-context learning.

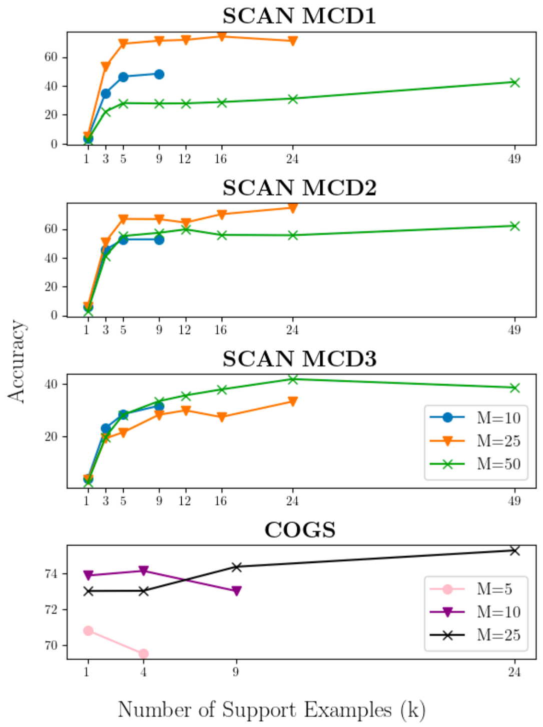

Figure 3: Exp. 3: Results for models evaluated with different number of support examples (i.e. k). M is maximum roll-out length of meta-training trajectories.

For COGS, however, we do not see such a clear trend, except for $M = 25$. This can be attributed to two factors: memorization and the informativeness of support examples. Since no shuffling was applied, the model memorized the examples, though it was being regularized in doing so. Also, it is much less likely for the support example to be informative as the dataset is much diverse compared to SCAN. We assume this is why we only see such a trend for the model trained with the large value of M (i.e. $M = 25$) where the likelihood of informative examples in the context becomes higher, which allows the model to retrieve relevant examples.

5.4 Exp. 4: Ability to Learn from New Distribution

Setup. Finally, we test how general the ability of our models to in-context learn by testing whether they can learn from new distribution. For the SCAN splits, we hold out 49 examples from the *test* set to sample our support examples during evaluation. We chose the value 49 as this is the maximum

[2] This was done as the test set of COGS is 21 times larger than SCAN. However, we found this to be representative of evaluating on the entire test set.

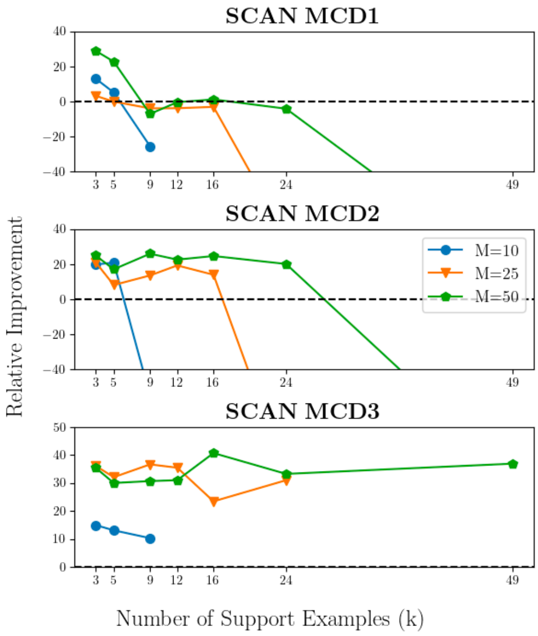

Figure 4: Exp. 4: Results for models evaluated with different number of support examples (i.e. k) sampled from the held-out portion of the *test* set. Relative improvement (RI) is calculated using this formula: $\frac{new-old}{old} \times 100$. For all models, $k = 1$ leads to a significant positive RI (omitted for clarity).

value of k for any model. We then evaluate our models on the rest of the test set by sampling from the held out *test* examples to form the context. We repeat Exp. 3 and report on the relative improvement (RI), $\frac{new-old}{old} \times 100$.

Result. As Figure 4 shows, for all models, there exists a value of k where the model shows an improved performance. This suggests that the in-context learning ability is *general* to a degree, being able to learn from *test* examples. This is especially true for MCD2 and MCD3, where the model improves for most (MCD2) and all (MCD 3) values of k. This is interesting, because these are the more difficult splits of SCAN, where the *test* support examples can be much more informative. However, we see that for MCD1 and MCD2 models, the model catastrophically fails when the maximum number of test examples are provided in its context. Upon qualitative inspection, we observe that all fail to predict the end-of-sequence token properly, after which the whole sequence prediction counts as wrong (cf. Section 4.3). This is probably due to the model having to predict in positions far exceeding the maximum value of position seen during training, which has been known to be difficult for neural sequence models including Transformers (Newman et al., 2020; Csordás et al., 2021).

6 Conclusion

In this paper, we have studied the emergence of compositional generalization from a meta-learning regime which forces a sequence-to-sequence model to learn to 'in-context learn'. We have investigated performance on two difficult datasets: the MCD splits of SCAN and COGS. Our main results showed that the meta-trained models show substantially better compositional generalization than the baselines in SCAN and closely matching in COGS. Different from related prior work such as Conklin et al. (2021), we completely eliminate the possibility of memorization through label shuffling to better understand the impact of in-context learning on compositional generalization without any knowledge about the structure of the datasets. Furthermore, we sample from the training set for the model to also in-context learn for inferencing.

Our results provide positive evidence that in-context learning can induce compositional generalization. We confirm this relationship through various ablative studies, illustrating the effect of changing the task distribution and providing different number and type of support examples for the model to in-context learn. In this way, our study represents one step towards to a deeper understanding of in-context learning. This can arguably improve our handling of out-of-distribution generalization, which is a fundamental challenge for effective machine learning (Ye et al., 2023), as well as making our learning models more plausible on the cognitive side, given the very limited memorization capabilities of humans (Fodor and Pylyshyn, 1988).

In future work, we plan to investigate the effect of using relative positional encoding (Dai et al., 2019) which are widely applied in recent LLMs (Chowdhery et al., 2022) as it has been shown to improve compositional generalization (Ontañón et al., 2022). Second, we believe that it would be worthwhile to investigate the effect of equipping the model with retrieval method to better choose the support examples in the future.

Ethics Statement

Our work is concerned with foundational questions of learning generalizable models. It does not introduce new risks, nor does it involve sensitive applications. We do not use any pre-trained models, and the used datasets are publicly available. Computational costs for the required training are relatively cheap. In sum, we do not believe there to be substantive ethical concerns regarding our work.

References

Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. 2021. Learning to recombine and resample data for compositional generalization. In *Proceedings of International Conference on Learning Representations*.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. ArXiv:1607.06450.

Marco Baroni. 2020. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375(1791):20190307.

Y. Bengio, S. Bengio, and J. Cloutier. 1991. Learning a synaptic learning rule. In *International Joint Conference on Neural Networks*, volume ii, Seattle, WA.

Leon Bergen, Timothy O' Donnell, and Dzmitry Bahdanau. 2021. Systematic generalization with edge transformers. In *Advances in Neural Information Processing Systems*, volume 34, pages 1390–1402. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*, volume 33, pages 1877–1901.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, Jay McClelland, and Felix Hill. 2022. Data Distributional Properties Drive Emergent In-Context Learning in Transformers. In *Proceedings of NeurIPS*.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. The MIT Press, Cambridge.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. ArXiv:2204.02311.

Henry Conklin, Bailin Wang, Kenny Smith, and Ivan Titov. 2021. Meta-Learning to Compositionally Generalize. 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 3322–3335, Online. Association for Computational Linguistics.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. 2021. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2019. Universal transformers. In *Proceedings of International Conference on Learning Representations*.

Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. 2017. RL^2: Fast reinforcement learning via slow reinforcement learning. ArXiv:1611.02779.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. 2016. Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization. ArXiv:1603.00448 [cs].

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and Cognitive Architecture: A Critical Analysis. *Cognition*, 28:3–71.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *Proceedings of the International Conference on Learning Representations*.

Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. 2021. Compositional Generalization in Semantic Parsing: Pre-training vs. Specialized Architectures. ArXiv:2007.08970 [cs].

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems*, volume 35, pages 30583–30598. Curran Associates, Inc.

Jonathan Gordon, David Lopez-Paz, Marco Baroni, and Diane Bouchacourt. 2020. Permutation equivariant models for compositional generalization in language. In *Proceedings of International Conference on Learning Representations*.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Yinuo Guo, Zeqi Lin, Jian-Guang Lou, and Dongmei Zhang. 2020a. Hierarchical poset decoding for compositional generalization in language. In *Advances in Neural Information Processing Systems*, volume 33, pages 6913–6924. Curran Associates, Inc.

Yinuo Guo, Hualei Zhu, Zeqi Lin, Bei Chen, Jian-Guang Lou, and Dongmei Zhang. 2020b. Revisiting iterative back-translation from the perspective of compositional generalization. In *Proceedings of AAAI*.

Jonathan Herzig, Peter Shaw, Ming-Wei Chang, Kelvin Guu, Panupong Pasupat, and Yuan Zhang. 2021. Unlocking Compositional Generalization in Pre-trained Models Using Intermediate Representations. ArXiv:2104.07478 [cs].

Sepp Hochreiter, A. Steven Younger, and Peter R. Conwell. 2001. Learning to learn using gradient descent. In *Proceedings of the International Conference on Artificial Neural Networks*, page 87–94, Berlin, Heidelberg. Springer-Verlag.

Arian Hosseini, Ankit Vani, Dzmitry Bahdanau, Alessandro Sordoni, and Aaron Courville. 2022. On the compositional generalization gap of in-context learning. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 272–280, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795.

Yichen Jiang and Mohit Bansal. 2021. Inducing transformer's compositional generalization ability via auxiliary sequence prediction tasks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6253–6265, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *Proceedings of ICLR*.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Louis Kirsch, James Harrison, Jascha Sohl-Dickstein, and Luke Metz. 2022. General-purpose in-context learning by meta-learning transformers. In *Sixth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*.

Brenden M Lake. 2019. Compositional generalization through meta sequence-to-sequence learning. In *Proceedings of NeurIPS*, volume 32. Curran Associates, Inc.

Brenden M. Lake and Marco Baroni. 2017. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of ICML*.

David B. Leake. 1996. *Case-Based Reasoning: Experiences, Lessons and Future Directions*, 1st edition. MIT Press, Cambridge, MA, USA.

Yuanpeng Li, Liang Zhao, Jianyu Wang, and Joel Hestness. 2019. Compositional generalization for primitive substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4293–4302, Hong Kong, China. Association for Computational Linguistics.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020a. On the variance of the adaptive learning rate and beyond. In *Proceedings of the International Conference on Learning Representations*.

Qian Liu, Shengnan An, Jian-Guang Lou, Bei Chen, Zeqi Lin, Yan Gao, Bin Zhou, Nanning Zheng, and Dongmei Zhang. 2020b. Compositional Generalization by Learning Analytical Expressions. ArXiv:2006.10627 [cs].

Benjamin Newman, John Hewitt, Percy Liang, and Christopher D. Manning. 2020. The EOS decision and length extrapolation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 276–291, Online. Association for Computational Linguistics.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4547–4562, Online. Association for Computational Linguistics.

Maxwell I. Nye, Armando Solar-Lezama, Joshua B. Tenenbaum, and Brenden M. Lake. 2020. Learning Compositional Rules via Neural Program Synthesis. ArXiv:2003.05562 [cs].

Santiago Ontanon, Joshua Ainslie, Zachary Fisher, and Vaclav Cvicek. 2022. Making transformers solve compositional tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3591–3607, Dublin, Ireland. Association for Computational Linguistics.

Santiago Ontañón, Joshua Ainslie, Vaclav Cvicek, and Zachary Fisher. 2022. Making Transformers Solve Compositional Tasks. ArXiv:2108.04378 [cs].

Pedro A. Ortega, Jane X. Wang, Mark Rowland, Tim Genewein, Zeb Kurth-Nelson, Razvan Pascanu, Nicolas Heess, Joel Veness, Alex Pritzel, Pablo Sprechmann, Siddhant M. Jayakumar, Tom McGrath, Kevin Miller, Mohammad Azar, Ian Osband, Neil Rabinowitz, András György, Silvia Chiappa, Simon Osindero, Yee Whye Teh, Hado van Hasselt, Nando de Freitas, Matthew Botvinick, and Shane Legg. 2019. Meta-learning of Sequential Strategies. ArXiv:1905.03030 [cs, stat].

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of NeurIPS*, pages 8024–8035. Curran Associates, Inc.

Arkil Patel, Satwik Bhattamishra, Phil Blunsom, and Navin Goyal. 2022. Revisiting the compositional generalization abilities of neural sequence models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 424–434, Dublin, Ireland. Association for Computational Linguistics.

Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2639–2649, Florence, Italy. Association for Computational Linguistics.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Pawel Nowak, Tal Linzen, Fei Sha, and Kristina Toutanova. 2022. Improving compositional generalization with latent structure and data augmentation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4341–4362, Seattle, United States. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Jake Russin, Jason Jo, Randall C. O'Reilly, and Yoshua Bengio. 2019. Compositional generalization in a deep seq2seq model by separating syntax and semantics. ArXiv: 1904.09708.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *Proceedings of ICML*, page 1842–1850, New York, NY.

Juergen Schmidhuber, Jieyu Zhao, and Marco Wiering. 1996. Simple principles of metalearning. Technical report, Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Jane X. Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. 2017. Learning to reinforcement learn. ArXiv:1611.05763 [cs, stat].

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 *Proceedings of NeurIPS*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An Explanation of In-context Learning as Implicit Bayesian Inference. In *Proceedings of ICLR*.

Nanyang Ye, Lin Zhu, Jia Wang, Zhaoyu Zeng, Jiayao Shao, Chensheng Peng, Bikang Pan, Kaican Li, and Jun Zhu. 2023. Certifiable out-of-distribution generalization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):10927–10935.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *Proceedings of ICLR*.

A Hyperparameters and Computing Resource

Causal Transformer We use the PyTorch (Paszke et al., 2019) implementation of RAdam (Liu et al., 2020a) as our choice of optimizer with the learning rate of 1×10^{-4} and $\beta = (0.9, 0.99)$ for all of our experiments. For stable training, we apply linear warm-up for 500 steps for SCAN and 5000 for COGS. We clip the gradient whenever the norm exceeds 5. We apply dropout rate of 0.1, ReLU activation, and batch-size of 5. Low batch size was used due to the limited available computing resource. As parameters are initialized according to the default initialization method of PyTorch along with word embeddings. This means that the word embeddings are initialized by drawing from a standard normal. The embeddings have the same dimension as the model. We use the variant where the LayerNorm (Ba et al., 2016) is applied before each sub-block for SCAN and a normal configuration for COGS. The resulting model size is 25.2 million parameters for both datasets. For the causal baseline (i.e. $M = 1$), the use of low batch-size leads to unstable training, hence we increase the batch-size to 256. Finally, we note that we did not perform any systematic hyperparameter tuning and most of the used hyperparameters were initial guesses.

Transformer and Universal Transformer Both Transformer and Universal Transformer baselines are a 3-layer encoder-decoder architecture, and it is adapted from the code release of Csordás et al. (2021). We use the same learning rate 1×10^{-4} and $\beta = (0.9, 0.99)$ using Adam optimizer (Kingma and Ba, 2014) and not the default learning rate value of PyTorch as we saw the results to be more stable. The dimension of 128 is used for both model state and word embeddings with 8 heads and 256 for feed-forward dimension. We use the dropout rate of 0.1 and batch size of 256.

Computing Resources We used a single GeForce RTX 2080 Ti 11G for our SCAN experiments and a single GeForce GTX TITAN X 12G for our COGS experiments.

B Checkpoint Selection for Evaluation

For SCAN, we follow the checkpoint selection method of Conklin et al. (2021) and use the available development set to pick the checkpoint for testing by training for 20k steps evaluating every 1000 steps. Usually, each model with meta-training takes around 10k steps to converge. The causal baseline, Transformer, and Universal Transformer all take much less time to converge, hence we only train for 10k steps.

For COGS, we simply train the models for 150k steps and take the last checkpoint for evaluation. This was similarly done in Conklin et al. (2021), but they use 10% of the test set to tune their hyperparameters. There is a validation set associated with the training set in COGS, but it is a widely known that tuning on this set does not work well Csordás et al. (2021) as the model continues to improve on the test even when the model scores perfectly on the train and validation.

C Datasets and Preprocessing

SCAN The preprocessing decreases the average output length of the dataset from 14.3 to 12.2 and the maximum sequence length from 48 to 17. In the new format, the overall vocabulary size of SCAN is 30 with 11 output words, 4 special symbols and 15 input words. Table 2 shows a few examples of results.

COGS The resulting preprocessing is illustrated in Table 3. Before preprocessing, the average length is 51.07 and maximum of 175 which becomes 28.01 and 96 respectively after preprocessing. The resulting vocabulary size is 871.

Command (Input)	**Before (Output)**	**After (Output)**
run twice	RUN RUN	RUN * 2
jump after run	RUN JUMP	RUN + JUMP
jump around right	RTURN JUMP RTURN JUMP RTURN JUMP RTURN JUMP	(RTURN JUMP) * 4
jump around right and walk twice	RTURN JUMP RTURN JUMP RTURN JUMP RTURN JUMP WALK WALK	(RTURN JUMP) * 4 + WALK * 2

Table 2: Example SCAN action sequences (outputs) before and after preprocessing.

Sentence (Input)	**Before (Output)**	**After (Output)**
A rose was helped by a dog .	rose (x_1) AND help . theme (x_3 , x_1) AND help . agent (x_3 , x_6) AND dog (x_6)	rose 1 AND help . theme 3 1 AND help . agent 3 6 AND dog 6
Charlie loaned the cake in a house to the girl .	* cake (x_3) ; * girl (x_9) ; loan . agent (x_1 , Charlie) AND loan . theme (x_1 , x_3) AND loan . recipient (x_1 , x_9) AND cake . nmod . in (x_3 , x_6) AND house (x_6)	* cake 3 ; * girl 9 ; loan . agent 1 Charlie AND loan . theme 1 3 AND loan . recipient 1 9 AND cake . nmod . in 3 6 AND house 6

Table 3: Example COGS semantic parsing results before and after preprocessing.