Compositional Learning of Addition in Neural Networks

Interdisciplinary Biosciences Rotation 1 Report

Mia Whitefield
mia.whitefield@linacre.ox.ac.uk

March 2023

Department of Experimental Psychology
University of Oxford

Supervisor Sophie Arana (sophie.arana@psy.ox.ac.uk)
Supervisor Chris Summerfield (christopher.summerfield@psy.ox.ac.uk)

Abstract

Compositional learning, the ability to recompose elements of prior knowledge to produce novel combinations, is a characteristic feature of human cognition. However, neural networks typically fail to match human performance in tasks that demand this capability. Recent studies have used primitive training to encourage human-like compositional generalisation in neural networks. A deeper understanding of how this occurs may provide insights into the mechanisms underlying compositional learning in the brain. In this study, we compared the performance of networks that had received primitive training vs. those that did not, in an arithmetic task and found that primitive training improved their sample efficiency, systematicity and productivity. We also studied their hidden layer representations and found that the different training curricula gave rise to distinct representational geometries.

Keywords— Compositional learning; generalisation; shaping; primitive training; representational geometry

Introduction

Compositional learning is the ability to recompose elements of prior knowledge to generate new combinations. This is a characteristic feature of human cognition and is evident in human learning across a wide range of domains, including problem solving, logical reasoning, natural language, and music. For example, natural language has a complex, hierarchically compositional structure consisting of sentences composed of phrases and words. The meaning of a sentence is determined by its constituent words and the grammatical rules describing the semantic relationships between them. Given a prior knowledge of the vocabulary and grammar, an unlimited number of new sentences can be understood and formulated. It has been hypothesised that compositional learning may be key to flexible cognition and generalisation as it facilitates the acquisition of new concepts from only few examples and the transfer of prior knowledge to novel contexts. For example, after encountering a single example of the mythical creature the "Jabberwocky", a person can integrate the noun into their vocabulary, and be able to meaningfully describe it in completely novel contexts, such as: "the Jabberwocky was playing tennis on the moon".

Although neural networks display powerful predictive capabilities, their generalisation performance does not resemble human-like compositional generalisation. While humans can infer the underlying rules of a task from a small number of examples, neural networks are typically trained on vast data sets and few-shot learning remains a challenge (Ravi and Larochelle, 2016). While human learning is flexible and robust, neural networks trained on one task frequently fail to transfer this knowledge to different but related tasks, and they can be highly sensitive to perturbations in the input data, as Szegedy et al. (2015) demonstrated with the use of adversarial examples. In recent years there has been significant progress in the field of transfer learning, particularly in image recognition and natural language processing, but it remains unclear whether these models are actually learning systematic compositional rules or are just fitting to the statistical patterns in the data (Lake et al., 2016). It has even been argued that neural networks are intrinsically incapable of systematic compositional generalisation (Fodor and Pylyshyn, 1988). This remains an open and active area of research in machine learning, with several approaches being taken to tackle the problem, including specialised architecture (Chang et al., 2019, Márton et al., 2022), data augmentation (Andreas, 2020), meta-learning paradigms (Conklin et al., 2021, and shaping through training (Krueger and Dayan, 2009; Ito et al., 2022; Dekker et al., 2022

Shaping is a teaching method, ubiquitous in human learning, that involves breaking down a complex task into simpler sub tasks, which are learned sequentially in order of increasing complexity. This speeds up the process of learning and makes explicit the compositional structure of the task by directing the learner's efforts towards important features over less relevant details Elman, 1993. For example, a guitar student will first learn to play individual notes and chords, followed by simple chord progressions before eventually learning to play full songs. This training method has been applied to neural networks to support compositional generalisation, and has been shown to improve learning speed, flexibility, sample efficiency and few-shot learning accuracy (Krueger and Dayan, 2009; Ito et al., 2022; Zou et al., 2020).

In this study, we investigated how primitive training facilitates compositional learning in neural networks. Previous work in this area has dominantly focused on the domain of natural language and semantic reasoning (Baroni, 2019). However, we chose to explore the network performance in an arithmetic task as mathematics offers a well defined compositional structure and unambiguous rules, allowing more precise tests for composition. Composition is a broad term that has been used to describe a wide variety of behaviours across modalities. Here we tested two of the five elements of composition mapped out by Hupkes et al. (2020). (1) Systematicity: the ability to understand the rules for combining known parts, and (2) productivity: the limitless capacity to generate new examples from a finite set of components.

We trained recurrent neural networks (RNNs) on a symbolic arithmetic task in which they had to predict the outcome of a sequence of characters that corresponded to a numerical calculation. The RNNs had to infer the numerical values of the symbol characters, the function of the operation characters, and the compositional rules of the sequence to reliably predict the correct output. We compared the zero-shot generalisation performance of networks that had been trained with two different curricula: a Primitive curriculum that included sequences corresponding to the primitive components of the full length sequences, and a control curriculum containing only the full length sequences. We investigated the systemeticity of the networks by testing them on sequences consisting of new combinations of primitives, and we tested their productivity by evaluating them on longer sequences than they had been exposed to in training. We also studied the hidden layer activations to gain an insight into the effect of primitive training on the representation of the sequences.

Understanding how primitive training in neural networks facilitates compositional generalisation may shed light on how shaping aids compositional learning in the brain (Kepple et al., 2022), and generate biologically testable predictions. This could contribute to the development of a computational account of compositional learning and a deeper understanding of this phenomenon. Furthermore, these insights have applications in machine learning as they could provide inspiration for the development of models that are more robust, flexible and multi-modal.

Results

Arithmetic Task

We set out to explore how RNNs learned the compositional rules of arithmetic, and whether primitive training facilitated generalisation to novel problems. We designed a symbolic arithmetic task that involved predicting the outputs of sequences representing numerical calculations. The task sequences had the general structure: $[[+, X]_n =]$. Where $+$ represents the addition operation, X represents a letter symbol from the set $\{A, B, C, D\}$, n represents the number of operation-symbol pairs in the sequence, and the character $=$ occurred at the end of each sequence, acting as a query mark. The operation-symbol pairs were the primitive units of the sequence, and each one corresponded to an integer

value, where the operation signified the sign and the letter signified the magnitude. The RNNs were trained on a set of sequences and evaluated on an unseen test set. In order for the networks to reliably predict the correct output of any novel sequence, they had to infer the numerical value of each symbol, the function of the operations, and the grammar of the sequence. We used this task to study how the RNNs generalised to (1) novel combinations of primitives as a test of systematicity, and (2) longer sequences with multiple addition steps as a test of productivity.

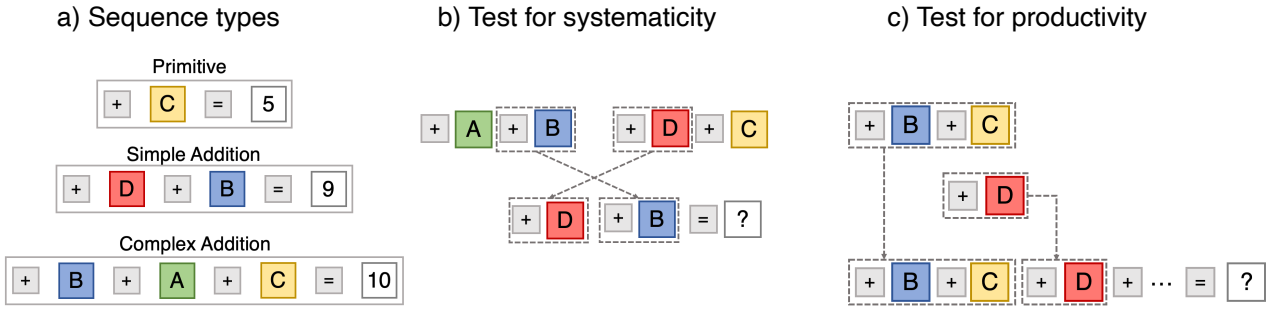

Figure 1: Schematic of the arithmetic task. a) 3 kinds of addition sequences. Primitive sequences consisted of one operation-symbol pair, Simple addition consisted of two, and complex addition sequences were made up of more than 2. Elements of composition b) systematicity, and c) productivity

Training Curricula

In this task, we trained the networks on a number of simple addition sequences, which refers to sums consisting of two primitives added together, Fig 1. In a simple addition sequence, an augend (the first value) is added to an addend (the second). For example, in the sequence $+C + D =$, the augend is $+C$ and the addend is $+D$. We used 4 symbols $\{A, B, C, D\}$, which were ranked such that A always had the smallest value and D the largest. The total simple addition set consisted of 16 unique sequences for all permutations of the 4 augends and 4 addends.

We designed two training curricula: the Primitive training curriculum and the Balanced curriculum as a control. For a given set of simple addition "base sequences", the Primitive curriculum contained an additional set of 4 primitive sequences, while the Balanced curriculum contained 2 simple addition "balancing sequences". We designed the primitive sequences to explicitly map the primitive units of the sequences (the operation-symbol pairs) to their numerical values. These sequences took the form: $+A =$, $+B =$, $+C =$, and $+D =$; these were interleaved with the base sequences. In the Balanced curriculum, two "balancing sequences", e.g. $A + C =$, and $B + D =$, were included instead to balance for the number of occurrences of each primitive per epoch. For example, if the Primitive curriculum contained 1 base sequence and 4 primitives, the matching Balanced curriculum would replace the primitives with two simple addition sequences such that both curricula were matched on the overall frequency of each symbol. The key difference between curricula was that the networks directly learned the values of the symbols in the Primitive curriculum, while these had to be inferred in the Balanced curriculum.

We programmed a solver to ensure that the training sequences provided sufficient information to infer the values of the symbols, assuming an understanding of addition. For any number of base sequences, the Primitive curriculum always contained sufficient information as it explicitly trained the networks on each symbol value. However, the Balanced curriculum required at least two sequences, representing linearly independent equations, in order for the networks to have sufficient information to infer the values of all 4 symbols.

Experiment 1: Effect of Training Set Size on Learning Simple Addition

First, we investigated how increasing the training set size impacts generalisation performance in the Primitive vs Balanced curricula. We compared the performance of 200 networks trained in the Primitives vs. Balanced curriculum as a function of the number of unique base sequences used in training. We employed paired initialisation, where one copy of each initialised network was trained with the Primitives curriculum and the other with the Balanced. The RNNs were trained for 1000 epochs on the training set and then evaluated on the held-out test set (the sequences not present in the training set of either curriculum).

The accuracy of a model's predictions for the test set provided a measure of its one-shot generalisation accuracy to novel combinations. We used two metrics to quantify the prediction accuracy: (1) the root mean squared error (RMSE) and (2) the coefficient of determination (R^2 score). The RMSE quantifies how closely the model predictions match the ground truth; the higher the RMSE, the lower the generalisation performance. The R^2 score describes how well the model captures the variance of the ground truth outputs; the greater the score (i.e. closer to 1), the more accurate the predictions.

As the training set size increased, generalisation performance increased for both the Primitive and the Balanced group. Fig 2 illustrates that the RMSE decreased, and the R squared score increased with increasing numbers of base training sequences. As the networks were trained on a larger fraction of the sequence space, they learned more general solutions that resulted in more accurate predictions for novel sequence combinations.

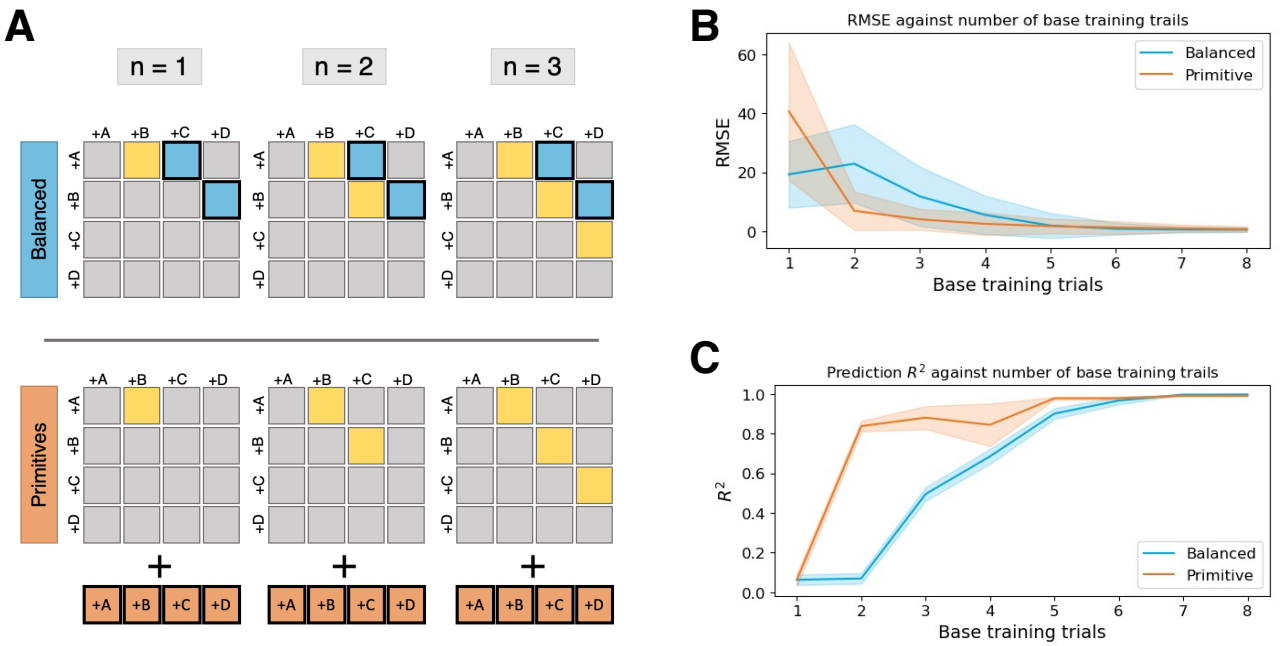

Figure 2: Experiment 1. **A** Schematic of the training sets. Grids represent the 16 unique simple addition sequences with rows as augends and columns as addends. The coloured squares are the sequences used in each training set (yellow: base, blue: balancing, orange: primitive sequences). Balanced (top) and Primitive (bottom) curricula training sets are shown for n=1, n=2, and n=3 base sequences. Group mean **B** root mean square error (RMSE), and **C** R^2 score of Balanced (blue) vs Primitive (orange) RNN predictions as a function of training set size (i.e. number of base sequences). The shaded band shows the standard deviation.

At low numbers of base training sequences, the Primitive group performed significantly better than the Balanced. The Balanced curriculum required 2 or more base sequences to have sufficient information to infer the value of all symbols, therefore, it was expected that the networks would fail to generalise with one base sequence. When trained on between 2 and 5 base sequences, the Primitive group outperformed the Balanced group. The greater sample efficiency of the Primitive group suggests that the primitive training helped the network learn a more general solution when provided with fewer training examples.

The Balanced curriculum trained the networks on a larger fraction of the sequence space than the Primitive group, as an additional 2 simple addition sequences were included in the Balanced training set. This experiment clarifies that the greater accuracy of the Primitive group was due to the benefits of primitive training, rather than any negative impact introduced by the inclusion of balancing sequences.

Experiment 2: Performance with Balanced vs Primitive Two Base Sequence Curricula

We analysed the RNN performance with the 2 base sequence curricula in greater detail to investigate how primitive training aided generalisation for small training set sizes. We chose this training set as the difference in performance between the two curricula was at a maximum. The specific training sequences used in each curriculum are displayed in Fig 3A. In the primitive curriculum, no simple addition sequences containing the symbol C or D were included in training. This provided the opportunity to study whether the networks could incorporate a primitive into a sum indicating it has learned to chain primitives together as a precise test of systematicity.

We trained 800 networks in each curriculum for 1500 epochs, and the 677 of these that converged with a mean squared error (MSE) loss < 1.0 were included in the following analysis. Fig 3B displays all predictions plotted against the ground truth and shows that, as a group, the primitive curriculum predictions were closer to the ground truth and the variance in the aggregated predictions was smaller. On an individual network level, the RMSE was siginificantly lower Fig 3C, and the R^2 scores were significantly higher, Fig 3D, for the Primitives vs Balanced curriculum, confirming that the Primitive group better generalised to the test set than the Balanced.

We calculated the mean prediction accuracy for each individual sequence, where the accuracy score is one if the rounded prediction is equal to the ground truth, and zero if not. This stricter accuracy score indicates which sequences the RNNs could predict with high accuracy. The Balanced RNNs could only reliably predict the sequences where they had been trained on both the augend and the addend. Fig 3E shows the only sequences they accurately predicted were: $+B+B$ and $+A+D$, which were the only sequences composed of addends and augends they had previously been trained on (with $+A+B$ and $+B+D$). This indicates the networks failed to recompose the order of the primitive units and could not generalise knowledge of an addend to the augend position. In contrast to this, the Primitive group networks

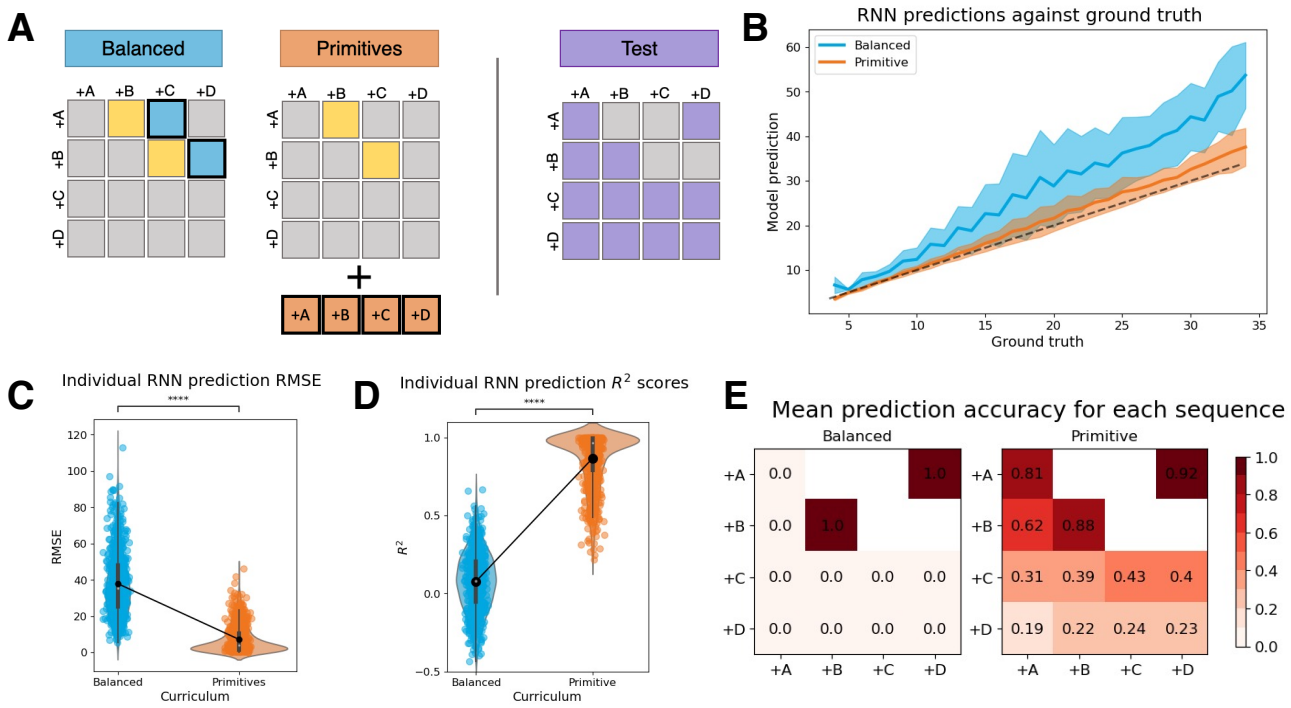

Figure 3: Experiment 2. **A** Schematic of the balanced vs primitive two base sequence curricula. Both curricula contain the same 2 base sequences (yellow), with additional primitive sequences (orange) or balancing sequences (blue). The RNNs were tested with the held-out sequences (purple), which appeared in neither training set. **B** Group mean RNN predictions against the ground truth outputs for the test sequences. The shaded region represents the standard deviation. Individual RNN prediction **C** RMSE and **D** R^2 score for Balanced (blue) vs Primitive (orange) group. **E** Heat maps showing the mean prediction accuracy for each unique sequence of the test set for the Balanced and Primitive groups. The darker the red, the higher the accuracy. The white squares correspond to the training sequences that were not tested on.

were able to predict sequences across the total sequence space, (with variable accuracies indicating heterogeneity in the group), which included sequences with C and D even though they had not been trained on any simple addition sequences with these primitives. For example, 92% of the networks could predict (to the nearest integer) the output of the sequence $+A + D$ even though they had never been trained on +D embedded in a simple addition sequence. This suggests the primitive trained networks displayed systematicity as they learned the systematic rules for combining primitives.

Representation Similarity Analysis

Next, we analysed the networks' hidden layer representations to identify the geometric basis for the differences in performance observed in the previous sections. We conducted representation similarity analysis (RSA) on the hidden layer representations for the full set of 16 unique simple addition sequences. We calculated the pairwise euclidean distances between the hidden layer activations for the 16 sequences to obtain the representation dissimilarity matrix (RDM) for each RNN.

Principle component analysis indicated that over 95% of the variance between the representations was explained by 2 principle components. 2D-MDS (multidimensional scaling) plots of the averaged RDM for each curriculum are shown in Fig 4. The three plots correspond to the sequence time steps 3, 4, and 5, and each point on the plot represents each of the 16 simple addition sequences.

At the third time step of the sequence, the network had received the first 3 characters in the addition sequence, e.g. $+A+$. The points on the MDS plot are arranged in 4 clusters as all sequences in the same augend group are identical at this time step. An augend group refers to the group of sequences sharing an augend e.g.$+A+A$, $+A+B$, $+A+C$, and $+A+D$. At step 4, the networks had received the addend symbol as well e.g $+A + B$, so the points within the augend group were no longer identical and spread out arranged in an order corresponding to the rank of the addends (from the smallest - A (green) to the greatest

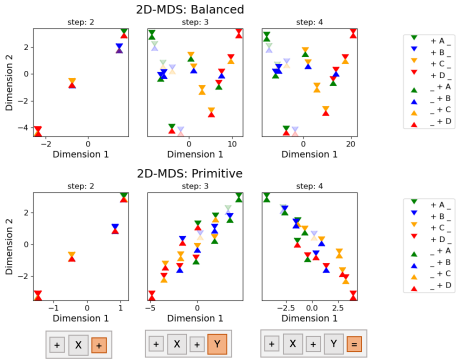

Figure 4: 2D-MDS plots of the RDMs for the 16 simple addition sequences, corresponding to the hidden layer activations at time steps 3, 4, and 5. Semi-transparent points correspond to the training sequences.

D (red)). This spreading out is more extensive in the Primitive curriculum than in the Balanced. Fig 5A shows the MDS plots with construction lines drawn across augend and addend groups. The Primitive representation formed a sheared grid-like arrangement, while the points in the Balanced representation were clustered within the augend groups. At step 5, at the end of the sequence marked by the query mark =, the points appear to be compressed onto the diagonal. This line between the largest and smallest value sequence can be viewed as a 'number line' representing the magnitude of the sequence output. In the balanced curriculum, the rank within each augend cluster is correct with respect to the number line - however, the global order of points across clusters is incorrect. In contrast to this, the order of the points in the Primitive group MDS plot is a closer approximation to the ground truth ranking.

Recent work in machine learning has identified "rich" and "lazy" regimes under which a network can learn two kinds of solutions (Flesch et al., 2021). In the rich regime, where the initialised weights are very small, the networks learn low-dimensional, highly structured solutions that are specific to the task. These tend to be more robust to noise and generalise better to novel examples. In the lazy regime, where initialised weights have a large variance, the networks learn faster but form high dimensional, unstructured representations that are sensitive to noise and generalise poorly. The networks in the previous sections were initialised in the rich regime, as we set out to analyse their structured representations. However, the lazy regime serves as case for comparison.

We initialised a set of 200 networks in the lazy regime, setting the variance of the weights to be 5000 times greater than in the rich regime. We included the 176/200 networks that converged with an MSE loss < 1.0 in the subsequent analysis. As is typical for the lazy regime, the networks overfit to the training set and failed to generalise to the test sequences, performing significantly worse than the rich regime networks. Lazy vs rich Balanced group test prediction R^2: -1.421 ± 0.123 vs 0.078 ± 0.007 (Mann–Whitney U statistic = 10216.0, p value ¡ 0.0001). Lazy vs rich Primitive group test prediction R^2: -2.141 ± 0.361 vs 0.867 ± 0.007 (Mann–Whitney U statistic = 698.0, p value ¡ 0.0001). At time step 4, the lazy initialised networks displayed strong augend clustering, Fig 5A. However, in contrast to the Balanced group, there was not a consistent ranking of points within the augend clusters. Additionally, these representations were higher dimensional, with 3 principle components required to explain 95% of the variance across the mean representations at all time steps, compared to the 2 required in the rich regime.

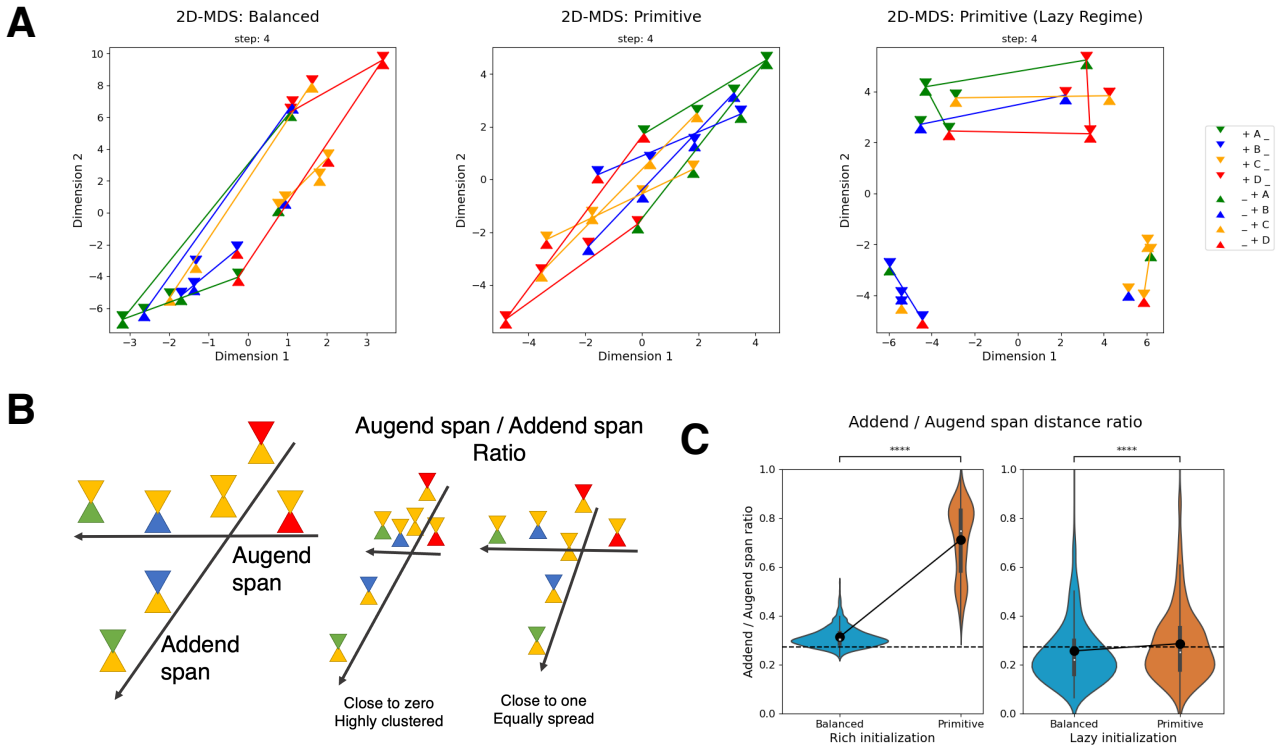

Figure 5: **A** 2D-MDS plots of mean RDM for the Primitive, Balanced and Lazy (Primitive) groups at time step 4 of the sequence. Construction lines are plotted across augend and addend groups. **B** Schematic of the augend/addend span ratio metric. **C** Violin plots of individual RNN augend/addend span ratio for Primitive and Balanced groups in the rich and lazy regimes.

Augend Clustering

In order to quantify the degree of augend clustering, we calculated the ratio between the augend span and the addend span. Fig 5B illustrates that the augend span is the greatest distance between smallest and largest valued points in augend group (i.e. between $+A+X$ and $+D+X$ for an augend X), and the addend span is the corresponding distance between points in an addend group (i.e. between $+X+A$, $+X+D$ for an addend X). A ratio of zero indicates the

complete clustering of an augend group and a ratio of 1 indicates the points are equally spread out across the augend and addend directions. We calculated the addend/augend span ratio for the individual RNNs and compared the values across training curricula. The lazy networks had low addend/augend span ratios, below 0.3, confirming that this measure indicates high clustering. This acts as a standard to compare the networks initialised in the rich regime. Consistent with the MDS plots, we found the Primitive group mean addend/augend span ratio was significantly higher than the Balanced group. This confirms that the hidden representations of the Primitive group were more spread out across the addend and augend directions, while the representations of the Balanced regime were more clustered in augend groups.

Control models

We constructed two control RDMs to characterise how closely the RNNs' hidden representations corresponded to the numerical differences between the sequences. The first control was the ground truth RDM in which the distances between sequences corresponded to the numerical difference between their outputs. We also constructed an augend control RDM, where the distances between sequences corresponded to the numerical differences between the augends of the sequences only. We predicted that this RDM would correspond to a representation that fully clusters the sequences by their augend, and so the output is equal to the augend value. We conducted a multiple linear regression analysis, regressing the two control models against the RDMs of each individual RNN. We calculated the partial coefficient of determination for each control to determine which control best fitted the RDM.

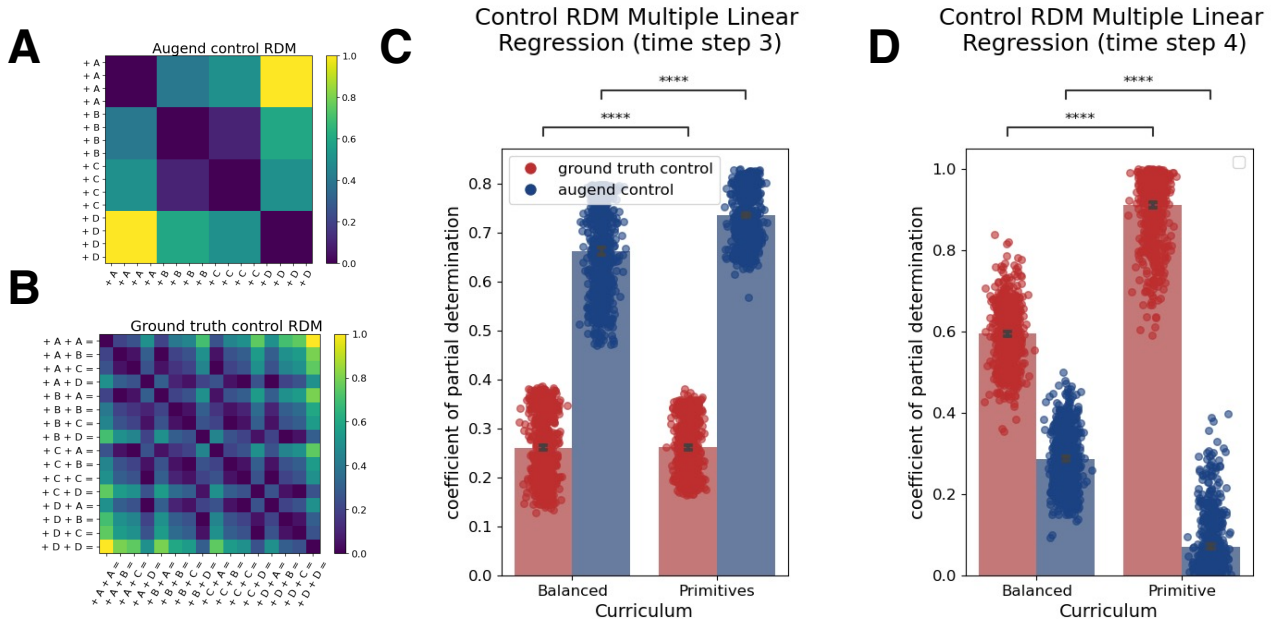

Figure 6: Experiment 2 Control model regression. **A** Augend control and **B** ground truth control RDMs. Partical coefficients of determination for ech control model in the Multiple linear regression with the RNN RDMs at **C** time step 3 and **D** time step 4 in the sequence.

At time step 3, the network had received the first 3 sequence characters (e.g. $+A+$), and so the optimal solution at this stage would be the augend control. Fig 6C shows that for both curricula, the augend control was the best fitting model, and it provided a better fit for for the Primitives curriculum than the Balanced indicating the primitive training helped the networks learn the numerical differences between the symbol values more accurately.

At step 4 the network has seen both the augend and addend, therefore, the optimal RDM would correspond to the ground truth control. Fig 6D shows that for both curricula the ground truth control was the best fitting model. Simple linear regression demonstrated that the ground truth control alone provided a better fit for the Primitive curriculum than the Balanced, indicating the Primitive group represented the numerical differences between sequences more accurately. For the Balanced curriculum, the augend control explained some of the variance in the RDMs. This indicates that the balanced curriculum RNNs are explained by a combination of the augend and ground truth controls, consistent with the augend clustering observed in the MDS plots.

Experiment 3: Generalisation to Longer Sequences

The previous experiments considered how the RNNs could generalise to novel combinations of one step (simple) addition sequences, however, another compositional feature of the sequences was the number of addition steps in the sequence. We next investigated how RNNs trained on simple addition generalised to longer, complex addition sequences. Here, complex addition refers to sequences where more than two symbols are added together e.g. $+A+B+C$ is a two step complex addition sequence, and $+A+B+C+D=$ is a three step complex addition sequence.

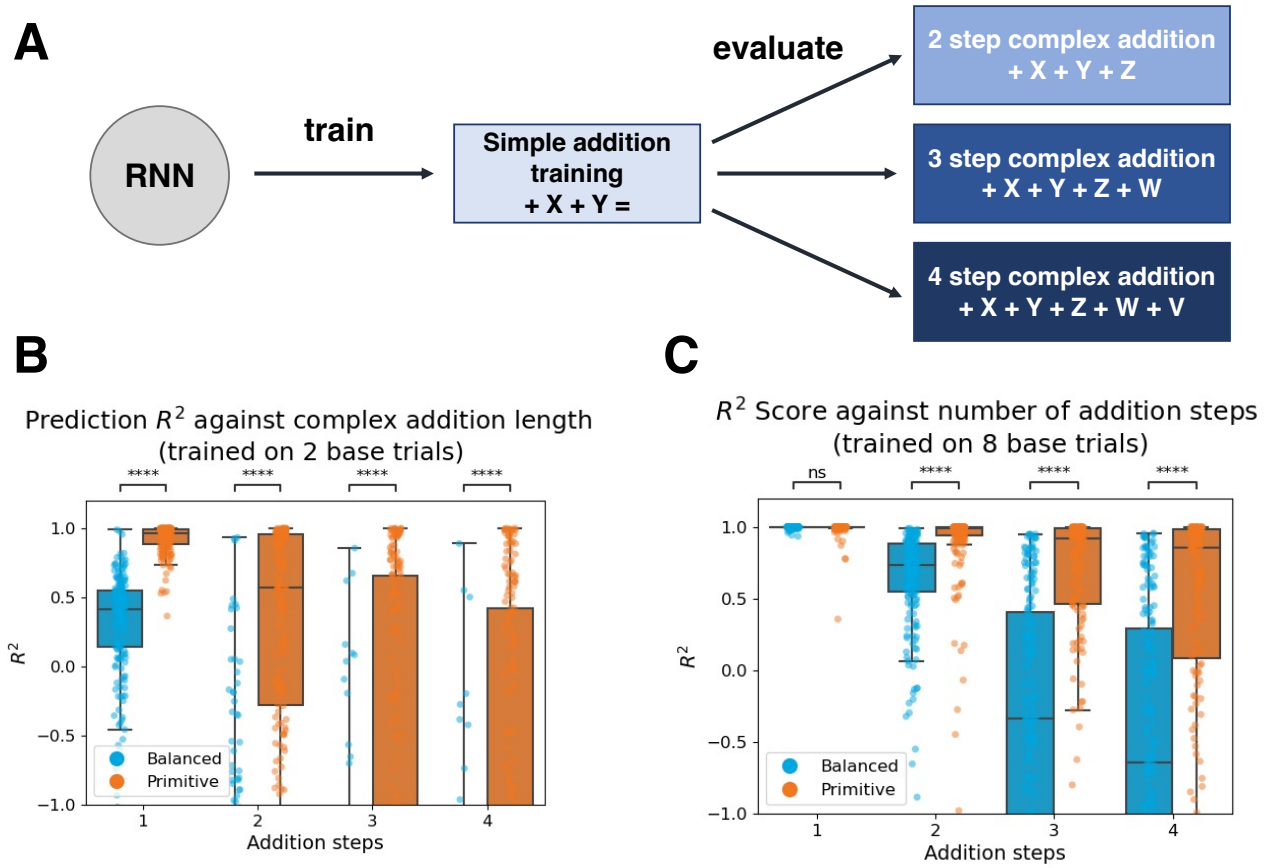

Figure 7: **A** Schematic of experiment 3. RNNs trained on a set of simple addition sequences are tested on 2 step, 3 step, and 4 step complex addition sequences. Boxplots of individual RNN prediction R^2 scores against complex addition test sequence length for RNNs trained on **B** 2 base sequences, and **C** 8 base sequences.

We evaluated the pretrained networks from experiment 1 on 2 step, 3 step and 4 step complex addition sequences, Fig 7A, and calculated the R^2 score for each individual model's predictions. Broadly, The quality of the model predictions decreased as the length of the test sequences increased, but the fall in R^2 scores for the Balanced group was more pronounced than for the Primitive. The networks that had been trained on 2 simple addition base set sequences generalised poorly to the longer complex addition sequences. However, networks that had been trained on larger sets generalised better to the complex addition sequences. Fig 7B-C shows the performance of RNNs trained with 2 base sequences degraded more significantly than than those trained on 8.

For the group of networks trained on 8 base sequences, Fig 7C shows that, while both curricula perform similarly on the simple addition test sequences with near perfect accuracy, the primitive trained RNNs generalise significantly better to longer sequences. For this large training set size, primitive training provided no significant benefit for generalising to novel simple addition sequences. However, this experiment indicates that this training improved productivity. Figs 8C-D depict how the Primitive group's 2 step complex addition sequence representations are qualitatively more structured and less clustered than the Balanced.

We fit 3 control RDMs, Fig 8A, to the 2 step complex addition representations (for the RNNs trained on 2 base sequences). These control RDMs corresponded to the true numerical differences between the sequences at each stage of addition: (1) The "augend control" - differences between the augends, (2) the "simple addition control" - differences between the sum of the first two primitives, and (3) the "ground truth control" - differences between the total output of the sequences (the sum of all 3 primitives). We fit each control to the individual RNNs using simple linear regression to determine how much of the variance was explained by each individual control. We also fit all controls simultaneously with multiple linear regression to determine how much variance was captured by a linear combination of these controls, Fig 8B. In the Primitive curriculum, the best fitting control at each step corresponded to the optimal control. However, the variance of the quality of fit across the group increased with time steps. This suggests the predictions deviated from the optimum as the sequence progressed. In the Balanced curriculum, the augend control was the best fitting model across all time steps indicating the output of the sequence was strongly dependent on the augend value. Additionally, for the Primitive curriculum, the total variance explained by the multiple regression model was close to 1, indicating most of the variance could be explained by a combination of the control models. In contrast to this, the Balanced group had a significantly higher proportion of unexplained variance.

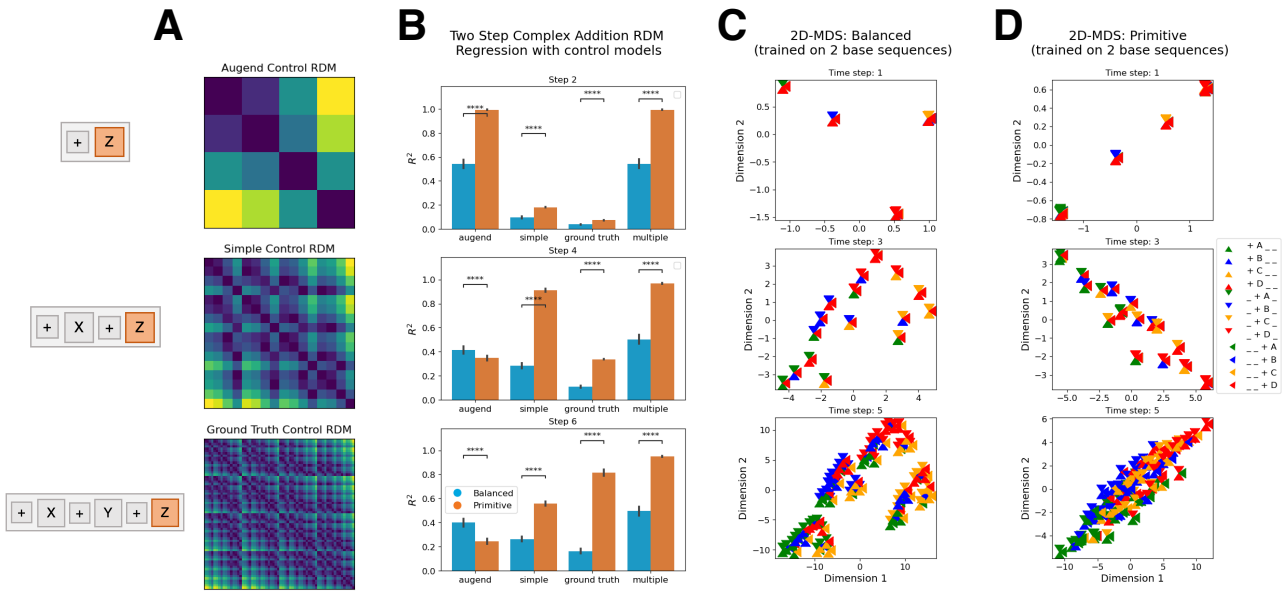

Figure 8: 2 step complex addition RSA. **A** Heatmaps representing the augend, simple and ground truth RDMs. **B** Coefficient of determination of the three controls and the multiple linear regression model for the RNN representations at time step 2, 4, and 6 of the sequence. 2D-MDS plots of the 2 step complex addition sequences at time step 2, 4, and 6of the sequence for the **C** Blanced and **D** Primitive groups.

Discussion

In this study we explored how primitive training facilitated compositional learning. We set out to disambiguate systematic compositional learning from pattern recognition by using an arithmetic task that could clearly disentangle these behaviours.

Systematicity

We first investigated the systematic generalisation of the RNNs by evaluating them on simple addition sequences comprised of novel combinations of known primitives. We found that RNNs both with and without primitive training were able to generalise to the novel combinations when trained on a large enough fraction of the unique sequence set. This is consistent with prior work, which has demonstrated that neural networks can learn to solve arithmetic problems when trained on large data sets, such as the Mathematics Dataset (Saxton et al., 2019; Russin et al., 2021). However, analysing the error patterns of the networks when trained on a small number of examples revealed that only the primitive trained networks displayed systematicity, correctly predicting sequences with new orders of trained primitives. This demonstrated that primitive training supports generalisation with fewer training examples, consistent with findings by Ito et al. (2022) that primitive training increases sample efficiency.

Productivity

We also tested the productivity of the networks by evaluating how they generalised to longer sequences and found that the accuracy of the networks degraded with increasing sequence length. Prior work has also demonstrated that network performance decreases when extrapolating to longer sequences, including for both language tasks (Lake, 2019) and mathematical tasks (Schlag et al., 2020). This exposes that the networks have not perfectly learned the systematic rules of the task, as this would allow the networks to predict arbitrarily long sequences. However, we found that, with primitive training, the performance degraded to a lesser extent. One reason for this could be that the particular structure of the primitive trained representations was more robust to extrapolation. Additionally, the more accurate numerical representations may have resulted in a smaller accumulated error for the longer sequences.

Representational Geometry

Studying the hidden layer representation geometry revealed distinct differences in the sequence representations between training curricula. Without primitive training, the sequence representations were more strongly clustered by their augend value, suggesting that these networks emphasised the first value in the addition sequence. In contrast to this, the representations of the primitive trained networks were more spread out across the augend and addend directions, mapping out the sequence space in a way that more accurately corresponded to the numerical differences, as confirmed formally with the ground truth control regression analysis. Krueger & Dayan (2009) demonstrated that shaping procedures can influence which elements of a sequence are most pronounced in the neural network representation. This is consistent

with our findings that primitive training reduced the emphasis on the first primitive in the sequence such that the representation was more symmetrical with respect to the augend and addend.

Limitations and Future Work

In this study, we only considered the limited scope of addition problems. The RNN architecture we used meant that the networks could not form distinct abstract representations for the magnitude and the operation. Therefore, the RNNs would not be able to systematically generalise across multiple operations. A more complex architecture would be required to represent magnitude and operations as distinct components in order to investigate compositional learning of complex arithmetic with multiple operations.

Conclusion

In conclusion, we have shown that primitive training facilitated compositional generalisation in neural networks in an arithmetic task. Specifically, we identified an improvement in sample efficiency, systematicity, and productivity. Additionally, we found that Primitive trained networks had a distinct representational geometry that corresponded to the ground truth numerical differences more closely.

Methods

Arithmetic Sequences

In the arithmetic sequences, the values of the letter symbols were drawn with a uniform probability from a range of integers from 2 to 17, without replacement. These numbers were sorted and assigned to the letters such that the numerical rank reflected the alphabetical rank, i.e. "A" was assigned the smallest integer, "B" the second smallest and so on. For each RNN, the integer values for the symbols were redrawn.

In this experiment we used three categories of sequences: primitive, simple, and complex. Primitive sequences had one operation-letter pair (n=1), simple sequences had 2 (n=2), and complex sequences had 3 or more (n ¿= 3).

Experiment 1

This experiment compared RNN performance as the number of base trials increased from 1 to 8. The base trials were chosen so that two different symbols were present in each simple addition sequence. The base trials were selected in order from the range: $+A+B$, $+B+C$, $+C+D$, $+D+A$, $+A+C$, $+B+D$, $+C+A$, $+D+B$. In the Primitive curriculum, the primitive seqeuences: $+A$, $+B$, $+C$, and $+D$ were interleaved with the base trials, and in the balanced curriculum, 2 balancing sequences, not present in the base set, were randomly selected as balancing sequences.

Experiment 2

The two base trials +A+B and +B+C were present in both curricula. The same primitive sequences were used as in experiment 2. The simple addition sequences $+A+C$ and $+B+D$ were used as balancing sequences.

Experiment 3

We compared the one-shot generalisation to complex addition performance of 2 sets of networks, which had been trained in experiment 1: those trained on 2 and those trained on 8 base trials. We evaluated the networks on 2 step, 3 step, and 4 step complex addition trials. We randomly selected a set of 64 sequences from each complex addition permutation set to test the networks on.

Neural Networks

RNN Architecture

The recurrent neural networks (RNNs) had 1 hidden layer of size 20 hidden units, with a ReLu activation function. For all RNN experiments the learning rate was set to 0.005 and a batchsize of 1 was used. The initial weights were drawn from a Xavier normal distribution with mean of 0 and standard deviation = $gain \times \sqrt{2/(no.inputs + no.outputs)}$, where $gain$ is a scaling constant. The $gain$ value was set to 0.0001 in the rich regime, and 5 in the lazy regime.

RNN inputs

The input sequences were converted to one hot encoded vectors, where each vector represented a single character. The RNNs received an input vector of 22 units corresponding to the one-hot encoded character at each time step, and it produced a single scaler output value. The number of time steps per sequence depended on the number of characters. For example, primitive sequences, e.g. " + A = " had 3 time steps, and simple addition, e.g. " + A + B = ", had 5 time steps.

Analysis

Statistical Tests

On all figures, we used the non-parametric, paired-difference, statistical hypothesis test: the Wilcoxon signed rank test for all group difference significance tests.

Control model regression

We included elements of the RDMs corresponding to the sequences of the test set only in the regression analyses. In the multiple regression analyses, the coefficient of partial determination was calculated by dividing the variance explained by the individual feature by the variance explained by the total model. This value was used as a metric of feature importance.

References

Andreas, J. (2020). Good-enough compositional data augmentation.

Baroni, M. (2019). Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *375*(1791), 20190307.

Chang, M. B., Gupta, A., Levine, S., & Griffiths, T. L. (2019). Automatically composing representation transformations as a means for generalization.

Conklin, H., Wang, B., Smith, K., & Titov, I. (2021). Meta-learning to compositionally generalize.

Dekker, R. B., Otto, F., & Summerfield, C. (2022). Curriculum learning for human compositional generalization. *Proceedings of the National Academy of Sciences*, *119*(41), e2205582119.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, *48*, 71–99.

Flesch, T., Juechems, K., Dumbalska, T., Saxe, A., & Summerfield, C. (2021). Rich and lazy learning of task representations in brains and neural networks. *bioRxiv*. https://doi.org/10.1101/2021.04.23.441128

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*(1-2), 3–71. https://doi.org/10.1016/0010-0277(88)90031-5

Ito, T., Klinger, T., Schultz, D. H., Murray, J. D., Cole, M. W., & Rigotti, M. (2022). Compositional generalization through abstract representations in human and artificial neural networks.

Kepple, D. R., Engelken, R., & Rajan, K. (2022). Curriculum learning as a tool to uncover learning principles in the brain. *International Conference on Learning Representations*. https://openreview.net/forum?id=TpJMvo0_pu-

Krueger, K. A., & Dayan, P. (2009). Flexible shaping: How learning in small steps helps. *Cognition*, *110*(3), 380–394.

Lake, B. M. (2019). Compositional generalization through meta sequence-to-sequence learning.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2016). Building machines that learn and think like people. *CoRR*, *abs/1604.00289*. http://arxiv.org/abs/1604.00289

Márton, C. D., Gagnon, L., Lajoie, G., & Rajan, K. (2022). Efficient and robust multi-task learning in the brain with modular latent primitives.

Ravi, S., & Larochelle, H. (2016). Optimization as a model for few-shot learning. *International Conference on Learning Representations*.

Russin, J., Fernandez, R., Palangi, H., Rosen, E., Jojic, N., Smolensky, P., & Gao, J. (2021). Compositional processing emerges in neural networks solving math problems.

Saxton, D., Grefenstette, E., Hill, F., & Kohli, P. (2019). Analysing mathematical reasoning abilities of neural models.

Schlag, I., Smolensky, P., Fernandez, R., Jojic, N., Schmidhuber, J., & Gao, J. (2020). Enhancing the transformer with explicit relational encoding for math problem solving.

Zou, Y., Zhang, S., Chen, K., Tian, Y., Wang, Y., & Moura, J. M. F. (2020). Compositional few-shot recognition with primitive discovery and enhancing.