COGNITIVE SCIENCE
A Multidisciplinary Journal

Cognitive Science 48 (2024) e13492

ISSN: 1551-6709 online
DOI: 10.1111/cogs.13492

Exploring Early Number Abilities With Multimodal Transformers

Alice Hein, Klaus Diepold

Chair of Data Processing, TUM School of Computation, Information and Technology, Technical University of Munich

Received 5 September 2023; received in revised form 17 July 2024; accepted 7 August 2024

Abstract

Early number skills represent critical milestones in children's cognitive development and are shaped over years of interacting with quantities and numerals in various contexts. Several connectionist computational models have attempted to emulate how certain number concepts may be learned, represented, and processed in the brain. However, these models mainly used highly simplified inputs and focused on limited tasks. We expand on previous work in two directions: First, we train a model end-to-end on video demonstrations in a synthetic environment with multimodal visual and language inputs. Second, we use a more holistic dataset of 35 tasks, covering enumeration, set comparisons, symbolic digits, and seriation. The order in which the model acquires tasks reflects input length and variability, and the resulting trajectories mostly fit with findings from educational psychology. The trained model also displays symbolic and non-symbolic size and distance effects. Using techniques from interpretability research, we investigate how our attention-based model integrates cross-modal representations and binds them into context-specific associative networks to solve different tasks. We compare models trained with and without symbolic inputs and find that the purely non-symbolic model employs more processing-intensive strategies to determine set size.

Keywords: Deep learning; Numerical cognition; Computational modeling; Number processing

Correspondence should be sent to Alice Hein, Chair of Data Processing, TUM School of Computation, Information and Technology, Technical University of Munich, 80333, Munich, Germany. E-mail: alice.hein@tum.de

1. Introduction

For many adults, tasks such as counting objects or sorting a set of digits appear simple. For children, however, early number abilities take several years to learn. Mastering these skills involves developing a network of concepts that encompasses language, visuospatial abilities, and executive functions (Zhang, 2016). This knowledge later forms the basis for more complex capabilities, for example, arithmetic. Given the integral role of numbers in our daily lives, questions surrounding the way we learn, represent, and process them have occupied cognitive scientists, neuroscientists, and psychologists for decades, forming the multidisciplinary field of numerical cognition.

In this field, connectionist computational models have long played an important part. Often referred to as artificial neural networks, they take inspiration from the way information is stored and processed in the brain via neurons and synapses. As such, they represent concrete implementations of ideas on how at least small subsystems in the brain acquire and process concepts, which can be evaluated against behavioral and neural data. Connectionist models are thus invaluable tools in elucidating critical aspects of learning processes. Their outputs and behavior are inherently shaped by their architecture, training algorithms, and hyperparameters. Additionally, and perhaps more insidiously, these characteristics are influenced by the designers' choice of input modalities as well as the complexity and variety of tasks addressed.

As we show in a brief literature review in Section 2.1, numerical cognition researchers have been able to reproduce observations from human experimental data using a wide range of approaches. However, many previous computational models operate only on binary images or vectors. When multiple modalities are involved, these are usually processed via specialized modules that are sometimes trained separately. Furthermore, computational modeling studies have mainly focused on a single task type, such as comparing quantities or counting. This setup contrasts with the way humans acquire number knowledge. Children learn through interaction with complex multimodal environments where they encounter number and magnitude concepts in various contexts (Fuson, Richards, & Briars, 1982).

Faithfully reconstructing a child's brain and experiences is, of course, outside our current abilities. Still, using overly abstracted inputs may artificially impose a stricter separation of input pre-processing and task solving than would naturally occur. Furthermore, considering only isolated skills neglects the interactions between concepts that characterize natural learning and information processing. The main purposes of this work are to introduce a greater but still controlled realism into the modeling of early number abilities and to analyze the points of similarity and difference with empirical research and other models in the literature. Our approach entails training a model on 35 tasks related to enumeration, set relations, symbolic digits, and seriation. Our goal is not to optimize model accuracy or training times on these tasks—in fact, we are precisely interested in where the model struggles or learns more slowly. The tasks draw inspiration from a suite of tests designed to assess young children's early number abilities, which includes hypothesized and empirically validated learning trajectories to serve as comparisons. The model learns end-to-end from video demonstrations in a synthetic environment with visual and language inputs.

Specifically, we examine the following questions: (1) In which order does the model acquire tasks, and how does this compare with findings from educational psychology? (2) On a behavioral level, how do the model's outputs and error patterns compare with human data and previous computational modeling studies? (3) On a mechanistic level, to what extent does input modality or task specialization emerge in the model? (4) On a behavioral and mechanistic level, what is the effect of removing tasks involving symbolic numbers from the model's training data? We begin with an overview of previous connectionist models in numerical cognition.

2. Background and related work

2.1. Computational models in numerical cognition

Most early connectionist models in numerical cognition focused on numerosity detection and comparison. One of the first such studies was that of Dehaene and Changeux (1993). Their modular architecture processed simple non-verbal visual and auditory inputs using hand-crafted connections and accounted for several psychophysical effects observed in humans. Peterson and Simon (2000) conducted a computational study on enumeration and proposed two models, one based on the Adaptive Character of Thought-Rational (ACT-R) theory (Anderson, 1983) and one a feedforward architecture, which provided good qualitative fits to results obtained in empirical studies. Ahmad, Casey, and Bale (2002) introduced a multi-network modular system and also focused on determining input numerosity. The architecture used various independently trained neural network types, including recurrent connections and self-organizing maps, and showed some adherence with experimental data from children. Verguts and Fias (2004) and Verguts, Fias, and Stevens (2005) studied the mental representation of numbers using connectionist models inspired by neuroscientific findings. They proposed a number representation system using place coding, linear scaling, and constant variability on the mental number line, reproducing error patterns similar to humans on number comparison tasks.

Several computational studies have also focused on spatial aspects of numerical cognition. Mareschal and Shultz (1999) designed a modular cascade-correlation generative network for sorting arrays of numbers. Similar to children, the model showed soft stage transitions and variation in performance within stages. Gevers, Verguts, Reynvoet, Caessens, and Fias (2006) extended the work of Verguts et al. to study the interaction between number and space representations in parity judgment and number comparison tasks. Their model exhibited the Spatial-Numerical Association of Response Codes (SNARC) effect (Dehaene, Bossini, & Giraux, 1993), a phenomenon where people tend to respond faster to small numbers located to their left and large numbers located to their right. Chen and Verguts (2010) further expanded the model, adding hand-crafted biologically inspired layers to explicitly represent space and associate numbers with it. The resulting model simulated various experimental data and effects related to spatial attention and dysfunction. Finally, McGonigle-Chalmers and Kusel (2019) proposed a set of models that combined aspects of Bayesian, dynamical systems, and cognitive architectural approaches to model the shift in children's size sequencing and ordinal search competencies.

Many initial computational models had relatively few parameters and sometimes involved hand-crafted connections. Recently, researchers have increasingly embraced the paradigm of "deep learning," inspired by the complex, layered organization and functioning of the human cerebral cortex. Stoianov and Zorzi (2012) investigated the emergence of visual number sense using a deep neural network (DNN) trained on binary images. They observed that some neural units acted as "emergent numerosity detectors," resembling the response profiles of monkey parietal neurons. Since then, several studies have found number-selective neurons even in randomly initialized, entirely untrained DNNs (Kim, Jang, Baek, Song, & Paik, 2021; Nasr & Nieder, 2021), suggesting that signals that covary with numerosity can emerge spontaneously from the statistical properties of bottom-up projections in multilayered architectures. When explicitly trained on number tasks, a range of DNN models have been shown to estimate numerosity at a level comparable to humans. Architectures proposed so far include deep feedforward networks and differentiable recurrent attention models (Chen, Zhou, Fang, & McClelland, 2018), stacked autoencoders (Testolin, Zou, & McClelland, 2020), recurrent neural networks (RNNs) (Sheahan, Luyckx, Nelli, Teupe, & Summerfield, 2021), deep belief networks (DBNs), and hierarchical convolutional neural networks (CNNs) (Creatore, Sabathiel, & Solstad, 2021).

The computational models discussed so far have been systems trained to classify or reconstruct static inputs usually limited to one modality, such as vision. Several studies have taken a more embodied approach to number learning, exploring the implications of training agents that carry out actions in an environment. Most of these investigations have been in the area of developmental cognitive robotics, where the main focus has been on the benefits of gestures, such as pointing or finger counting, for learning number representations faster, more accurately, and more in line with psychological phenomena observed in humans (Di Nuovo, De La Cruz, Cangelosi, & Di Nuovo, 2014; De La Cruz, Di Nuovo, Di Nuovo, & Cangelosi, 2014; Di Nuovo, Vivian, & Cangelosi, 2015; Di Nuovo, 2017, 2018; Di Nuovo & McClelland, 2019; Rucinski, Cangelosi, & Belpaeme, 2011, 2012). Furthermore, Dulberg, Webb, and Cohen (2021) trained an emergent symbol binding network on a subset counting task using a two-step training curriculum. Although the model was not physically embodied, it was trained by interacting with an environment via reinforcement learning.

Most closely related to our work is that of Sabathiel, McClelland, and Solstad (2020a). Their model consisted of a long short-term memory (LSTM) and a convolutional LSTM module and was trained on four tasks: counting objects, counting events, reciting numbers, and counting out a subset. The model learned these tasks in a supervised manner in an environment consisting of a 4×4 grid with two binary features at each location, denoting the presence of an object and the agent's hand, respectively. The network developed a strategy of "mentally tagging" objects during counting (Sabathiel et al., 2020a) and abstract number representations employed across tasks (Sabathiel, McClelland, & Solstad, 2020b). We follow a similar approach in that we train a DNN on multiple number-related tasks from demonstrations and investigate the model's learned representations. However, we significantly expand the number of tasks and use more complex visual inputs. Motivated by the recent successes of attention-based models in processing sequential, multimodal data, we also use a different architecture, namely, a transformer. We provide some background on transformers in the following section.

2.2. Transformers

Transformers are a family of deep learning architectures first proposed by Vaswani et al. (2017). While they originated in natural language processing (NLP), transformers have since spread to other domains; they are now applied to many non-textual forms of data, including images, videos, audio signals, and protein structures (Paaß & Giesselbach, 2023). They also form the backbone of the now ubiquitous large language models (LLMs). The main change transformers introduced to the field was a shift from sequential to parallel processing of time series data. Before transformers, most NLP models used RNNs. In an RNN, inputs, such as tokenized words or characters, are added one after the other. The model then learns which inputs and intermediate computation results to retain for how long in order to succeed on a given task. To do this, it must update its hidden states after each time step, as they form the inputs for any following calculations.

In contrast to RNNs, transformers receive an entire context window, such as a sentence or paragraph, at a time. They maintain access to all the information in this window without having to learn to "remember" it. Because a transformer essentially treats all time steps independently, it can process them in parallel, leading to considerably faster computation than the recurrent approach. The main units that carry out the input processing are a transformer's attention heads. As the following sections presuppose an understanding of the attention mechanism, we seek to provide some intuition on the topic in Box 1.

BOX 1 The attention mechanism

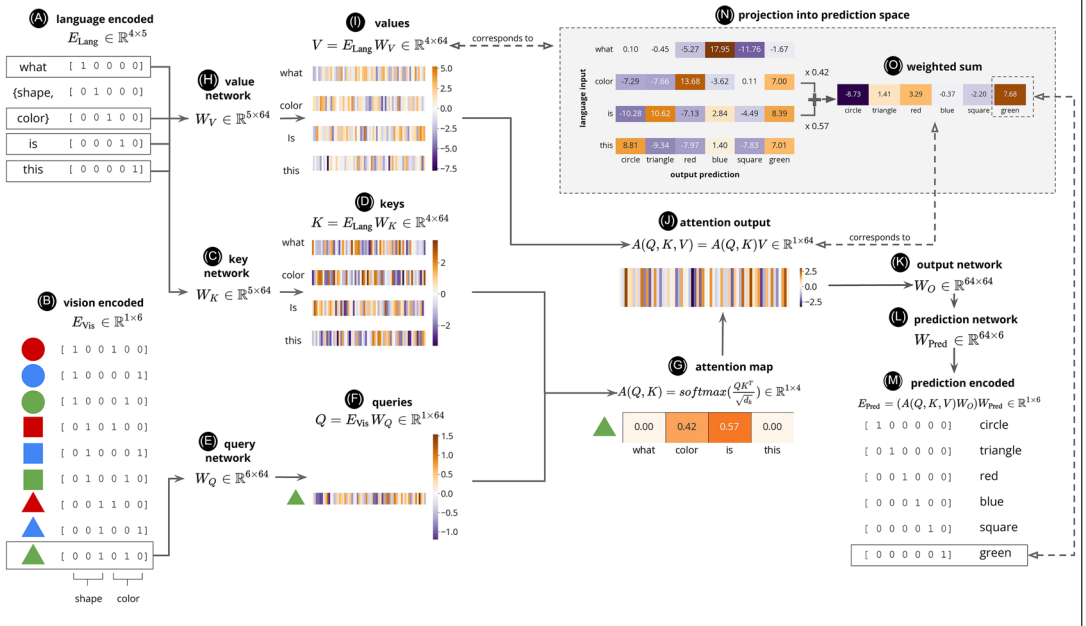


Fig. 1. Toy example illustrating the workings of a single attention head.

In Fig. 1, we illustrate the workings of a single attention head with a toy task: The model receives a visual input consisting of a circle, rectangle, or triangle, which may be red, blue, or green. It is asked about this input's shape or color. Let us assume that our inputs are a green triangle and the question "What color is this?" We first translate language and vision inputs into binary vectors E_{Lang} Ⓐ and E_{Vis} Ⓑ. Note that this is a simplification for illustrative purposes, not how we encode visual inputs in our actual model (see Section 3.2). The binary vectors serve as inputs to the attention head. One attention head consists of five single-layer neural networks: W_V, W_K, W_Q, W_O, and W_{Pred}. The networks' weights are initially random and learned through training on question–answer pairs via backpropagation. In our illustration, model weights have already been optimized. Each network serves a different function. W_K Ⓒ receives language input and produces activation vectors K Ⓓ. W_Q Ⓔ receives visual input and produces an activation vector Q Ⓕ. Inspired by information retrieval terminology, K and Q are referred to as "keys" and "queries" (Vaswani et al., 2017). Query vectors represent what the model is looking for, whereas keys act as signals to match against the queries. Because W_K and W_Q have the same number of output neurons $d = 64$, K and Q have the same dimensionality and can be combined via their inner product. This combination allows the model to relate information from both modalities. We divide the key–query product by a scaling factor $\sqrt{d_k}$ and apply a softmax function to keep values between 0 and 1. The result, $A(Q, K)$ Ⓖ, is often referred to as an "attention heatmap" (Rush, Chopra, & Weston, 2015). It shows the strength of the match between the query and each key. $A(Q, K)$ is specific to the context, that is, combining the same question with another visual input would result in a different heatmap. $A(Q, K)$ is combined with the output V Ⓘ (values) of the value network W_V Ⓗ. Analogous to how values in databases are the actual data associated with a key, a "value" in the attention mechanism is a transformed representation of the input (in this case E_{Lang}) that contains the actual content to be focused on. Multiplying $A(Q, K)$ with V yields the attention output $A(Q, K, V)$ Ⓙ, which represents the weighted sum of values, where the weights are determined by the attention heatmap. We pass $A(Q, K, V)$ through the output network W_O Ⓚ and feed the result to the prediction network W_{Pred} Ⓛ. This gives us the correct answer to the question: "green" Ⓜ. While the activation vectors in Fig. 1 are not human-interpretable, we can translate them into intermediate predictions by directly inputting them to W_O and W_{Pred}. Doing this for V shows that each word in the question triggers different answers Ⓝ. For example, the "color" vector activates the output "red." This pairing is arbitrary—with a different random weight initialization, "red" might, for example, be maximally activated by "this." If we linearly combine the activations according to our attention heatmap [0.0 0.42 0.57 0.0], we obtain a vector Ⓞ that translates to the correct output "green." $A(Q, K)$ can thus be seen as a "selector" of the most likely answer among the options encoded in V, based on the linguistic and visual context.

3. Methods

3.1. Data

3.1.1. Tasks

Our dataset is based on a curriculum proposed by Resnick, Wang, and Kaplan (1973). Inspired by Gagné's framework of "learning hierarchies" (Gagne, 1968), the authors operationalized early number concepts as a suite of tasks, ordered by what they hypothesized to be an optimal match for children's natural sequence of acquisition. In two empirical studies, they turned many of these tasks into diagnostic tests, which they administered to 80–150 pre-kindergartners, kindergartners, and students in their second week of elementary school (Wang, Resnick, & Boozer, 1971; Wang, 1973). The authors applied multiple scalogram analysis (Lingoes, 1963) to the test scores to identify dependencies in the relationships among children's abilities. They then compared the empirical patterns of acquisition they found against their hypothesized learning hierarchies. Resnick, Wang, and Kaplan's task suite constitutes an excellent basis for our dataset, as it encompasses a wide range of skills related to the concept of numbers, including hypothesized and, in part, psychometrically validated results from human studies. The suite covers enumeration, set comparison, symbolic numerals, and sorting.

Table 1 gives an overview of the tasks we used, ordered by difficulty as hypothesized by Resnick et al. (1973). Table 2 shows the developmental trajectories in children's learning found by Wang et al. (1971) and Wang (1973) for those tasks that were psychometrically validated. The grouping into task families in Table 1 is not a perfect partition, and some tasks may integrate skills from other task types. We distinguish between three numerical concepts that may be involved in a task: quantity, rank, or label (Nieder, 2005). Quantity refers to cardinality, that is, the number of elements in a set. Rank refers to the serial order of an element. In label tasks, numbers are used categorically to identify an object. As can be seen, the dataset encompasses all three usages of the number and some tasks that do not explicitly involve numbers but are believed to support the acquisition of number concepts. We go through the tasks in more detail in the following, starting with those related to enumeration.

The first three tasks introduce two important counting principles. A1 asks the agent to recite the count list, starting and stopping at a specified number. Knowing the number sequence, the so-called stable order principle (Gelman & Gallistel, 1986), is a crucial numerical concept and arguably the first mathematical skill a child acquires (Sabathiel et al., 2020a). Children typically learn this principle over several years, starting around age 2 and going up to age 6 (Mussolin, Nys, Content, & Leybaert, 2014). In A2, the agent must point at each object in a set exactly once. A3 combines A1 and A2. It requires the agent to say the correct count word as it touches each object—the so-called one-to-one principle (Gelman & Gallistel, 1986). In the original curriculum, the child can remove objects as it counts them to decrease the strain on working memory. In our computational implementation, objects disappear after being grabbed and released.

The last four tasks involve the enumeration of fixed sets. A4 and A5 are analogous to A3, except objects do not disappear after being tagged. The set is linearly arranged in A4, reducing

Table 1
Overview of the task types in our dataset.

		Visual Input		Representation of Relevant Number/Magnitude					Numerical Concept		
ID	Task Description	Number of Sets	Number of Digits	Word (Symbolic)	Digit (Symbolic)	Set Size (Analog)	Object Size (Analog)	Object Color (Analog)	Quantity (Cardinal)	Rank (Ordinal)	Label (Nominal)
A1	recite count list	0	0	x						x	
A2	touch all fixed, unordered Os in turn	1	0								
A3	enumerate movable Os	1	0	x						x	
A4	enumerate fixed, ordered Os	1	0	x						x	
A5	enumerate fixed, unordered Os	1	0	x						x	
A6	touch exactly {number word} Os	1	0	x					x		
A7	select set of size {number word}	2-5	0	x		x			x		
B1	select larger of two sets	2	0			x			x		
B2	select smaller of two sets	2	0			x			x		
B3	select larger of a digit and a set	1	1		x	x			x		
B4	select smaller of a digit and a set	1	1		x	x			x		
B5	select all digits larger than size {set}	1	5		x	x			x		
B6	select all digits smaller than size {set}	1	5		x	x			x		
B7	select larger of two rows	1	0			x			x		
B8	select smaller of two rows	1	0			x			x		
B9	select largest of three sets	3	0			x			x		
B10	select smallest of three sets	3	0			x			x		
C1	match digits	0	2-10		x						x
C2	select stated digit	0	5	x	x						x
C3	name given digit	0	1	x	x						x
C4	select set of size {digit}	2-5	1		x	x			x		
C5	select larger of two digits	0	2		x					x	
C6	select smaller of two digits	0	2		x					x	
C7	sort digits	0	2-4		x					x	
C8	touch exactly {digit} Os	1	1		x				x		
D1	select largest O in set	1	0				x				
D2	select smallest O in set	1	0				x				
D3	sort Os by size (Os differ in 1 attribute)	1	0				x	x			
D4	sort Os by size (Os differ in > 1 attributes)	1	0				x	x			
D5	select darkest O in set	1	0					x			
D6	select lightest O in set	1	0					x			
D7	sort Os by color (Os differ in 1 attribute)	1	0				x	x			
D8	sort Os by color (Os differ in > 1 attributes)	1	0				x	x			
D9	sort sets by size	2-4	0			x			x	x	
D10	name position of an O in ordered set	1	0	x						x	

Table 2
Comparison of hypothesized and observed developmental trajectories in children's learning, based on Wang et al. (1971) and Wang (1973). To be read from left to right. Only psychometrically validated tasks with direct counterparts in the current study are shown

Hypothesized Trajectory	A1	A3	A4	A5	A6	A7	B1	B2	B9	B10	C1	C2	C3	C4	C5	C6	C7	D9
Empirical trajectory for numbers zero to five	C1	A1	A3	A5	A4	B2	B1	B9*	B10*	C2	A7	A6	C3	C4	C5/C6†		C7	D9
Empirical trajectory for numbers six to ten	C1	A1	B1	B2	B9*	B10*	A4	A3	A5	A6	C2	C3	A7	C4	C7	C5/C6†		D9

Note. ∗ Excluded because too few subjects mastered the task † Not distinguished in psychometric analysis

the difficulty of tracking which objects have been counted (Potter & Levy, 1968; Schaeffer, Eggleston, & Scott, 1974). In A6, the agent must touch a stated number of objects without uttering any number words, then stop. A6 is a version of the give-N task, which has been used in previous studies of children (Sarnecka & Carey, 2008; Wynn, 1992) and neural networks (Dulberg et al., 2021; Sabathiel et al., 2020a). In A7, the agent must point at a set of a given size, selecting from two to five options. Unlike the other tasks in this unit, which are primarily concerned with the rank of an element in the count sequence, A6 and A7 require determining the cardinality of a set without counting aloud.

The second unit involves comparing quantities. In B1 and B2, the agent must point at one of two sets containing more or fewer objects, respectively. Resnick et al. (1973) considered B2 more challenging than B1, arguing that B2 requires finding a set with extra objects, then choosing its counterpart. It thus involves negative information, which can be difficult for young children. In B3 and B4, the agent receives a digit and an object set and must point at whichever represents the higher (B3) or lower (B4) number. In B5 and B6, inputs consist of five digits and one set. The agent must point at all digits denoting numbers larger (B5) or smaller (B6) than the set. In B7 and B8, the agent must decide which of two rows of objects contains more (B7) or fewer (B8) objects. This task is reminiscent of the Piagetian number conservation test, where two sets are linearly arranged such that equivalence is easy to determine via a 1-to-1 comparison. The arrays are then spaced differently to test whether a child still recognizes the sets' equivalence (Piaget, Gattegno, & Hodgson, 1952). B9 and B10 are analogous to B1 and B2 but involve three instead of two sets.

The third unit relates to symbolic numerals. Children have been shown to start recognizing and manipulating Arabic digits at around 4 or 5 years of age (Gilmore, McCarthy, & Spelke, 2007; Kolkman, Kroesbergen, & Leseman, 2013; Li et al., 2018; Mussolin et al., 2014). In the first three digit tasks, numbers serve a purely nominal role. In C1, the agent receives one to five pairs of digits and needs to match them by placing corresponding digits atop each other. In C2, the agent must point at one of five numerals denoting a stated number. In C3, the agent is asked to state the name of a given digit. C4 is analogous to A7, except the subset size specification is now given by a digit rather than a number word, connecting the numeral to set cardinality for the first time. The following three tasks are ordinal tasks concerned with relations between numbers. C5 and C6 require the agent to point at the larger and smaller of two digits, respectively. In C7, the agent must sort two to four digits in ascending order by

dragging them into the correct linear configuration. C8 is similar to A7, except for the set size being denoted by a digit.

The last set of tasks is related to sorting, one of the skills thought to mark a child's entrance into the stage of concrete operations (Resnick, 1973). Although most tasks in this unit involve magnitudes rather than numerosity, it has been suggested that seriation is an essential ability for understanding the properties of numbers (Piaget, 1961). Sorting is generally considered a difficult skill to acquire, learned around 7–8 years of age (Jeske, 1978; McGonigle-Chalmers & Kusel, 2019). D1, D2, D5, and D6 require the agent to point at the largest, smallest, darkest, or lightest object in a set, respectively. Resnick et al. (1973) considered these tasks prerequisites for D3, D4, D7, and D8, where the agent must sort two to six objects according to size by placing them in the correct order. In D3 and D7, objects differ only in the attribute according to which they are to be sorted. In D4 and D8, they vary in more attributes, for example, shape, size, and luminance. Adding irrelevant cues to objects should make seriation more challenging (Tomic & Kingma, 1997). D9 requires the agent to seriate two to four whole sets by their size and thus involves both cardinality and rank. In D10, objects are arranged in one or two rows. The agent must verbally specify the ordinal position of a pointed-to object.

3.1.2. Data generation

We translate the tasks of Resnick et al. (1973) into an environment of 259 × 259 pixels with 4×4 black panels, each of size 64×64. The panels are separated by white lines of width 1 pixel and can contain 1–10 gray-scale objects or a digit from 1 to 10, depending on the task. Objects can be rectangles, triangles, circles, or ellipses. They are randomly assigned sizes, luminances, and positions. Sizes vary between 8 and 32 pixels in height or width. Luminances vary between 0.1 and 1.0 to ensure sufficient contrast with the background. Objects are initially non-overlapping, but occlusion can occur as the agent moves them around. We represent the agent with the icon of a yellow hand spawned in the upper left corner of a random panel at environment initialization. The hand can be in one of three states: open, pointing, or grabbing. At each time step, there are a total of 24 output options.

The agent can move up, down, left, or right by either small, 8-pixel, or large, 64-pixel steps (eight options). It can interact with its environment by grabbing or releasing an object, grabbing or releasing a whole set, or pointing (five options). It can also output number words from 1–10 and the word "stop" (11 options). Unlike previous work, where task IDs were encoded via binary vectors, we prompt the agent with language inputs such as "sort the numbers" or "which row has fewer objects." For each task, we collect 10,000 training examples, 1,000 test examples, and 500 validation examples using a solver which produces demonstration sequences deterministically.

The solver navigates to its target panel in 64-pixel steps, moving first to the correct row, then to the correct column. If necessary, it moves on to its target object within the set, following the same logic. In enumeration tasks, it targets the next untagged object that is closest horizontally, then vertically. If two objects have the same absolute distance, it prioritizes objects to the right, resulting in a row-wise tagging order. This is representative of linear spatial strategies employed by older children (Shannon, 1978; Wellman, Fabricius, & Chuan-Wen, 1987) and adults (Potter & Levy, 1968) in enumeration tasks. For tasks C1, B5, and B6, the solver

targets the next eligible panel with the smallest Manhattan distance to the agent and prioritizes panels below, above, to the right, and the left, in that order. When sorting objects (D3, D4, D7, D8), it goes from darkest or smallest to lightest or largest and places them next to each other at the top of the panel. It orders them from left to right, which is the preferred seriation order in many industrialized groups (Pitt et al., 2021). When sorting whole panels (C7, D9), it proceeds similarly, but may first have to remove any blocking panels from the top row in the grid.

We programmatically checked for and removed any exact duplicates in the training, test, and validation sets. For some tasks, such as A1 and C3, duplicates were unavoidable due to the limited number of task configurations. In these cases, we held out certain combinations that we only allowed to occur in the train, test, or validation split, respectively. We upsampled these combinations such that the overall number of examples remained the same across tasks. Depending on the task, we ensured a uniform distribution of set sizes, prompts, or the number of non-empty panels. This runs counter to the suggestion of Piantadosi (2016) that the developmental trajectory of number knowledge in children is influenced by the Zipfian distribution of numbers they encounter in everyday experience. However, Testolin et al. (2020) found that human-like psychophysical effects also occurred for DNNs trained with flat number frequencies.

We constructed two additional datasets. The first consists of test tasks B1, B2, and D9, with the difference that one set has 11–15 objects rather than 1–10. We use this to test the model's extrapolation on comparison tasks to larger set sizes. The second excludes all tasks involving digits or number words. The construction of this dataset was motivated by proposals in the literature that language plays a key role in learning numeracy skills (Hornburg, Schmitt, & Purpura, 2018; Purpura & Reid, 2016; Toll & Van Luit, 2014) and is a prerequisite for forming certain concepts (Carey, 2011; Gelman & Gallistel, 2004). Support for this idea comes from studies of cultures without words for larger, exact quantities, such as the Pirahã, the Mundurukú, the Tsimane, and Nicaraguan Homesigners. In adults from these cultures, the ability to represent exact numbers has been found to be limited to the range for which verbal labels are available (Pitt, Gibson, & Piantadosi, 2022). We are, therefore, interested in the effect that training a model only on non-symbolic tasks has on its performance and inner representations.

3.2. Model

Having described the tasks we aimed to solve, we now present the architecture we designed to do so. Fig. 2 shows a visualization of the model. The example in Box 1 illustrated the workings of a single attention head—our full model has 512: four attention blocks, each containing eight so-called attention layers with 16 heads. Each head can be thought of as a specialized unit that learns to focus on specific aspects of the input data during training. Using multiple heads in an attention layer allows the model to focus, in parallel, on different aspects of the input within one processing step (where a processing step is all the computations performed in one attention layer). The outputs of all heads in an attention layer are concatenated and then transformed linearly. This aggregation synthesizes the information from all heads.

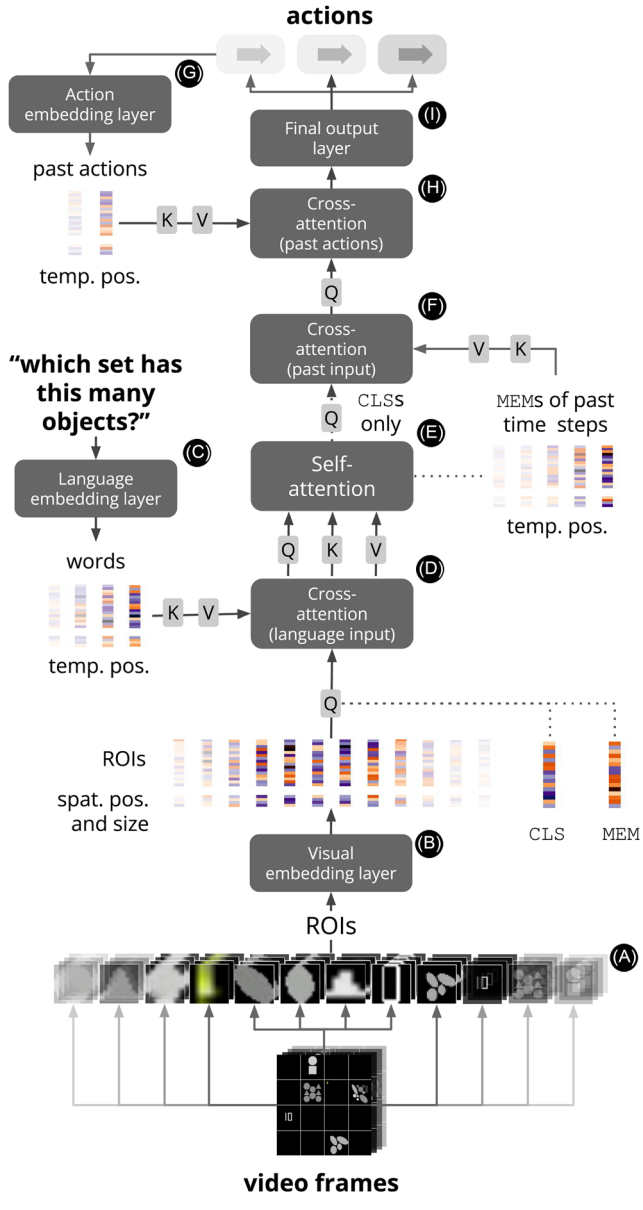


Fig. 2. Schematic of our attention-based model. Inputs consist of regions of interest extracted from each frame in the demonstration videos and a language prompt. They are processed via four attention blocks, the first two of which attend over a single time step. The third and fourth blocks take into account past inputs, compressed into the special `MEM` token, and the model's past actions, respectively.

The result is passed through a feedforward block, which consists of a small two-layer neural network. Inputs to the attention heads and the feedforward block first undergo normalization. Normalization and feedforward block were omitted from the example in Box 1 for simplicity but are commonly used components of attention layers in deeper models as they have been found to stabilize training (Lin, Wang, Liu, & Qiu, 2022). We also employ so-called "residual connections," where the attention layer's output is added to its original input before being passed to the next layer (He, Zhang, Ren, & Sun, 2016). This approach allows later heads to operate on both original inputs and results from previous heads. Multiple attention layers in an attention block enable the model to learn increasingly abstract representations of the input data.

Similar to our toy example, the model receives language and visual inputs. The language input consists of a question or instruction. The visual input is a series of video frames showing the demonstration sequence produced by the deterministic solver. To pre-process the language input, we encode each word into a binary vector, analogous to the example in Fig. 1. To pre-process the video frames, we extract regions of interest (ROIs), which may contain individual objects, digits, or an entire panel of objects (Fig. 2 Ⓐ). Such an ROI-based transformer approach has previously been applied to tasks like visual navigation (Du, Yu, & Zheng, 2021). We find our ROIs by identifying contours through morphological transformations and thresholding. Specifically, we extract ROIs by eroding each video frame with a 2×2 kernel, dilating it with a 1×1 kernel, and applying a binary threshold of value 15. Fifteen is the darkest Red, Green, Blue (RGB) color value which objects can take in our task environment. We then apply the Douglas–Peucker algorithm (Douglas & Peucker, 1973) to obtain object contours and their bounding boxes. We found that this yields ROIs of sufficient quality for our task environment; for more naturalistic inputs, CNN-based object detectors could be used. We resize all ROIs to RGB patches of size $28 \times 28 \times 3$, then flatten them into 2,352-dimensional vectors. We limit the maximum number of ROIs per frame to 85 due to computational constraints.

Having converted our linguistic and visual inputs to vector form, we feed them into separate embedding layers (Fig. 2 Ⓑ and Ⓒ) with 60 and 48 output neurons, respectively. These are single-layer neural networks, which produce an activation vector, or "embedding," for each input. So far, those vectors contain no positional information. Therefore, we concatenate the visual embedding of each ROI with its central x and y coordinates and original width and height. For each word embedding, we append a sinusoidal 16-dimensional encoding (Vaswani et al., 2017) representing its relative position in the sentence. The result is a set of 64-dimensional visual and linguistic embeddings. They are passed into the first attention block alongside two special inputs: the class token `CLS` and the memory token `MEM`. The `CLS` contains the model's prediction, that is, which action to take. The `MEM` vector compresses relevant information in each time step to be used later in the model. These are initially random, "blank" vectors, which each attention layer can modify by adding its output to them.

The first two attention blocks integrate language and visual information for individual frames. Similar to the example in Box 1, query networks in the first block's first attention layer receive visual input, and key and value networks receive language input (Fig. 2 Ⓓ). Merging multiple modalities in this way is referred to as cross-attention. A self-attention block follows

(Fig. 2 Ⓔ). Self-attention means that inputs do not come from different sources (e.g., vision and language). Instead, query, key, and value networks all receive the same inputs—in this case, the outputs from the first block. Up to this point, we process all frames in parallel but separately. That is, each frame is treated independently from previous frames. However, many tasks set out in Section 3.1 require knowledge of past time steps.

We address this need for a memory mechanism with the last two attention blocks. In the third block (Fig. 2 Ⓕ), we give the model access to inputs from past frames. The query networks of this block's first attention layer receive the `CLS` tokens output by the second block. The key and value networks receive the `MEM` tokens, concatenated with temporal position encodings (analogous to the word embeddings). We do this because there can be up to 85 ROIs in a frame and up to 100 frames in a video. Due to the quadratic complexity of the matrix multiplications involved in the naive attention mechanism, attending over every object of every previous frame would be computationally prohibitive. By forcing the model to compress relevant information into a single `MEM` vector per time step, we only need to attend over up to 99 instead of 99×85 vectors.

In the last attention block, we give the model access to its past outputs. This information is, for example, important for tasks that involve counting. Similar to the language input, past actions are converted to binary vectors and processed by an embedding layer (Fig. 2 Ⓖ) to yield 48-dimensional embeddings, which we concatenate with 16-dimensional temporal position encodings. These action embeddings serve as input to the key and value networks in the fourth block's first attention layer (Fig. 2 Ⓗ). Query networks receive the `CLS` tokens output by the third attention block (one for each time step). Finally, the `CLS` tokens are processed by an output layer (Fig. 2 Ⓘ), yielding a sequence of action predictions.

3.3. Training

We trained four models in total. The first three were trained on the dataset containing non-symbolic and symbolic tasks. We used multiple models to determine whether they would display similar final accuracies and training trajectories. The models shared the same architecture and training setup, but their random weight initialization differed. Due to the random shuffling of the dataset, they also received training samples in a slightly different order. The fourth model was trained on the dataset containing only non-symbolic tasks in order to investigate whether it would display differences in performance or internal representations.

All models were implemented in PyTorch (Paszke et al., 2019). They were trained to predict the deterministic solver's next output at each time step. This was done by minimizing cross-entropy loss, which is a measure of the difference between model predictions $\hat{y}$ and correct answers y:

$$L(y, \hat{y}) = -\sum_i y_i \cdot \log(\hat{y}_i). \tag{1}$$

We used the rectified Adam optimizer (Liu et al., 2020) and gradually adjusted the learning rate using a schedule with cosine annealing and warm restarts (Loshchilov & Hutter, 2017). The scheduler exponentially decayed the learning rate from an initial value of

Table 3
Model accuracy on the test set. Tasks are considered solved correctly if the model's predictions are identical to the deterministic solver's action sequence

Counting and Enumeration			Set Comparison			Numerals			Seriation and Ordinal Position		
ID	S	N-S	ID	S	N-S	ID	S	N-S	ID	S	N-S
A1	0.99 ± 0.01	—	B1	0.99 ± 0.00	0.99	C1	0.91 ± 0.00	—	D1	0.93 ± 0.00	0.91
A2	0.97 ± 0.01	0.86	*B1+*	*1.00 ± 0.00*	*0.99*	C2	1.00 ± 0.00	—	D2	0.96 ± 0.00	0.95
A3	0.97 ± 0.01	—	B2	0.99 ± 0.00	0.98	C3	1.00 ± 0.00	—	D3	0.86 ± 0.01	0.79
A4	1.00 ± 0.01	—	*B2+*	*1.00 ± 0.00*	*0.99*	C4	0.80 ± 0.02	—	D4	0.82 ± 0.01	0.74
A5	0.97 ± 0.01	—	B3	1.00 ± 0.00	—	C5	1.00 ± 0.00	—	D5	1.00 ± 0.01	0.99
A6	0.95 ± 0.00	—	B4	1.00 ± 0.00	—	C6	1.00 ± 0.00	—	D6	0.98 ± 0.00	0.98
A7	0.82 ± 0.02	—	B5	0.91 ± 0.01	—	C7	0.99 ± 0.01	—	D7	0.99 ± 0.00	0.97
			B6	0.88 ± 0.01	—	C8	0.96 ± 0.01	—	D8	0.98 ± 0.00	0.96
			B7	1.00 ± 0.01	0.99				D9	0.83 ± 0.03	0.64
			B8	1.00 ± 0.00	1.00				*D9+*	*0.84 ± 0.03*	*0.69*
			B9	0.96 ± 0.01	0.91				D10	1.00 ± 0.00	—
			B10	0.95 ± 0.02	0.87						

Note. S = models trained on symbolic and non-symbolic tasks N-S = model trained on non-symbolic tasks only B1+, B2+, D9+ = datasets requiring extrapolation to larger sets of size 11–15

0.005–0.0002 over four passes through the dataset (epochs), after which it was kept constant. This annealing scheme served to speed up initial training. To prevent the model from memorizing the training data too much (overfitting), we used dropout. Dropout is a technique where randomly selected neurons are temporarily disabled to prevent overreliance on individual units. We used a dropout probability of 0.1. We also applied early stopping, meaning we performed validations after every half epoch and stopped training if the model had not improved over three checks. Performance usually stagnated after around 28 epochs. We trained the models in batches of 512 samples at a time. Each epoch, including validation, took ca. 7 h on a 16-core AMD EPYC 7282 server with six GeForce RTX 2080 GPUs.

4. Results

4.1. Performance

As shown in Table 3, the model performs well on most tasks, with an overall average accuracy of 93%. Variation across models trained on symbolic and non-symbolic tasks (denoted as S in Table 3) is minimal, indicating that performance is not sensitive to weight initialization or batch ordering. When tested on comparison and seriation tasks with sets of larger size than seen in training, performance is slightly higher, presumably because of the increased contrast between set sizes. There are, however, tasks on which it consistently reaches lower accuracies, namely, A7, B5, B6, C4, D3, D4, and D9.

A7, B5, B6, C4, and D9 all require the integration of several subskills: determining the cardinality of, in the case of A7, C4, and D9, up to five sets and comparing them against either a number or multiple other sets, as well as keeping track of already tagged or obscured panels. Transformers have no recurrent connections; thus, their number of attention layers determines

the number of "reasoning" steps they can perform. While the model reaches high accuracy on prerequisites such as, for example, comparing two sets (B1 and B2), the above-mentioned tasks that require multistep combinations of such subtasks appear to strain its capacity.

The lower performance when sorting objects by size (D3, D4) seems to be due to an issue with size discrimination as the model successfully sorts objects by luminance (D7, D8). In our environment, an object's size equals its surface area, and differences may be as minor as a few pixels, whereas we enforced larger spacings for color. The model, therefore, needs to retain very granular information about each shape. This may be why the model's accuracy when choosing the smallest or largest object (D1, D2) is 2–7% below its accuracy for choosing the lightest or darkest object (D5, D6). Errors compound when the model has to compare up to six objects during seriation.

The model trained only on non-symbolic tasks (denoted as N-S in Table 3) does about as well as the model trained on the full dataset on most tasks related to object attributes, namely D1, D2, D5, D6, D7, and D8. The lower performance on tasks D3 and D4 can again be ascribed to an issue of size discrimination—while the accuracy on tasks D1 and D2 is only 1–2% below the S model, the difference compounds in the case of seriation. The N-S model also achieves similar accuracies as the S model on the two-set comparison tasks B1, B2, B7, and B8, including in the case of extrapolation to sets of larger size. The fact that its performance is not affected by a lack of symbolic training is in line with findings that comparing the cardinality of sets without having mastered symbolic counting is feasible; see studies on cultures with a smaller number lexicon (Pica, Lemer, Izard, & Dehaene, 2004), non-verbal infants (Xu, 2003), animals (Brannon & Terrace, 1998; Dadda, Piffer, Agrillo, & Bisazza, 2009; Hauser, Carey, & Hauser, 2000; Nieder, Freedman, & Miller, 2002), and neural networks without counting knowledge (Dehaene & Changeux, 1993).

However, the N-S model achieves lower performance than the S model for pointing out all objects in turn (A2), comparing three sets (B9, B10), and seriation by set size (D9). In the case of A2, this drop might be because, without the inclusion of the enumeration tasks A3–A6, the proportion of tasks that require going through a set one by one is lower, thus putting less emphasis on this skill. In the case of B9, B10, and D9, part of the issue may be that, without symbolic tasks, the model has less exposure to tasks that involve multiple non-empty panels. We also hypothesize that the S model can more efficiently parse and use numerosity information. We investigate this idea further in Section 4.4.

4.2. Training trajectories

In addition to the final accuracies reached by the model, we are interested in the order in which the model's performance progresses on the different tasks and whether this aligns with findings from educational psychology. As mentioned in Section 3.3, our dataset was randomly shuffled. While this contrasts with the sequential way children encounter tasks, neural networks trained on multiple tasks simultaneously have been found to consistently learn easier samples first (Graves, Bellemare, Menick, Munos, & Kavukcuoglu, 2017; Wu, Dyer, & Neyshabur, 2021). This allows us to compare the "implicit curriculum" that emerges for our models with the order of acquisition empirically found by Resnick within and across

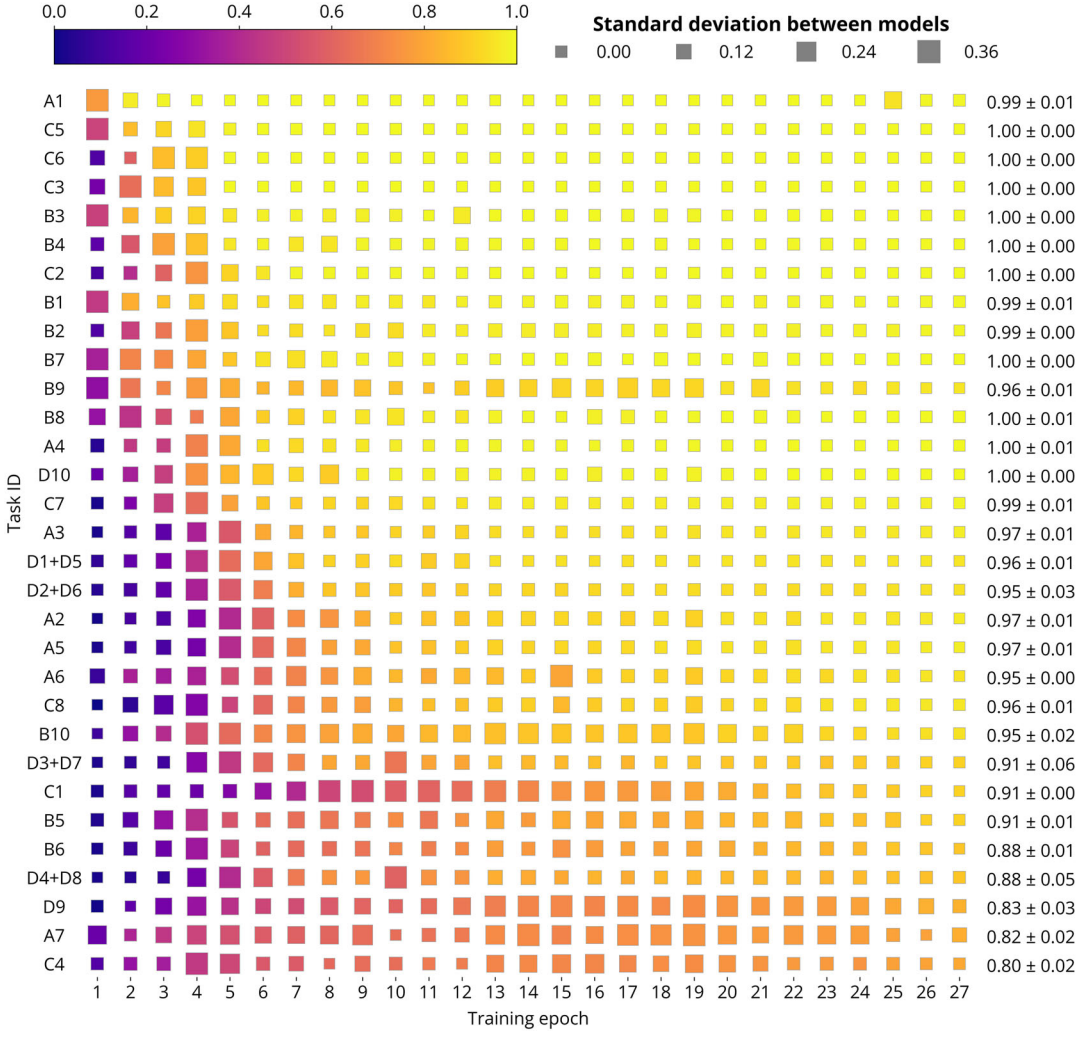


Fig. 3. Accuracy development across tasks in the course of training. Color encodes performance, while size encodes the standard deviation between the three (architecturally identical) models. Task IDs are listed on the left, and final model accuracy is listed on the right.

task families. We show the development of model performance for each task in the course of training in Fig. 3.

4.2.1. Overview: All tasks

The order of acquisition for the enumeration tasks follows the order found by Wang et al. (1971) for numbers from 6 to 10: the count list is learned first (A1), followed by counting ordered objects (A4), movable objects (A3), unordered sets (A5), subsets (A6), and finally choosing a set of specified size (A7). Wang et al. (1971) did not validate the task of touching

each object in turn (A2), but this skill was hypothesized to emerge before tasks A4 and A3. However, our model acquires A2 simultaneously with A5—likely reflecting the tasks' similar demands on memory, which plays less of a role in A3 and A4.

For set relation tasks, a direct comparison with human data is only possible in some cases, as B3–B8 were not empirically validated. Regarding the tasks that were tested with children (B1, B2, B9, B10), accuracies progress as expected: "More" tasks (B1, B7, B9) are learned before "less" tasks (B2, B8, B10) (Resnick, 1973; Resnick et al., 1973), and two-set comparisons (B1, B2, B7, B8) are learned before three-set comparisons (B9, B10). In fact, three-set comparisons were excluded from analysis by Wang et al. (1971) because too few subjects mastered them. However, unlike children who first acquire non-symbolic comparisons, the model begins by learning to select between a digit and a set. We discuss this in more detail towards the end of the section. B5 and B6 are learned last, reflecting the higher demands on the model: it needs to compare a set and multiple digits and keep track of tagged digits, making the task more challenging than just navigating to a single panel and pointing.

For tasks involving numerals, training trajectories only partially align with those found by Wang et al. (1971) and Wang (1973). Digit identification (C3) does precede digit comparison (C5, C6), which precedes seriation (C7). However, matching digits (C1), which were mastered by human subjects before any other numeral task, is acquired last by the model. The reason may be that, in our setup, C1 is the only task of its kind and involves longer and more complex navigational sequences. Learning to state (C2) and select digits (C3) is also switched compared to children, likely because outputting a number word simply means activating a single node for our model. In contrast, speech production in humans involves more complex articulatory coordination.

Seriation and ordinal position tasks were also not empirically validated by the authors of the original curriculum. However, Jeske (1978) investigated prerequisite skills in children tasked with ordering plastic strips of different lengths and found that the selection of the longest strip preceded correct seriation. In line with these findings and the hypothesized training trajectory, selecting the largest, darkest, lightest, or smallest object (D1, D2, D5, D6) is achieved first, followed by object seriation (D3, D4, D7, D8), and finally, set seriation by cardinality (D9). Naming an object's ordinal position (D10) is learned earlier than was hypothesized by Resnick (1973). However, other studies have found ordinal concepts to precede cardinal concepts (Brainerd, 1973) and seriation (Siegel, 1971) in children.

We now turn to the training trajectories across task families. The first tasks the model learns are mostly symbolic (A1, C5, C6, C3, B3, B4, C2), followed by non-symbolic two-set comparison (B1, B2, B7, B8), then enumeration and ordinal position tasks (A4, D10, A3, A2, A5, A6, C8). Concurrently, the model learns to select single objects by a specified attribute (D1, D5, D2, D6). The tasks that develop the latest require comparing or manipulating more than two sets of objects (B9, B10, D3, D7, C1, B5, D4, D8, B6, D9, A7, C4). Training trajectories show gradual development, characteristic of neural networks, and are consistent with findings from various aspects of mathematical cognition (Mareschal & Shultz, 1999; McClelland, Mickey, Hansen, Yuan, & Lu, 2016). Furthermore, students' development of early numerical competencies is not always linear, and their skill acquisition timelines may differ (Powell & Fuchs, 2012). Similarly, a model's performance on a task will sometimes

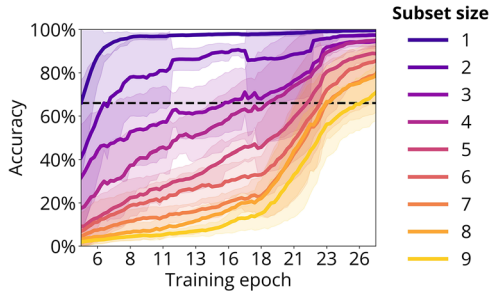


Fig. 4. Accuracy development (smoothed) for the A6 task (give-N), grouped by subset size. Shaded regions indicate standard deviation. The dashed line represents the threshold at which a learner is typically considered an N-knower.

drop momentarily (see, e.g., A6), leading to a dip in average performance and an increased standard deviation.

In general, the tasks acquired faster by the model are ones with less variability across examples, shorter sequence length, fewer memory requirements, and more exposure—either because there are limited task configurations that were upsampled or because there are very similar tasks that can serve as a scaffold. These are features of most tasks involving number words and digits, which is likely why they are acquired earlier than purely non-symbolic ones. This contradicts the order observed in children, who typically develop non-symbolic numerical representations before symbolic ones (Li et al., 2018; Matejko & Ansari, 2016; Wang et al., 1971).

4.2.2. Spotlight: Give-N task

Having looked at training trajectories across the dataset, we now focus on a task that has received considerable attention in the numerical cognition literature: The give-N task. A prominent proposal for the developmental trajectory on this kind of task is a series of six performance levels: pre-numeral-knower, one-knower, two-knower, three-knower, four-knower, and cardinal-principle (CP) knower (Carey & Sarnecka, 2006; Sarnecka & Carey, 2008; Wynn, 1992). Pre-numeral-knowers will give random amounts in response to a give-N instruction. One-, two-, three, and four-knowers can give out one, two, three, and four objects, respectively, but fail at all other numbers. CP-knowers can solve any give-N task. According to the knower-level theory, children learn the meanings of numbers one through three or four one after the other. However, once they uncover the cardinal principle, tasks with higher numbers are mastered simultaneously. Several studies support this view, although some have questioned whether a true semantic inductive leap underlies the transition to CP-knower (Davidson, Eng, & Barner, 2012). Others have found that early stages may be noisier than previously assumed (Wagner, Chu, & Barner, 2019).

We show the training trajectory of our model on task A6 separately for each subset size in Fig. 4. The order of acquisition goes from smallest to largest numbers. Performance on subset size one increases first and remains high. The training trajectory for subset size two shows

the same concave shape but with an accuracy gap of 10–25%, which is only closed towards the end of training. Subset sizes three and four are learned relatively simultaneously, with an almost linear development slope. Training trajectories for tasks with subsets of size five and up form a group of convex-shaped curves. Although the graph shows no instantaneous transitions, there is a point around epoch 18 during which the performance on subsets larger than two begins to rise more steeply. This behavior is somewhat in line with the knower-level stages observed in children. However, it may not necessarily reflect any realization of a fundamental underlying principle. The training trajectories are likely also shaped by sequence length and the fact that the "visuo-motor" routines needed to complete tasks with smaller subsets are implicitly contained in those with larger subsets, leading to more training exposure.

The CP trajectory has previously been modeled computationally. Instead of using a connectionist approach, where knowledge is encoded in a set of weights, Piantadosi, Tenenbaum, and Goodman (2012) proposed a model based on Bayesian program induction. The model learned to combine pre-defined operations, a so-called language of thought, to count occurrences in a set. Its training trajectory mimicked the proposed CP leap. The model by Sabathiel et al. (2020a) also successfully learned a give-N task, although its learning curves did not follow the CP trajectory. Dulberg et al. (2021) trained a reinforcement learning agent consisting of specialized pre-trained modules to select N items from a binary vector using a curriculum approach. Similar to our model, the agent showed a gradual progression characteristic of neural networks but did exhibit an inflection during training.

4.3. Analyzing model predictions

Having inspected the model's training trajectories, we now turn to analyzing the trained model on a "behavioral" level, that is, investigating its output predictions and error patterns.

4.3.1. Recognizing exact numerosity

Two core numerical systems are often distinguished in the literature on numerical cognition (Feigenson, Dehaene, & Spelke, 2004): The object tracking system (OTS) and the approximate numerical system (ANS). The OTS is said to sustain the fast and precise enumeration of sets with up to five objects without counting, an ability referred to as subitizing. The intuitive estimation and approximation of larger sets are proposed to rely on the ANS. This dichotomy has received support from many investigations of humans and non-human animals (Agrillo, Piffer, Bisazza, & Butterworth, 2012; Burr, Turi, & Anobile, 2010; Hyde & Spelke, 2009; Mandler & Shebo, 1982; Revkin, Piazza, Izard, Cohen, & Dehaene, 2008). However, it has been challenged by some who suggest that a single system is responsible for both subitizing and counting (Piazza, Mechelli, Butterworth, & Price, 2002). Whatever the underlying mechanisms, it has been widely shown that processing smaller numerosities is more precise than processing larger numerosities.

To see whether this is also the case in our model, we let it interact with 1,000 instances of a task environment requiring it to select a set of a given size (A7). We generate an equal number of tasks for each prompt and plot the target set size against the size of the set chosen by the

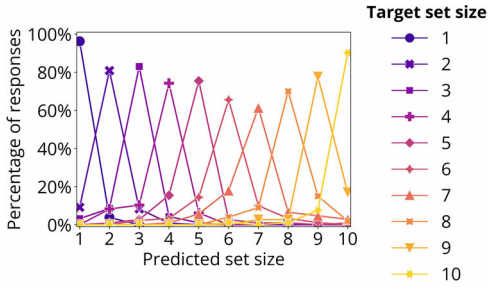


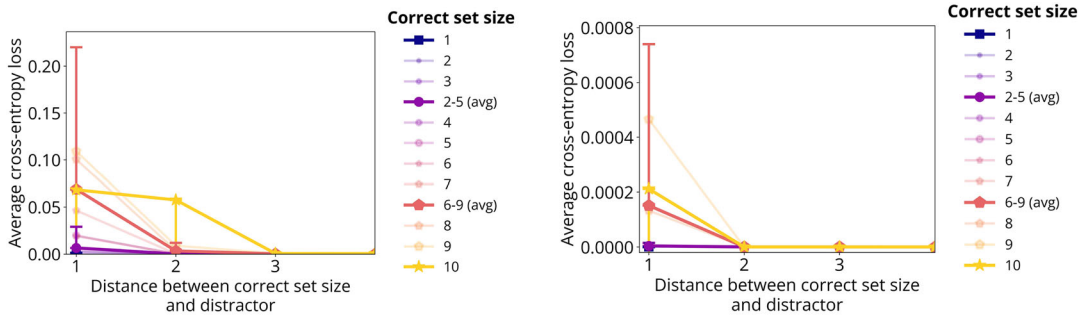
Fig. 5. Target set size plotted against the size of the set chosen by the model on 1,000 instances of the A7 task environment (choosing a set of stated size).

model in Fig. 5. The model shows decreasing accuracy and broader response variability with increasing target numerosity, in line with human experimental data. However, performance increases again for larger numerosities. Creatore et al. (2021) trained a DBN on an enumeration task and observed a similar effect. They noted that this was an artifact of the limited range of numerosities used, which is also the likely explanation in our case.

4.3.2. Size and distance effects

In infants, adult humans, and a variety of animal species, numerosity comparisons are characterized by size and distance effects: comparisons are faster and more accurate when there is a larger difference between two numbers (distance effect) and when numbers are smaller (size effect) (Dehaene, Dehaene-Lambertz, & Cohen, 1998). That is, comparing 1 versus 9 is less error-prone than 1 versus 2, and 1 versus 3 is easier than 7 versus 9. A prominent explanation for this phenomenon is that numbers are stored on a "mental number line," where close-by numbers overlap, and their noise is proportional to their value (Verguts & Fias, 2004). In humans, size and distance effects hold for symbolic and non-symbolic stimuli (Lyons & Ansari, 2015), although they are minute for judgments on number symbols (Buckley & Gillman, 1974).

We analyze whether our model displays symbolic and non-symbolic size and distance effects by evaluating its performance on two-set (B1, B2) and two-digit comparison tasks (C5, C6). Since the model performs very well on these tasks, accuracy is not a meaningful metric to compare. Instead, we use the model's cross-entropy loss on the test data, averaged over time steps within a task. We plot this against the distance between the correct number and its distractor, shown in Fig. 6. In both the symbolic (Fig. 6b) and non-symbolic (Fig. 6a) cases, target size one has the lowest error and almost no variation, followed by target sizes two to five. Errors and variations increase for target sizes six to nine, particularly in non-symbolic tasks. Similar to task A7 (Section 4.3.1), performance increases for target size 10—again, likely an artifact of the limited range of numbers used. In line with human behavioral studies, the error range for non-symbolic comparisons is higher than for symbolic comparisons.



(a) Non-symbolic set comparisons (B1, B2)

(b) Symbolic digit comparisons (C5, C6)

Fig. 6. The model's cross-entropy loss on symbolic and non-symbolic comparisons, averaged over the time steps within a task. Error bars indicate standard deviation. The distance between the correct set or digit and its distractor is shown on the *x*-axis. Color and markers encode the correct size. We aggregate similar graphs by average for better visibility but keep the individual plots in the background for completeness.

4.3.3. Applying the logit lens

As mentioned in Section 3.2, the special token `CLS` contains the model's prediction. Each attention head can contribute to `CLS`, gradually refining the prediction until it is translated to an action by the model's output layer. However, it is possible to directly read out the prediction's state in any intermediate attention layer. This approach has been dubbed the "logit lens" and shown to provide relatively coherent internal prediction trajectories for LLMs such as GPT-2 (nostalgebraist, 2020). Although our model differs from purely text-based language models in that it operates in multiple modalities, it shares the same architecture. It thus lends itself to applying the logit lens.

We evaluate our model on each test task, decode the nascent prediction in `CLS` at every attention layer, and log its accuracy. The result is shown in Fig. 7. We also include the logit lens for the model trained without symbolic tasks, denoted as N-S. How early or late a task reaches high accuracy can be seen as a measure of difficulty—analogous to reaction time in humans: Some tasks require more processing steps, that is, attention layers, to arrive at a solution. Alternatively, the model may resort to higher attention layers because information about past inputs is only provided after the second attention block (see Fig. 2).

Outputs of attention layers in the first attention block indicate that they prime the model for the type of answer called for by a prompt. For example, when asked for an ordinal position (D10) or a digit's name (C3), the initial prediction is a default number such as 5 or 2. For tasks requiring recognition of a final state, the default output is "stop," while for those calling for selecting a panel or object, the default is to point. Any correct predictions in these first attention layers are by chance, for example, when the agent starts off positioned correctly, then points. The second attention block shows a decrease in correct default answers, suggesting the involvement of inhibitory mechanisms at this stage.

The order of prediction trajectories mostly fits with the sequence of acquisition found in Section 4.2. The fastest tasks to reach high accuracies are comparisons (B1, B2, B3, B4, B9, B10, C5, C6) and pure digit tasks (C2, C3). In contrast, tasks involving comparing or

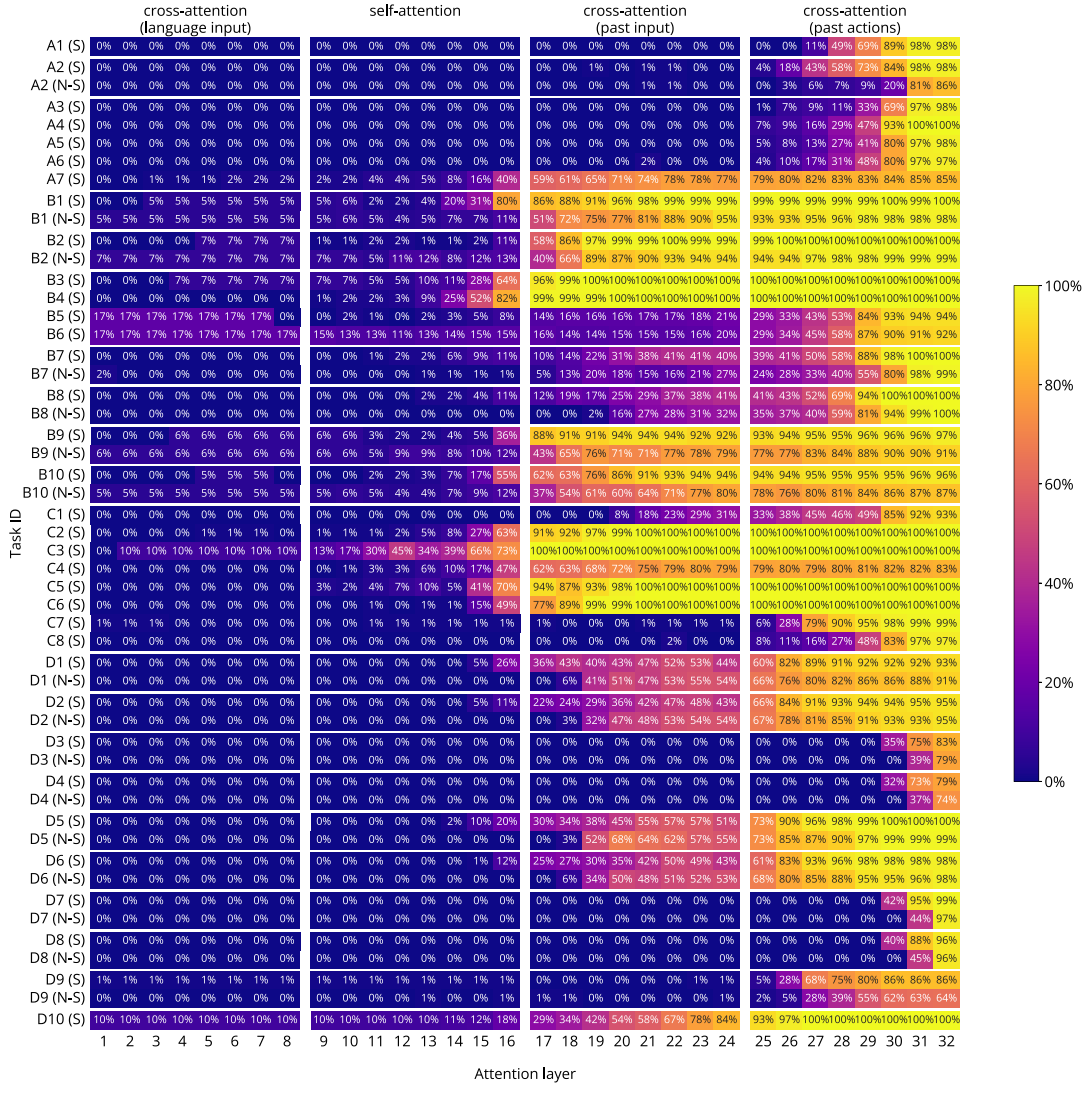


Fig. 7. Performance of the model on each of the test tasks when cutting off the prediction process at a certain attention layer, specified on the *x*-axis. The "S" in the *y*-axis labels denotes the model trained on all tasks, while "N-S" denotes the model trained only on non-symbolic tasks. Accuracy is encoded via color.

manipulating multiple sets, objects, digits, or knowledge of past time steps require more processing steps. This is generally congruent with event-related potential (ERP) studies showing that comparison is associated with modulations of an early component while spatial mappings are associated with later ERP components (Toomarian & Hubbard, 2018). Less congruently, digit-set comparisons (B3, B4) are among the first tasks to reach high accuracy, whereas studies show high switching costs when humans are asked to compare symbolic and non-symbolic numbers (Finke et al., 2021; Lyons, Ansari, & Beilock, 2012).

The N-S model requires more processing steps than the S model, even when the final accuracy on a task is similar, indicating differences in internal processing. Notably, the accuracy progression in the N-S model is gradual for comparisons of two or three sets (B1, B2, B9, B10). This contrasts with the S model's prediction trajectories on these tasks, which show sudden performance increases between attention layers. However, on the ordinal position (D10) and row comparison (B7 and B8) tasks, the S model's accuracies increase more steadily. This linear progression is particularly striking for D10, which also has the benefit of involving only a symbolic output, making intermediate predictions more human-interpretable. We, therefore, use this task to investigate the processing underlying such gradual prediction trajectories in the following section.

4.3.4. Determining ordinal position

It has yet to be understood how humans and animals process non-verbal serial order information. However, behavioral and neuronal data suggest an imprecise representation of discrete numerical rank, similar to an analog magnitude mechanism proposed for cardinality (Nieder, 2005). Studies in humans and macaques have identified brain areas similarly activated by numerical quantity and rank order information, suggesting a shared neural system for these processes (Marshuetz, Smith, Jonides, DeGutis, & Chenevert, 2000; Nieder et al., 2002; Ninokura, Mushiake, & Tanji, 2003, 2004).

Determining an object's position in a sequence also seems to involve a mixture of cardinal and ordinal number usage in our model. Fig. 8 shows two D10 example tasks and how the model's prediction changes after each attention layer of the third attention block. Predictions take the form of probabilistic distributions centered on one or more outputs. These distributions gradually move along the number line. Note that this happens "silently," that is, the model is not trained to output numbers at each time step, only to produce the final answer. The strategy it develops to do so is evocative of an internal counting procedure. However, the model does not necessarily go through the count list individually. In Fig. 8b, it starts directly at the end of the first row with "5," from where it moves up towards "8" (the correct answer), essentially skipping over "6." This behavior is similar to adaptive grouping strategies people employ when enumerating larger groups of objects or solving number line estimation tasks (Camos, 2003; Newman, Friedman, & Gockley, 1987; Starkey & McCandliss, 2014; Schneider et al., 2018). The model may also start at the end of a row, following the number line in reverse order (see Fig. 8a).

These examples indicate that the gradual internal progression found for D10 in Fig. 7 stems from the model internally tagging one object (or group of objects) per processing step until it has identified the requested ordinal position. To provide additional support for this assumption, we plot the accuracy on the D10 test set throughout the third and fourth attention blocks as a function of the target object's distance from the nearest row start or end. The result is shown in Fig. 9. The model internally reaches its conclusion faster on tasks with target objects closer to a row's edge and with target objects in the first row—consistent with the hypothesis that tasks requiring less "internal counting" involves fewer processing steps.

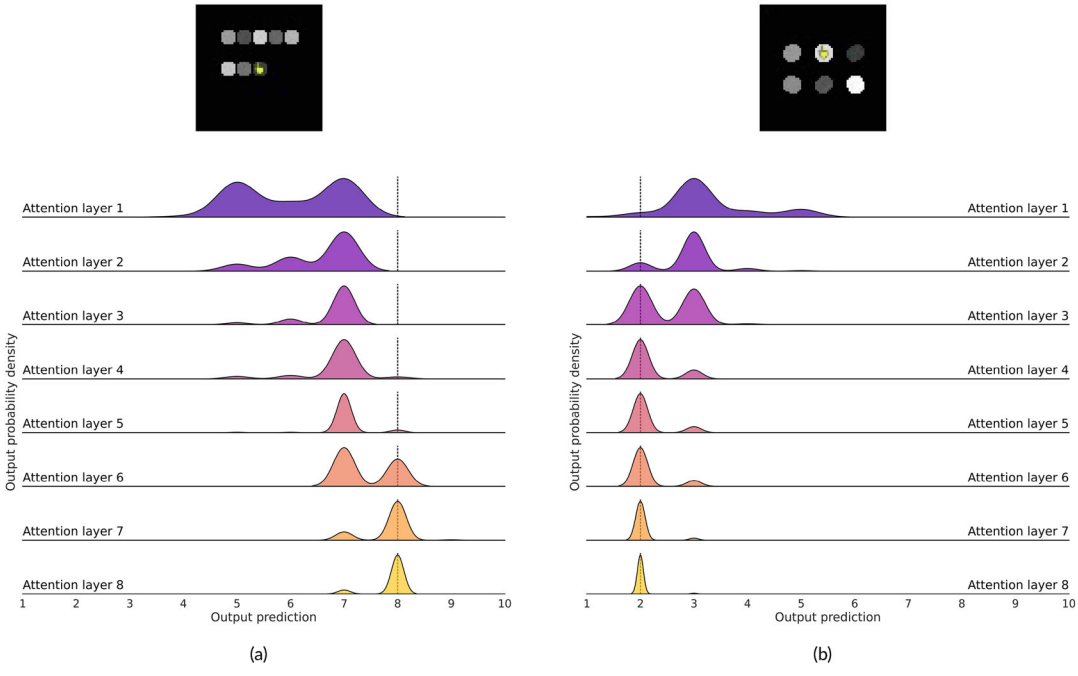


Fig. 8. Examples of the progression of an individual prediction on task D10 throughout the attention layers of the model's third attention block. Visual task input is shown at the top. The agent has to name the ordinal position of the object to which the yellow hand is pointing. The *x*-axis shows the 10 number word outputs. The *y*-axis shows the density of the probability distribution over these outputs, as predicted by the model, in each attention layer. A dashed line marks the correct output.

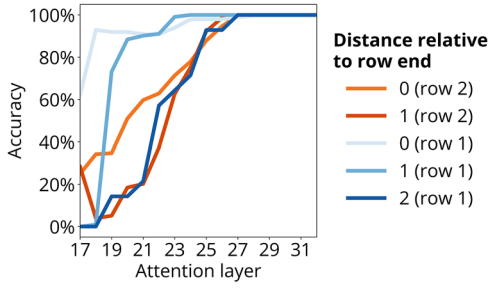


Fig. 9. Accuracy on the D10 test set throughout the third and fourth attention blocks as a function of the target object's distance from the nearest row start or end. Tasks with target objects in the first and second rows are plotted separately.

4.4. Analyzing model representations

In the previous sections, we looked at the model's outputs and error patterns in the context of human behavioral data. We now turn our attention to its internal representations.

4.4.1. Integrating multiple modalities

Many neuroimaging and behavioral studies have investigated where and how the human brain processes numerical inputs. One prominent proposal, the triple-code model (Dehaene, 1992), argues for three codes with which we mentally represent numbers: symbolic digits, verbal number words, and non-symbolic quantity representations. The codes are thought to depend on distinct neural substrates, with visual inputs such as Arabic numerals most likely depending on ventral occipitotemporal structures, verbal representations depending on left frontal and temporal language areas, and analog magnitudes depending on the parietal cortex (Hubbard, Piazza, Pinel & Dehaene, 2005). However, functional magnetic resonance imaging (fMRI) studies have shown that numerical tasks, even those involving only one representational format, activate a distributed network of areas, including the frontal and parietal lobes (Hubbard et al., 2005).

In this section, we seek to analyze how our model processes and integrates information from different modalities and whether a similar picture of specialized and integrative areas emerges. We begin by creating isolated probes of input stimuli from different modalities. We then feed these isolated inputs to the key, query, and value networks of every attention head in the model and measure how strongly they react to each probe.

Our visual probes consist of 1,051 representative patches, including digits, different luminances, the agent's hand in its three states, shapes of varying resolutions, and panels with object sets of sizes 1–10. We apply the visual embedding layer (Fig. 2 Ⓑ) to each probe but do not add size or position information. Instead, we create separate size probes, spaced evenly from 4×4 to 64×64, and position probes, spanning 65 locations across the input grid. The language probes consist of 107 vectors representing every word in the vocabulary, encoded by the language embedding layer (Fig. 2 Ⓒ) and concatenated with each position at which a word may appear in the task prompts. Probes for previous actions consist of all 24 possible outputs encoded by the action embedding layer (Fig. 2 Ⓖ) and 100 isolated temporal position embeddings. Finally, we create probes that measure sensitivity to the state of the `CLS` token. We translate all possible actions E_{Pred} back into internal model representations by applying the output layer W_{Pred} (Fig. 2 Ⓘ) "in reverse." Specifically, we subtract W_{Pred}'s bias term b_{Pred} from E_{Pred} and apply the pseudo-inverse $W_{\text{Pred}}^{\dagger}$:

$$W_{\text{Pred}}^{\dagger}(E_{\text{Pred}} - b_{\text{Pred}})^{T}.$$

For each network in each attention head, we record the 10 probes that evoke the largest response, quantified as the sum of the network activations' absolute values. In Fig. 10, we show which modality these inputs belong to and the strength of the response they elicited. The first two attention blocks integrate language and visual information. Query networks of heads in the very first attention layer receive visual input. Key and value networks receive language input. As might be expected, query networks in the first block mainly respond to image patches, and key networks mainly respond to words. The value networks react to a mix of language, visual, and output predictions (`CLS`). The partial sensitivity to nascent predictions fits our observations from Section 4.3.3 that the model forms "default" outputs at this level.

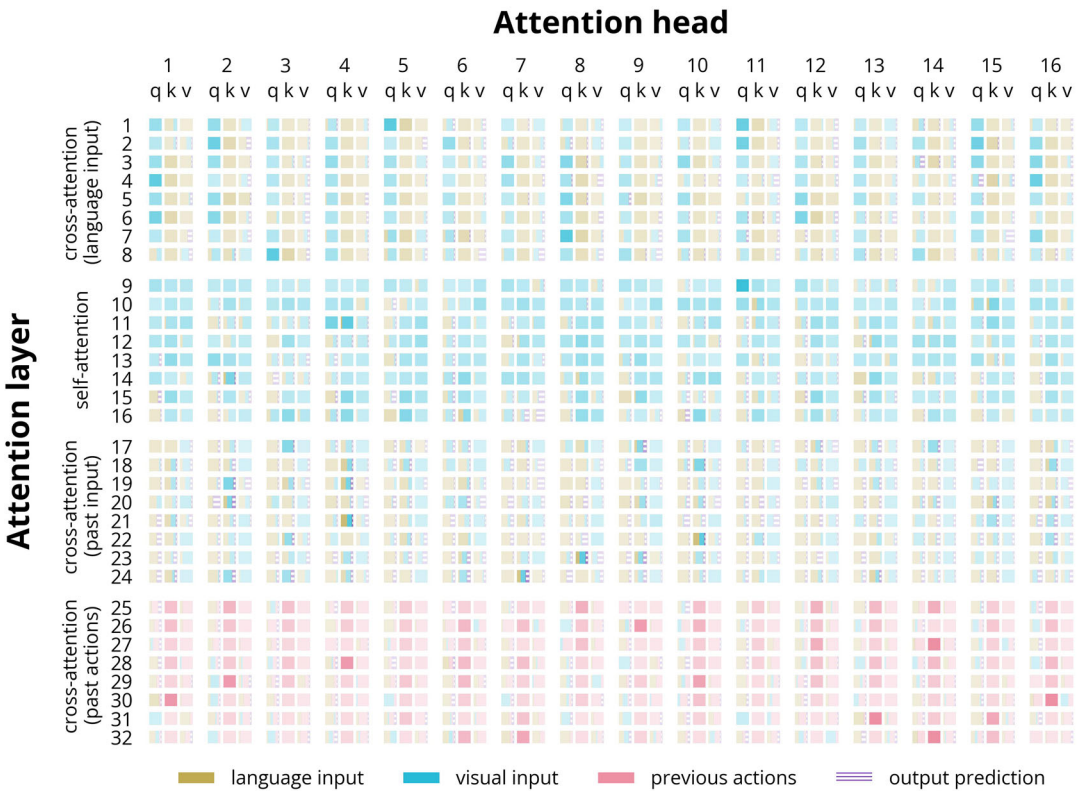


Fig. 10. Visualization of the sensitivity of every query, key, and value networks in the model to isolated probes from different input modalities. Opacity indicates the strength of activation exhibited by a network in response to the input probes.

Heads in the second attention block primarily integrate visual information, although some exhibit sensitivity to words.

In the last two attention blocks, information from past time steps enters the picture. The third block is of particular interest because the key and value networks in its first attention layer receive `MEM` vectors, that is, the time step representations produced by model 2 Ⓕ). Fig. 10 shows that these compressed representations seem to contain a mix of visual, linguistic, and output prediction information. We also see more sensitivity to output predictions, which matches our finding from Section 4.3.3 that many tasks are already solved at this stage, and predictions undergo little to no further refinement. In the last block, query networks process a mix of linguistic, visual, and output prediction input, while key and value networks are predominantly sensitive to previous actions.

Overall, Fig. 10 paints a picture of a distributed network of specialized processing units integrating multimodal information. There are very few unimodal heads—primarily in the second attention block. Most heads consist of an unimodal query network interacting with key and value networks sensitive to different modalities. In a few heads, particularly

in higher attention layers, single key, query, or value networks respond to inputs from multiple modalities.

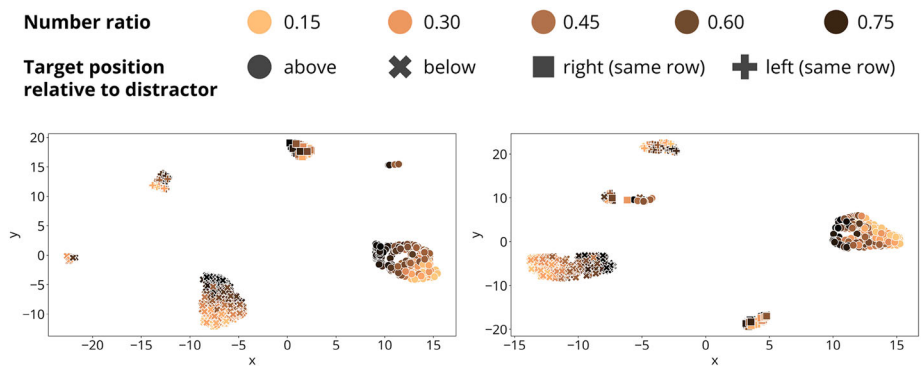
4.4.2. The effect of symbolic training

Having observed in Section 4.3.3 that set comparison tasks require more processing steps in the N-S model than in the S model, we here investigate this finding further. We run both models on the test data for tasks B1 and B2 and collect the inputs to the third attention block, as our analysis in Section 4.3.3 showed this to be the point where two-set comparison predictions begin to form. We collect only the time step where the agent is positioned at the correct set but has not yet selected it to facilitate cross-task comparison. We visualize the collected `CLS` and `MEM` vectors using pairwise controlled manifold approximation (PaCMAP) (Wang, Huang, Rudin, & Shaposhnik, 2021). PaCMAP is a method for transforming high-dimensional data into a lower dimensional space while still preserving the data's local and global structure. Fig. 11 gives insight into the differences between the internal representations of the S and N-S models and the role of `MEM` vectors. Proximity of points indicates similarity.

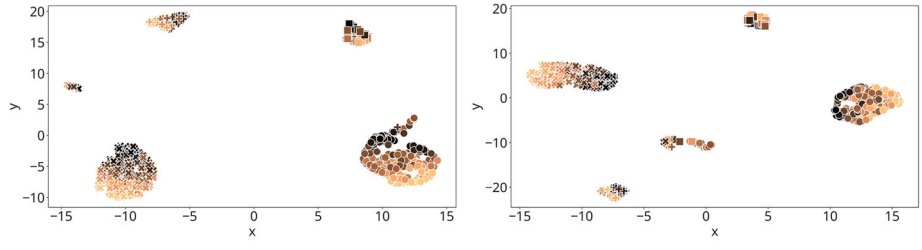
We begin with the `CLS` vectors, which encode the model's predictions. For task B1, these form distinct clusters according to the position of the target set relative to its distractor (Fig. 11a). Within the clusters, tasks with similar number ratios, calculated as the smaller set size divided by the larger set size, are grouped closer together. However, for the N-S model, this stratification is slightly less pronounced. There is also a collection of "miscellaneous" predictions that are not yet well clustered, indicating that further processing steps are needed. `CLS` vectors for task B2 (Fig. 11c) are less neatly grouped than for B1, which fits with the observation from Section 4.3.3 that "less" comparisons are solved in higher attention layers than "more" comparisons. The PaCMAP for the N-S case is almost circular, reflecting that many `CLS` tokens have few neighbors of high similarity. The arrangement indicates that, at this stage, the vectors still contain perceptual details that have already been abstracted away in the S model.

We now turn to the `MEM` vectors (Figs. 11b and 11d, which contain compressed information the model deemed relevant enough to "remember" about a time step. The `MEM` PaCMAPs closely resemble the `CLS` PaCMAPs in their differences between tasks B1 and B2 and S and N-S models, as well as their stratification according to number ratio and target position. This suggests that `MEM` and `CLS` contain similar information. To test this hypothesis, we evaluate the models on tasks B1 and B2 as before but replace the `MEM` vectors with `CLS` vectors after the second attention block. We see no decrease in performance, confirming that the two are interchangeable, at least for set comparison. For other tasks, such as A5, doing this does cause a significant accuracy drop from around 98% to 13%, showing that `MEM` vectors carry crucial additional or complementary information in some cases.

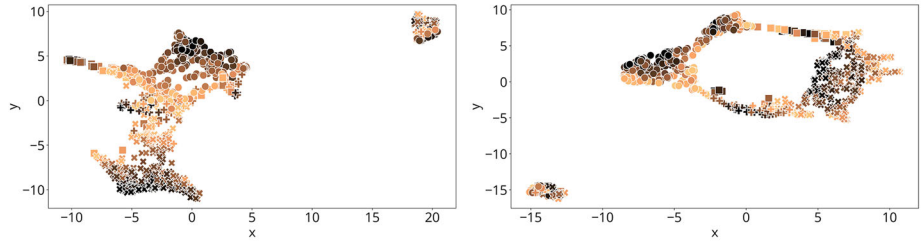
We can conclude that set relations are implicitly quite well defined by attention layer 16, although slightly less so for the N-S model. To quantify this gap further, we train two linear regression models to predict the size of a task's larger and smaller set based on the models' B1 `CLS` and `MEM` vectors. We do this for each attention layer in the second attention block. For the N-S model, the coefficient of determination goes from an average of 83% in the first to 93% in the eighth attention layer. For the S model, it goes from 83% to 97%, suggesting



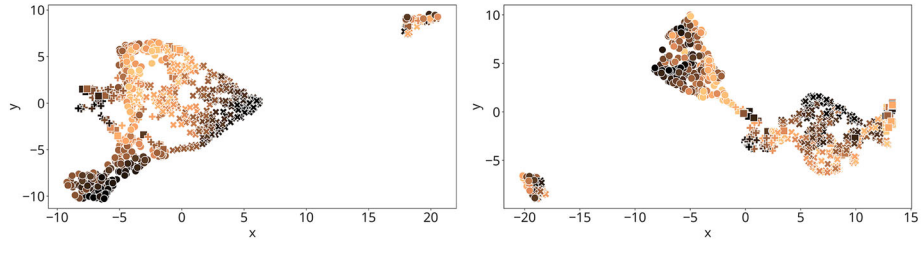
(a) PaCMAP of CLS vectors produced by the S model (left) and N-S model (right) on task B1.



(b) PaCMAP of MEM vectors produced by the S model (left) and N-S model (right) on task B1.



(c) PaCMAP of CLS vectors produced by the S model (left) and N-S model (right) on task B2.



(d) PaCMAP of MEM vectors produced by the S model (left) and N-S model (right) on task B2.

Fig. 11. Pairwise controlled manifold approximation (PaCMAP) applied to the CLS and MEM representations produced by the models trained with both symbolic and non-symbolic tasks (S) and on non-symbolic tasks only (NS), collected after the second attention block during the processing of two-set comparison tasks (B1 and B2).

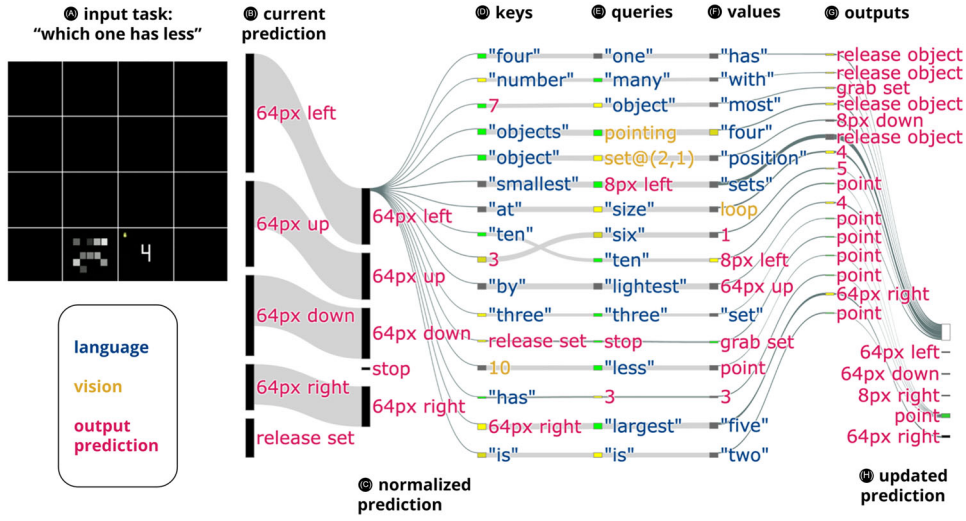


Fig. 12. A graph of the information flow in the style of Katz & Belinkov (2023) of the third attention layer of the first attention block while processing one time step of a B4 task. Nodes represent groups of activated neurons. Edges represent interactions, with width indicating interaction strength. Nodes are labeled with the most likely prediction when processed by the model's final output layer, except for keys, queries, and values, where we use the probe from Section 4.4.1 eliciting the most similar activation. Node color represents whether activations, when interpreted as predictions, have the correct action (pointing) as either the most likely (green) or second-most likely (yellow) output. The graph should be read from left to right and omits the attention layer's feedforward block for simplicity.

that it produces slightly more precise representations of set cardinality earlier, on which its higher levels can operate. In the N-S model, which appears to require more processing steps, that is, attention layers, to determine set size, fewer attention layers are available for higher level operations once cardinality information has been determined. This leads to a lower performance on tasks like set seriation (D9).

4.4.3. Visualizing an information flow

The analyses presented so far have mostly looked at static model weights for one or more entire task families. We now want to provide a glimpse into the dynamics that unfold while processing a single task. We use an information flow graph in the style of Katz and Belinkov (2023), who recently proposed this kind of visualization for LLMs. We adapt their tool to our multimodal case to show a snapshot of the information flow in the model's 11th attention layer during one time step of the first B4 task in the test set (Fig. 12). We choose task B4 as it is relatively simple but involves the comparison of a set size and a digit—in this case, a set of size 10 and the numeral four (Fig. 12 Ⓐ). This makes it an interesting case for investigating the two number formats' representations. We choose the 11th attention layer because it is the first point in the feedforward pass in which the correct action enters the model's top five most likely predictions, indicating that the attention layers' heads play a role in solving the

task. Nodes represent groups of activated neurons. Edges represent interactions, with width encoding interaction strength.

The attention layer receives the model's current state as input. This state is a high-dimensional vector that is not human-interpretable. However, we can translate it to an action prediction by directly applying the model's final output layer (Fig. 2 Ⓘ). We show the five most likely outputs as separate bars (Fig. 12 Ⓑ). Length indicates certainty. The correct answer is to point because the agent is in the right panel. This action is not yet among the top outputs. The prediction undergoes normalization (Fig. 12 Ⓒ), which has been found to act as a "semantic filter" in LLMs by dampening the effect of common inputs and boosting the signal of rare tokens (Katz & Belinkov, 2023). In our case, normalization does little except increase the likelihood of the "stop" action.

What follows are the outputs of the key, query, and value networks in the attention layer's 16 heads (Fig. 12 Ⓓ–Ⓕ), each represented by a node. As we saw in Fig. 10, the networks may encode linguistic or visual information. To "decode" their outputs, we compare their activations with those they exhibited in response to the probes in Section 4.4.1 and use the closest match as node labels. Labels are colored according to modality. We also translate each network output to an action prediction, as we did for the attention layer input (Fig. 12 Ⓑ). The color of each node represents whether this translation yields the correct action (pointing) as the most likely (green) or second-most likely (yellow) output. This color-coding indicates whether a network contributes to the correct prediction.

The results of the interactions between keys, queries, and values pass through the heads' output layers (Fig. 12 Ⓖ). The individual heads' outputs are aggregated into an updated prediction (Fig. 12 Ⓗ). This updated prediction is added to the attention layer's original input and processed by further normalization and a feedforward block, which we do not depict for simplicity. The attention layer shown in Fig. 12 is a relatively early one, and the updated prediction it produces is still almost uniform. However, the correct action, pointing, has now entered the model's top five predictions due to the contributions from the attention layer's heads.

If we consider the mechanism formed by keys, queries, values, and outputs as an associative process, we see that the model retrieves relevant information, including representations learned from other tasks. For example, there are activations for the visual digit 10, a pointing hand, and the number words "10" or "4"—none of which are in the task's immediate input. Most heads output the action "point," which modifies the model's top five predictions to include pointing. However, the output predictions "three," "four," or "five" also appear across the attention head. This activation of surrounding number outputs can be explained by looking at the weights in the model's final output layer. Fig. 13 shows the cosine similarity of the incoming weights for each possible output. Similarity for weights of neighboring numbers is higher than for numbers further away, leading to a coactivation of close-by numbers in line with Verguts & Fias (2004)'s proposal of a noisy mental number line. The fact that various visual, spatial, and output prediction nodes appear in the graph also fits well with Abrahamse, Braem, Notebaert, and Verguts (2016)'s proposal that performing a task coactivates perceptual, motor, and goal representations in the brain, binding them into a context-specific associative network which allows for cognitive control.

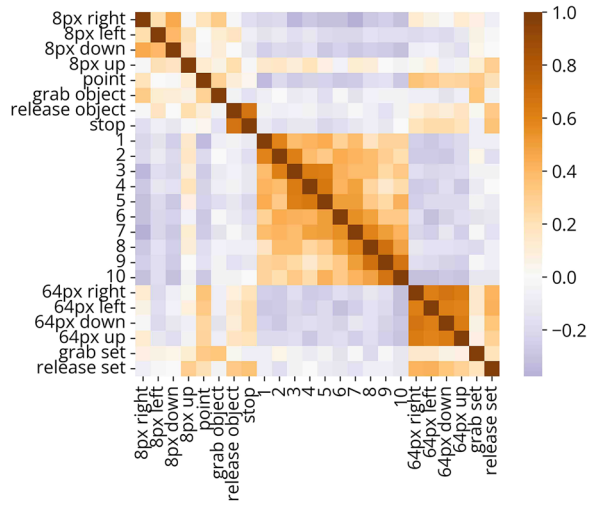


Fig. 13. Cosine similarity of each output's incoming weights in the model's final output layer.

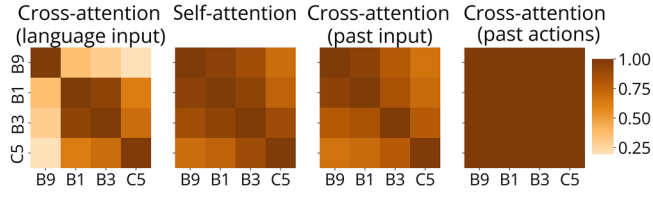


Fig. 14. Cosine similarity between aggregated activation trajectories for the four tasks in the dataset involving the "more" relation, in each of the four attention blocks.

4.4.4. Comparing task-processing sequences

Our dataset spans a range of number concepts and task families, which enables us to compare them from various perspectives. So far, we have looked at the order of acquisition during training in Section 4.2 and within-model prediction trajectories in Section 4.3.3. Finally, we want to compare the model's internal activations while processing different tasks. We run the model on all tasks in the test set and collect each attention layer's 64-dimensional attention head outputs at every time step. We average the recorded activations over the time steps of a single task and sum over the 1,000 tasks in a task family. We take the pairwise cosine similarity for the aggregated activation vectors of each task family as a measure of similarity between their activation trajectories. In Fig. 15, we present the results in a hierarchically clustered heatmap.

Two over-arching clusters form—one cluster of mainly cardinal tasks that involve set comparisons or exact cardinality (upper left) and one of within-panel seriation and ordinal tasks (lower right). Within the second cluster, there is a subcluster of set enumeration tasks (A3, C8, A6, A4, A2, A5), object seriation tasks (D7, D8, D3, D4), and object selection tasks (D5, D2, D1, D6). Notably, although sorting objects differing in one (D3, D7) and more than

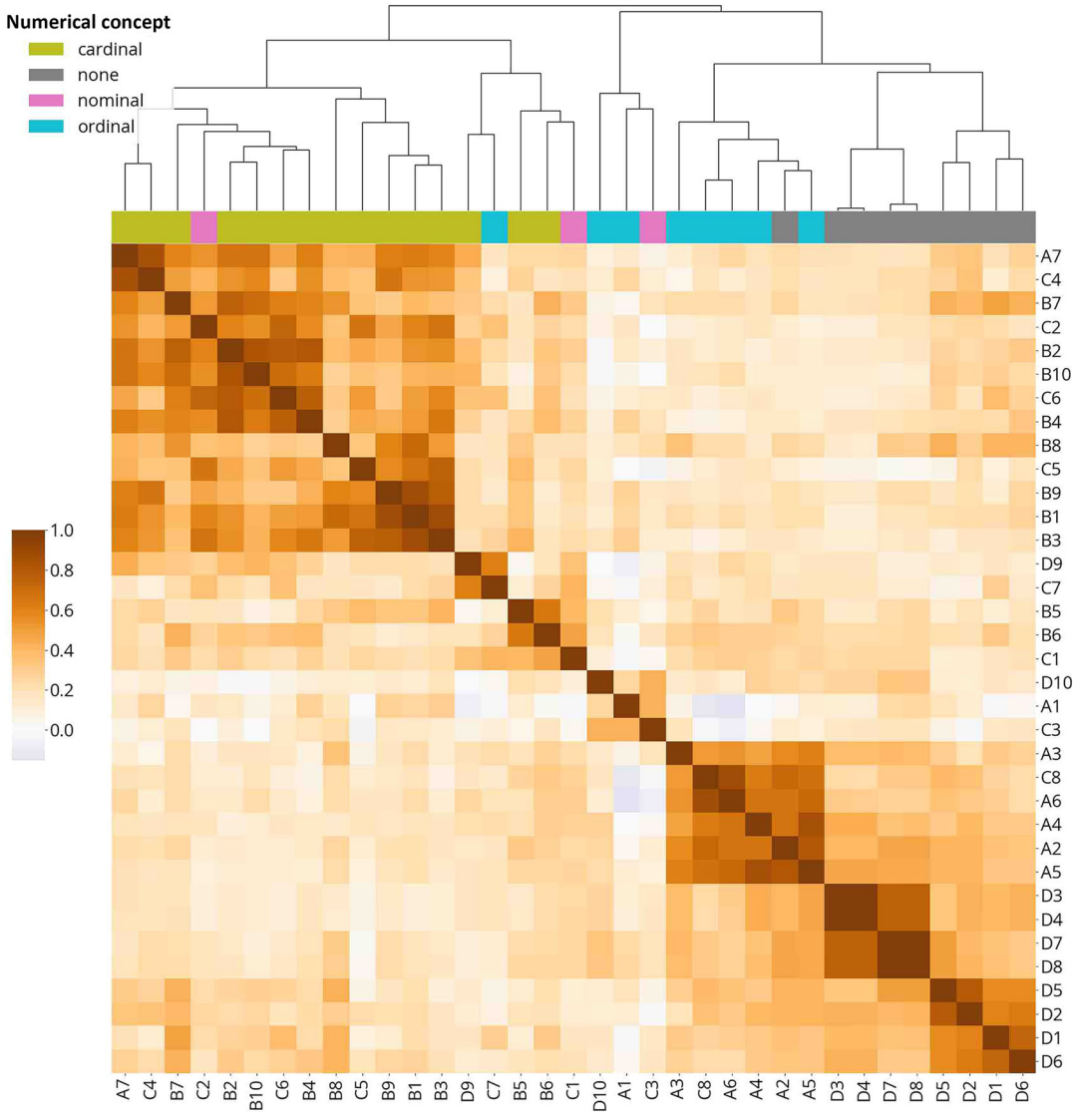


Fig. 15. Hierarchically clustered heatmap of the cosine similarity between aggregated activation trajectories for each task in the dataset.

one (D4, D8) attribute showed different training trajectories, the activation trajectories in the trained model are almost identical. The map also includes a small cluster of tasks with purely verbal outputs (D10, A1, C3) and one cluster of tasks requiring the manipulation or selection of multiple panels (D9, C7, B5, B6, C1).

Tasks that involve different number modalities but are otherwise identical show high similarity. Examples include A7 and C4, C8 and A6, or D9 and C7. This suggests that the network

has learned knowledge and procedures employed similarly in tasks involving different number formats. We investigate where the model processing diverges when solving versions of the same task with different number representations in Fig. 14. The plot compares four tasks involving the "more" relation (B9, B1, B3, and C5) broken down by attention block. Activation trajectories diverge most in the lower attention layers, then form clusters according to input representation formats: B9 and B1 involve only object sets, and C5 involves only digits. B3, which involves objects and digits, shows equal similarity to both. Activations in the fourth block are almost identical, most likely because its attention layers do not contribute much to these tasks and are essentially skipped during processing (see Section 4.3.3).

Several neuroimaging studies have done comparable analyses to investigate activations in the brain during tasks involving different number representations and magnitudes. Results indicate that neural overlap depends on task demands (Lyons & Ansari, 2015) and that, besides areas thought to represent numbers, numerical tasks activate more non-specific brain areas related to, for example, general visuospatial skills (Hubbard et al., 2005). These findings generally fit with the fact that clusters in activation trajectories in Figs. 14 and 15 in part reflect number representation format and in part similarities in other visual inputs and action sequences.

5. Conclusion

In summary, our work reinforces and amplifies previous findings that early number skills can emerge from the general learning mechanisms of DNNs. The model's training trajectories within and across tasks mostly fit with empirical findings from children, where available. This "implicit curriculum" forms without imposing an order of task presentation or explicitly modeling maturational changes, which have been hypothesized to underlie transitions in children's learning (McGonigle-Chalmers & Kusel, 2019). In line with human behavioral data, the model shows decreasing accuracy and broader response variability with increasing target numerosity and non-symbolic and symbolic size and distance effects. It produces these effects without an innate, spatially organized "mental number line," a prevalent explanation in humans (Harvey, Klein, Petridou, & Dumoulin, 2013; Zorzi, Priftis, & Umiltà, 2002).

Qualitative analysis of the model suggests an intricately entwined network of specialized and more general processing units. Using isolated probes, we show where in the model information is integrated via multimodal attention heads. We explore the interplay between attention heads in action by visualizing an exemplary information flow. The visualization illustrates how attention heads retrieve cross-modal information related to, but not necessarily present in, the model's immediate input. We compare aggregated activations across tasks and find that overlap in activation trajectories reflects similarities in inputs and task demands. This functional organization emerges from objective-based training without enforcing topological constraints on model connections. Of course, these findings do not preclude the presence of certain neural structures supporting number skills in the brain. However, they demonstrate that innate circuitry is not the only possible source of explanation.

Inspired by discussions in the literature on the role of language in numerical cognition, we train a model only on non-symbolic tasks. The model performs well on two-set comparisons and tasks related to object attributes, in line with findings that some proto-quantitative skills can develop without language. However, it performs less well on tasks involving more than two sets. We compare the internal processing and embeddings of the models trained with and without symbolic tasks. We conclude that the model trained without symbolic tasks requires more processing steps to determine set sizes, leaving fewer capacities for more advanced operations involving multiple sets. This offers a concrete, computationally implemented demonstration of how differences in exposure to symbolic number tasks can give rise to differences in internal representations and processing strategies.

Given that the model reaches high accuracy on most tasks, including comparisons requiring extrapolation to larger set sizes, it could serve as a starting point for further in silico exploration of hypotheses about the biological mind. Discrepancies between model and human behavior are particularly interesting in this regard because they provide clues about factors at play in human learning that may be missing in the setup (McClelland, 2009). For example, our model learns symbolic tasks faster than non-symbolic ones. We attributed this to symbolic tasks involving less variability and, often, shorter sequence length. A more realistic dataset where digits vary in appearance and outputting number words requires producing individual phonemes might thus lead to a more human-like acquisition order. Alternatively, introducing symbolic tasks later in training or changes to the architecture may be needed. Furthermore, future work could investigate hypotheses about the role of maturational changes in learning by gradually increasing model capacity and comparing internal representations or processing strategies to those emerging from a priori full-scale models. The model could also be ablated to simulate hypotheses about developmental disorders.

On a broader level, we hope this work can serve as an example of how DNNs can be used for cognitive modeling both despite and because of their inherent complexity. Many of the studies outlined in Section 2.1 used comparatively small models, abstracted inputs, and few specific tasks, as this was conducive to their goal of understanding model representations and processing. In deep learning, models are trained on naturalistic data and evermore general tasks. However, the focus is generally on performing well on benchmark datasets rather than analyzing the models' inner workings. While there is undoubtedly room and good reason for both approaches, we have tried to find a middle ground: We use a large, relatively general-purpose DNN but train it on the circumscribed domain of early number knowledge, then analyze it in depth. Many of our analyses reveal representations and processing strategies that could only emerge from a sufficiently complex setup. Despite this complexity, we hope we have shown that DNNs are not the entirely impenetrable black boxes they are often made out to be.

We believe that employing DNNs in smaller, controlled environments that capture essential properties of natural experience and focusing more on the "how" than the "how well" can benefit both cognitive science and AI. Cognitive scientists can use AI developments to broaden their models' scope, allowing them to analyze phenomena that cannot emerge when studying isolated concepts. For AI researchers, better insights into DNNs can yield a more realistic assessment of model capabilities and motivate improvements in architectures or input

data. For example, analyses of our model pointed to the limitations of its purely feedforward nature, underscoring the importance of recent efforts to introduce recurrent weight sharing and adaptive halting mechanisms to transformer-based architectures (Messina, Amato, Carrara, Gennaro, & Falchi, 2022; Cognolato & Testolin, 2022). As seen throughout this paper, design decisions at every level significantly impact what a model can be taught. Even two models with identical architectures and similar task performance may develop diverging internal processing mechanisms if trained on different inputs. Fields such as cognitive science and developmental psychology have long studied the experiences that shape what and how we learn. This expertise can inform the design of training inputs that induce more human-like representations and processing in DNNs, ultimately making them more understandable and, therefore, easier to trust.

Acknowledgments

Open access funding enabled and organized by Projekt DEAL.

Conflict of interest and financial disclosures

The authors have no relevant financial or non–financial interests to disclose.

References

Abrahamse, E., Braem, S., Notebaert, W., & Verguts, T. (2016). Grounding cognitive control in associative learning. *Psychological Bulletin*, *142*(7), 693–728.

Agrillo, C., Piffer, L., Bisazza, A., & Butterworth, B. (2012). Evidence for two numerical systems that are similar in humans and guppies. *PloS ONE*, *7*(2), e31923.

Ahmad, K., Casey, M., & Bale, T. (2002). Connectionist simulation of quantification skills. *Connection Science*, *14*(3), 165–201.

Anderson, J. R. (1983). *The architecture of cognition*. Mahwah, NJ: Lawrence Erlbaum.

Brainerd, C. J. (1973). Mathematical and behavioral foundations of number. *The Journal of General Psychology*, *88*(2), 221–281.

Brannon, E. M., & Terrace, H. S. (1998). Ordering of the numerosities 1 to 9 by monkeys. *Science*, *282*(5389), 746–749.

Buckley, P. B., & Gillman, C. B. (1974). Comparisons of digits and dot patterns. *Journal of Experimental Psychology*, *103*(6), 1131–1136.

Burr, D. C., Turi, M., & Anobile, G. (2010). Subitizing but not estimation of numerosity requires attentional resources. *Journal of Vision*, *10*(6), 20.

Camos, V. (2003). Counting strategies from 5 years to adulthood: Adaptation to structural features. *European Journal of Psychology of Education*, *18*, 251–265.

Carey, S. (2011). Précis of "The Origin of Concepts." *Behavioral and Brain Sciences*, *34*(3), 113–124.

Carey, S., & Sarnecka, B. W. (2006). The development of human conceptual representations: A case study. *Processes of Change in Brain and Cognitive Development: Attention and Performance XXI*, 473–496.

Chen, Q., & Verguts, T. (2010). Beyond the mental number line: A neural network model of number-space interactions. *Cognitive Psychology*, *60*(3), 218–240.

Chen, S., Zhou, Z., Fang, M., & McClelland, J. (2018). Can generic neural networks estimate numerosity like humans? In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society (CogSci) 2018, Madison, USA*, *40*, 202–207.

Cognolato, S., & Testolin, A. (2022). Transformers discover an elementary calculation system exploiting local attention and grid-like problem representation. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN) 2022, Padova, Italy* (pp. 1–8). Piscataway, NJ: IEEE.

Creatore, C., Sabathiel, S., & Solstad, T. (2021). Learning exact enumeration and approximate estimation in deep neural network models. *Cognition*, *215*, 104815.

Dadda, M., Piffer, L., Agrillo, C., & Bisazza, A. (2009). Spontaneous number representation in mosquitofish. *Cognition*, *112*(2), 343–348.

Davidson, K., Eng, K., & Barner, D. (2012). Does learning to count involve a semantic induction? *Cognition*, *123*(1), 162–173.

De La Cruz, V. M., Di Nuovo, A., Di Nuovo, S., & Cangelosi, A. (2014). Making fingers and words count in a cognitive robot. *Frontiers in Behavioral Neuroscience*, *8*, 13.

Dehaene, S. (1992). Varieties of numerical abilities. *Cognition*, *44*(1–2), 1–42.

Dehaene, S., Bossini, S., & Giraux, P. (1993). The mental representation of parity and number magnitude. *Journal of Experimental Psychology: General*, *122*(3), 371–396.

Dehaene, S., & Changeux, J.-P. (1993). Development of elementary numerical abilities: A neuronal model. *Journal of Cognitive Neuroscience*, *5*(4), 390–407.

Dehaene, S., Dehaene-Lambertz, G., & Cohen, L. (1998). Abstract representations of numbers in the animal and human brain. *Trends in Neurosciences*, *21*(8), 355–361.

Di Nuovo, A. (2017). An embodied model for handwritten digits recognition in a cognitive robot. In *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI) 2017, Honolulu, USA* (pp. 1–6). Piscataway, NJ: IEEE.

Di Nuovo, A. (2018). Long-short term memory networks for modelling embodied mathematical cognition in robots. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN) 2018*, Rio de Janeiro, Brazil (pp. 1–7). Piscataway, NJ: IEEE.

Di Nuovo, A., De La Cruz, V. M., Cangelosi, A., & Di Nuovo, S. (2014). The iCub learns numbers: An embodied cognition study. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN) 2014, Beijing, China* (pp. 692–699). Piscataway, NJ: IEEE.

Di Nuovo, A., & McClelland, J. L. (2019). Developing the knowledge of number digits in a child-like robot. *Nature Machine Intelligence*, *1*(12), 594–605.

Di Nuovo, A., Vivian, M., & Cangelosi, A. (2015). A deep learning neural network for number cognition: A bi-cultural study with the iCub. In *Proceedings of the Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob) 2015, Providence, RI* (pp. 320–325). Piscataway, NJ: IEEE.

Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *10*(2), 112–122.

Du, H., Yu, X., & Zheng, L. (2021). VTNet: Visual transformer network for object goal navigation. In *Proceedings of the 9th International Conference on Learning Representations (ICLR) 2021, Virtual Event, Austria* (pp. 1–16). Appleton WI: ICLR.

Dulberg, Z., Webb, T., & Cohen, J. (2021). Modelling the development of counting with memory-augmented neural networks. In *Proceedings of the 43rd Annual Meeting of the Cognitive Science Society (CogSci) 2021, Virtual Event* (pp. 868–874). Seattle WA: Cognitive Science Society.

Feigenson, L., Dehaene, S., & Spelke, E. (2004). Core systems of number. *Trends in Cognitive Sciences*, *8*(7), 307–314.

Finke, S., Kemény, F., Clayton, F. J., Banfi, C., Steiner, A. F., Perchtold-Stefan, C. M., Papousek, I., Göbel, S. M., & Landerl, K. (2021). Cross-format integration of auditory number words and visual-arabic digits: An ERP study. *Frontiers in Psychology*, *12*, 765709.

Fuson, K. C., Richards, J., & Briars, D. J. (1982). *The acquisition and elaboration of the number word sequence* (pp. 33–92). New York: Springer.

Gagne, R. M. (1968). Presidential address of division 15—learning hierarchies. *Educational Psychologist*, *6*(1), 1–9.

Gelman, R., & Gallistel, C. R. (1986). *The child's understanding of number*. Cambridge, MA: Harvard University Press.

Gelman, R., & Gallistel, C. R. (2004). Language and the origin of numerical concepts. *Science*, *306*(5695), 441–443.

Gevers, W., Verguts, T., Reynvoet, B., Caessens, B., & Fias, W. (2006). Numbers and space: A computational model of the SNARC effect. *Journal of Experimental Psychology: Human Perception and Performance*, *32*(1), 32.

Gilmore, C. K., McCarthy, S. E., & Spelke, E. S. (2007). Symbolic arithmetic knowledge without instruction. *Nature*, *447*(7144), 589–591.

Graves, A., Bellemare, M. G., Menick, J., Munos, R., & Kavukcuoglu, K. (2017). Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning, (ICML) 2017, Sydney, Australia*, volume 70 of *Proceedings of Machine Learning Research* (pp. 1311–1320). Cambridge, MA: PMLR.

Harvey, B. M., Klein, B. P., Petridou, N., & Dumoulin, S. O. (2013). Topographic representation of numerosity in the human parietal cortex. *Science*, *341*(6150), 1123–1126.

Hauser, M. D., Carey, S., & Hauser, L. B. (2000). Spontaneous number representation in semi–free–ranging rhesus monkeys. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *267*(1445), 829–833.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770–778). Piscataway, NJ: IEEE.

Hornburg, C. B., Schmitt, S. A., & Purpura, D. J. (2018). Relations between preschoolers' mathematical language understanding and specific numeracy skills. *Journal of Experimental Child Psychology*, *176*, 84–100.

Hubbard, E. M., Piazza, M., Pinel, P., & Dehaene, S. (2005). Interactions between number and space in parietal cortex. *Nature Reviews Neuroscience*, *6*(6), 435–448.

Hyde, D. C., & Spelke, E. S. (2009). All numbers are not equal: An electrophysiological investigation of small and large number representations. *Journal of Cognitive Neuroscience*, *21*(6), 1039–1053.

Jeske, P. J. (1978). *The effects of modeling, imitative performance, and modeling feedback on hierarchical seriation learning* [Doctoral dissertation]. The University of Arizona, Tucson, AZ.

Katz, S., & Belinkov, Y. (2023). VISIT: Visualizing and interpreting the semantic information flow of transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore* (pp. 14094–14113). Stroudsburg, PA: Association for Computational Linguistics.

Kim, G., Jang, J., Baek, S., Song, M., & Paik, S.-B. (2021). Visual number sense in untrained deep neural networks. *Science Advances*, *7*(1), eabd6127.

Kolkman, M. E., Kroesbergen, E. H., & Leseman, P. P. (2013). Early numerical development and the role of non-symbolic and symbolic skills. *Learning and Instruction*, *25*, 95–103.

Li, Y., Zhang, M., Chen, Y., Deng, Z., Zhu, X., & Yan, S. (2018). Children's non-symbolic and symbolic numerical representations and their associations with mathematical ability. *Frontiers in Psychology*, *9*, 1035.

Lingoes, J. C. (1963). Multiple scalogram analysis: A set-theoretic model for analyzing dichotomous items. *Educational and Psychological Measurement*, *23*(3), 501–524.

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, 3, 111–132. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S2666651022000146 doi: 10.1016/j.aiopen.2022.10.001

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Han, J. (2020). On the variance of the adaptive learning rate and beyond. In *Proceedings of the 8th International Conference on Learning Representations, (ICLR) 2020, Addis Ababa, Ethiopia* (pp. 1–13). Appleton WI: ICLR.

Loshchilov, I., & Hutter, F. (2017). SGDR: Stochastic gradient descent with warm restarts. In *Proceedings of the 5th International Conference on Learning Representations, (ICLR) 2017, Toulon, France*. Appleton WI: ICLR.

Lyons, I. M., & Ansari, D. (2015). Foundations of children's numerical and mathematical skills: The roles of symbolic and nonsymbolic representations of numerical magnitude. *Advances in Child Development and Behavior*, *48*, 93–116.

Lyons, I. M., Ansari, D., & Beilock, S. L. (2012). Symbolic estrangement: Evidence against a strong association between numerical symbols and the quantities they represent. *Journal of Experimental Psychology: General*, *141*(4), 635–641.

Mandler, G., & Shebo, B. J. (1982). Subitizing: An analysis of its component processes. *Journal of Experimental Psychology*, *111*(1), 1–22.

Mareschal, D., & Shultz, T. R. (1999). Development of children's seriation: A connectionist approach. *Connection Science*, *11*(2), 149–186.

Marshuetz, C., Smith, E. E., Jonides, J., DeGutis, J., & Chenevert, T. L. (2000). Order information in working memory: fMRI evidence for parietal and prefrontal mechanisms. *Journal of Cognitive Neuroscience*, *12*(Supplement 2), 130–144.

Matejko, A. A., & Ansari, D. (2016). Trajectories of symbolic and nonsymbolic magnitude processing in the first year of formal schooling. *PloS ONE*, *11*(3), e0149863.

McClelland, J. L. (2009). The place of modeling in cognitive science. *Topics in Cognitive Science*, *1*(1), 11–38.

McClelland, J. L., Mickey, K., Hansen, S., Yuan, A., & Lu, Q. (2016). A parallel-distributed processing approach to mathematical cognition [Unpublished Manuscript]. Stanford University.

McGonigle-Chalmers, M., & Kusel, I. (2019). The development of size sequencing skills: An empirical and computational analysis. *Monographs of the Society for Research in Child Development*, *84*(4), 7–202.

Messina, N., Amato, G., Carrara, F., Gennaro, C., & Falchi, F. (2022). Recurrent vision transformer for solving visual reasoning problems. In *Proceedings of the 21st International Conference in Image Analysis and Processing (ICIAP) 2022, Lecce, Italy*, Lecture Notes in Computer Science, Volume 13233 (pp. 50–61). Cham: Springer.

Mussolin, C., Nys, J., Content, A., & Leybaert, J. (2014). Symbolic number abilities predict later approximate number system acuity in preschool children. *PLoS One*, *9*(3), e91839.

Nasr, K., & Nieder, A. (2021). Spontaneous representation of numerosity zero in a deep neural network for visual object recognition. *iScience*, *24*(11), 103301.

Newman, R. S., Friedman, C. A., & Gockley, D. R. (1987). Children's use of multiple-counting skills: Adaptation to task factors. *Journal of Experimental Child Psychology*, *44*(2), 268–282.

Nieder, A. (2005). Counting on neurons: The neurobiology of numerical competence. *Nature Reviews Neuroscience*, *6*(3), 177–190.

Nieder, A., Freedman, D. J., & Miller, E. K. (2002). Representation of the quantity of visual items in the primate prefrontal cortex. *Science*, *297*(5587), 1708–1711.

Ninokura, Y., Mushiake, H., & Tanji, J. (2003). Representation of the temporal order of visual objects in the primate lateral prefrontal cortex. *Journal of Neurophysiology*, *89*(5), 2868–2873.

Ninokura, Y., Mushiake, H., & Tanji, J. (2004). Integration of temporal order and object information in the monkey lateral prefrontal cortex. *Journal of Neurophysiology*, *91*(1), 555–560.

nostalgebraist. (2020). Interpreting GPT: The logit lens.

Paaß, G., & Giesselbach, S. (2023). Foundation models for speech, images, videos, and control. In *Foundation models for natural language processing: Pre-trained language models integrating media* (pp. 313–382). Cham: Springer.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 32nd Annual Conference on Advances in Neural Information Processing Systems (NeurIPS) 2019, Vancouver, Canada*, (pp. 8024–8035). Red Hook, NY: Curran Associates.

Peterson, S. A., & Simon, T. J. (2000). Computational evidence for the subitizing phenomenon as an emergent property of the human cognitive architecture. *Cognitive Science*, *24*(1), 93–122.

Piaget, J. (1961). The genetic approach to the psychology of thought. *Journal of Educational Psychology*, *52*(6), 275–281.

Piaget, J., Gattegno, C., & Hodgson, F. (1952). *The child's conception of number*. London, England: Routledge & Kegan Paul Ltd.

Piantadosi, S. T. (2016). A rational analysis of the approximate number system. *Psychonomic Bulletin & Review*, *23*, 877–886.

Piantadosi, S. T., Tenenbaum, J. B., & Goodman, N. D. (2012). Bootstrapping in a language of thought: A formal model of numerical concept learning. *Cognition*, *123*(2), 199–217.

Piazza, M., Mechelli, A., Butterworth, B., & Price, C. J. (2002). Are subitizing and counting implemented as separate or functionally overlapping processes? *Neuroimage*, *15*(2), 435–446.

Pica, P., Lemer, C., Izard, V., & Dehaene, S. (2004). Exact and approximate arithmetic in an Amazonian indigene group. *Science*, *306*(5695), 499–503.

Pitt, B., Ferrigno, S., Cantlon, J. F., Casasanto, D., Gibson, E., & Piantadosi, S. T. (2021). Spatial concepts of number, size, and time in an indigenous culture. *Science Advances*, *7*(33), eabg4141.

Pitt, B., Gibson, E., & Piantadosi, S. T. (2022). Exact number concepts are limited to the verbal count range. *Psychological Science*, *33*(3), 371–381.

Potter, M. C., & Levy, E. I. (1968). Spatial enumeration without counting. *Child Development*, *39*(1), 265–272.

Powell, S. R., & Fuchs, L. S. (2012). Early numerical competencies and students with mathematics difficulty. *Focus on Exceptional Children*, *44*(5), 1–16.

Purpura, D. J., & Reid, E. E. (2016). Mathematics and language: Individual and group differences in mathematical language skills in young children. *Early Childhood Research Quarterly*, *36*, 259–268.

Resnick, L. B. (1973). Hierarchies in children's learning: A symposium. *Instructional Science*, *2*(3), 311–361.

Resnick, L. B., Wang, M. C., & Kaplan, J. (1973). Task analysis in curriculum design: A hierarchically sequenced introductory mathematics curriculum. *Journal of Applied Behavior Analysis*, *6*(4), 679–709.

Revkin, S. K., Piazza, M., Izard, V., Cohen, L., & Dehaene, S. (2008). Does subitizing reflect numerical estimation? *Psychological Science*, *19*(6), 607–614.

Rucinski, M., Cangelosi, A., & Belpaeme, T. (2011). An embodied developmental robotic model of interactions between numbers and space. In *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society (CogSci) 2011, Boston, USA* (pp. 237–242). Seattle WA: Cognitive Science Society.

Rucinski, M., Cangelosi, A., & Belpaeme, T. (2012). Robotic model of the contribution of gesture to learning to count. In *Proceedings of the IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob) 2012, San Diego, USA* (pp. 1–6). Piscataway, NJ: IEEE.

Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 379–389). Stroudsburg, PA: Association for Computational Linguistics.

Sabathiel, S., McClelland, J., & Solstad, T. (2020a). A computational model of learning to count in a multimodal, interactive environment. In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society (CogSci) 2020, Virtual Event* (pp. 1425–1431). Seattle WA: Cognitive Science Society.

Sabathiel, S., McClelland, J. L., & Solstad, T. (2020b). Emerging representations for counting in a neural network agent interacting with a multimodal environment. In *Proceedings of the Conference on Artificial Life (ALife) 2020, Montréal, Canada* (pp. 736–743). Cambridge MA: MIT Press.

Sarnecka, B. W., & Carey, S. (2008). How counting represents number: What children must learn and when they learn it. *Cognition*, *108*(3), 662–674.

Schaeffer, B., Eggleston, V. H., & Scott, J. L. (1974). Number development in young children. *Cognitive Psychology*, *6*(3), 357–379.

Schneider, M., Merz, S., Stricker, J., De Smedt, B., Torbeyns, J., Verschaffel, L., & Luwel, K. (2018). Associations of number line estimation with mathematical competence: A meta-analysis. *Child Development*, *89*(5), 1467–1484.

Shannon, L. (1978). Spatial strategies in the counting of young children. *Child Development*, *49*(4), 1212.

Sheahan, H., Luyckx, F., Nelli, S., Teupe, C., & Summerfield, C. (2021). Neural state space alignment for magnitude generalization in humans and recurrent networks. *Neuron*, *109*(7), 1214–1226.

Siegel, L. S. (1971). The development of the understanding of certain number concepts. *Developmental Psychology*, *5*(2), 362–363.

Starkey, G. S., & McCandliss, B. D. (2014). The emergence of "groupitizing" in children's numerical cognition. *Journal of Experimental Child Psychology*, *126*, 120–137.

Stoianov, I., & Zorzi, M. (2012). Emergence of a 'visual number sense' in hierarchical generative models. *Nature Neuroscience*, *15*(2), 194–196.

Testolin, A., Zou, W. Y., & McClelland, J. L. (2020). Numerosity discrimination in deep neural networks: Initial competence, developmental refinement and experience statistics. *Developmental Science*, *23*(5), e12940.

Toll, S. W., & Van Luit, J. E. (2014). The developmental relationship between language and low early numeracy skills throughout kindergarten. *Exceptional Children*, *81*(1), 64–78.

Tomic, W., & Kingma, J. (1997). The relationship between seriation and number line comprehension: A validation study. *Curriculum and Teaching*, *12*(2), 59–69.

Toomarian, E. Y., & Hubbard, E. M. (2018). On the genesis of spatial-numerical associations: Evolutionary and cultural factors co-construct the mental number line. *Neuroscience and Biobehavioral Reviews*, *90*, 184–199.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 30th Annual Conference on Advances in Neural Information Processing Systems (NeurIPS) 2017, Long Beach, USA*, (pp. 5998–6008).

Verguts, T., & Fias, W. (2004). Representation of number in animals and humans: A neural model. *Journal of Cognitive Neuroscience*, *16*(9), 1493–1504.

Verguts, T., Fias, W., & Stevens, M. (2005). A model of exact small-number representation. *Psychonomic Bulletin & Review*, *12*, 66–80.

Wagner, K., Chu, J., & Barner, D. (2019). Do children's number words begin noisy? *Developmental Science*, *22*(1), e12752.

Wang, M. C. (1973). Psychometric studies in the validation of an early learning curriculum. *Child Development*, *44*(1), 54–60.

Wang, M. C., Resnick, L. B., & Boozer, R. F. (1971). The sequence of development of some early mathematics behaviors. *Child Development*, *42*(6), 1767.

Wang, Y., Huang, H., Rudin, C., & Shaposhnik, Y. (2021). Understanding how dimension reduction tools work: An empirical approach to deciphering t-SNE, UMAP, TriMAP, and PaCMAP for data visualization. *The Journal of Machine Learning Research*, *22*(1), 9129–9201.

Wellman, H. M., Fabricius, W. V., & Chuan-Wen, W. (1987). Considering every available instance: The early development of a fundamental problem solving skill. *International Journal of Behavioral Development*, *10*(4), 485–500.

Wu, X., Dyer, E., & Neyshabur, B. (2021). When do curricula work? In *Proceedings of the 9th International Conference on Learning Representations (ICLR) 2021, Virtual Event* (pp. 1–23). Appleton WI: ICLR.

Wynn, K. (1992). Children's acquisition of the number words and the counting system. *Cognitive Psychology*, *24*(2), 220–251.

Xu, F. (2003). Numerosity discrimination in infants: Evidence for two systems of representations. *Cognition*, *89*(1), B15–B25.

Zhang, X. (2016). Linking language, visual-spatial, and executive function skills to number competence in very young Chinese children. *Early Childhood Research Quarterly*, *36*, 178–189.

Zorzi, M., Priftis, K., & Umiltà, C. (2002). Neglect disrupts the mental number line. *Nature*, *417*(6885), 138–139.