Review

The relational bottleneck as an inductive bias for efficient abstraction

Taylor W. Webb [1,*], Steven M. Frankland[2], Awni Altabaa[3], Simon Segert[4], Kamesh Krishnamurthy[4], Declan Campbell[4], Jacob Russin[5], Tyler Giallanza[4], Randall O'Reilly[6], John Lafferty[3], and Jonathan D. Cohen[4]

A central challenge for cognitive science is to explain how abstract concepts are acquired from limited experience. This has often been framed in terms of a dichotomy between connectionist and symbolic cognitive models. Here, we highlight a recently emerging line of work that suggests a novel reconciliation of these approaches, by exploiting an inductive bias that we term the relational bottleneck. In that approach, neural networks are constrained via their architecture to focus on relations between perceptual inputs, rather than the attributes of individual inputs. We review a family of models that employ this approach to induce abstractions in a data-efficient manner, emphasizing their potential as candidate models for the acquisition of abstract concepts in the human mind and brain.

Modeling the efficient induction of abstractions

Human cognition displays a remarkable ability to transcend the specifics of limited experience to entertain highly general, abstract ideas. Understanding how the mind and brain accomplish this has been a central challenge of cognitive science and a major preoccupation of philosophy before that [1–4]. Of particular importance is the central role played by relations, which enable human reasoners to abstract away from individual entities and identify higher-order patterns across distinct domains [5,6]. For instance, when presented with the images in Figure 1, one can easily determine that a common relational pattern (ABA) is displayed on both the left and right, despite the involvement of completely different objects. This capacity is a major component underlying the human capacity for fluid reasoning [7,8] and has been proposed as a key factor distinguishing human intelligence from that of other species [9].

A long tradition in cognitive science and artificial intelligence (AI) [10–12] holds that this capacity for abstraction depends on processes akin to symbolic programs. A major appeal of this approach is that symbols are, by design, abstracted away from the content to which they refer, thus naturally accounting for the flexibility and systematicity of human concepts [13]. More recently, **program induction** (see Glossary) has provided an account of how symbolic concepts might be learned directly from data [14–20], formalizing learning as a search for the program that maximizes the likelihood of observed data. However, while this approach is capable in principle of representing any possible set of concepts [21], the discovery of these concepts using traditional search methods often proves intractable, making it difficult in practice to identify programs with the richness and complexity of human natural concepts (though see Box 1 for discussion).

An alternative approach, **connectionism**, has for decades explored how cognitive abstractions might emerge through experience in general-purpose neural architectures [22–25]. This endeavor has taken on new relevance with the advent of large language models, demonstrating that it is possible, in some cases, for a human-like capacity for abstraction to emerge given sufficient scaling of architecture and training data [26–29]. For instance, it has recently been shown that large language

[1]University of California, Los Angeles, CA, USA
[2]Dartmouth College, Hanover, NH, USA
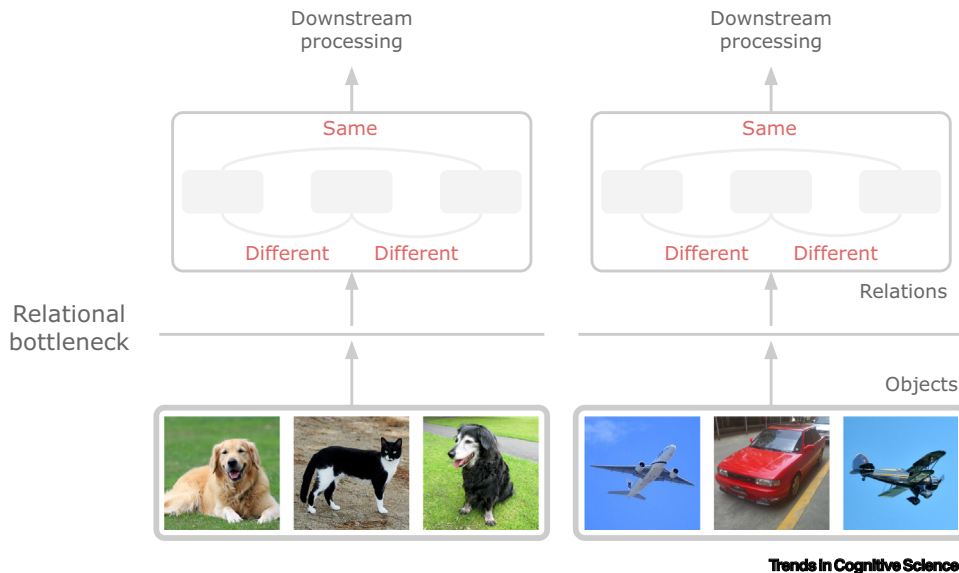[3]Yale University, New Haven, CT, USA
[4]Princeton University, Princeton, NJ, USA
[5]Brown University, Providence, RI, USA
[6]University of California, Davis, CA, USA

*Correspondence:
taylor.w.webb@gmail.com (T.W. Webb).

Figure 1. The relational bottleneck. An inductive bias that prioritizes the representation of relations (e.g., 'same' versus 'different') and discourages the representation of the features of individual objects (e.g., the shape or color of the objects in the images). The result is that downstream pr ocessing is driven primarily, or even exclusively, by patterns of relations and can therefore systematically generalize those patterns across distinct instances (e.g., the common ABA pattern displayed on both left and right), even for completely novel objects. The approach is illustrated here with same/different relations, but other relations can also be accommodated. Note that this example is intended only to illustrate the overall goal of the relational bottleneck framework. Figure 2 in the main text depicts neural architectures that implement the approach.

models can solve various analogy problems at a level equal to that of college students [30]. However, the ability of these models to perform abstract tasks depends on exposure to a much larger training corpus than individual humans receive in an entire lifetime [31,32], thus failing to account for the data efficiency of human concept learning.

Box 1. Neuro-symbolic modeling approaches

Many approaches have been proposed for hybrid systems that combine aspects of both neural and symbolic computing. Early work in this area focused on incorporating a capacity for variable-binding (a key property of symbolic systems) into connectionist systems. Notable examples of this approach include binding-by-synchrony [35], tensor product variable-binding [34], and BoltzCONS [89]. A number of vector symbolic architectures have since been proposed that build on the tensor product operation, but enable more elaborate symbolic structures to be embedded in a vector space of fixed dimensionality [90–93]. These approaches have all generally relied on the use of pre-specified symbolic primitives.

More recently, hybrid systems have been developed that combine deep learning with symbolic programs. In this approach, deep learning components are typically employed to translate raw perceptual inputs, such as images or natural language, into symbolic representations, which can then be processed by traditional symbolic algorithms [94–96]. This approach is complemented by recent neuro-symbolic approaches to probabilistic program induction, in which symbolic primitives are pre-specified (following earlier symbolic-connectionist modeling efforts) and then deep learning is used to assemble these primitives into programs [17].

An alternative approach (which might also be viewed as neuro-symbolic in some sense) involves the integration of key features of symbolic computing within the framework of end-to-end trainable neural systems. Examples of this approach include neural production systems [97], graph neural networks [46], discrete-valued neural networks [98] (see [99] for further discussion of normative considerations regarding discrete-valued representations), sparse causal graphs [100], and efforts to incorporate tensor product representations into end-to-end systems [101,102]. The relational bottleneck falls into this broad category, as it incorporates key elements of symbolic computing (variable-binding and relational representations) into fully differentiable neural systems that can be trained end-to-end without the need for pre-specified symbolic primitives. Relative to these other approaches, the primary innovation of the relational bottleneck framework is the emphasis on architectural components that promote the development of genuinely relational representations.

Glossary

Attention: in the context of deep learning, an operation that allows information to be flexibly shared between a set of embeddings [128] (popularized by the Transformer architecture [50]), also closely related to content-addressable memory (see external memory). This operation shares some, but not all, properties of formal treatments of attention in cognitive psychology (see [129] for an example of how the psychological notion of attention may be implemented in neural networks; see [130] for discussion of the relationship between these different senses of 'attention').

Backpropagation: a technique used to train multilayer neural networks, in which the connection strengths between processing units in intermediate layers are updated based on an error signal in downstream layers, allowing intermediate representations to be automatically learned in a manner that is most useful for downstream tasks (without having to specify those intermediate representations).

Connectionism: a modeling framework in cognitive science that emphasizes the emergence of complex cognitive phenomena from the interaction of simple, neuron-like elements organized into networks, in which connections are formed through learning.

Embedding: a real-valued vector that represents a particular input (e.g., a word or image), often instantiated as the state (set of activation values) of a particular layer in a neural network.

Episodic memory (EM): a form of memory in which arbitrary, but durable, associations can be rapidly formed. Often thought to be implemented by hippocampal mechanisms for rapid synaptic plasticity and similarity-based retrieval.

External memory: in the context of deep learning, an approach that combines neural networks with separate external stores of information, typically with learnable mechanisms for writing to and reading from these stores and in which retrieval is usually similarity-based (i.e., 'content-addressable'). Often used to implement a form of episodic memory [131].

Inductive bias: an assumption made by a machine learning model about the distribution of the data. In deep learning models, this often takes the form of architectural features that bias learning

In this review, we highlight an emerging approach that suggests a novel reconciliation of these two traditions. The central feature of this approach is an **inductive bias** that we refer to as the relational bottleneck: a constraint that biases neural network models to focus on relations between objects rather than the attributes of individual objects. This approach enables the data efficiency associated with symbolic cognitive models, while retaining the scalable training procedures associated with neural network models (see Box 1 for further discussion of neuro-symbolic approaches). In the following sections, we first provide a general characterization of this approach, drawing on information theory, and discuss recently proposed neural network architectures that implement the approach. We then discuss the potential of the approach for modeling human cognition, relating it to existing theories and considering mechanisms through which it might be implemented in the brain.

The relational bottleneck

We define the relational bottleneck as any mechanism that restricts the flow of information from perceptual to downstream reasoning systems to consist only of relations (see Box 2 for a formal definition). For example, consider the relational patterns depicted in Figure 1. In this example, the images on the left and right are governed by the same abstract pattern (ABA), but these patterns contain different sets of objects (dogs and cats versus planes and cars). Given inputs representing individual objects (e.g., representations of the dog and cat images on the left), a relational bottleneck would constrain the representations passed to downstream reasoning processes to capture only the relations between these objects (e.g., whether the objects have the same shape) and discard information about the individual objects (e.g., information about dogs and cats). The result is that the images on the left will have the same representation as those on the right, despite the different objects depicted in these images. This encourages downstream processes to identify relational patterns in a manner that is abstracted away from specific instances of those patterns and can therefore be systematically generalized to novel inputs (see Concluding remarks for discussion of non-relational factors in human cognition). In the following section, we highlight three recently proposed neural architectures that instantiate this approach in different guises, illustrating how they utilize a relational bottleneck to induce abstract concepts in a data-efficient manner. It is also worth emphasizing that, although we focus here on artificial neural networks, the approach may also be applicable to other modeling approaches (see Outstanding questions).

The relational bottleneck in neural architectures

Figure 2, Key figure depicts three architectures that implement the relational bottleneck through architectural inductive biases. Here, we discuss how the distinct mechanisms in these models implement the same underlying principle. In particular, a common element is the use of **inner products** to represent relations, which ensures that the resulting representations are genuinely relational. In each case, we also contrast these architectures with related approaches that do not incorporate a relational bottleneck, emphasizing how this key architectural feature enables the data-efficient induction of abstractions.

Emergent symbol binding

We first consider the emergent symbol binding network (ESBN) (Figure 2A) [33], a deep neural network architecture inspired by the notion of role-filler variable binding in cognitive models of relational reasoning [34–36]. In those models, relational reasoning is supported by the existence of separate 'roles', representing information about abstract variables, and 'fillers', representing information about concrete entities bound to those variables. This coding scheme enables roles and fillers to be flexibly combined in new ways, capturing a key property of symbol-processing systems: the ability of symbolic variables to be associated with any potential values. Previous work focused on how binding might be performed in neural circuits (e.g., units coding

toward certain (typically desirable) outcomes. Genetically pre-configured aspects of brain structure can be viewed as a form of inductive bias.
Inner product: an operation in which two vectors are converted to a scalar, often interpreted as representing the similarity of those two vectors.
Out-of-distribution generalization: in machine learning, generalization to a distribution that differs from the distribution observed during training.
Program induction: a modeling approach in which concepts are represented as symbolic programs and learned via a search process that seeks to maximize the likelihood of observed data (typically subject to some constraints, e.g., parsimony).

Box 2. The relational bottleneck principle

Information bottleneck theory [103] provides a normative framework for formalizing the notion of a relational bottleneck. Consider an information processing system that receives an input signal X and aims to predict a target signal Y. X is processed to generate a compressed representation $Z = f(X)$ (the 'bottleneck'), which is then used to predict Y. At the heart of information bottleneck theory is the idea of 'minimal-sufficiency'. Z is sufficient for predicting Y if it contains all the information X encodes about Y. That is, $I(Z; Y) = I(X; Y)$, where $I(\cdot; \cdot)$ is the mutual information. If Z is sufficient, then we write $X \rightarrow Z \rightarrow Y$, meaning that Y is conditionally independent of X given the compressed representation Z. Z is minimal-sufficient if it is sufficient for Y and does not contain any extraneous information about X that is not relevant to predicting Y. That is, $I(X; Z) \leq I(X; \tilde{Z})$ for any other sufficient compressed representation $\tilde{Z}$.

Achieving maximum compression while retaining as much relevant information as possible is a tradeoff. It is captured by the information bottleneck objective,

$$\text{minimize } \mathcal{L}(Z) = I(X; Z) - \beta I(Z; Y). \tag{I}$$

This objective reflects the tension between compression, which favors discarding information as captured by the first term, and the preservation of relevant information in Z, captured by the second term. The parameter β controls this tradeoff.

While this objective is well-defined when the joint distribution (X, Y) is known, obtaining a minimal-sufficient compressed representation from data is, in general, very challenging for the high-dimensional signals that are often of interest. However, it may be possible to implicitly enforce a desirable information bottleneck for a large class of tasks through architectural inductive biases.

In particular, we hypothesize that human cognition has been fundamentally optimized for tasks that are relational in nature. We define a 'relational task' as any task for which there exists a minimal-sufficient representation R that is purely relational. Suppose the input signal represents a set of objects,

$$X = (x_1, \ldots, x_N). \tag{II}$$

A relational signal is a signal of the form,

$$R = \{r(x_i, x_j)\}_{i \neq j} = \{r(x_1, x_2), r(x_1, x_3), \ldots, r(x_{N-1}, x_N)\}, \tag{III}$$

where $r(x_i, x_j)$ is a relation function. A 'relation function' is a function that takes a pair of objects as input and returns a relation between them (see 'Modeling more complex relations' in the main text for discussion of higher-order relations). We say that a task is relational if there exists some relational signal R that is sufficient for predicting the target Y (i.e., $X \rightarrow R \rightarrow Y$). A relational bottleneck is any mechanism that restricts or biases the learned compressed representation of the input to be a relational representation, separated from object-level features. This gives the model a smaller space of possible compressed representations over which it must search. Moreover, this restricted space is guaranteed to contain a sufficient representation for the task and excludes many representations that encode extraneous information about X, promoting efficient learning of relational abstractions.

In practice, a relational bottleneck can be implemented in a learning model through the use of architectural inductive biases. One particularly useful operation is inner products of the form $\langle \phi(x_i), \phi(x_j) \rangle$, which naturally capture a notion of relations in terms of similarity between learned attributes. Inner products can also capture asymmetric relations through the use of separate encoders for x_i and x_j [i.e., $\langle \phi(x_i), \psi(x_j) \rangle$] and indeed can be shown to be universal approximators [104]. The class of functions that can be represented in this way is thus fully general, but the approach provides a useful inductive bias for disentangling feature extraction (implemented by the encoders ϕ and ψ from comparison (implemented by the inner product operator).

for a particular role may be temporarily 'bound' to units coding for a particular filler by firing synchronously [35]). However, role and filler representations were typically pre-specified by the modeler, leaving open the question of how such symbolic representations might be learned.

The ESBN adopts this key idea of separate roles and fillers, but integrates them into a system that can be trained end-to-end (via **backpropagation**, averting the need to pre-specify these representations. The ESBN contains three major components: (i) a feedforward encoding pathway ('Encoder' in Figure 2A), which generates object **embeddings** (i.e., fillers) from perceptual inputs

Key figure

Implementing the relational bottleneck
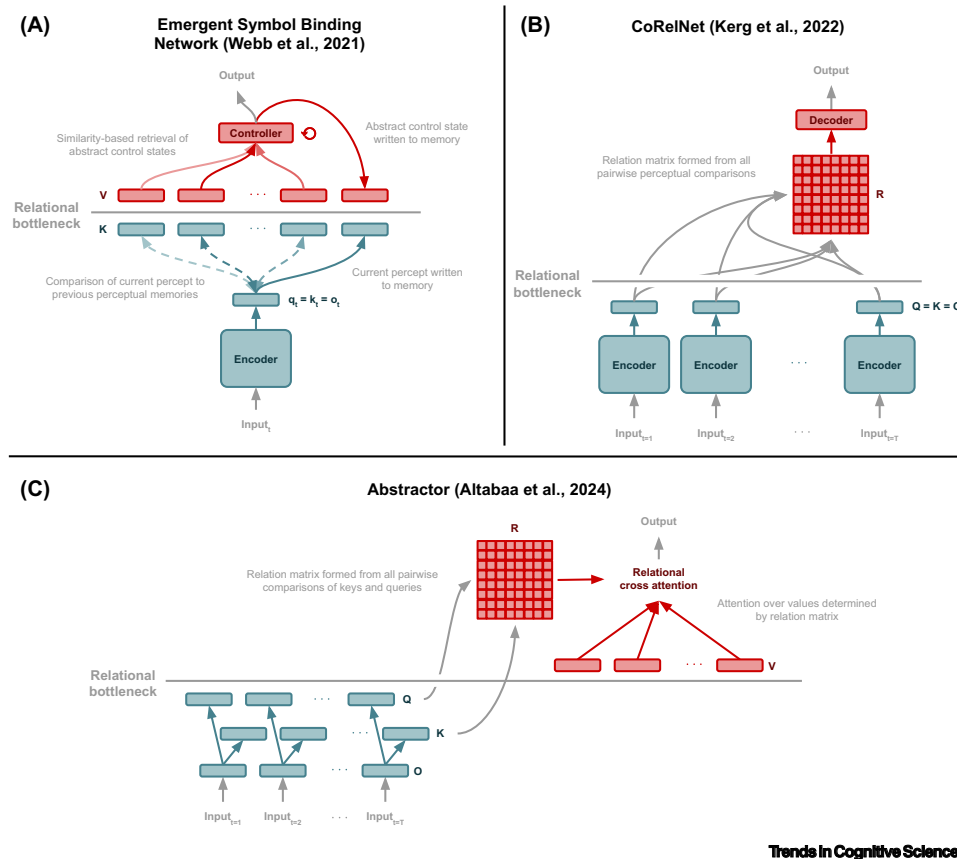
Figure 2. Three neural architectures that implement the relational bottleneck. (A) Emergent symbol binding network (ESBN) [33]. (B) Compositional relation network (CoRelNet) [43]. (C) Abstractor [49]. In all cases, high-dimensional inputs (e.g., images) are processed by a neural encoder (e.g., a convolutional network), yielding a set of object embeddings O. These are projected to a set of keys K and queries Q, which are then compared yielding a relation matrix R, in which each entry is an inner product between a query and key. Abstract values V are isolated from perceptual inputs (the core feature of the relational bottleneck) and depend only on the relations between them.

(e.g., images); (ii) a recurrent controller ('Controller' in Figure 2A), which operates over learned representations of abstract variables (i.e., roles; though the system is not explicitly trained to represent any particular variables, but instead learns these representations through backpropagation); and (iii) an **external memory** system responsible for binding representations of roles and fillers (i.e., abstract variables and perceptual embeddings). The ESBN processes inputs sequentially. For each observation, a pair of representations is appended to memory, one from the perceptual pathway (referred to as a key) and one from the control pathway [referred to as a value; note that the use of the terms 'key' and 'value' here is reversed relative to the original paper [33] in order to be consistent with their usage in describing the compositional relation network (CoRelNet) and Abstractor architecture]. To read from this memory, the embedding for the current observation (referred to as a query) is compared with all keys in memory via an inner product, yielding a set of scores (one for each key) that indicate the similarity of the current object embedding (i.e., filler) to

each entry in memory. These scores are then used to compute a weighted average of the value embeddings (i.e., roles) in the abstract pathway, which is then retrieved and passed to the controller.

Importantly, in this retrieval operation, the control pathway cannot access the content of the representations in the perceptual pathway (i.e., the fillers). Instead, the interaction is mediated only by the comparison of perceptual representations with each other. In other words, to the extent that a particular perceptual memory embedding (key) is similar to the current perceptual embedding (filler, used as a query), the corresponding control memory (value) will be retrieved and passed to the controller to be used as the role; critically, the perceptual embedding itself is not passed to the controller. The ESBN thus implements the relational bottleneck as an architectural prior, separating the learning and use of abstract representations (roles) by the controller from the embeddings of perceptual information (fillers). It is precisely this separation that guarantees the representations in the control pathway are abstract. This feature enables the ESBN to rapidly learn relational patterns (such as the identity rules displayed in Figure 1) and generalize them to **out-of-distribution** inputs (e.g., previously unseen shapes; Figure 3A) [33]. Indeed, the ESBN can learn relational patterns from only a handful of examples (as few as four examples), mirroring the data efficiency of relational learning in young children (who can typically learn to perform relational tasks with fewer than ~20 examples [37], as opposed to the thousands of examples that are necessary for standard neural network architectures). Critically, the ESBN can be shown to use precisely the same representation for a given role, irrespective of filler, thus exhibiting a critical feature of abstract, symbolic processing [13]. In this sense, the representations in the model's control pathway can be viewed as a form of learned 'symbols'.

It is instructive to compare this model with similar approaches without a relational bottleneck. The ESBN is part of a broader family of architectures that use content-addressable external memory: a separate store of information with which a neural network can interact via learnable read and write operations [38,39]. Notably, these read and write operations rely on a similarity computation (based on inner products). These have often been cast as simplified implementations of the brain's **episodic memory (EM)** system [40,41]. Standard external memory architectures do not typically isolate the control and perceptual pathways. Instead, perceptual inputs are passed directly to a central controller, which is responsible for writing to and reading from a single, monolithic memory. Though it is possible for a role-filler structure to emerge in these systems given a sufficiently large amount of training data [42], they take much longer to learn relational tasks (requiring approximately an order of magnitude more training data) and do not generalize as well as the ESBN [33]. Thus, although external memory plays an important role in the ESBN, the presence of external memory alone is insufficient to enforce a relational bottleneck. Rather, it is the isolation of the perceptual and abstract processing components from one another that does so. Furthermore, as we illustrate in the following sections, it is possible to achieve this isolation without the use of external memory.

Relation matrices

An alternative approach to implementing the relational bottleneck is illustrated by CoRelNet (Figure 2B) [43]. In that approach, a set of perceptual observations are first processed by an encoder, yielding a sequence of object embeddings. A relation matrix is then computed over all pairs of objects (within a given problem instance, e.g., all pairs of objects found within the ABA pattern in Figure 1), in which each entry consists of the inner product between a pair of object embeddings (thus capturing the similarity between each pair of objects). Finally, this relation matrix is passed to a downstream decoder network (the architecture of this network can vary, e.g., using a multilayer perceptron or Transformer). The relation matrix thus forms a relational bottleneck: all

(A)

(B)

(C)
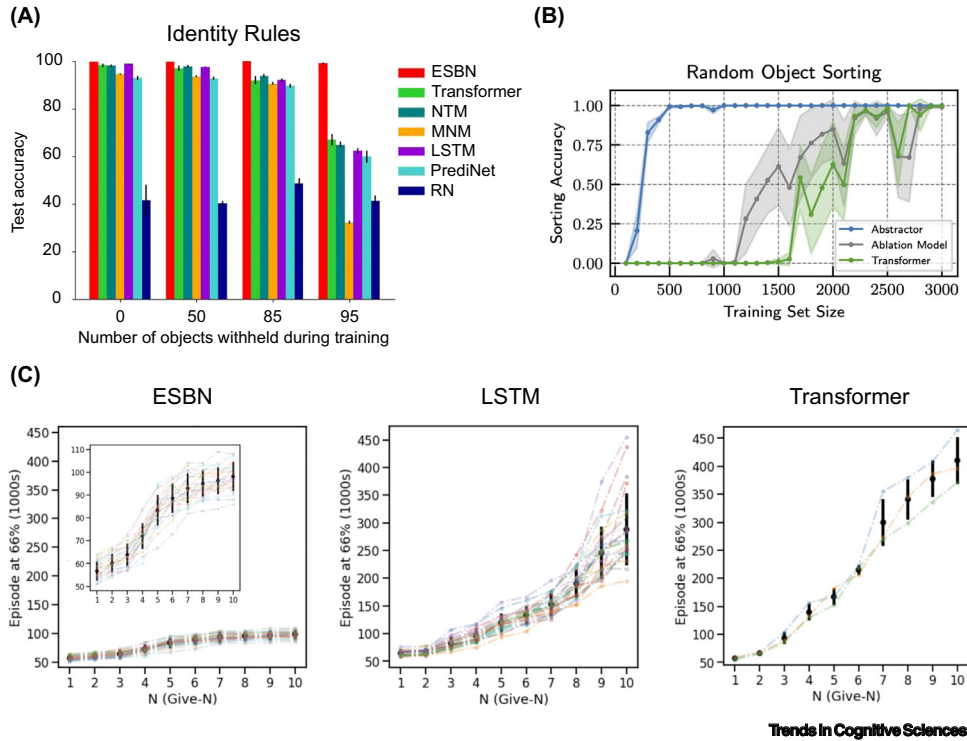
Figure 3. The relational bottleneck encourages data-efficient and generalizable relation learning. (A) Results for the emergent symbol binding network (ESBN) and baseline architectures [Transformer, neural Turing machine (NTM), metalearned neural memory (MNM), long short-term memory (LSTM), PrediNet, and the relation network (RN)] on the identity rules task, reproduced from [33]. The *x* axis represents the number of potential objects (out of 100 possible objects) withheld during training. When all objects are observed during training (0 withheld), most baselines perform well on the task. When most objects are withheld (95 withheld; test set includes only objects withheld during training), only the ESBN generalizes well to new objects. (B) Results for an object sorting task involving an asymmetric relation (greater-than/less-than), reproduced from [49]. The Abstractor learns this task significantly faster than both the Transformer and an ablation model in which relational cross-attention is replaced by standard cross-attention. (C) Results from the give-*n* task, reproduced from [63]. The *x* axis represents the target number *n* (desired number of objects). The *y* axis represents the episode at which the model reaches a particular criterion for the ability to count to each value of *N*. ESBN learns the task significantly faster than the LSTM or Transformer baselines. ESBN also displays inductive transition (rapid learning for $N > 5$) similar to that observed in human development.

perceptual information flows into this matrix, converting it into a form that only preserves relational information (about the similarity between objects) and only this relational information is then passed on to the decoder. As with the ESBN, this relational bottleneck enables CoRelNet to rapidly learn and systematically generalize relational patterns.

CoRelNet can be viewed as a feedforward, parallelized implementation of the sequential process (of encoding and similarity-based retrieval from external memory) carried out by the ESBN. This results in performance benefits, as CoRelNet does not suffer from the vanishing gradient problem that is a challenge for recurrent neural networks used to implement sequential processing [44]. That is, the gradients associated with all relations in a given scene are processed in parallel via a single step of backpropagation, rather than being diluted over several iterations, as is the case for the sequential processing carried out by the ESBN. This approach also makes the key relational inductive bias underlying the ESBN more explicit. The ESBN's memory retrieval procedure, in which the current observation is compared with the entries in memory, can be interpreted as computing a single row of the relation matrix. In both architectures, downstream processing is

constrained so as to depend only on this relation matrix (which forms the relational bottleneck), though the details of this dependency differ.

Here too, a useful contrast can be made with related architectures that do not incorporate a relational bottleneck. In particular, architectures such as the Relation Net [45] (see [46] for related approaches) explicitly perform a comparison between each pair of inputs, leading to improved performance in relational tasks. However, whereas CoRelNet represents pairwise relations using inner products, the Relation Net utilizes generic neural network components (e.g., multilayer perceptrons) that are learned in a task-dependent manner. While this is in principle more flexible, it does not constrain the network to learn representations that only capture relational information. As a consequence, this architecture is susceptible to learning shortcuts consistent with the training data (i.e., overfitting to perceptual details), compromising its ability to efficiently learn and reliably generalize relations to out-of-distribution inputs [33,47,48]. This is in contrast to the inner product operation employed by the ESBN and CoRelNet, which is inherently relational and, therefore, guarantees that downstream processing is based only on relations.

Relational attention

The recently proposed Abstractor architecture (Figure 2C) [49] illustrates how the relational bottleneck can be implemented within the broader framework of **attention-based** architectures (including the Transformer [50]). The Abstractor is built on a novel attention operation termed relational cross-attention. In this operation, a set of object embeddings (which may be produced by an encoder given perceptual observations) is converted to form keys and queries, using separate linear projections. A relation matrix is then computed, in which each entry corresponds to the inner product between a query and key. The relation matrix is used to attend over a set of learned values. The attention operation itself is identical to standard self-attention, in which each value embedding is replaced with a weighted average of the set of all value embeddings, where the weights are determined by the match between queries and keys (i.e., the relation matrix). However, the value embeddings used for relational cross-attention are formed from a separate set of representations that are learned through backpropagation (based on the downstream task for which the network is trained) which reference objects but are independent of their attributes (in the simplest scheme, these values reference objects via their position, though more sophisticated referencing schemes are also possible [49]). This is in contrast to standard self-attention, in which the values are computed via a linear projection of the perceptual inputs (just as the queries and keys are computed). As with the CoRelNet architecture, the relation matrix thus forms a relational bottleneck, in the sense that it converts all perceptual information to relational information, and downstream processing (in this case, the relational cross-attention operation) then depends only on this relational information.

Relational cross-attention can be contrasted with the standard forms of attention employed in Transformers: self-attention and cross-attention. In self-attention, the same set of object embeddings are used to generate keys, queries, and values. In cross-attention, object embeddings are used to generate keys and values and queries are generated by a separate decoder network. In both cases, the values over which attention is performed are based directly on the object embeddings and the information contained in these embeddings is, therefore, passed on for downstream processing (thus contravening the relational bottleneck). By contrast, in relational cross-attention, keys and queries are generated from object embeddings, but a separate set of learned vectors are used as values (note that more complex architectures can be created by combining multiple forms of attention, e.g., object embeddings can first be processed by standard self-attention before applying relational cross-attention). As in the ESBN, these values can be viewed as learned

'symbols', in the sense that they are isolated from the perceptual content of the objects with which they are associated.

This implementation of the relational bottleneck yields the same benefits observed in others: the Abstractor learns relational patterns faster than the Transformer and displays better **out-of-distribution generalization** of those patterns. The Abstractor also has a few advantages relative to existing implementations of the ESBN and CoRelNet. Because the relation matrix is computed using separate key and query projections, the Abstractor is capable of representing asymmetric relations (e.g., can capture the difference in meanings between 'A is greater than B' and 'B is greater than A'; Figure 3B). In addition, multihead relational cross-attention enables the Abstractor to model multidimensional relations. As proposed, ESBN and CoRelNet are limited to relations along a single feature dimension only. Finally, similar to Transformers, the Abstractor is a generative architecture, whereas the ESBN and CoRelNet are purely discriminative (although an alternative implementation of the ESBN has been proposed that can perform generative tasks [51]). This enables the Abstractor to perform a broader range of tasks, including the sequence-to-sequence tasks that are common in natural language processing.

Modeling more complex relations

The neural network modeling work discussed in the previous sections was focused primarily, though not exclusively, on relational patterns involving same/different relations. Although similarity is fundamental to human reasoning, and has indeed been a central focus of theories of relational and analogical reasoning [52–54], human cognition is also characterized by more complex relation types, including asymmetric relations [55] and higher-order relations [5] (relations between relations). It is worth emphasizing that the relational bottleneck can also account for many of these more complex relation types. First, although the inner product between two vectors is inherently symmetric ($a \cdot b$ is identical to $b \cdot a$), the relational bottleneck can also account for asymmetric relations by computing the inner product between separate key and query embeddings ($q_a \cdot k_b$ is not the same as $q_b \cdot k_a$). This use of separate key and query embeddings enables the Abstractor to model asymmetric relations such as greater-than/less-than, as well as other asymmetric relations found in mathematical reasoning problems [49] (and a similar architectural modification is also possible for the ESBN and CoRelNet). Second, although the architectures discussed here focused on pairwise relations, higher-order relations can be straightforwardly accommodated through the recursive application of the relational bottleneck (i.e., by treating the outputs of one relational bottleneck as the inputs to another relational bottleneck, thus computing relations between other relations). Along these lines, recent work proposed relational convolutional networks [56] and demonstrated that a hierarchical relational architecture can learn representations of higher-order relations, outperforming both non-hierarchical relational architectures (e.g., CoRelNet) and deep non-relational architectures (e.g., Transformers). These results illustrate how the relational bottleneck can account for more complex relation types, but it remains an important avenue for future work to investigate whether and how such architectures can account for the full space of relations that characterize human cognition.

As the examples we have considered illustrate, the relational bottleneck can be implemented in a diverse range of architectures, each with their own strengths and weaknesses. In each case, the inclusion of a relational bottleneck enables rapid learning of relations without the need for pre-specified relational primitives. In the remainder of the review, we discuss the implications of this approach for models of cognition and consider how the relational bottleneck may relate to the architecture of the human brain (see Box 3).

> **Box 3. Brain mechanisms supporting the relational bottleneck**
>
> How might the relational bottleneck principle be implemented in the human brain? A central element of this framework is the presence of segregated systems for representing abstract versus perceptual information (i.e., abstract values versus perceptual keys/queries in the ESBN or Abstractor). A large body of findings from cognitive neuroscience suggests the presence of distinct neocortical systems for representing abstract structure (e.g., of space or events) versus concrete entities (e.g., people or places), located in the parietal and temporal cortices, respectively [105–109]. This factorization has also been explored in a number of recent computational models [110–112].
>
> However, this segregation raises the question of how representations in these distinct neocortical systems are flexibly bound together. Though many proposals have been made for how the brain might solve this variable-binding problem (see Box 1 in the main text), one intriguing possibility involves use of the episodic memory (EM) system [40]. A common view holds that EM is supported by rapid synaptic plasticity in the hippocampus, which complements slower statistical learning in the neocortex [41,113]. According to this view, episodes are encoded in the hippocampus by the rapid binding of features that co-occur within an episode, while the features themselves are represented in neocortical systems. This same mechanism could in principle support an architecture similar to the ESBN, by enabling rapid binding of abstract and perceptual neocortical representations. This is in fact very similar to models of cognitive map learning, in which conjunctive representations are rapidly formed in the hippocampus [114]. These hippocampal conjunctive codes bind structural and sensory information, which are thought to be encoded in the medial and lateral entorhinal cortices, respectively, and are commonly understood as extensions of the parietal and temporal neocortical systems referenced earlier. More generally, the involvement of EM in relational reasoning would be consistent with a growing body of recent findings suggesting the potential involvement of EM in tasks traditionally associated with working memory [115–117].
>
> That said, the extent to which variable-binding relies on the hippocampus remains an open question. Some lesion evidence suggests that hippocampal damage does not lead to impairments of abstract reasoning [118]. Other alternatives are that variable-binding may be supported by other structures capable of rapid synaptic plasticity (e.g., the cerebellum, which has been increasingly implicated in higher cognitive functions [119–121]), or by other structures (such as the prefrontal cortex) that use other mechanisms for binding (such as selective attention [122] or working memory gating [123]). The latter possibilities are consistent with findings that prefrontal damage often leads to severe deficits in abstract reasoning tasks [124,125] and prefrontal activity is frequently implicated in neuroimaging studies of abstract reasoning [126,127]. However, this may also reflect the role of prefrontal cortex in representing abstract structure (along with the parietal system described earlier), rather than the binding of that structural information to concrete content. Of course, it is also possible that variable-binding is supported by a collection of distinct mechanisms, rather than a single mechanism alone. These are all important questions for future work that we hope will be usefully guided by the formalisms and computational models reviewed here.

The relational bottleneck in the mind and brain

Modeling the development of counting: a case study in learning abstractions

A core requirement for cognitive models of abstract concept acquisition is to account for the time-course of acquisition during human development. A useful case study can be found in the early childhood process of learning to count [57–59]. Children typically learn to recite the count sequence (i.e. 'one, two, three...' etc.) relatively early, but their ability to use this knowledge to count objects then proceeds in distinct stages (as measured by the 'give n' task [59], in which the child is asked to give the experimenter n objects). Each stage is characterized by the ability to reliably count sets up to a certain size (i.e., first acquiring the ability to reliably count only single objects, then to count two objects, and so on). Around the time that children learn to count sets of five, an inductive transition occurs, in which children rapidly learn to count sets of increasing size. It has been proposed that this transition corresponds to the acquisition of the 'cardinality principle' (the understanding that the last word used when counting corresponds to the number of items in a set [59]) though the exact nature and scope of this inductive transition has been the subject of debate [60,61]. Previous work found that data accumulation played an essential role in supporting this inductive transition [62], but that work employed a symbolic modeling approach, leaving open the question of how such symbol-like processes might be implemented in a neural system.

To address this, a recent study investigated the development of counting in deep neural network architectures [63]. These included the ESBN, the Transformer, and long short-term memory

(LSTM) [64] (a type of recurrent neural network). Each architecture displayed a distinct developmental time-course (Figure 3C). The Transformer displayed a roughly linear time-course, taking approximately the same amount of time to master each number. The LSTM displayed an exponentially increasing time-course, taking more time to learn each new number. Only the ESBN displayed a human-like inductive transition, gradually learning to count each number from one to four and then rapidly acquiring the ability to count higher after learning to count to five. This was due to the ability of the ESBN to learn a procedure over the representations in its control pathway that was abstracted away from the specific numbers in the count sequence (represented in the model's perceptual pathway), allowing it to rapidly and systematically generalize between numbers. Specifically, the ESBN learned a procedure in which it stopped counting once the count sequence matched the desired number of objects (once the target value of n had been reached in the give-n task). The ESBN's use of symbol-like representations (in its control pathway, representing 'item at which to stop') allowed this procedure to be abstracted away from the particular target value (represented in the perceptual pathway and bound to the symbol-like representation in external memory), facilitating rapid generalization to higher values. This case study illustrates how the relational bottleneck can facilitate a human-like developmental trajectory for learning abstract concepts.

Capacity limits and the curse of compositionality

The relational bottleneck principle may also help to explain the limited capacity of some cognitive processes (e.g., working memory) [65]. Recent work has proposed a normative explanation of capacity-limited processes, according to which these capacity limits arise from the use of compositional representations, implemented in an architecture that employs a relational bottleneck [66]. In that architecture, two separate representational pools (each representing distinct feature spaces, e.g., color and location) interact via a dynamic variable-binding mechanism (in that case, implemented using rapid Hebbian learning [67]) in a manner that is conceptually similar to the ESBN. This mechanism enables the model to flexibly construct compositional representations (e.g., representing a visual scene by binding together spatial locations and visual features). However, this flexibility comes at the cost of relying on compositional representations that, by definition, are shared across many different, potentially competing processes. For instance, the representation of a blue object in the upper left corner (formed by binding together representations of 'blue' and 'upper left') will overlap with the representation of a blue object in the lower right corner (formed by binding together representations of 'blue' and 'lower right'), leading to interference between these two representations (note that this interference is also further exacerbated by the use of maximally low-dimensional codes to represent features). This can be viewed as an instance of the more general tradeoff between processing capacity and the use of shared representations [68]and also relates to classic considerations regarding the 'binding problem' in psychology [69], though it has not been previously appreciated that this tradeoff can explain the severely capacity-limited nature of processes such as working memory (and others, including subitizing [70], and absolute judgment [71]). This work illustrates how one of the most notable strengths of human cognition (compositionality) may explain one of its most notable weaknesses (capacity-limited processing) and how both can be implemented in neural networks via the relational bottleneck.

Concluding remarks and future directions

The human mind has a remarkable ability to acquire abstract relational concepts from relatively limited and concrete experience. Here, we have proposed the relational bottleneck as a functional principle that may explain how the human brain accomplishes such data-efficient abstraction and highlighted recently proposed computational models that implement this principle. We have also considered how the principle relates to a range of cognitive phenomena and how it might be implemented by the mechanisms of the human brain.

Outstanding questions

Can a more graded version of the relational bottleneck capture 'content effects', in which abstract reasoning processes are influenced by the specific content under consideration and therefore are not purely abstract or relational, while preserving a capacity for relational abstraction?

Can the relational bottleneck principle be usefully applied to symbolic (or neuro-symbolic) models, in a manner similar to its application to neural network models? For instance, could program induction models benefit from a constraint that forces perceptual inputs to be recoded in terms of relations?

What is the relationship between the relational bottleneck and traditional cognitive models of analogical reasoning? Does the relational bottleneck provide a useful inductive bias toward learning to implement processes such as analogical mapping?

How can other cognitive processes (attention, memory, etc.) be integrated with the relational bottleneck?

How is the relational bottleneck implemented in the brain? To what extent does this rely on mechanisms responsible for episodic memory, attentional mechanisms, and/or other mechanisms that remain to be identified? What role do the hippocampus, prefrontal cortex, and/or other structures play in these computations?

How do architectural biases toward relational processing interact with the influence of training curricula and other cultural sources of abstraction (e.g., formal education)?

It should be noted that the framework reviewed here is not necessarily at odds with the existence of certain forms of domain-specific innate knowledge. In particular, a range of evidence from developmental psychology has suggested that humans possess certain 'core knowledge' systems, such as an innate capacity to represent objects [72–74]. These findings have motivated the development of neuro-symbolic models endowed with these innate capacities [75], although it is also possible that these findings may ultimately be accounted for by the inclusion of additional inductive biases into connectionist systems, such as mechanisms for object-centric visual processing [76–79] (which have also been combined with the relational bottleneck [80]). Critically, however, it is important to emphasize that the relational bottleneck is, in principle, orthogonal to questions about these domain-specific capacities and is focused instead on explaining the induction of abstract, domain-general concepts and relations.

There are a number of important avenues for further developing the relational bottleneck framework. One major question concerns how the framework relates to cognitive models of analogical reasoning, which have traditionally afforded a central role to the process of analogical mapping, driven by patterns of similarity over entities and relations [5,52–54]. An intriguing possibility is that the relational bottleneck encourages neural networks to learn to implement such algorithms, by re-representing their inputs in terms of patterns of similarity (represented as inner products), though future work should aim to more precisely establish this link. Future work should also consider how the proposed framework relates to other theoretical perspectives on out-of-distribution generalization in neural networks, including group theoretic approaches [81] (S. Segert, PhD thesis, Princeton University, 2024), as well as other cognitive processes relevant to abstraction, including attentional processes [82] and semantic cognition [83]. Additionally, much work has suggested that human reasoning is not purely relational, but instead depends on a mixture of concrete and abstract influences [84–87]. This suggests the potential value of a more graded formulation that controls the amount of non-relational information allowed to pass through the bottleneck. Finally, the human capacity for abstraction surely depends not only on architectural biases such as those that we have discussed here, but also on the rich educational and cultural fabric that allows us to build on the abstractions developed by others [88]. In future work, it will be important to explore the interaction between education, culture, and relational inductive biases.

Declaration of interests

The authors declare no competing interests.

References

1. Descartes, R. (1988) Rules for the Direction of our native intelligence. In *Descartes: Selected Philosophical Writings* (Cottingham, J. *et al.*, eds), Cambridge University Press
2. Locke, J. (1894) In *An Essay Concerning Human Understanding* (Fraser, A.C., ed.), Clarendon Press
3. Leibniz, G. (1996) In *New essays on human understanding* (Remnant, P. and Bennett, J., eds), Cambridge University Press, New York
4. Chomsky, N. (1980) *A review of BF Skinner's Verbal Behavior*. The Language and Thought Series pp. 48–64
5. Gentner, D. (1983) Structure-mapping: A theoretical framework for analogy. *Cogn. Sci.* 7, 155–170
6. Holyoak, K.J. (2012) Analogy and relational reasoning. In *The Oxford Handbook of Thinking and Reasoning* (Holyoak, K.J. and Morrison, R.G., eds), pp. 234–259, Oxford University Press
7. Cattell, R.B. (1971) *Abilities: Their structure, growth, and action*, Houghton Mifflin
8. Snow, R.E. *et al.* (1984) The topography of ability and learning correlations. *Adv. Psychol. Hum. Intell.* 2, 103
9. Penn, D.C. *et al.* (2008) Darwin's mistake: Explaining the discontinuity between human and nonhuman minds. *Behav. Brain Sci.* 31, 109–130
10. Newell, A. *et al.* (1972) *Human Problem Solving*, (vol. 104). Prentice-hall Englewood Cliffs
11. Fodor, J.A. (1975) *The Language of Thought*, (vol. 5). Harvard University Press

12. Anderson, J.R. (1996) ACT: A simple theory of complex cognition. *Am. Psychol.* 51, 355
13. Fodor, J.A. and Pylyshyn, Z.W. (1988) Connectionism and cognitive architecture: A critical analysis. *Cognition* 28, 3–71
14. Lake, B.M. *et al.* (2015) Human-level concept learning through probabilistic program induction. *Science* 350, 1332–1338
15. Lake, B.M. *et al.* (2017) Building machines that learn and think like people. *Behav. Brain Sci.* 40, e253
16. Rule, J.S. *et al.* (2020) The child as hacker. *Trends Cogn. Sci.* 24, 900–915
17. Ellis, K. *et al.* (2021) *Dreamcoder: Bootstrapping inductive program synthesis with wake-sleep library learning.* Proceedings of the 42nd ACM Sigplan International Conference on Programming Language Design and Implementation. https://doi.org/10.1145/3410302
18. Dehaene, S. *et al.* (2022) Symbols and mental programs: a hypothesis about human singularity. *Trends Cogn. Sci.* 26, 751–766
19. Yang, Y. and Piantadosi, S.T. (2022) One model for the learning of language. *Proc. Natl. Acad. Sci.* 119, e2021865119
20. Quilty-Dunn, J. *et al.* (2022) The best game in town: The reemergence of the language of thought hypothesis across the cognitive sciences. *Behav. Brain Sci.* https://doi.org/10.1017/S0140525X22002849
21. Piantadosi, S.T. (2021) The computational origin of representation. *Mind. Mach.* 31, 1–58
22. McClelland, J.L. and Rumelhart, D.E. (1989) *Explorations in parallel distributed processing: A handbook of models, programs, and exercises*, MIT Press
23. Elman, J.L. (1990) Finding structure in time. *Cogn. Sci.* 14, 179–211
24. McClelland, J.L. and Rogers, T.T. (2003) The parallel distributed processing approach to semantic cognition. *Nat. Rev. Neurosci.* 4, 310–322
25. McClelland, J.L. *et al.* (2010) Letting structure emerge: connectionist and dynamical systems approaches to cognition. *Trends Cogn. Sci.* 14, 348–356
26. Brown, T. *et al.* (2020) Language models are few-shot learners. *Adv. Neural Inf. Proces. Syst.* 33, 1877–1901
27. Wei, J. *et al.* (2022) *Emergent Abilities of Large Language Models*, Transactions on Machine Learning Research. Survey Certification (issn: 2835-8856)
28. Piantadosi, S. (2023) Modern language models refute Chomsky's approach to language. *Lingbuzz*, Preprint, lingbuzz 7180
29. Bubeck, S. *et al.* (2023) Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv*, preprint arXiv:2303.12712. https://doi.org/10.48550/arXiv.2303.12712
30. Webb, T. *et al.* (2023) Emergent analogical reasoning in large language models. *Nat. Hum. Behav.* 7, 1526–1541
31. Griffiths, T.L. (2020) Understanding human intelligence through human limitations. *Trends Cogn. Sci.* 24, 873–883
32. Frank, M.C. (2023) Bridging the data gap between children and large language models. *Trends Cogn. Sci.* 27, 990–992
33. Webb, T.W. *et al.* (2021) *Emergent Symbols through Binding in External Memory.* 9th International Conference on Learning Representations (ICLR)
34. Smolensky, P. (1990) Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artif. Intell.* 46, 159–216
35. Hummel, J.E. and Holyoak, K.J. (1997) Distributed representations of structure: A theory of analogical access and mapping. *Psychol. Rev.* 104, 427
36. Marcus, G.F. (2001) *The algebraic mind: Integrating connectionism and cognitive science*, MIT Press
37. Kotovsky, L. and Gentner, D. (1996) Comparison and categorization in the development of relational similarity. *Child Dev.* 67, 2797–2822
38. Graves, A. *et al.* (2014) Neural turing machines. *arXiv*, preprint arXiv:1410.5401. https://doi.org/10.48550/arXiv.1410.5401
39. Graves, A. *et al.* (2016) Hybrid computing using a neural network with dynamic external memory. *Nature* 538, 471–476
40. Tulving, E. (2002) Episodic memory: From mind to brain. *Annu. Rev. Psychol.* 53, 1–25
41. McClelland, J.L. *et al.* (1995) Why there are complementary learning systems in the hippocampus and neocortex: insights

from the successes and failures of connectionist models of learning and memory. *Psychol. Rev.* 102, 419
42. Chen, C. *et al.* (2021) Learning to perform role-filler binding with schematic knowledge. *PeerJ* 9, e11046
43. Kerg, G. *et al.* (2022) On neural architecture inductive biases for relational tasks. *arXiv*, preprint arXiv:2206.05056. https://doi.org/10.48550/arXiv.2206.05056
44. Hochreiter, S. (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 6, 107–116
45. Santoro, A. *et al.* (2017) A simple neural network module for relational reasoning. *Adv. Neural Inf. Proces. Syst.* 30, 4974–4983
46. Battaglia, P.W. *et al.* (2018) Relational inductive biases, deep learning, and graph networks. *arXiv*, preprint arXiv:1806.01261. https://doi.org/10.48550/arXiv.1806.01261
47. Kim, J. *et al.* (2018) Not-So-CLEVR: learning same–different relations strains feedforward neural networks. *Interface Focus* 8, 20180011
48. Ichien, N. *et al.* (2021) *Visual analogy: Deep learning versus compositional models.* Proceedings of the 43rd Annual Meeting of the Cognitive Science Society. https://doi.org/10.48550/arXiv.2105.07065
49. Altabaa, A. *et al.* (2024) *Abstractors and relational cross-attention: An inductive bias for explicit relational reasoning in Transformers.* 12th International Conference on Learning Representations (ICLR)
50. Vaswani, A. *et al.* (2017) Attention is all you need. *Adv. Neural Inf. Proces. Syst.* 30, 6000–6010
51. Sinha, I. *et al.* (2020) A memory-augmented neural network model of abstract rule learning. *arXiv*, preprint arXiv:2012.07172. https://doi.org/10.48550/arXiv.2012.07172
52. Falkenhainer, B. *et al.* (1989) The structure-mapping engine: Algorithm and examples. *Artif. Intell.* 41, 1–63
53. Lu, H. *et al.* (2022) Probabilistic analogical mapping with semantic relation networks. *Psychol. Rev.*,
54. Webb, T.W. *et al.* (2023) Zero-shot visual reasoning through probabilistic analogical mapping. *Nat. Commun.* 14, 5144
55. Lu, H. *et al.* (2019) Emergence of analogy from relation learning. *Proc. Natl. Acad. Sci.* 116, 4176–4181
56. Altabaa, A. and Lafferty, J. (2023) Relational Convolutional Networks: A framework for learning representations of hierarchical relations. *arXiv*, preprint arXiv:2310.03240. https://doi.org/10.48550/arXiv.2310.03240
57. Wynn, K. (1992) Children's acquisition of the number words and the counting system. *Cogn. Psychol.* 24, 220–251
58. Carey, S. (2001) Cognitive foundations of arithmetic: Evolution and ontogenisis. *Mind Lang.* 16, 37–55
59. Sarnecka, B.W. and Carey, S. (2008) How counting represents number: What children must learn and when they learn it. *Cognition* 108, 662–674
60. Davidson, K. *et al.* (2012) Does learning to count involve a semantic induction? *Cognition* 123, 162–173
61. Carey, S. and Barner, D. (2019) Ontogenetic origins of human integer representations. *Trends Cogn. Sci.* 23, 823–835
62. Piantadosi, S.T. *et al.* (2012) Bootstrapping in a language of thought: A formal model of numerical concept learning. *Cognition* 123, 199–217
63. Dulberg, Z. *et al.* (2021) *Modelling the development of counting with memory-augmented neural networks.* Proceedings of the 43rd Annual Meeting of the Cognitive Science Society. https://doi.org/10.48550/arXiv.2105.10577
64. Hochreiter, S. and Schmidhuber, J. (1997) Long short-term memory. *Neural Comput.* 9, 1735–1780
65. Miller, G.A. (1956) The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychol. Rev.* 63, 81
66. Frankland, S.M. *et al.* (2021) No coincidence, George: Capacity-limits as the Curse of Compositionality. *PsyArXiv*, preprint. https://doi.org/10.31234/osf.io/cjuxb
67. Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci.* 79, 2554–2558
68. Musslick, S. and Cohen, J.D. (2021) Rationalizing constraints on the capacity for cognitive control. *Trends Cogn. Sci.* 25, 757–775

69. Treisman, A.M. and Gelade, G. (1980) A feature-integration theory of attention. *Cogn. Psychol.* 12, 97–136

70. Mandler, G. and Shebo, B.J. (1982) Subitizing: an analysis of its component processes. *J. Exp. Psychol. Gen.* 111, 1

71. Pollack, I. (1952) The information of elementary auditory displays. *J. Acoust. Soc. Am.* 24, 745–749

72. Spelke, E.S. *et al.* (1992) Origins of knowledge. *Psychol. Rev.* 99, 605

73. Spelke, E.S. and Kinzler, K.D. (2007) Core knowledge. *Dev. Sci.* 10, 89–96

74. Baillargeon, R. and Carey, S. (2012) Core cognition and beyond: The acquisition of physical and numerical knowledge. In *Early Childhood Development and Later Outcome* (Pauen, S.M., ed.), pp. 35–65, Cambridge University Press

75. Smith, K. *et al.* (2019) Modeling expectation violation in intuitive physics with coarse probabilistic object representations. *Adv. Neural Inf. Proces. Syst.* 32, 8985–8995

76. Burgess, C.P. *et al.* (2019) Monet: Unsupervised scene decomposition and representation. *arXiv*, preprint arXiv:1901.11390

77. Locatello, F. *et al.* (2020) Object-centric learning with slot attention. *Adv. Neural Inf. Proces. Syst.* 33, 11525–11538

78. Piloto, L.S. *et al.* (2022) Intuitive physics learning in a deep-learning model inspired by developmental psychology. *Nat. Hum. Behav.* 6, 1257–1267

79. Mondal, S.S. *et al.* (2023) *Learning to reason over visual objects.* 11th International Conference on Learning Representations (ICLR)

80. Webb, T.W. *et al.* (2023) Systematic Visual Reasoning through Object-Centric Relational Abstraction. *Adv. Neural Inf. Proces. Syst.* 37

81. Bronstein, M.M. *et al.* (2021) Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv*, preprint arXiv:2104.13478

82. Vaishnav, M. and Serre, T. (2022) *GAMR: A guided attention model for (visual) reasoning.* 11th International Conference on Learning Representations (ICLR)

83. Giallanza, T. *et al.* (2023) An Integrated Model of Semantics and Control. *PsyArXiv*, preprint. https://doi.org/10.31234/osf.io/jq7ta

84. Wason, P.C. (1968) Reasoning about a rule. *Q. J. Exp. Psychol.* 20, 273–281

85. Johnson-Laird, P.N. *et al.* (1972) Reasoning and a sense of reality. *Br. J. Psychol.* 63, 395–400

86. Bassok, M. *et al.* (1998) Adding apples and oranges: Alignment of semantic and formal knowledge. *Cogn. Psychol.* 35, 99–134

87. Goldberg, A.E. (2003) Constructions: A new theoretical approach to language. *Trends Cogn. Sci.* 7, 219–224

88. McClelland, J.L. (2022) Capturing advanced human cognitive abilities with deep neural networks. *Trends Cogn. Sci.* 26, 1047–1050

89. Touretzky, D.S. (1990) BoltzCONS: Dynamic symbol structures in a connectionist network. *Artif. Intell.* 46, 5–46

90. Plate, T.A. (1995) Holographic reduced representations. *IEEE Trans. Neural Netw.* 6, 623–641

91. Kanerva, P. (2009) Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cogn. Comput.* 1, 139–159

92. Eliasmith, C. *et al.* (2012) A large-scale model of the functioning brain. *Science* 338, 1202–1205

93. Schlegel, K. *et al.* (2022) A comparison of vector symbolic architectures. *Artif. Intell. Rev.* 55, 4523–4555

94. Johnson, J. *et al.* (2017) *Inferring and executing programs for visual reasoning.* Proceedings of the IEEE International Conference on Computer Vision pp. 2989–2998

95. Yi, K. *et al.* (2018) Neural-symbolic VQA: Disentangling reasoning from vision and language understanding. *Adv. Neural Inf. Proces. Syst.* 31, 1039–1050

96. Nye, M. *et al.* (2020) Learning compositional rules via neural program synthesis. *Adv. Neural Inf. Proces. Syst.* 33, 10832–10842

97. Goyal, A. *et al.* (2021) Neural production systems. *Adv. Neural Inf. Proces. Syst.* 34, 25673–25687

98. Liu, D. *et al.* (2021) Discrete-valued neural communication. *Adv. Neural Inf. Proces. Syst.* 34, 2109–2121

99. Feldman, J. (2012) Symbolic representation of probabilistic worlds. *Cognition* 123, 61–83

100. Ke, N.R. *et al.* (2019) Learning neural causal models from unknown interventions. *arXiv*, preprint arXiv:1910.01075. https://doi.org/10.48550/arXiv.1910.01075

101. Palangi, H. *et al.* (2018) *Question-answering with grammatically-interpretable representations.* Proceedings of the AAAI Conference on Artificial Intelligence 32 pp. 5350–5357

102. Jiang, Y. *et al.* (2021) Enriching transformers with structured tensor-product representations for abstractive summarization. *arXiv*, preprint arXiv:2106.01317. https://doi.org/10.48550/arXiv.2106.01317

103. Tishby, N. *et al.* (2000) The information bottleneck method. *arXiv*, preprint physics/0004057. https://doi.org/10.48550/arXiv.physics/0004057

104. Altabaa, A. and Lafferty, J. (2024) Approximation of relation functions and attention mechanisms. *arXiv*, preprint arXiv:2402.08856

105. Mishkin, M. *et al.* (1983) Object vision and spatial vision: two cortical pathways. *Trends Neurosci.* 6, 414–417

106. Goodale, M.A. and Milner, A.D. (1992) Separate visual pathways for perception and action. *Trends Neurosci.* 15, 20–25

107. Frankland, S.M. and Greene, J.D. (2020) Concepts and compositionality: in search of the brain's language of thought. *Annu. Rev. Psychol.* 71, 273–303

108. Summerfield, C. *et al.* (2020) Structure learning and the posterior parietal cortex. *Prog. Neurobiol.* 184, 101717

109. O'Reilly, R.C. *et al.* (2022) The structure of systematicity in the brain. *Curr. Dir. Psychol. Sci.* 31, 124–130

110. Russin, J. *et al.* (2019) Compositional generalization in a deep seq2seq model by separating syntax and semantics. *arXiv*, preprint arXiv:1904.09708. https://doi.org/10.48550/arXiv.1904.09708

111. O'Reilly, R.C. *et al.* (2021) Deep predictive learning in neocortex and pulvinar. *J. Cogn. Neurosci.* 33, 1158–1196

112. Bakhtiari, S. *et al.* (2021) The functional specialization of visual cortex emerges from training parallel pathways with self-supervised predictive learning. *Adv. Neural Inf. Proces. Syst.* 34, 25164–25178

113. Sun, W. *et al.* (2023) Organizing memories for generalization in complementary learning systems. *Nat. Neurosci.* 26, 1438–1448

114. Whittington, J.C. *et al.* (2020) The Tolman-Eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell* 183, 1249–1263

115. Hoskin, A.N. *et al.* (2019) Refresh my memory: Episodic memory reinstatements intrude on working memory maintenance. *Cogn. Affect. Behav. Neurosci.* 19, 338–354

116. Beukers, A.O. *et al.* (2021) Is activity silent working memory simply episodic memory? *Trends Cogn. Sci.* 25, 284–293

117. Beukers, A. *et al.* (2023) When working memory may be just working, not memory. *Psychol. Rev.*

118. Dzieciol, A.M. *et al.* (2017) Hippocampal and diencephalic pathology in developmental amnesia. *Cortex* 86, 33–44

119. Ravizza, S.M. *et al.* (2006) Cerebellar damage produces selective deficits in verbal working memory. *Brain* 129, 306–320

120. D'Mello, A.M. *et al.* (2020) Evidence for hierarchical cognitive control in the human cerebellum. *Curr. Biol.* 30, 1881–1892

121. McDougle, S.D. *et al.* (2022) Continuous manipulation of mental representations is compromised in cerebellar degeneration. *Brain* 145, 4246–4263

122. Miller, E.K. and Cohen, J.D. (2001) An integrative theory of prefrontal cortex function. *Annu. Rev. Neurosci.* 24, 167–202

123. Kriete, T. *et al.* (2013) Indirection and symbol-like processing in the prefrontal cortex and basal ganglia. *Proc. Natl. Acad. Sci.* 110, 16390–16395

124. Waltz, J.A. *et al.* (1999) A system for relational reasoning in human prefrontal cortex. *Psychol. Sci.* 10, 119–125

125. Cipolotti, L. *et al.* (2023) Graph lesion-deficit mapping of fluid intelligence. *Brain* 146, 167–181

126. Christoff, K. *et al.* (2001) Rostrolateral prefrontal cortex involvement in relational integration during reasoning. *Neuroimage* 14, 1136–1149

127. Knowlton, B.J. *et al.* (2012) A neurocomputational system for relational reasoning. *Trends Cogn. Sci.* 16, 373–381

128. Bahdanau, D. *et al.* (2015) *Neural machine translation by jointly learning to align and translate.* 3rd International Conference on Learning Representations (ICLR)

129. Cohen, J.D. *et al.* (1990) On the control of automatic processes: a parallel distributed processing account of the Stroop effect. *Psychol. Rev.* 97, 332

130. Lindsay, G.W. (2020) Attention in psychology, neuroscience, and machine learning. *Front. Comput. Neurosci.* 14, 29

131. Medin, D.L. and Schaffer, M.M. (1978) Context theory of classification learning. *Psychol. Rev.* 85, 207