Relational reasoning and generalization using non-symbolic neural networks

Atticus Geiger[a,1], **Alexandra Carstensen**[b], **Michael C. Frank**[b], **and Christopher Potts**[a]

[a]Department of Linguistics, Stanford University; [b]Department of Psychology, Stanford University

This manuscript was compiled on June 11, 2020

Humans have a remarkable capacity to reason about abstract relational structures, an ability that may support some of the most impressive, human-unique cognitive feats. Because equality (or identity) is a simple and ubiquitous relational operator, equality reasoning has been a key case study for the broader question of abstract relational reasoning. This paper revisits the question of whether equality can be learned by neural networks that do not encode explicit symbolic structure. Earlier work arrived at a negative answer to this question, but that result holds only for a particular class of hand-crafted feature representations. In our experiments, we assess out-of-sample generalization of equality using both arbitrary representations and representations that have been pretrained on separate tasks to imbue them with abstract structure. In this setting, even simple neural networks are able to learn basic equality with relatively little training data. In a second case study, we show that sequential equality problems (learning ABA sequences) can be solved with only positive training instances. Finally, we consider a more complex, hierarchical equality problem, but this requires vastly more data. However, using a pretrained equality network as a modular component of this larger task leads to good performance with no task-specific training. Overall, these findings indicate that neural models are able to solve equality-based reasoning tasks, suggesting that essential aspects of symbolic reasoning can emerge from data-driven, non-symbolic learning processes.

Keyword 1 | Keyword 2 | Keyword 3 | ...

One of the key components of human intelligence is our ability to reason about abstract relations between stimuli. Many of the most unremarkable human activities – scheduling a meeting, following traffic signs, assembling furniture – require a fluency with abstraction and relational reasoning that is unmatched in nonhuman animals. An influential perspective on human uniqueness holds that relational concepts are critical to higher-order cognition (e.g., 1). By far the most common case study of abstract relations has been equality.* Equality is a valuable case study because it is simple and ubiquitous, but also completely abstract in the sense that it can be evaluated regardless of the identity of the stimuli being judged.

Equality reasoning has been studied extensively across a host of systems and tasks, with wildly variant conclusions. In some studies, equality is very challenging to learn: only great apes with either extensive language experience or specialized training succeed in matching tasks in which a *same* pair, AA, must be matched to a novel same pair, BB (2, 3). Preschool children struggle to learn the same regularities in a seemingly similar task (4). In contrast, other studies suggest that equality is simple: bees are able to learn abstract identity relationships from only a small set of training trials (5), and human infants can generalize identity patterns (6) and succeed in relational matching tasks (7). We take the central challenge of this literature to be characterizing the conditions that lead to success or failure in learning an abstract relation in a way that can be productively generalized to new stimuli.

The learning task in all of these cases can be described using the predicate *same* (or equivalently, =), which operates over two inputs and returns TRUE if they are identical in some respect. One perspective in the literature is that success in these learning tasks implies the presence of an equivalent symbolic description in the mind of the solver (2, 8). This view does not provide a lever to distinguish which of these tasks are trivial and which are difficult, however. Further, it can fall prey to circularity: because newborns show sensitivity to identity relations (9), then it would follow from this argument that they must have symbolic representations. If this logic applies also to bees, then we presuppose symbolic representations universally and have no account of the gradient difficulty of different tasks for different species.

An explanation of when same/different tasks are trivial and when they are difficult requires a theoretical framework beyond the symbolic/non-symbolic distinction. To make quantitative predictions about task performance, such a framework should ideally be instantiated in a computational model that takes in training data and learns a solution that generalizes when assessed with stimuli analogous to those used in experimental assessments. Symbolic computational models (e.g., 10) can be used to make contact with data about the breadth of generalization, but they require the existence of a symbolic equality predicate and hence again presuppose symbolic abilities in every case of success. Ideally, we would want a model that describes under what conditions *same* is easy and under what conditions it is hard or unlearnable – and how learning proceeds in these hard cases. We take the development of such an account to be our goal here.

We are inspired by an emergent perspective in the animal learning literature that suggests that the representations un-

Significance Statement

Authors must submit a 120-word maximum statement about the significance of their research paper written at a level understandable to an undergraduate educated scientist outside their field of speciality. The primary goal of the Significance Statement is to explain the relevance of the work in broad context to a broad readership. The Significance Statement appears in the paper itself and is required for all research papers.

Please provide details of author contributions here.

Please declare any conflict of interest here.

*We use the term "equality" here, though different literatures have also used "identity."

[1]To whom correspondence should be addressed. E-mail: author.twoemail.com

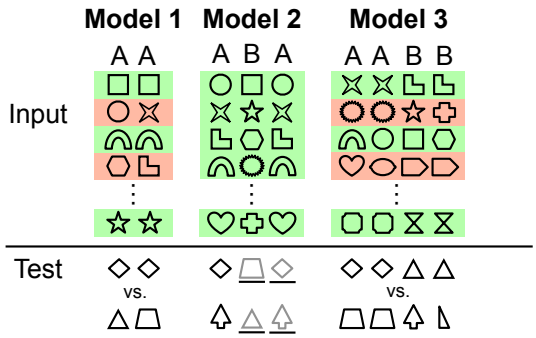


Fig. 1. Relational reasoning tasks. Green and red mark positive and negative training examples, respectively. The sequential task (Model 2) uses only positive instances, and a model succeeds if, prompted with α, it produces a sequence $\beta\ \alpha$ for $\beta \neq \alpha$. For the hierarchical task (Model 3), we show that a model trained on the basic task (Model 1) is effective with no additional training.

derlying non-human animals' and human infants' successes in equality reasoning tasks are graded (11). This view acknowledges the increasing evidence that other species like pigeons (12), crows (13), and baboons (14) can make true, out-of-sample generalizations of same and different relations, but it also recognizes that the observed patterns of behavior do not show the hallmarks of all-or-none symbolic representations. Instead, performance is graded. Out of sample generalization is possible but the level of performance depends critically on the diversity of the training stimuli (e.g., 15). Success typically requires hundreds, thousands, or even tens of thousands of training trials. And the eventual outcome of learning is noisy and imperfect.

These learning signatures appear to be a close match to the kind of learning exhibited by neural network models, a flexible framework for arbitrary function learning that has enjoyed a huge resurgence of interest in recent years (16). Contra this proposal, however, Marcus et al. (8) argued that a broad class of recurrent neural networks were unable to learn equality relations. These claims were subsequently challenged by the presentation of evidence that neural networks are able to learn (at least aspects of) the tasks that Marcus et al. (8) posed (17–21). The subsequent debate (reviewed in 22) revealed a striking lack of consensus on some of the ground rules regarding what sort of generalization would be required to show that the learned function was suitably abstract. In addition, only a narrow range of network architectures and representations was explored.

We revisit this debate here, adopting stringent criteria for generalization and considering a broader range of representations across three models. We model three cases of identity-based reasoning that have featured prominently in discussions of the role of symbols in relational reasoning (Figure 1): (1) learning to discriminate pairs of objects that exemplify the relation *same* or *different*, (2) learning sequences with repeated *same* elements (8), and (3) learning to distinguish hierarchical *same* and *different* relations in a context with pairs of pairs exemplifying these relations (2). This last problem is more challenging and requires vastly more data. We show how using pretrained representations – an active area of development in recent artificial intelligence research (23–27) – can help achieve far faster learning.

Across these three models, we find strong support for the ability of learn equality relations. These results should serve to revise the conclusions of the earlier debate. Marcus and colleagues (8, 28) showed experimentally that neural networks using feature representations cannot generalize to binary features unseen in training. We agree with this claim (and support it with a direct mathematical argument in Appendix **??**). However, they concluded from this result that neural networks will need to have primitive symbolic operators to solve hard relational reasoning tasks. On this point, we disagree. Our experiments show that networks without these primitives can solve a range of these tasks, as long as they use non-featural representations. Overall, these findings suggest that essential aspects of symbolic reasoning can emerge from entirely data-driven, non-symbolic learning processes.

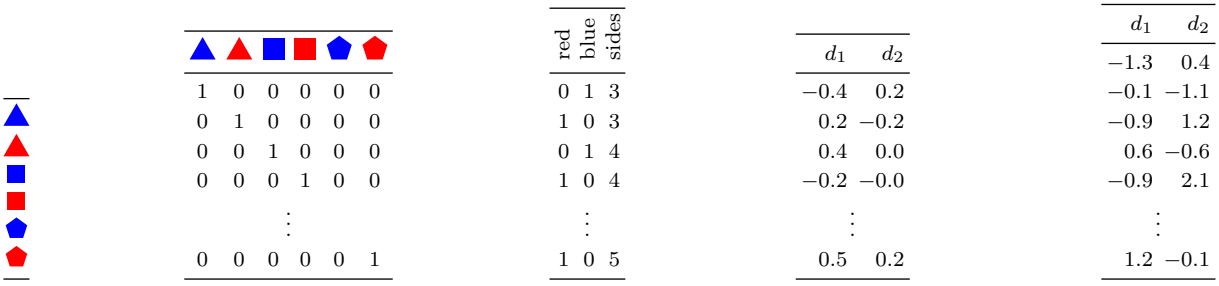
Designing theoretical models of equality learning

We begin by discussing two critical design considerations for our models: standard for generalization by which models should be evaluated and the type of representations they should use. To summarize this discussion: We select generalization tasks with fully disjoint training and test vocabularies to provide the most stringent test of generalization. Further, we adopt both randomly initialized non-featural representations and pretrained non-featural representations for our subsequent models, showing analytically that localist representations cannot make successful out-of-sample generalizations.

Generalization. The standard approach to training and evaluating neural networks is to choose a dataset, divide it randomly into training and assessment sets, train the system on the training set, and then use its performance on the assessment set as a proxy for its capacity to generalize to new data.

The standard approach is fine for many purposes, but it raises concerns in a context in which we are trying to determine whether a network has truly acquired a global solution to a target function. In particular, where there is any kind of overlap between the training and assessment vocabularies (primitive elements), we can't rule out that the network might be primarily taking advantage of idiosyncrasies in the underlying dataset to effectively cheat – to memorize aspects of the training set and learn a local approximation of the target function that happens to provide traction during assessment.

To address this issue, we follow **(author?)** (8) in proposing that networks must be evaluated on assessment sets that are completely disjoint in every respect from the train set, all the way down to the entities involved. For example, below, we train on pairs (a, a) and (a, b), where a and b are representations from a train vocabulary V_T. At test time, we create a new assessment vocabulary V_A, derive equality and inequality pairs (α, α) and (α, β) from that vocabulary, and assess the trained network on these new examples. In adopting these methods, we get a clear picture of the system's capacity to generalize, and we can safely say that its performance during assessment is a window into whether a global solution to identity has been learned. This is a very challenging setting for any machine learning model. For the sequential same–different task, we will see that it even requires us to departure from usual model formulations.



1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
⋮					
0	0	0	0	0	1

(b) Localist.

red	blue	sides
0	1	3
1	0	3
0	1	4
1	0	4
⋮		
1	0	5

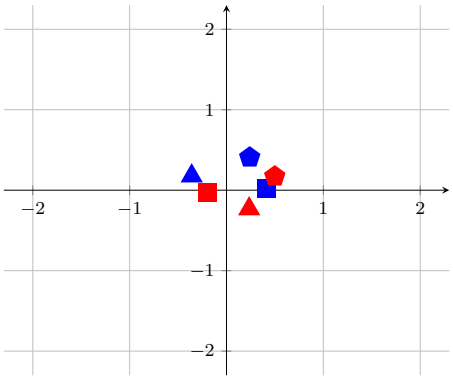
(c) Distributed.

d_1	d_2
−0.4	0.2
0.2	−0.2
0.4	0.0
−0.2	−0.0
⋮	
0.5	0.2

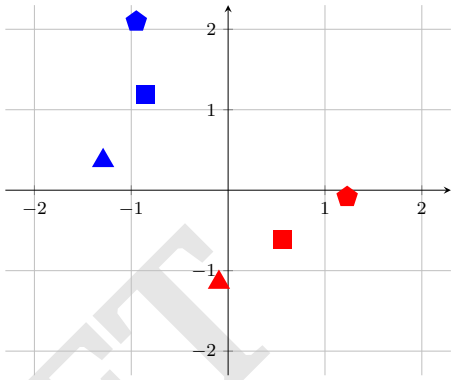
(d) Random.

d_1	d_2
−1.3	0.4
−0.1	−1.1
−0.9	1.2
0.6	−0.6
−0.9	2.1
⋮	
1.2	−0.1

(e) Pretrained.



(f) The matrix in Figure 2d (before pretraining).



(g) The matrix in Figure 2e (after pretraining).

Fig. 2. Each matrix is a method for representing the shapes in Figure 2a, where each row is a vector representation of one shape. Localist and distributed are feature representations where each vector unit encodes the value of a single property. Random and pretrained are non-featural representations where the value of properties are encoded implicitly in two vector units. Random representations and localist representations encode only identity, whereas distributed representations and pretrained representations encode color and number of sides.

Representations. Essentially all modern machine learning models represent objects using vectors of real numbers. However, there are important differences in how these vectors are used to encode the properties of objects. In this section, we characterize two broad approaches to such property encoding – *featural representations* and *non-featural representations* – and argue that the differences between them have not been given sufficient attention in the debate about the ability of neural networks to perform relational reasoning.

To ground our discussion, we consider a hypothetical universe of blocks which vary by shape and color. Figure 2a is a partial view of them, and Figure 2b–Figure 2e present four different ways of encoding the properties of these objects in vectors.

Featural Representations. The defining characteristic of *featural* vector representations is that each dimension encodes the value of a single property. The properties can be binary, integer-valued, or real-valued.

We use the term *localist* for the special case of featural representations in which only identity properties are represented and there is an identity property corresponding to each object. In Figure 2b, each column represents the property of being an object, and every object is represented as a vector that has a single unit with value 1. There is no shared structure across objects; all are equally (un)related to each other as far as the model is concerned.

A featural representation that is not localist is often called a *distributed representation*. Here, column dimensions encode specific properties of objects. Coming back to our universe of blocks, we can represent the properties of being red and being blue with two different binary features, and the property of having a certain number of sides as a single analog feature, as in Figure 2c. Unlike with localist representations, objects in this space can have complex relationships to each other, as encoded in the shared structure given by the columns.

Featural representations – both localist and distributed – have the appealing property that they are easy for researchers to interpret because of the tight correspondence between column dimensions and properties. However, this transparency actually inhibits neural networks from discovering general solutions; such models work far better with representations that have property values implicitly encoded in the abstract structure of the vector space. We demonstrate this result analytically in Appendix **??** for the case of binary features. The core insight is that networks cannot learn anything about column dimensions that are not represented in their training data; whatever weights are associated with those dimensions are unchanged by the learning process, so predictions about those dimensions remain random at test time. The density of non-featural representations helps to avoid this limitation, and pretraining can fully overcome it.

Non-Featural Representations. A *non-featural representation* is a vector that encodes property values implicitly across many dimensions. Perhaps the simplest non-featural representations are *completely random* vectors, as in Figure 2d. Random representations can be seen as the non-featural counterpart

to localist representations. In both of these representation schemes, all the objects are equally (un)related to each other, since column-wise patterns are unlikely in random representations and, to the extent that they are present, they exist completely by chance. However, in random representations, all the column dimensions can contribute meaningfully to identifying objects, whereas a localist representation has only one vector unit that determines the identity of any given object object.

Random representations are a starting point that encodes object identity. To encode more diverse information, we can *pretrain* random representations via a learning process. This will imbue them with rich structure that implicitly encodes property values across many dimensions. Figure 2e provides a simple example. This matrix is the results of pretraining the representations in Figure 2d on the task of predicting whether the object is blue, whether the object is red, and the number of sides the object has. (Appendix **??** provides technical details on our pretraining approach.) Superficially, the two matrices look equally random. However, this is only because our pretraining process leads to property values being implicitly represented by linear structures in the vector representation space. The random representations in Figure 2f have no such linear structure (e.g., no line separates blue and red objects). However, the pretrained representations in Figure 2g have a structure such that straight lines can be drawn to separate objects by color and by number of sides.

Pretraining need not be restricted to input representations. All the parameters of a model can be pretrained. In this setting, a small amount of task-specific training (often called "fine-tuning") might suffice. This offers the possibility that networks might be used as modular components to solve more complex tasks. We realize this possibility in Section , where a model pretrained on a simple equality is used as a modular component to compute hierarchical equality.

Model 1: Same–different relations

In our first model, we consider whether a supervised, feed-forward classification model can learn equality (and inequality) relations in the strict setting we describe above: random representations with disjoint generalization examples.

Model. Our basic model for equality is a feed-forward neural network with a single hidden representation layer:

$$h = \mathrm{ReLU}([a;b]W_{xh} + b_h) \quad [1]$$

$$y = \mathbf{softmax}(hW_{hy} + b_y) \quad [2]$$

The input is a pair of vectors (a, b), each of dimension m, which correspond to the two stimulus objects. These vectors are non-featural representations that do not have features encoding properties of the objects or their identity. These are concatenated to form a single vector $[a;b]$ of dimension $2m$, which is the simplest way of merging the two representations to form a single input.

This representation is multiplied by a matrix of weights W_{xh} of dimension $2m \times n$ and a bias vector b_h of dimension n is added to this result, where n is the hidden layer dimensionality. These two steps create a linear projection of the input representation, and the bias term is the value of this linear projection when the input representation is the zero vector. Then, the non-linear activation function ReLU ($\mathrm{ReLU}(x) = \max(0, x)$) is applied element-wise to this linear projection. This non-linearity is what gives the neural model more expressive power than a logistic regression. The result is the hidden representation h.

The hidden representation is the input to the classification layer: h is multiplied by a second matrix of weights W_{hy}, dimension $n \times 2$, and a bias term b_y (dimension 2) is added to this. This second bias term encodes the probabilities of each class when the hidden representation is 0. The result is fed through the softmax activation function: $\mathbf{softmax}(x)_i = \frac{\exp x_i}{\sum_j \exp x_j}$. This creates a probability distribution over the classes (positive and negative). For a given input, the model computes this probability distribution and the input is categorized as the class with the higher probability.

During training, this model is presented with positive and negative labeled examples and the parameters W_{xh}, W_{hy}, b_y, and b_h are learned using back propagation with a cross entropy loss function. This function is defined as follows, for a corpus of N examples and K classes:

$$\max(\theta) \quad \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y^{i,k}\log(h_\theta(i)^k) \quad [3]$$

where θ abbreviates the model parameters (W_{xh}, W_{hy}, b_y, b_h), $y^{i,k}$ is the actual label for example i and class k, and $h_\theta(i)^k$ is the corresponding prediction.

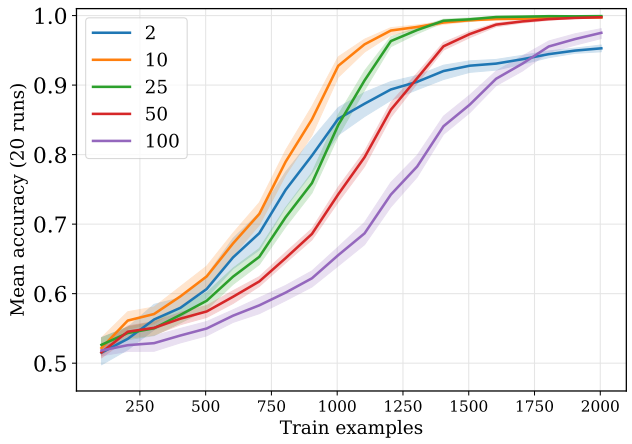
During testing, this model is tasked with categorizing inputs unseen during training. It is straightforward to show that a network like this is capable of learning equality as we have defined it. Appendix **??** provides an analytic solution to the equality relation using this neural model. Here we illustrate with a small example network that maps all identity pairs to $[0.5, 0.5]$ and all non-identity pairs to $[y, 1-y]$ where $y > 0.5$, which supports a trivial classification rule:

$$\mathrm{ReLU}\left([a;b]\begin{pmatrix}1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\\ -1 & 0 & 1 & 0\\ 0 & -1 & 0 & 1\end{pmatrix} + \mathbf{0}\right)\begin{pmatrix}1 & 0\\ 1 & 0\\ 1 & 0\\ 1 & 0\end{pmatrix} + \mathbf{0} \quad [4]$$

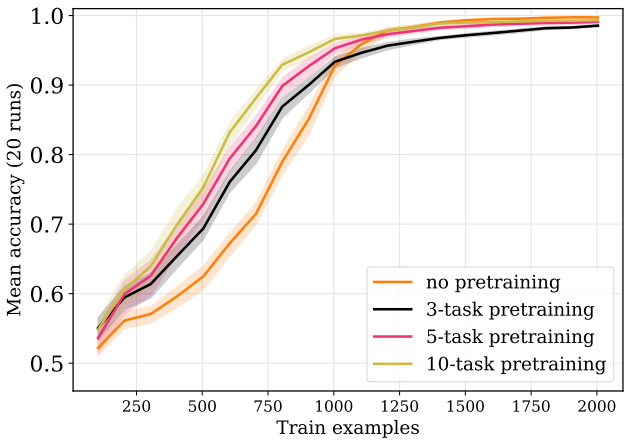
This result shows that equality in our sense is learnable in principle, but it doesn't resolve the question of whether networks can find this kind of solution given finite training data. To address this, we train the network on a stream of pairs of random vectors. Half of these are identity pairs (a, a), labeled with 1, and half are non-identity pairs (a, b), labeled with 0. Trained networks are assessed on the same kind of balanced dataset, with vectors that were never seen in training so that, as discussed earlier, we get a clear picture of whether they have found a generalizable solution. Further optimization details are in Appendix **??**.

Results. Figure 3a presents typical results. This is for the case where the hidden layer dimensionality is 100, and we plot results for different embedding dimensionalities m and different amounts of training data. The picture is comparable with hidden dimensionalities at 10, 25, and 50, but those models require more training data to reach (near) perfect performance (Appendix **??**).

While all the networks in Figure 3a reach above-chance performance almost immediately, they require upwards of



(a) Results for a model in which the hidden layer has dimensionality 100. The lines correspond to different dimensions for the entities a and b in the input pairs (a, b). These models largely pass 90% accuracy with 1,000 training examples and reach (near) perfection by 1,250.



(b) Results where random representations are grounded in property domains of color and number of sides via pretraining learning tasks. The 'no pretraining' model is the best of the models at left (10-dimensional embeddings, 100-dimensional hidden layers), repeated to facilitate comparisons. The pretraining models use those same dimensionalities.

Fig. 3. Same–different results with and without pretraining.

1,000 examples to truly solve these tasks. We additionally ran the above experiments using random representations that were pretrained using a linear classifier for 0, 3, 5, or 10 different feature discrimination tasks. For example, following Figure 2, a two-task model would be trained to encode the properties of color, number of sides, and size.

Figure 3b summarizes the results of these experiments for 10-dimensional embeddings and 100-dimensional hidden representations, as this seems to be the network that learns the fastest with random inputs. Interestingly, we see a clear speed-up, with more pretraining tasks resulting in the largest gains; by grounding our representations in "property domains" (as represented by the different task dimensions), we imbue them with implicit structure that makes learning easier.

Discussion. Our assessment pairs have nothing in common with the training pairs except insofar as both involve vectors of real numbers of the same dimensionality. During training, the network is told (via labels) which pairs are equality pairs and which are not, but the pairs themselves contain no information about equality per se. It thus seems fair to us to say that these networks have learned equality, or at least how to simulate that relation with near perfect accuracy. Further, the use of representations that are structured by pre-training results in a speed-up of learning.

Model 2: Sequential same–different (ABA task)

Our first model is simple and successfully learns equality. However, this model is supervised with both positive and negative evidence. In the initial debate around these issues, supervision with negative evidence was dismissed as an unreasonably strong learning regime (e.g., 29). While this argument likely holds true for language learning (in which supervision is generally agreed not to be binary or direct; 30, 31), it is not necessarily true for learning more generally. Nevertheless, learning of sequential rules without negative feedback is possible for infants (8, 32). In experiments of this type, infants are presented with a set of positive examples. Our next model explores whether neural network models can learn this task in a challenging regime with no negative supervision.

Model. To explore learning with only positive instances, we use neural language models. These models are sequential: at each timestep, they predict an output given their predictions about the preceding timesteps. As typically formulated, the prediction function is just a classifier: at each timestep, it predicts a probability distribution over the entire vocabulary of options, and the item with the highest probability is chosen as a symbolic output. This outpiut becomes the input at the next timestep, and the process continues.

This formulation will not work in situations in which we want to make predictions about test itesm with an entirely disjoint vocabulary from the training sample. The classifier function will get no feedback about these out-of-vocabulary items during training, and so it will never predict them during testing. To address this issue, we reformulate the prediction function. Our proposal is to have the model predict output vector representations – instead of discrete vocabulary items – at each timestep. During training, the model is trained to minimize the distance between these output predictions and the representations of the actual output entities. During assessment, we take the prediction to be the item in the entire vocabulary (training and assessment) whose representation is closest to the predicted vector (in terms of Euclidean distance). This fuzzy approach to prediction creates enough space for the model to predict sequences from an entirely new vocabulary.

The specific model we use for this is as follows:

$$h_t = \mathbf{LSTM}(x_t, h_{t-1}) \quad [5]$$

$$y_t = h_t W + b \quad [6]$$

This holds for $t > 0$, and we set $h_0 = \mathbf{0}$. **LSTM** is a long short-term memory cell (33). Full details on these cells are given in Section **??**.

The input is a sequence of vectors $x_1, x_2, x_3, \ldots$, each of dimension m, which correspond to a sequence of stimulus

objects. These vectors are, again, non-featural representations that do not have features encoding properties of the objects or their identity.

At each timestep t, the input vector x_t is fed into the **LSTM** cell along with the previous hidden representation h_{t-1}. The defining feature of an **LSTM** is the ability to decide whether to store information from the current input, x_t, and whether to remember or forget the information from the previous timestep h_{t-t}. The output of the **LSTM** cell is the hidden representation for the current time step h_t. The dimension of the hidden representations is n. The hidden representation is multiplied by a matrix W with dimensionality $n \times m$ to produce y_t. This result, y_t, is a linear projection of the hidden representation into the input vector space, which is necessary because y_t is a prediction of what the next input, x_{t+1}, will be.

The objective function is as follows:

$$\max(\theta) \quad -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\left\|h_\theta\left(x^{i,0:t-1}\right)-x^{i,t}\right\|^2 \qquad [7]$$

for N examples. Here, T_i is the length of example i. As before, θ abbreviates the parameters of the model as specified in (5)–(6). We use $h_\theta(x^{i,0:t-1})$ for the vector predicted by the model for example i at timestep t, which is compared to the actual vector at timestep t via squared Euclidean distance (i.e., the mean squared error).

Appendix **??** provides an analytic solution to the ABA task using this model. To see how well the model performs in practice, we trained networks on sequences `<s> a b a </s>`, where `b` ≠ `a`. We show the network every such sequence during training, from an underlying vocabulary of 20 items (creating a total of 380 examples). To assess how well the model learns this pattern, we seed it with `<s> x` where `x` is an item from a disjoint vocabulary from that seen in training, and we say that a prediction is accurate if the model continues with `y x </s>`, where `y` is any character (from the training or assessment vocabulary) except `x`.

Results. Figure 4a shows the results for a model with a 100-dimensional hidden representation. As before, the results are comparable for smaller networks. Unlike for the previous equality experiments, we found that we had to allow the model to experience multiple epochs of training on the same set in order to succeed. Figure 4b shows results with pre-trained representations across a range of different tasks (as in Model 1); these yielded no increase in performance.

Discussion. These sequential models are given no negative examples and they must predict into a totally new vocabulary. Despite these challenges, they succeed at learning the underlying patterns in our data, even though they use only random representations. On the other hand, the learning process is slow and data-intensive. We hypothesized that grounding representations in property domains via pretraining might lead to noticeable speed-ups, as it did in Section , but we did not see this effect in practice. We speculate that there may be model variants that reduce these demands, given that learning is in principle possible in this architecture, but we did not discover them and leave them to future work.

Model 3: Hierarchical same–different relations

Given the strong results found for simple equality relations in Section , we can ask whether more challenging equality problems are also learnable in our setting. The hierarchical equality task used by **(author?)** (2) is an interesting test case: given a pairs of pairs $((a,b),(c,d))$, the label is 1 if $(a = b) = (c = d)$, else 0. **(author?)** (2) suggested that the ability exemplified by this task – reasoning about hierarchical *same* and *different* relations – could represent a form of symbolic abstraction uniquely enabled by language. Given the non-symbolic nature of our models, our simulations provide a test of this hypothesis, though we should look critically at their ability to find good solutions with reasonable amounts of training data.

Model. We can approach this task using the same model and methods as we used for equality. The only change required to equations (1)–(2) is that we create inputs $[a;b;c;d]$: the flat concatenation of all the elements of the two pair of vectors. This change in turn leads W_{xh} to have dimensionality $4m \times n$.

These models are able to find nearly perfect solutions, but vastly more training data is required for this task than was required for simple equality, and the network configuration matters much more. For example, our model with 10-dimensional entity representations and 100-dimensional hidden representations reached near perfect accuracy, but only with over 95,000 training instances. A comparable model with 50-dimensional entity representations failed to get traction at all with this amount of training data, and pretraining led to only minor improvements. Appendix **??** provides a full picture of these learning trends.

We hypothesized that the flat input representations $[a;b;c;d]$ might be suboptimal here. This task is intuitively hierarchical: if one works out the equality labels for each of the two pairs, then the further classification decision can be done entirely on that basis. Our current neural network might be too shallow to find this kind of decomposition. To address this, we can simply add another intermediate layer:

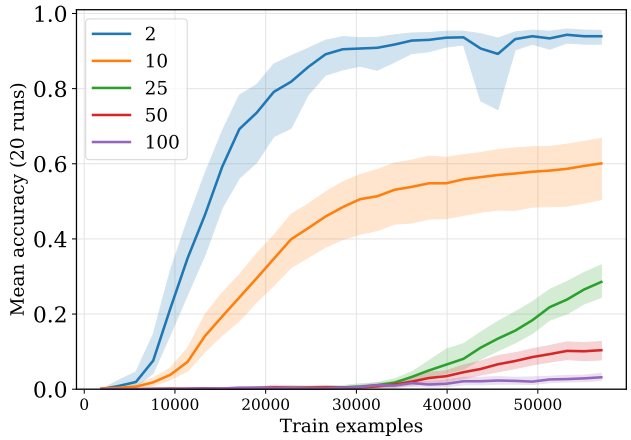
$$h_1 = \mathrm{ReLU}([a;b;c;d]W_{xh} + b_{h_1}) \qquad [8]$$

$$h_2 = \mathrm{ReLU}(h_1 W_{hh} + b_{h_2}) \qquad [9]$$

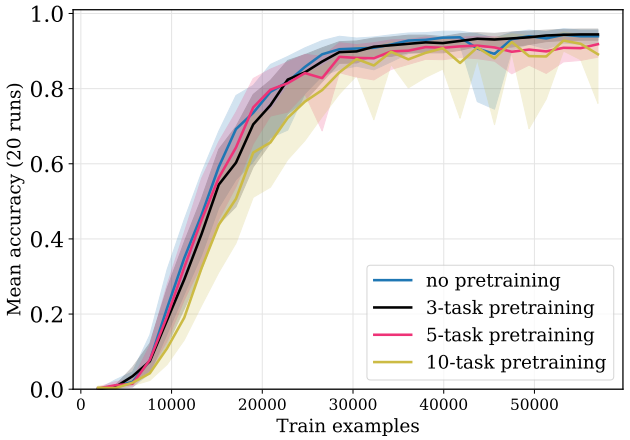
$$y = \mathbf{softmax}(h_2 W_{hy} + b_y) \qquad [10]$$

Results. Figure 5a shows that these deeper networks learn faster and are more robust to different network configurations than the single-layer variant. Intuitively, these models are better structured to find hierarchical solutions to this hierarchical problem. Pretraining led to further gains in training speed (Figure 5b).

Discussion. These results are encouraging, but we still require more than 20,000 training instances to reach top performance and upwards of 10,000 examples even with pretraining. This amount is still vastly more data than human participants get in similar experiments, which typically involve short exposures in the range of dozens to hundreds of examples (e.g., 8, 34). Thus, it is worth asking whether there are other solutions that would be more data efficient and more in line with human capabilities. We next sought to capitalize even more on the hierarchical nature of this task by defining a modular pretraining regime in which previously learned capabilities are recruited for new tasks.

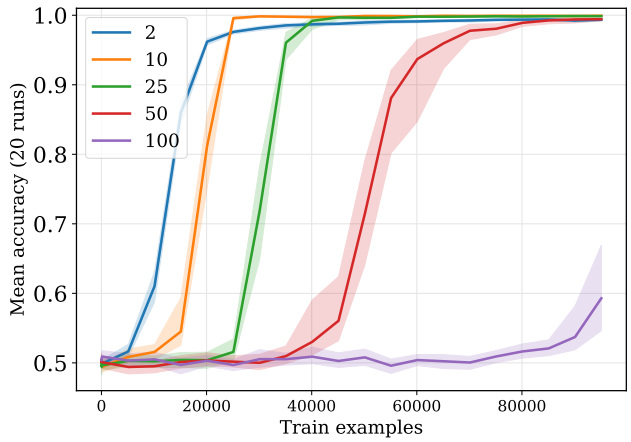


(a) Results for a model in which each h_t in Eq. (5) has dimensionality 100. The lines correspond to the dimensionality of the input representations (x_t in Eq. (5)). All the training examples are presented at once over multiple epochs.

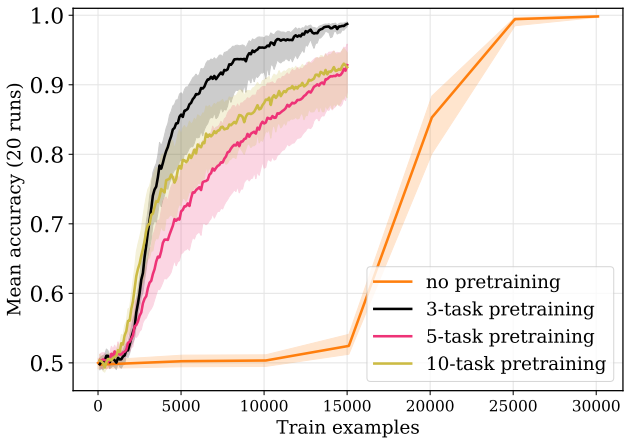


(b) Results where random representations are grounded via pretraining learning tasks. Simulations adopt the best-performing settings from the no-pretraining condition (2-dimensional embeddings, 100-dimensional hidden representations).

Fig. 4. Sequential same–different results with and without pretraining.

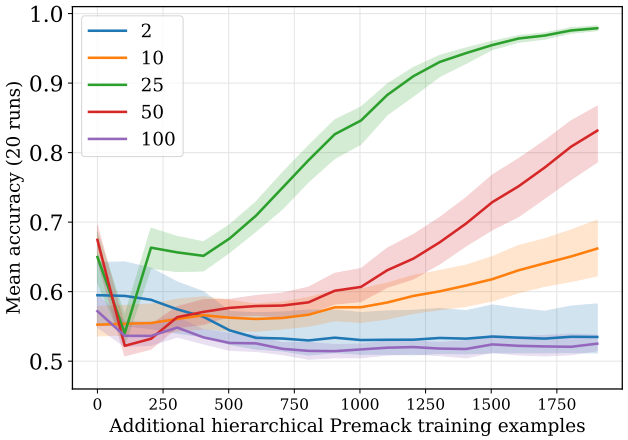


(a) Results for a network with two 100-dimensional hidden layers. Nearly all the networks solve the task, but they require very large training sets to do so.

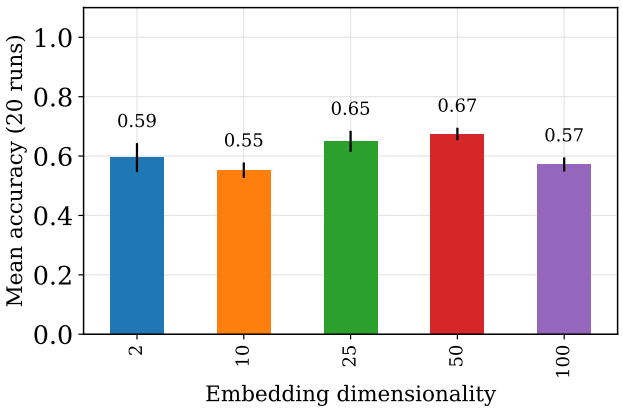


(b) Results from pre-trained networks, adopting the best of the models at left (10-dimensional embeddings, 100-dimensional hidden representations).

Fig. 5. Hierarchical same–different results with and without pretraining.



(a) Basic equality networks applied to the hierarchical equality task. Each line represents a different embedding dimensionality, which is constrained in this model to match the hidden dimensionality. Even with no additional training instances for this task, all models achieve greater than chance accuracy, and even modest amounts of additional training on the task lead to excellent performance.



(b) The results from the plot at left where the x-axis is 0, that is, where we are testing zero-shot generalization from the pretrained equality network to the hierarchical setting. All the models perform well above chance in this setting, with the 50-dimensional version achieving a mean of 67% accuracy.

Fig. 6. Modular network results for the hierarchical same–different task.

The critical role of experience. Our successful results training neural networks on simple equality suggested another strategy for solving the hierarchical equality task. Rather than requiring our networks to find solutions from scratch, we pretrained them on basic equality tasks and then used those parameters as a starting point for learning hierarchical equality. This set of simulations was conceptually similar to our previous experiments with pretraining, but now we pretrained an entire subpart of the model, rather than just input representations.

Model. The hierarchical equality task requires computing the equality relation three times: compute whether the first two inputs are equal, compute whether the second two inputs are equal, then compute whether the truth value outputs of these first two computations are equal. We propose to use the same network pretrained on basic equality to perform all three equality computations. More precisely, we define

$$h_1 = \text{ReLU}([a; b]W_{xh} + b_h) \quad [11]$$
$$h_2 = \text{ReLU}([c; d]W_{xh} + b_h) \quad [12]$$
$$h_3 = \text{ReLU}([h_1; h_2]W_{xh} + b_h) \quad [13]$$
$$y = \textbf{softmax}(h_3 W_{hy} + b_y) \quad [14]$$

where W_{xh}, W_{hy}, b_h, and b_y are the parameters from the model in equations (1)–(2) already trained on basic equality. Crucially, the same parameters, W_{xh} and b_h, are used three times: twice to compute representations encoding whether a pair of input entities are equal (h_1, h_2), and once to compute a representation (h_3) encoding whether the truth values encoded by h_1 and h_2 are equal. This final representation is then used to compute a probability distribution over two classes, and the class with the higher probability is predicted by the model.

Results and discussion. Figure 6 shows that this model succeeds very quickly at the hierarchical equality task. As we see in Figure 6a, after just a few thousand examples, the majority of the model configurations perform with near perfect accuracy. The findings in Figure 6b are even more striking: all the models have above chance performance after being trained only on the simple equality task – that is, they achieve zero-shot generalization to the hierarchical task. It is remarkable that a model trained only on equality between entities is able to get traction on a problem that requires determining whether equality holds between the truth values encoded in two learned representations.

It might be possible to effectively combine network pretraining with input pretraining as in the previous experiments. An initial exploration of this idea is presented in Appendix **??**. While we have not yet found a way to use this combination of pretraining regimes to improve over Figure 6, we are optimistic about such combinations for future work.

General Discussion

Equality is a key case study for understanding the origins of human relational reasoning. This case study has been puzzling for symbolic accounts of reasoning because such accounts do not provide a compelling explanation for why some equality tasks are so easy to learn and others are so hard. In addition, evidence of graded learning and generalization in non-human species suggests that a gradual learning account might provide more traction in explaining the empirical data (11). Inspired by this work, we revisited a long-standing debate about whether neural network models can learn equality relations from data (22). We presented a series of such models that succeeded even in stringent assessments, and, in the case of neural language models, even without being shown negative examples.

In some settings, our current models require more training instances than humans seem to need. However, our pretraining approach suggests a path forward: by using pretrained models as modular components, we can get traction on challenging tasks without any training specifically for those tasks. In some cases, even a small amount of additional training can make a substantial difference. Perhaps modular, pretrained components could serve as the basis for more complex cognitive abilities more generally.

One implication of these pretraining findings is that it should be possible to scaffold performance in complex, hierarchical equality tasks via training on simpler ones. Indeed, Smirnova et al. (13) show just this result in crows, consistent with our findings. Although we do not discount the potential role of linguistic labels in informing adult humans' expertise in such tasks (35), pretraining also provides a potential account of how infants and young children might succeed in a range of equality reasoning tasks without access to specific linguistic symbols like "same" (4, 7, 36).

More broadly still, our work suggests a possible way forward in understanding the acquisition of logical semantics. Graded logical functions like those our models learned here could form the foundation for a semantics of words like "same" (37). Such an option is appealing because it escapes from the circularity of defining the semantics of linguistic symbols as originating in a mental primitive SAME. A semantics for "same" requires defining its inputs and outputs as well as how it composes with other symbols. The assertion that there is a primitive identity computation does not provide these; it further fails to explain the flexibility that allows us to call two Toyota Corollas "the same" but two twin sisters "different." In contrast, the kinds of networks we propose here could in principle be conditioned contextually to provide flexible, context-sensitive interpretation of logical meanings (**?**).

Earlier debates about the nature of equality computations centered around the question of whether models included symbolic elements. We believe ours do not; but it is of course possible to quibble with this judgment. For example, since the supervisory signal used in Models 1 and 3 is generated based on a symbolic rule, perhaps that makes these models symbolic under some definition. (Of course, the same argument could be applied to the supervision signal that is provided to crows or baboons). We view this kind of argument as terminological, rather than substantive. In the end, our goal is an explicit learning theory for relational reasoning. Our hope is that the work described here takes a first step in this direction.

Materials and Methods

Please describe your materials and methods here. This can be more than one paragraph, and may contain subsections and equations as required. Authors should include a statement in the methods section describing how readers will be able to access the data in the paper.

Subsection for Method. Example text for subsection.

ACKNOWLEDGMENTS. This work is supported in part by a Facebook Robust Deep Learning for Natural Language Processing Research Award and by the McDonnell Foundation grant "The Ontogeny of Propositional Thought".

1. Gentner D, Goldin-Meadow S (2003) *Language in mind: Advances in the study of language and thought.* (MIT press).
2. Premack D (1983) The codes of man and beasts. *Behavioral and Brain Sciences* 6(1):125–136.
3. KR Thompson R, Rattermann MJ, L Oden D (2001) Perception and judgement of abstract same-different relations by monkeys, apes and children: Do symbols make explicit only that which is implicit? *Hrvatska revija za rehabilitacijska istraživanja* 37(1):9–22.
4. Walker CM, Bridgers S, Gopnik A (2016) The early emergence and puzzling decline of relational reasoning: Effects of knowledge and search on inferring abstract concepts. *Cognition* 156:30–40.
5. Avarguès-Weber A, Deisig N, Giurfa M (2011) Visual cognition in social insects. *Annual review of entomology* 56:423–443.
6. Anderson EM, Chang YJ, Hespos S, Gentner D (2018) Comparison within pairs promotes analogical abstraction in three-month-olds. *Cognition* 176:74–86.
7. Ferry AL, Hespos SJ, Gentner D (2015) Prelinguistic relational concepts: Investigating analogical processing in infants. *Child Development* 86(5):1386–1405.
8. Marcus GF, Vijayan S, Rao SB, Vishton PM (1999) Rule learning by seven-month-old infants. *Science* 283(5398):77–80.
9. Gervain J, Berent I, Werker JF (2012) Binding at birth: The newborn brain detects identity relations and sequential position in speech. *Journal of Cognitive Neuroscience* 24(3):564–574.
10. Frank MC, Tenenbaum JB (2011) Three ideal observer models for rule learning in simple languages. *Cognition* 120(3):360–371.
11. Wasserman E, Castro L, Fagot J (2017) Relational thinking in animals and humans: From percepts to concepts. in *APA HAndbook of Comparative Psychology: Perception, Learning, and Cognition*, eds. Call J, Burghardt GM, Pepperberg IM, Snowdon CT, Zentall T. (American Psychological Association) Vol. 2.
12. Cook RG, Wasserman EA (2007) Learning and transfer of relational matching-to-sample by pigeons. *Psychonomic Bulletin & Review* 14(6):1107–1114.
13. Smirnova A, Zorina Z, Obozova T, Wasserman E (2015) Crows spontaneously exhibit analogical reasoning. *Current Biology* 25(2):256–260.
14. Fagot J, Thompson RK (2011) Generalized relational matching by guinea baboons (papio papio) in two-by-two-item analogy problems. *Psychological Science* 22(10):1304–1309.
15. Castro L, Kennedy PL, Wasserman EA (2010) Conditional same-different discrimination by pigeons: Acquisition and generalization to novel and few-item displays. *Journal of Experimental Psychology: Animal Behavior Processes* 36(1):23.
16. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *nature* 521(7553):436–444.
17. Dienes Z, Altmann GT, Gao SJ (1999) Mapping across domains without feedback: A neural network model of transfer of implicit knowledge. *Cognitive Science* 23(1):53–82.
18. Seidenberg MS, Elman JL (1999) Networks are not 'hidden rules'. *Trends in Cognitive Sciences* 3(8):288–289.
19. Seidenberg MS, Elman JL (1999) Do infants learn grammar with algebra or statistics? *Science* 284(5413):433–433.
20. Elman JL (1999) Generalization, rules, and neural networks: A simulation of Marcus et. al. HTML document.
21. Negishi M (1999) Do infants learn grammar with algebra or statistics? *Science* 284(5413):435.
22. Alhama RG, Zuidema W (2019) A review of computational models of basic rule learning: The neural-symbolic debate and beyond. *Psychonomic bulletin & review* 26(4):1174–1194.
23. Collobert R, et al. (2011) Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12:2493–2537.
24. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality in *Advances in Neural Information Processing Systems 26*, eds. Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ. (Curran Associates, Inc.), pp. 3111–3119.
25. Pennington J, Socher R, Manning CD (2014) GloVe: Global vectors for word representation in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (Association for Computational Linguistics, Doha, Qatar), pp. 1532–1543.
26. Peters M, et al. (2018) Deep contextualized word representations in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. (Association for Computational Linguistics), pp. 2227–2237.
27. Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. (Association for Computational Linguistics, Minneapolis, Minnesota), pp. 4171–4186.
28. Marcus GF (2001) *The Algebraic Mind: Integrating Connectionism and Cognitive science.* (MIT Press).
29. Marcus GF (1999) Rule learning by seven-month-old infants and neural networks. Response to Altmann and Dienes. *Science* 284:875.
30. Brown R, Hanlon C (1970) Derivational complexity and order of development in speech in *Cognition and the development of language.*, ed. Hayes JR. (Wiley).
31. Chouinard MM, Clark EV (2003) Adult reformulations of child errors as negative evidence. *Journal of child language* 30(3):637–669.
32. Rabagliati H, Ferguson B, Lew-Williams C (2019) The profile of abstract rule learning in infancy: Meta-analytic and experimental evidence. *Developmental science* 22(1):e12704.
33. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural computation* 9(8):1735–1780.
34. Endress AD, Scholl BJ, Mehler J (2005) The role of salience in the extraction of algebraic rules. *Journal of Experimental Psychology: General* 134(3):406.
35. Gentner D (2003) Why we're so smart. *Language in mind: Advances in the study of language and thought* 195235.
36. Hochmann JR, Mody S, Carey S (2016) Infants' representations of same and different in match-and non-match-to-sample. *Cognitive psychology* 86:87–111.
37. Potts C (2019) A case for deep learning in semantics: Response to pater. *Language.*