A Theory of Relation Learning and Cross-Domain Generalization

Leonidas A. A. Doumas1, Guillermo Puebla1, Andrea E. Martin2, 3, and John E. Hummel4

1 School of Philosophy, Psychology, and Language Sciences, The University of Edinburgh
2 Donders Centre for Cognitive Neuroimaging, Radboud University
3 Language and Computation in Neural Systems Group, Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands
4 Department of Psychology, University of Illinois

People readily generalize knowledge to novel domains and stimuli. We present a theory, instantiated in a computational model, based on the idea that cross-domain generalization in humans is a case of analogical inference over structured (i.e., symbolic) relational representations. The model is an extension of the Learning and Inference with Schemas and Analogy (LISA; Hummel & Holyoak, 1997, 2003) and Discovery of Relations by Analogy (DORA; Doumas et al., 2008) models of relational inference and learning. The resulting model learns both the content and format (i.e., structure) of relational representations from nonrelational inputs without supervision, when augmented with the capacity for reinforcement learning it leverages these representations to learn about individual domains, and then generalizes to new domains on the first exposure (i.e., zero-shot learning) via analogical inference. We demonstrate the capacity of the model to learn structured relational representations from a variety of simple visual stimuli, and to perform cross-domain generalization between video games (Breakout and Pong) and between several psychological tasks. We demonstrate that the model’s trajectory closely mirrors the trajectory of children as they learn about relations, accounting for phenomena from the literature on the development of children’s reasoning and analogy making. The model’s ability to generalize between domains demonstrates the flexibility afforded by representing domains in terms of their underlying relational structure, rather than simply in terms of the statistical relations between their inputs and outputs.

Keywords: relation learning, generalization, learning relational content, learning structured representations, neural oscillations

Supplemental materials: https://doi.org/10.1037/rev0000346.supp

Many children learn the apocryphal story of Newton discovering his laws of mechanics when an apple fell from a tree and hit him on the head. We were told that this incident gave him the insight that ultimately led to a theory of physics that, hundreds of years later, would make it possible for a person to set foot on the moon. The point of this story is not about the role of fruit in humankind’s quest to understand the universe; the point is that a single incident—a child’s proxy for a small number of observations—led Newton to discover laws that generalize to an unbounded number of new cases involving not only apples, but also cannonballs and celestial bodies. Without being told, we all understand that generalization—the realization that a single set of principles governs falling apples, cannonballs, and planets alike—is the real point of the story.

Although the story of Newton and the apple is held up as a particularly striking example of insight leading to broad generalization, it resonates with us because it illustrates a fundamental property of human thinking: People are remarkably good—irresponsibly good, from a purely statistical perspective—at applying principles...
discovered in one domain to new domains that share little or no superficial similarity with the original. A person who learns how to count using stones can readily apply that knowledge to apples. A graduate student who learns to analyze data in the context of psychological experiments can readily generalize that knowledge to data on consumer preferences or online search behavior. And a person who learns to play a simple videogame like Breakout (where the player moves a paddle at the bottom of the screen horizontally to hit a ball toward bricks at the top of the screen) can readily apply that knowledge to similar games, such as Pong (where the player moves a paddle on the side of the screen vertically and tries to hit a ball past the opponent paddle on the other side of the screen).

This kind of “cross-domain” generalization is so commonplace that it is tempting to take it for granted, to assume it is a trivial consequence of the same kind of statistical associative learning taught in introductory psychology. But the truth is more complicated. First, there is no clear evidence that any species other than humans is capable of the kind of flexible cross-domain generalization we find so natural (see Penn et al., 2008). And second, while “cross-domain” generalization has also been the subject of substantial research in the field of machine learning (see, e.g., Gamrian & Goldberg, 2019; Kansky et al., 2017; Zhang et al., 2018), robust, human-like cross-domain generalization continues to frustrate even the most powerful deep neural networks (DNNS; see, e.g., Bowers, 2017; Geirhos et al., 2020; Greff et al., 2020).

In the following, we present a theory of human cross-domain generalization. Our primary claim is that human cross-domain generalization is a product of analogical inference performed over structured relational representations of multiple domains. We instantiate our theory in a unification and extension of two existing computational models of relational reasoning, Learning and Inference with Schemas and Analogy (LISA; Hummel & Holyoak, 1997, 2003) and Discovery of Relations by Analogy (DORA; Doumas et al., 2008). The resulting model accounts for how we acquire structured relational representations from simple visual inputs, integrates with current methods for reinforcement learning (RL) to learn how to apply these representations, and accounts for how we leverage these representations to generalize knowledge across different domains.

In what follows, we first review evidence for the role of structured relational representations in generalization, broadly defined, but especially in cross-domain generalization. Along the way, we discuss what it means for a representation to be structured and relational, contrasting the strengths and limitations of implicit and explicit representations of relations. Next, we outline our theory of human cross-domain generalization and describe the computational model that instantiates it. We then present a series of simulations demonstrating that the model learns structured representations of relations and uses these representations to perform zero-shot (i.e., first trial) cross-domain generalization between different video games and between completely different tasks (video games and analogy making). We also show that the model’s learning trajectory closely mirrors the developmental trajectory of human children. Finally, we discuss the implications for our account, contrasting it with purely statistical learning accounts such as DNNS, and we consider future extensions of the theory.

In addition to providing an account of cross-domain transfer, the model also represents theoretical advance in at least two other domains, (a) the discovery of the kind of invariant semantic properties that constitute the meaning of a relation and (b) the integration of explicitly relational representations with RL to produce representations with the expressive power of explicitly relational representations and the productive capacity of implicit representations of relations (e.g., as weight matrices).

### Relational Representations and Generalization

Psychologists have long observed that people’s concepts, even of “ordinary” domains like cooking and visual occlusion, are like theories or models of the domains in question (e.g., Carey, 2000, 2009; Murphy & Medin, 1985), in that they specify the relations among the critical variables in the domain. For example, our understanding of visual occlusion specifies that a larger surface can occlude a smaller surface more than vice-versa (one can hide behind a hundred-year-old oak better than a sapling; e.g., Hespos & Baillargeon, 2001); that an object continues to exist, even when it’s hidden behind an occluder (e.g., Piaget, 1954); and that the ability of an occluder to hide an object depends on the relative distances between the occluder, the observer, and the hidden object. Our model of biology tells us that the offspring are the same species as the parents (e.g., Gelman, 2003). Our understanding of cooking specifies that the amount of salt one adds to a dish should be proportional to the size of the dish. And our model of a game like tennis, baseball, or Pong tells us that the ability of the racket, bat, or paddle to hit the ball depends on the locations and trajectories of these objects relative to one another.

The critical property of all these models is that they specify—and thus depend on the capacity to learn and represent—an open-ended vocabulary of relations among variables: Whether $x$ can occlude $y$ from viewer $v$ depends on the relative sizes of $x$ and $y$, and the relative distances and angles between $x$, $y$, and $v$. Accordingly, learning a model of a domain entails learning a representation of the relations characterizing that domain. We take this claim—that an “understanding” of a domain consists of a representation of the relations characterizing that domain—to be uncontroversial. However, our claim is stronger than that. We claim that an understanding of a domain consists of structured relational representations of the domain. As elaborated shortly, by structured relational representation, we mean a representational format that explicitly captures both the semantic content (i.e., meaning) and the compositional structure (i.e., bindings of arguments to relational roles) of a relationship.

One of the most important manifestations of a capacity to reason about relations is analogical inference: inferences based on the relations in which objects are engaged, rather than just the literal features of the objects themselves. Analogical inference is evident in almost every circumstance in which a person demonstrates knowledge on which she was never explicitly trained. The physics student learning Newton’s law, $f = ma$, does not need to learn multiplication de novo in the context of this new equation, she knows it already. So instead of teaching her an enormous lookup table with all possible $m$ and $a$ as input and the corresponding $f$ as output, the physics teacher simply gives her the equation, knowing her knowledge of multiplication will generalize to the domain of physics. The student who knows she needs at least a grade of 70% to pass a course also knows, without additional training, that 69%, 68%, 67%, etc., are all failing grades, an inference based on the relation between 70 and all the numbers smaller than it. And if you have a meeting at 2:00 p.m., and the current time is 1:00 p.m, then you know you are not yet late. Importantly, we know these things without explicit training on each domain individually. An understanding of a relation
such as less-than or multiplied-by simultaneously confers understanding to all domains in which it applies (e.g., grades, appointments, recipes, automobile manufacture, etc.). In brief, cross-domain generalization based on structured relational representations is not the exception in human thinking, it is the default.

Learning Relational Representations

These facts have not escaped the notice of cognitive modelers (e.g., Doumas et al., 2008; Falkenhainer et al., 1989; Halford, Wilson, & Phillips, 1998; Hummel & Biederman, 1992; Hummel & Holyoak, 1997, 2003; Paccanaro & Hinton, 2001), and recent years have seen increased interest in getting neural networks trained by back propagation to learn useful representations of relations in domains such as relation extraction from pictures (Halderek et al., 2017), visual question answering (e.g., Cadene et al., 2019; Ma et al., 2018; Santoro et al., 2017; Xu & Saenko, 2016), same-different discrimination (Funk et al., 2021; Messina et al., 2021), and even visual (Hill et al., 2018; Hoshen & Werman, 2017) and verbal (Mikolov et al., 2013) analogy-making. The core assumptions underlying these efforts are that (a) the relevant relational properties can be discovered as a natural consequence of the statistical properties of the input–output mappings and that (b) the relevant relations will be represented in the learned weight matrices and will permit relational generalization.

This statistical approach to relation learning has met with some substantial successes. One strength of this approach is that because the learned relations are represented implicitly in the networks’ weight matrices, they are functional in the sense that they directly impact the model’s behavior: Given one term of a relation, for instance, along with a weight matrix representing a relation, a network of this kind can produce the other term of the relation as an output (see, e.g., Leech et al., 2008; Lu et al., 2012; but cf. Lu et al., 2021). By contrast, models based on more explicit representations of relations (e.g., Anderson, 2007; Doumas et al., 2008; Falkenhainer et al., 1989; Hummel & Holyoak, 1997, 2003), including the model presented here, must explicitly decide how to apply the relations it knows to the task at hand (e.g., by adding an inferred proposition to a database of known facts; see Anderson, 2007). As elaborated in the Simulations section, one advance presented in this article is a technique for using RL to choose which relations to use in what circumstances in the context of video game play.

Although statistical learning of implicit relations achieves impressive results when tested on examples that lie within the training distribution, their performance deteriorates sharply on out-of-distribution samples. For example, the relational network of Santoro et al. (2017) was trained to answer same-different questions about pairs of objects in an image. When the model was tested on shape-color combinations withheld from the training data (e.g., a test image with two identical cyan squares where the model had seen squares and the color cyan individually but not on combination), its performance dropped to chance (Kim et al., 2018). The limited applicability of the relations learned by these models holds across application domains (see, Ricci et al., 2021; Peterson et al., 2020; Sengupta & Friston, 2018; Stabinger et al., 2020, for recent reviews).

Why are useful relational representations so hard to learn using traditional statistical learning? The short answer is that although these approaches might capture the content of (some of) the relevant relations in their respective domains, they do not represent those relations in a form that supports broad relational generalization. We argue that flexible, cross-domain generalization relies, not on implicit representations (like weight matrices), but instead on explicitly relational representations, that simultaneously (a) represent the semantic content of a relation (e.g., the meaning of the relation left-of () and (b) represent that content in a format that makes it possible to dynamically bind the relation to arguments without altering the representation of either. The following expands on the distinction between form and content with the goal of clarifying our claims about the nature and utility of structured relational representations for the purposes of generalization. We return to this issue in much more detail in the Discussion, where we relate our simulation results to the differences between structured relational representations, as endorsed in the current theory, and implicit representations of relations, as represented in the weight matrices of some neural networks.

Relational Content

What does it mean to represent a relation such as left-of()? To a first approximation, it means having a unit (or pattern of activation over multiple units) that becomes active if and only if some part of the network’s input is to the left-of some other part. For example, imagine a neural network that learns to activate a node in its output layer or in some hidden layer if and only if the network’s input contains at least two objects, i and j, whose locations in the horizontal dimension of the display are unequal. Note that in order to represent left-of(), per se, as opposed to instances of left-of() at a specific location in the visual field, the unit or pattern must become active whenever any object i is left of any other object j, regardless of the specific objects in question, and regardless of their specific locations in the visual field. That is, a representation of a relation, such as left-of(i, j), is useful precisely to the extent that it is invariant with the specific conditions (e.g., the particular retinal coordinates, x_i and x_j) giving rise to it and with the objects (arguments, i and j) bound to it (Hummel & Biederman, 1992): Such a unit or pattern would, by virtue of its 1:1 correspondence with the presence of things that are left-of other things in the input, represent the semantic content of the relation left-of() .

The invariance of a relational representation is partly responsible for the flexible generalization afforded by such representations. A system that can represent left-of (paddle, ball) in a form that is invariant with the specific locations of the paddle and ball is well prepared to learn a rule such as “if the paddle is left of the ball, then move the paddle to the right” and then apply that rule regardless of the location of the paddle and ball in the game display. Such a representation would even permit generalization to a screen wider than the one used during training (i.e., with previously unseen values of x). Conversely, representing left-of (paddle, ball) with different units depending on where the paddle and ball are in the display will not permit generalization across locations: Having learned what to do when the paddle is at x = 10 and the ball is at x = 11 (represented by a unit we’ll call left-of(x10, 11)), such a network would not know what to do when the paddle is at x = 12 and the ball is at x = 15 (represented by a different unit, left-of(x12, 15)).

The Form of Structured Relational Representations

Being able to represent relational invariants such as left-of () and above () is extremely useful, if not necessary, for broad, cross-domain
generalization, but it is not sufficient. Simply activating a unit or pattern representing left-of () does not specify what is to the left of what: Is the paddle left of the ball, or the ball left of the paddle? Or is one of them, or some other object, to the left of some third or fourth object? Knowing only that something is left of something else provides no basis for deciding whether, for example, to move the paddle to the left or right.

Representing a relation such as left-of (i, j) in a way that can effectively guide reasoning or behavior entails representing both the relational content of the relation (e.g., that something is left-of something else, as opposed to, say, larger-than something else), and specifying that content in a format that makes the bindings of arguments to relational roles explicit. Following the literature on analogy and relational reasoning, we will use the term predicate to refer to a representation in this format. For our current purposes, a predicate is a representation (a symbol) that can be bound dynamically to its argument(s) in a way that preserves the invariance of both (see, e.g., Halford, Wilson, & Phillips, 1998; Hummel & Biederman, 1992; Hummel & Holyoak, 1997, 2003). By “dynamic binding” we mean a binding that can be created and destroyed as needed. To represent left-of (paddle, ball), the units representing the paddle must be bound to the units representing the first role of left-of () while the units representing the ball are bound to the units representing the second role; and to represent left-of (ball, paddle), the very same units must be rebound so that ball is bound to the first role and paddle to the second role. In propositional notation, these bindings are represented by the order of the arguments inside the parentheses. Neural networks need a different way to signal these bindings, and the work reported here follows Doumas et al. (2008; see also Hummel & Biederman, 1992; Hummel & Holyoak, 1997, 2003) and others in using systematic synchrony and asynchrony of firing for this purpose.

What matters about dynamic binding is not that it is based on synchrony or asynchrony of firing; one might imagine other ways to signal bindings dynamically in a neural network. What matters is only that the dynamic binding tag, whatever it is, is independent of the units it binds together. That is, the binding tag must be a second degree of freedom, independent of the units’ activations, with which to represent how those units (representing roles and objects) are bound together. Synchrony of firing happens to be convenient for this purpose, as well as neurally plausible (e.g., Hummel & Biederman, 1992; Hummel & Holyoak, 1997, 2003; Reichert & Serre, 2014; Rao & Cecchi, 2010, 2011; Shastri & Ajanagadde, 1993; von der Malsburg, 1982/1994). For example, to represent left-of (paddle, ball), neurons representing left-of would fire in synchrony with neurons representing the paddle, while neurons representing right-of fire in asynchrony with neurons representing the ball (and out of synchrony with the paddle and left-of neurons). The very same neurons would also represent left-of (ball, paddle), but the synchrony relations would be reversed. In this way, the representation captures the form of the representation (distinguishing left-of (ball, paddle) from left-of (paddle, ball)), without sacrificing the content of left-of (), ball, or paddle. This ability to bind ball and paddle dynamically to the roles of left-of () without changing the representation of either derives from the fact that when a unit fires, it is independent of how strongly it fires: The representation is explicitly structured and relational because timing (which carries binding) and activation (which carries content) are independent.

We posit that representing a relation as a structure that is invariant with its arguments and can be dynamically bound to arguments permits immediate (i.e., zero-shot) generalization to completely novel arguments—including arguments never seen during training. For example, a Breakout player who represents left-of () in a way that remains unaffected by the arguments of the relation could adapt rapidly if the paddle and ball were suddenly replaced by, say, a triangle and a square, or a net and a bunny. Armed with the capacity to map a given relation such as left-of () in one domain onto a different relation such as above () in another—that is, armed with a capacity for analogical mapping—a player could also rapidly adapt “keep the paddle aligned with the ball in the horizontal direction,” as in Breakout, to “keep the paddle aligned with the ball in the vertical direction,” as in Pong. Such a player would exhibit very rapid cross-domain generalization from Breakout to Pong.

In summary, representing relations explicitly, with a pattern or unit that remains invariant over different instantiations of the relation (e.g., left-of () in one location vs. another) and different role bindings (e.g., left-of (paddle, ball) vs. left-of (triangle, square)), and that can be dynamically bound to arguments, affords enormous flexibility in generalization. We argue that it is precisely this kind of relational generalization that gives rise to cross-domain transfer.

Learning Relations

An account of how people learn explicitly relational representations must explain how we learn both their content and their form (Doumas et al., 2008). To account for the discovery of relational content, it must specify how we come to detect the basic relational invariants that remain constant across instances of the relation. For example, how can we discover an invariant that holds true across all instances of left-of (), given that we only ever observe specific instances of left-of () at specific locations? To account for the learning of the form of a structured relational representation—that is, the capacity to bind relational roles to their arguments dynamically without destroying the invariant representations of either—the system must be able to solve two additional problems. First, it must be able to isolate the relevant invariants from the other properties of the objects engaged in the relation to be learned. Part of what makes relation learning difficult is that although a goal is to discover an invariant representation of the relation, during acquisition relations never occur in isolation, but always in the context of specific objects engaged in the relation (e.g., it is impossible to observe a disembodied example of left-of () as an abstract invariant). Second, having discovered and isolated the relevant invariants, the system must learn a structured predicate representation of the relation that can be stored in long-term memory and can be dynamically bound to arbitrary arguments while remaining independent of those arguments.

A Theory of Cross-Domain Generalization

We propose that human cross-domain generalization is a special case of analogical inference over explicitly relational representations. Accordingly, we propose that cross-domain generalization is subject to the constraints on relation learning summarized previously, plus the familiar constraints on analogical reasoning (see Holyoak & Thagard, 1995; Hummel & Holyoak, 1997, 2003). Specifically, we propose that cross-domain generalization is a consequence of four fundamental operations: (a) detecting (or learning to detect) relational invariants; (b) learning structured (i.e., predicate) representations of those invariants; (c) using these structured relational representations to
construct relational models of various domains (including arithmetic and physics, or video games) procedurally via processes like RL; and (d) using those representations to understand new domains by analogy to familiar ones.

We do not propose that all four of these operations take place de novo every time a person generalizes from one domain to another. In particular, if a given domain is familiar to a person, then they will already have performed steps (1) . . . (3) with respect to that domain. Moreover, even most novel domains are almost never completely novel. By the time a person learns Newton’s laws of mechanics, for instance, they have already mastered arithmetic, so although the equations themselves are new to the student, the arithmetic operations they represent are not. Although children likely engage actively in all four steps, by adulthood, the majority of cross-domain transfer—whether from arithmetic to physics, or from one game to another—likely relies to an extent on step (3), learning which known relations might be relevant in a given situation, and most heavily on step (4), using existing relational concepts and domain models to make inferences about novel domains that are themselves represented in terms of relations and objects already familiar to the learner.

The following presents our model of cross-domain transfer, which performs all four of the steps outlined above: invariant discovery, relation isolation and predication, model construction, and relational inference based on those models. The model is an integration and augmentation of the LISA model of analogical reasoning (Hummel & Holyoak, 1997, 2003; Knowlton et al., 2012) and the DORA model of relational learning and cognitive development (Doumas & Martin, 2018; Doumas et al., 2008). LISA and DORA account for over 100 major findings in human perception and cognition, spanning at least seven domains: (a) shape perception and object recognition (Doumas & Hummel, 2010; Hummel, 2001; Hummel & Biederman, 1992); (b) relational thinking (Choplin & Hummel, 2002; Hummel & Holyoak, 1997, 2003; Hummel et al., 2014; Krawczyk et al., 2004, 2005; Kroger et al., 2004; Kubose et al., 2002; Taylor & Hummel, 2009), (c) relation learning (Doumas & Hummel, 2012; Doumas et al., 2008; Jung & Hummel, 2015a, 2015b; Livins et al., 2015, 2016), (d) cognitive development (Doumas et al., 2008; Licato et al., 2012; Lim et al., 2013; Sandhofer & Doumas, 2008), (e) language processing (Doumas & Martin, 2018; Martin & Doumas, 2017, 2020; Rabaglia et al., 2017), (f) cognitive aging (Viskontas et al., 2004), and (g) decline due to dementia, stress, and brain damage (Morrison et al., 2004, 2011). Accordingly, we view these systems as a promising starting point for an account of human-level cross-domain generalization. Importantly, LISA provides a solution to Problem 4 above (the problem of inference), and DORA provides a solution to Problem 2 (learning structured representations from nonstructured inputs). The current model integrates LISA and DORA into a single framework, and then extends the resulting model to address Problem 1 (the discovery of abstract relational invariants) and Problem 3 (model construction via RL).

Our core theoretical claims and their instantiation in the proposed model are summarized in Table 1. These claims along with the core claims of LISA/DORA (see Doumas et al., 2008; Hummel & Holyoak, 1997, 2003) compose the primary assumptions of the approach. We claim that structured relational representations underlie cross-domain generalization as a natural consequence of (a) their general applicability across domains and (b) their ability to underlie analogies between domains. Cross-domain analogical inference occurs because we build models of domains consisting of an open-ended vocabulary of relations among the elements the domains (see also Murphy & Medin, 1985; Carey, 2009; Gentner, 2003). These representations are structured and relational in that they express the invariant content of the relations they specify in a structured (i.e., symbolic) format that dynamically binds arguments to relational roles. We learn both the content and the format of structured relational representations from experience by explicitly comparing examples. We learn which relations are important for characterizing and acting in a domain through a process of RL.

The remainder of the paper proceeds as follows. First, we summarize our integration of the LISA/DORA frameworks and describe the current extensions for invariant discovery and

### Table 1

| Core theoretical claim | Instantiation in DORA |
|------------------------|-----------------------|
| 1. Cross-domain generalization is a natural consequence of structured relational representations, which express the invariant content of relations and compose dynamic role-argument bindings into propositions. | DORA represents relations between elements in a format that makes explicit the relational invariants, the bindings of relational roles to arguments, and the integration of multiple bindings into propositions. DORA’s capacity for cross-domain generalization follows from operations over these representations. |
| 1a. Structured relational representations support cross-domain generalization because they are generally applicable across domains. | DORA’s representations can be applied promiscuously to characterize genuinely new situations. DORA uses the representations that it has learned in the past to represent novel situations by dynamically binding previously learned predicate representations to objects in the new situations. |
| 1b. Cross-domain generalization is a case of analogical inference over domain models. | DORA performs analogical mapping, discovering correspondences between situations based on shared relational structure, and generalizes across situations based on these mappings. |
| 2. Structured relational representations are acquired by a comparison process that reveals and isolates relational invariants and composes them into predicates that dynamically bind to arguments. | DORA discovers invariant relational properties by exploiting properties inherent to rate-coded representations of magnitude. DORA then learns structured relational representations of these properties through a process of comparison-based intersection discovery and Hebbian learning. |
| 3. Relations relevant for characterizing and acting in a domain are learned procedurally through reinforcement learning. | DORA and its representations integrate smoothly with existing methods for reinforcement learning. |

*Note. DORA = Discovery of Relations by Analogy.*
The Model

As noted previously, our model of cross-domain generalization is based on integration and augmentation of the LISA and DORA models of relational reasoning (henceforth, simply DORA). We begin by reviewing how DORA represents relational knowledge, how it uses those representations for reasoning, and how it learns structured representations of relations from unstructured vector-based inputs.

We next present a novel algorithm for discovering invariant relational properties—that is, the semantic content of relational representations—from simple, nonrelational visual inputs. The resulting model provides the first complete account of how structured representations of visual relations can be learned de novo from simple nonrelational inputs without feedback and without assuming a vocabulary of relations a priori. The resulting model also provides an account of human cross-domain generalization as a natural consequence.

The following description of the model presents published details of DORA’s operation only in broad strokes, going into detail only when those details are relevant to understanding the novel extensions of the work (e.g., relation discovery). Complete details of the model’s operation can be found in Appendix A (which includes a functional description of the model, pseudocode, and full computational details). The model’s source code is available online.¹

Representing Relational Knowledge: LISAese

We begin by describing the final (i.e., postlearning) state of DORA’s knowledge representations. These representations do not serve as the input to the model but are the result of its learning (as described below). DORA is a neural network consisting of four layers of bidirectionally connected units. DORA represents propositions using a format, LISAese, which is a hierarchy of distributed and progressively more localist units whose activations oscillate over progressively slower time scales (moving from the bottom to the top layer of the network; Figure 1). At the bottom of the hierarchy, feature units represent the basic features of objects and relational roles in a distributed manner. Token units (T1–T3) learn without supervision (see below) to conjunctively code collections of units from the layer below. Tokens at the lowest level of the hierarchy (T1) take inputs directly from feature units and learn to respond to objects or relational roles in a localist fashion. Tokens in the next layer (T2) take their inputs from PO tokens and learn to respond to pairs of PO units—that is, to roles and the objects (arguments) to which they are bound. Tokens in the highest layer (T3) learn to respond to collections T2 units, instantiating multiplicity relational propositions.

When they become active, units representing relational roles and their arguments independently (i.e., features and T1 units) must be bound together dynamically. DORA represents dynamic bindings using time: T1 units representing relational roles and their arguments fire out of synchrony but in close temporal proximity. These temporal relations arise from inhibitory interactions among token units. Within a single proposition, token units are laterally inhibitory (e.g., T1 units inhibit other T1 units, T2 units inhibit other T2 units; as elaborated below, lateral inhibition between token units extends beyond propositions but for the present illustration this simplification is appropriate). Each token unit is also yoked to an inhibitory unit, which causes the token’s activation to oscillate, even in response to a fixed excitatory input.

Together, the yoked inhibitors and lateral inhibitory interactions between tokens cause the constituents of a single proposition to fire in a systematic, hierarchical temporal pattern (Figure 1b and 1c). When a T3 unit representing a proposition such as above (ball, paddle) becomes active, it will excite the T2 units below itself (here, above + ball and below + paddle). These T2 units inhibit one another, and one of them (e.g., above + ball) will randomly win the initial inhibitory competition, becoming active and driving the other to inactivity. After a fixed number of iterations/milliseconds, k, the inhibitor yoked to that T2 unit (above + ball) will become active, driving it to inactivity, and allowing the other T2 unit (below + paddle) to become active for k iterations/milliseconds (until its own inhibitor drives it to inactivity).

This same pattern of lateral inhibition and yoked inhibitor activity causes the T1 units below above + ball and below + paddle—namely, the T1 units above and ball, and below and paddle, respectively—to oscillate out of phase with one another at twice the frequency (i.e., half the duration; specifically, k/2) at which the T2 units are oscillating. T1 units activate feature units, causing the semantic units’ activity to oscillate at the same frequency as the T1 units. The result on the T1 and semantic units is a repeating temporal pattern of the form [(above, ball), (below, paddle)], [(above, ball), (below, paddle)], etc., where units inside parentheses are oscillating out of phase with one another with duration k/2, and units inside brackets are out of phase with duration k (see Figure 1b and 1c). In this way, the network represents relational roles and fillers independently (in the features and T1 units) and simultaneously represents the binding of roles to fillers.

Interpreting LISAese

One advantage of the representations DORA learns is that they are easily interpretable. Units in T1 will learn to represent objects and relational roles, and by inspecting the features to which any given T1 unit is connected, it is possible to determine which object

¹ Source code is available online; see Author Note.
or role it represents. Units in T2 will learn to represent specific predicate–argument bindings, which are interpretable by inspecting the T1 units to which they are connected. And units in T3 will learn to represent complete propositions, which are interpretable by inspecting the connected T2 units. Accordingly, in the following, we will refer to the units DORA learns in terms of these interpretations. We do so solely for clarity of exposition. The labels we use have no meaning to DORA and no impact on its operation.

Computational Macrostructure

Figure 1 depicts the representation of an individual proposition in DORA’s working memory (as synchronized and desynchronized patterns of activation). Figure 2 provides an overview of DORA’s macrostructure.

A complete analog—situation, story, schema, etc.—consists of the collection of token units that collectively encode its propositional content. Within an analog, a single token represents a given object, role, role-binding, or proposition, no matter how many propositions refer to that entity. For example, in a single analog, the same T1 unit for left-of represents that role in all propositions containing left-of as a role. However, separate analogs do not share tokens. For example, one unit would represent left-of in DORA’s representation of the game it is currently playing (represented in one analog) and a completely separate token would represent left-of in DORA’s representation of a game it had played in the past (represented in a separate analog). Collections of token units (i.e., T1…T3) representing the situations and schemas (i.e., “analogs”) DORA knows collectively form its long-term memory (LTM; Figure 2a).

For the purposes of learning and reasoning—for example, when making an analogy between one situation and another—the propositions representing those analogs enter active memory (dashed box in Figure 2a), a state in which they are readily accessible for processing, but not fully active (see, e.g., Cowan, 2001; Hummel & Holyoak, 1997, 2003). As depicted in Figure 2b, the analogs in active memory are divided into independent sets: The driver corresponds to the current focus of attention (e.g., the state of the current game, as delivered by perceptual processing), and maps onto one or more recipients (e.g., an analog describing the model’s emerging understanding of the game).2 Token units are laterally inhibitory within but not between sets. The driver/recipient distinction in DORA is different from the more familiar source/target distinction discussed in the analogical reasoning literature. The target of analogical reasoning is the novel problem to be solved (or situation to be reasoned about), and the source is the analog from which inferences are drawn about the target. As summarized shortly, in DORA, the target analog tends to serve as the driver (i.e., the focus of attention)

2 The idea that mutually exclusive sets are fundamental for analogical reasoning goes back to Gentner (1983), and has been implemented in a variety of models (SME, LISA, etc.). As detailed in Knowlton et al. (2012), we assume these sets are implemented by neurons in posterior frontal cortex with rapidly modifiable synapses that act as “proxies” for larger structures represented elsewhere in cortex.
during memory retrieval and in the initial stages of analogical mapping, whereas the source serves as the driver during analogical inference.

**Operation**

In DORA activation generally starts in the driver, passing through the feature units (and any mapping connections, as detailed shortly) into other analogs in LTM (for memory retrieval), including any analogs in the recipient (for mapping, learning, and inference). Token units in the driver compete via lateral inhibition to become active (i.e., token units in the driver laterally inhibit other tokens in the same layer), generating patterns of activation on the feature units (as described previously; Figure 1). Units in the recipient (or LTM) compete via lateral inhibition to respond to the resulting patterns on the feature units. This inhibitory competition is hierarchical in time, reflecting the temporal dynamics of the driver and features: T1 units (relational roles and objects) in LTM/recipient compete to respond to patterns generated by individual roles and objects in the driver; T2 units (role bindings) in LTM/recipient compete to respond to specific role/filler bindings; and T3 units compete to respond to complete propositions. The result is a winner-take-all inhibitory competition operating at multiple temporal scales and serves as the foundation of all the functions DORA performs, including memory retrieval, analogical mapping (Hummel & Holyoak, 1997), analogical inference (Hummel & Holyoak, 2003), and relation discovery (Doumas et al., 2008).

**Memory Retrieval**

Patterns of activation imposed on the feature units by active tokens in the driver will tend to activate token units in LTM that have learned to respond to similar patterns (Appendix A—Retrieval section, for details). For example, the features activated by a paddle in the driver will tend to activate T1 units responsive to paddle features, and the features activated by leftmost in the driver will tend to activate T1 units connected to leftmost features. Together, these T1 units will tend to excite T2 units for leftmost + paddle. Features consistent with ball and rightmost would likewise activate T1 units for ball and rightmost, which would excite a T2 unit for rightmost + ball. Together, the T2 units for leftmost + paddle and rightmost + ball will tend to activate any T3 unit(s) encoding the proposition left-of (paddle, ball): The model will have recognized the desynchronized patterns of features as representing the fact that the paddle is left-of the ball, which can be retrieved into the recipient.

**Mapping**

One of the most important operations DORA performs is analogical mapping. During mapping, DORA discovers structural correspondences between tokens in the driver and recipient. When tokens in the driver become active, similar tokens are activated in the recipient via the shared feature units. Using a kind of Hebbian learning, the model learns mapping connections between co-active units in the same layer across driver and recipient (Hummel & Holyoak, 1997, 2003; see Appendix A—Update Mapping Hypotheses and Update Mapping Connections sections for details). The resulting connections serve both to represent the mappings DORA has already discovered, and to constrain its discovery of additional mappings. The algorithm provides an excellent account of human analogical mapping performance (Doumas et al., 2008; Hummel & Holyoak, 1997, 2003).

**Analogical Inference**

Augmented with a simple algorithm for self-supervised learning (Hummel & Holyoak, 2003), DORA’s mapping algorithm also provides a psychologically and neurally realistic account of analogical inference (making relational inferences about one situation based on knowledge of an analogous one; Appendix A—Relational Generalization section, for details). The algorithm implements a version of Holyoak et al.’s (1994) copy-with-substitution-and-generalization (CWSG) framework. In CWSG, when two situations are analogically mapped, information about one situation can be inferred about the other. For example, if one knows about situation-1, where chase (Fido, Rosie), and scared (Rosie) are true, and...
maps that onto situation-2, where chase (Spot, Bowser) is true, one can copy the representation of the scared predicate from situation-1 to situation-2, and then use the mapping of Bowser to Rosie, to copy Bowser as the argument of scared to infer scared (Bowser). As elaborated below, this process serves as the basis for our proposed solution to the problem of cross-domain generalization.

**Learning Relational Format: LISAese From Nonstructured Inputs**

DORA generalizes the operations described above to address the problem of learning structured representations of relations from unstructured “flat” vector/feature-based representations (Doumas et al., 2008). DORA represents relations as collections of linked roles, rather than as monolithic structures: For example, the relation above composes the roles higher and lower rather than consisting strictly of the single atom above, as it would in propositional notation or a labeled graph. This role-based approach to representing relations offers several advantages over alternative approaches (see Doumas & Hummel, 2005; for a review), one of which is that it makes it possible to learn relations by (a) first learning their roles and then (b) linking those roles together into multi-argument relational structures (as described in Doumas et al., 2008).

DORA’s unsupervised relation learning algorithm (Doumas et al., 2008) begins with representations of objects encoded as vectors of features. DORA learns single-place predicates—that is, individual relational roles—as follows (see Appendix A—Predicate and Relation Learning section, for details): (a) By the process of analogical mapping (summarized above) the model maps objects in one situation (the driver; e.g., a previous state of the game of Breakout) onto objects in a similar known situation (the recipient; e.g., the current state of a game). For example, DORA might map a T1 unit representing the ball in its previous location onto a T1 unit representing the paddle in its current location (Figure 3ai). Early in learning, these tokens will be holistic feature-based representations specifying an object’s attributes (e.g., its location, color, etc.) in a single vector; (b) As a result of this mapping connection, the T1 unit representing the ball will become co-active with the T1 unit representing the paddle in the recipient. T1 units in both the driver and recipient pass activation to the feature units to which they are connected, so any features connected to the active T1 units in both the driver and recipient will receive about twice as much input—and therefore become about twice as active—as any features unique to one or the other (Figure 3aii). As a result, the intersection of the two instances becomes highlighted as the collection of the most active features; (c) DORA recruits (activates) a T1 unit and a T2 unit in the recipient, and updates connections between units in adjacent layers via Hebbian learning (Figure 3aiii). Consequently, the recruited T1 unit will learn connections to active features in proportion to their activation, thereby encoding the shared features (the intersection) of the mapped objects. If the compared objects are both, say, higher than something—and so have the features of higher in their vectors—then DORA will learn an explicit representation of the features corresponding to being higher (the next section describes how such relational features can be learned from absolute location information delivered by the perceptual system); and (d) The resulting representation (Figure 3av) can now function as single-place predicate (i.e., relational role), which can be bound to new arguments (i.e., other units in T1) by asynchrony of firing (see Figure 1). Applied iteratively, this kind of learning produces progressively more refined single-place predicates (Doumas et al., 2008).

The same Hebbian learning algorithm links token units into complete propositions by allowing tokens units in successive layers to integrate their inputs over progressively longer temporal intervals (Doumas et al., 2008; see Appendix A—Predicate and Relation Learning section, for details). The algorithm exploits the fact that objects playing complementary roles of a single relation will tend to co-occur in the environment. For example, the representation of a ball that is higher than something (i.e., higher-than-something (ball)) will systematically co-occur with another object (e.g., the paddle), which is lower than something (e.g., lower-than-something (paddle); Figure 3bi). When two sets of co-occurring role-argument pairs are mapped (e.g., an instance where ball is higher-than-something and a paddle is lower-than-something is mapped to an instance where a paddle is higher-than-something and a ball is lower-than-something; Figure 3bi), a diagnostic pattern of firing emerges: (a) The T2 units coding each predicate–argument binding will oscillate systematically across both driver and recipient (Figure 3bii–iii); (b) In response, DORA recruits a unit in T3 that learns (via Hebbian learning) connections to the T2 units as they become active (Figure 3biii); (c) The resulting representation encodes a multiplace relational structure (equivalent to above (paddle, ball); Figure 3biv). Applied iteratively over many examples, this algorithm learns abstracted structured representations describing a domain in terms of the properties and relations that characterize that domain (Doumas et al., 2008).

Doumas et al. (2008) demonstrated that this algorithm for relation discovery provides a powerful account of numerous phenomena in cognitive development. However, although the Doumas et al. algorithm can learn relational representations with a structured form from repeated exposures to nonstructured inputs, it provides little basis for discovering the invariant content of those relations—that is, the relational features themselves. Instead, their simulations were based primarily on representations of objects with hand-coded invariant features. The next section describes a novel algorithm for discovering invariant relational features from nonrelational inputs.

**Learning Relational Content: Discovering Relational Invariance**

To learn an abstract representation of a relation that remains invariant with the relation’s arguments, there must be at least one invariant that characterizes that relation. For example, to learn a representation of right-of that captures every instance of right-of-ness, there must be a detectable property(ies) that remains constant over all instances of right-of-ness (see, e.g., Biederman, 2013; Hamad, 1990; Hummel & Biederman, 1992; Kellman et al., 1999). Most previous work on relational perception and thinking has tacitly assumed the existence of such invariants (e.g., Anderson, 2007; Doumas et al., 2008; Falkenhainer et al., 1989; Hummel, 2001; Hummel & Biederman, 1992; Hummel & Holyoak, 1997, 2003). But unless all these invariants are assumed to be innate, there must be

---

*In the above example, we describe learning a two-place relation composed of two role-filler pairs for the purposes of brevity. DORA learns relations of arity n by linking n role-filler pairs (e.g., a three-place relation is composed of three role-filler pairs; see Doumas & Hummel, 2012).*
(a) Learning a single-place predicate representation by comparing two objects. (i) A representation of a ball in the driver is mapped (gray double-headed arrow) to a different representation of a paddle (e.g., from a different game screen) in the recipient. (ii) The representation of the ball in the driver activates the mapped unit in the recipient (through shared features and mapping connection); as units pass activation to their features, shared features become more active (dark-gray units) than unshared features (light gray units). (iii) Units in T1 and T2 are recruited (activation clamped to 1; dark-gray units with white squiggle) in the recipient, and weighted connections are learned via Hebbian learning (i.e., stronger connections between more active units). (iv) The result is an explicit representation of the featural overlap of the ball and paddle—in this case, the property of being higher-than-something (see main text)—that can be bound to an argument (as in Figure 1).

(b) Learning a multi-place relational representation by linking a co-occurring set of role-argument pairs. (i) A representation of a ball that is higher-than-something and a paddle that is lower-than-something is mapped to a different representation of a paddle that is higher-than-something, and a paddle that is lower-than-something is mapped to a different representation of a paddle that is higher-than-something, and a
ball that is lower-than-something (e.g., from a different game screen). (ii) (Mapping connections, gray double-headed arrows, have been lightened to make the rest of the figure clearer.) When the representation of higher-than-something (ball) becomes active in the driver it activates mapped units in the recipient; a T3 unit is recruited (activation clamped to 1; dark-gray unit with white squiggle) in the recipient and learns weighted connections to units in T2 via Hebbian learning. (iii) When the representation of lower-than-something (paddle) becomes active in the driver, it activates corresponding mapped units in the recipient; the active T3 unit learns weighted connections to T2 units. (iv) The result of learning is a LISAsese representation of the relational proposition above (ball, paddle) (see Figure 1). Labels in units indicate what the unit encodes (see key). The labels on the units are provided for clarity and are meaningless to DORA. DORA = Discovery of Relations by Analogy.

Part of what makes invariant discovery difficult is that it poses a kind of chicken and egg problem: An invariant only seems to be discoverable in a noninvariant input if one knows to look for that invariant in the first place. Consider an invariant like “square.” Of all the possible arrangements of pixels on a computer screen, some of them form squares and others do not. Whether a set of pixels forms a square does not depend on the color of the pixels, the color of the background, the locations of the pixels on the screen, or their distances from one another: Provided they are arranged relative to one another in a way that forms a square, then they satisfy the invariant “square.” “Square” is a higher-order relational property that is independent of—that is, invariant with—the properties of any of the pixels composing it. Making matters more complicated, “square” is only one of an infinity of such higher-order relational invariants one could find in visual images. Others include rhombi, various triangles, and countless random-looking clouds of points. All these configurations are defined by the spatial relations among sets of points, so any one of them could, in principle, become a perceptual invariant like “square.” But not all of them do. Why do we recognize “square” as an invariant, but not any of the nearly infinite random-looking (but nonetheless invariant) clouds of points?

It is not our intent to fully answer to this question here, but one constraint that suggests itself is to start simple: Perhaps “square” is not itself a basic (“primitive”) invariant in the human cognitive architecture but is instead (at least initially) a composition of several simpler invariants (e.g., straight lines, equal lengths, right angles, and such) arranged in particular relations to one another (as proposed by Biederman, 1987, and many others). According to this account, the cognitive architecture might be biased to find a small number of very basic invariants (things such as equal-to, greater-than, and less-than, among others; Doumas et al., 2008; Kellman et al., 1999), and compose more complex relational invariants, such as above (x, y), right-of (x, y), and square (x), by applying the basic relations to specific perceptual and cognitive dimensions, and to other relations.

Following this intuition, we developed a simple relational invariant discovery circuit (henceforth, simply relational invariance circuit) to discover the invariants greater-than, equal-to, and less-than on any metric dimension, m. This circuit exploits computational properties that naturally emerge whenever magnitudes are rate coded, either in terms of the number of units that fire in response to a given magnitude, or in terms of the rates at which individual neurons fire. The basic idea is that for any magnitude represented as a rate code, computing relations such as greater-than, less-than, and equal-to is a straightforward matter of responding to the difference between two rates. The sign of this difference (+, 0, or −) becomes an invariant signature of the categorical relations greater-than, equal-to, and less-than respectively; and by summing over greater-than and less-than, this same operation yields the invariants same and different with respect to the dimension in question.

Let m be an n-dimensional vector space, for example, a collection of neurons that codes a simple magnitude, m, such as size. The vector a (in m) then represents an object with size a, and b represents size b. Armed with these rate codes, the difference between sizes a and b, Ea,b, is the directional difference (e.g., Gallistel & Gelman, 2000; Zorzi et al., 2005):

\[ E_{a,b} = \sum_i (a_i - b_i), \]

when a is larger than b, Ea,b will be positive; when a is smaller than b, Ea,b will be negative; and when they are equal, Ea,b will be zero.

Using unsupervised learning, it is straightforward to exploit this regularity to train units to respond explicitly to whether any metric values a and b are equal, unequal with a > b, or unequal with a < b. The resulting neurons will be invariant representations of the relations equal, greater-than, and less-than for any rate-coded metric dimension m. In the language of LISAsese, the resulting units could serve as feature units in dimension-specific relations such as right-of, above, larger, etc.\footnote{We have left the arguments a and b out of the relational expressions here because these units represent invariant relational content, but by themselves do not specify how the roles of those relations bind to arguments; that is, they do not specify the relational format. However, as noted above, the problem of learning representations with relational format from representations without this format is already solved in DORA.}

### Circuit Architecture

The circuit begins with a rate-coded representation of a metric dimension, m (e.g., size, or location in the horizontal dimension of the visual field; “Feature units m” in Figure 4a). These units share bidirectional excitatory connections with a collection of T1 units (e.g., role units in DORA), which mutually inhibit one another, and each of which excites a single “proxy” unit. The proxy units in turn excite a collection of four E units (for their relation to Equation 1), which excite feature units outside the set of features representing m (“Feature units non-m” in Figure 4a). All other connections depicted in Figure 4 start with weights of 1.0, with the exception of connections between feature units non-m (henceforth non-m features) and E units, and the connections between feature units non-m and T1 units. The connections to and from feature units change during learning. Initially, the connection weights, \( w_{ij} \), from each E unit, j, to a collection of 10 non-m features, i, are initialized randomly with values between 0 and 1. As detailed below, after learning each E unit is most strongly connected to a different subset of the non-m feature units, and these subsets become representations greater-than,
Figure 4
Relational Invariance Circuit

Note. (a) The relational invariance circuit. (b) (i) Activation flows from clamped feature units encoding a dimension or property to T1 units (dark-gray units indicate more active units). (ii) T1 units compete via lateral inhibition to respond to the active feature units (light gray units indicate less active units). (iii) T1 units activate proxy units, which feed activation to E units. E units pass activation to a subset of feature units non-m and connections between active feature units non-m and T1 units are updated via Hebbian learning. (iv) The active T1 unit is inhibited to inactivity by its inhibitor (black square). (v–vi) The process repeats for the second active T1 unit. T1 = TOKENS at the lowest level of the hierarchy.

less-than, and equal-to (as in Figure 4a). Connections between non-m feature units and T1 units are initially 0.

Circuit Operation

Processing in the relational invariance circuit is assumed to begin after perceptual processing has segmented the image into objects and encoded each object in terms of its various attributes (e.g., size, location in the horizontal and vertical dimensions, etc.). As elaborated under Simulations, DORA is equipped with a simple perceptual preprocessor that accomplishes this segmentation and encoding. The T1 units depicted in Figure 4 correspond to a subset of this encoding (i.e., representing each object’s size). For the purposes of illustration, we shall assume that each T1 unit in Figure 4 represents the size of one object (in a display containing two objects, a and b). (The T1 units in this circuit are otherwise identical to other T1 units in DORA.) For simplicity, we also assume that the pattern of activation on the feature units representing dimension m is the superposition (i.e., sum) of the vectors, m, representing the sizes of the two objects in the display. The learning algorithm does not require the feature inputs, m, to be segmented into separate objects; instead, it is sufficient to encode this information in the connections from m to T1.

Once the objects and their attributes are encoded by the preprocessor, the superimposed vector m = m_s + m_r is clamped on the subset of feature units, m, representing dimension m (Figure 4b). This vector serves as input to the T1 units, which compete via lateral inhibition to become active. The input to T1 unit i is given by:

\[ n_i = \sum_j a_j w_{ij} - \sum_k a_k - L I_i \]  
(2)

where j is feature units connected to T1 unit i, k are other active T1 units (k ≠ i), and LI is the activation of the local inhibitor, a refresh signal given when no active T1 units are active in the driver (as in, Doumas et al., 2008; Horn et al., 1991; Horn & Usher, 1990; Hummel & Holyoak, 1997, 2003; Usher & Niebur, 1996; von der Malsburg & Buhmann, 1992; see Appendix A—Update the Local and Global Inhibitors section, for details). Activation of T1 units is calculated as:

\[ \Delta a_i = \gamma n_i (1.1 - a_i) - \delta a_i, \]  
(3)

where \( \Delta a_i \) is the change in activation of unit i, \( \gamma (= 0.3) \) is a growth parameter, \( n_i \) is the net input to unit i, and \( \delta (= 0.1) \) is a decay parameter. At this juncture, a note about parameter values is warranted: All the standard DORA parameters have the same values as reported in previous papers, and previous work has shown that DORA’s behavior is robust to the values of these parameters (see Doumas et al., 2008). Where we have had to introduce new parameters for the purposes of the relational invariant circuit, their values were set solely for the purposes of having the circuit’s behavior match the main idea expressed in Equation 1. No attempt was made to optimize their values.

Since the vectors m_s and m_r are superimposed on the feature units, the T1 unit with more connections to the feature units (in this case, the unit coding for the larger size), say T1_L, will initially win the inhibitory competition, inhibiting T1_s to inactivity. For this reason, the T1 unit with the larger input vector will always win the initial inhibitory competition (Figure 4b).

Each proxy unit, i, has a connection weight of 1 from T1_L and a weight of 0 from all other T1 units. A proxy unit is a simple binary threshold unit whose activation is given by:

\[ p_i = \begin{cases} 1, & n_i \geq 0.4 \\ 0, & \text{otherwise} \end{cases} \]  
(4)

where \( p_i \) is the activation of proxy unit i.

The input, \( n_i \), to proxy unit i is calculated as:

\[ n_i = \sum_j a_j w_{ij} - p_i, \]  
(5)

where j is an active T1 unit and \( p_i \) is the refraction of unit i. The refraction, p_i, is given:
\[ \rho_i = \frac{1}{1 + te^x}, \]  
where \( x \) is the number of iterations since unit \( i \) last fired, and \( t (= .0000001) \) is a scaler. Proxy unit \( i \) will be active if and only if \( T_1a \) active (i.e., \( a_1 > 0.4 \)) and proxy unit \( i \) has not recently been active.

\( E \) units take their inputs from the proxy units. The connections from proxy units to \( E \) units have temporal delays built into them, so that each \( E \) unit has a Gaussian receptive field in the three-dimensional space formed by the two \( T_1 \) cells’ activations, plus time. Input to \( E \) units is given by Equation 5 such that \( j \) is proxy units, and change in activation is calculated using:

\[ \Delta a_i = \gamma \left( \frac{\alpha_i}{\pi} \right)^{0.5} - a_i - 5 \sum_j a_j - L_i, \]  
where \( \gamma \) is a growth parameter, \( \theta_\varepsilon \) is the threshold on unit in \( E \), \( k = .2, j \) is all other units in \( E \) \( j \neq i \). The circuit contains four \( E \) unit each with a \( \gamma \) of .1 or .3, and a \( \theta_\varepsilon \) of 1 or 2, such that all four combinations of \( \gamma \) and \( \theta_\varepsilon \) values are present in a single \( E \) unit. As a result, some \( E \) cells respond preferentially to proxy units firing early in processing, others respond preferentially to proxy units firing later, and still others respond preferentially to the two proxy units firing at the same time. Like the \( T_1 \) units, \( E \) units laterally inhibit one another to respond in a winner-take-all fashion so that only one \( E \) unit tends to become active in response to any (temporally extended) pattern of activation over the proxy units.

\( E \) units are randomly connected to a collection of feature units that are not part of the vector space \( m \) (Figure 4). Active \( E \) units, \( i, b \) both excite the non-\( m \) feature units, \( j \), and during learning, update their connections to them. Feature units connected to \( E \) units update their input by:

\[ n_i = \sum_j a_j w_{ij}, \]  
where \( i \) and \( j \) are feature units and \( E \) units, respectively. Feature unit activation is updated as:

\[ a_i = \frac{n_i}{\max(n_j)}, \]  
where \( a_i \) is the activation of feature unit \( i \), \( n_i \) is the net input to feature unit \( i \), and \( \max(n_j) \) is the maximum input to any feature unit. There is physiological evidence for divisive normalization in the feline visual system (e.g., Bonds, 1989; Heeger, 1992; Foley, 1994) and psychophysical evidence for divisive normalization in human vision (e.g., Thomas & Olizak, 1997). While the circuit is being learned, connections between units in \( E \) and feature units are updated by the equation:

\[ \Delta w_{ij} = \alpha_i (a_i - w_{ij}) \gamma, \]  
where \( i \) and \( j \) refer to units in \( E \) and feature units, respectively, and \( \gamma \) is the growth parameter.

When the circuit is running, connections between the non-\( m \) feature units and active \( T_1 \) units are updated by Equation 10 (Figure 4biii). As a result, \( T_1a \) (the \( T_1 \) unit we assumed won the initial inhibitory competition) will learn positive connections to whatever feature unit(s) the active \( E \) unit has learned to activate (recall that the connections from \( E \) units to non-\( m \) features are initially random). Importantly, the active non-\( m \) feature units, \( j \), to which \( T_1a \) has learned connections effectively represent greater-than: These features will become active whenever the \( E \) unit that responds to early firing proxy units becomes active. In short, \( T_1a \) has gone from representing a particular value on \( m \) to representing the conjunction of that value (it is still connected to the features in \( m \)) along with the relational invariant greater-than.

Recall that the \( T_1 \) units are oscillators (see also Appendix A). As a result, after some iterations have transpired with \( T_1a \) active, that unit’s inhibitor will become active, inhibiting \( T_1b \) and allowing \( T_1a \) to become active (Figure 4biv). The same operations described with respect to \( T_1a \) will take place with respect to \( T_1b \) which will learn connection(s) to feature unit(s) representing less-than (those non-\( m \) features that are strongly connected to the \( E \) unit that responds later in firing; Figure 4biv–vi).

After these operations have taken place on a single pair of objects, DORA will have constructed a representation of that pair of objects, each with a specific value on metric dimension \( m \) (represented by \( T_1a \) and \( T_1b \) in Figure 4) explicitly tagged with an invariant specifying that its value is either greater-than (in the case of \( T_1a \) or less-than (\( T_1b \)) some other value on \( m \). Importantly, these representations do not yet constitute an explicit representation of the relation greater-than (a, b), because they are not linked into a propositional structure (e.g., by \( T_2 \) and \( T_3 \) units) specifying that the individual roles, greater-than and less-than, are linked into a single relation. Moreover, the emerging representation of these roles, as instantiated in the feature units connected to \( T_1a \) and \( T_1b \), still retain a full representation of the specific metric values of \( T_1a \) and \( T_1b \) on \( m \). In other words, \( T_1a \) and \( T_1b \) do not represent greater-than and less-than in the abstract, but instead represent something closer to greater-than-and-value-a and less-than-and-value-b. However, this kind of almost-relational representation, in which relational invariants are present, but (a) are still associated with other, nonrelational features (e.g., specific values on \( m \)) and (b) are not yet composed into an explicitly relational structure, are precisely the kind of representations DORA uses as the starting point for learning explicitly structured relations (see above).

To illustrate, consider what will happen when a different pair of objects, \( c \) and \( d \), are engaged in the process described above. For the purposes of illustration, assume that \( c \) has a larger value on \( m \) than \( d \) does, but both have different values than \( a \) and \( b \). The processes described previously will attach \( T_1a \) to the same invariant greater-than feature(s) as \( T_1a \) and \( T_1b \) to the same less-than feature(s) as \( T_1b \). It is in this sense that those feature units represent greater-than and less-than, respectively. The relational invariance circuit will, by virtue of the \( E \) unit-to-feature connections learned in the context of objects \( a \) and \( b \), be biased to activate the feature unit(s) recruited for greater-than in the context of \( T_1a \) in response to the “greater-than-ness” of \( T_1a \), and the feature unit(s) recruited for less-than in response to the “less-than-ness” of \( T_1b \). If, subsequently, DORA compares the pair \([a, b]\) to the pair \([c, d]\), it will learn predicates (\( T_1 \) units) strongly connected to greater-than and less-than, and only weakly connected to the specific values of \( a \) and \( c \), and \( b \) and \( d \), respectively (see above). After exposure to as few as two pairs of objects, DORA has started to explicitly predicate the invariant relation greater-than \((x, y)\) with respect to dimension \( m \).
In fact, DORA has learned something much more general than that because the invariant features greater-than and less-than, learned in the context of metric dimension \( m \), will generalize to any other rate-coded metric dimension, \( n \neq m \). The reason is that the rate code that serves as the input to the relational invariance circuit operates on the magnitudes of the T1 units’ inputs, regardless of their origin: So long as the T1 units in question are (a) coupled oscillators that compete with one another to become active, (b) receive rate-coded inputs from whatever metric dimension they represent, so that (c) the T1 unit with the larger value fires earlier than the T1 unit connected to the smaller value, the T1 unit connected to the larger value of the dimension will become connected to greater-than and the T1 unit connected to the smaller value on the dimension will become connected to less-than (or, if the two T1 units encode the same value on \( n \), and become simultaneously co-active therefore activating the \( E \) unit that responds when multiple T1 units are active, they will become connected to the non-\( m \) features strongly connected to the active \( E \) unit, or equal-to).

This same property of the relational invariance circuit renders it vulnerable to incorrectly assigning the invariants more-than and less-than to any pair of object properties (represented as T1 units) that get passed into the circuit, even if those properties do not lie on a metric dimension. For this reason, there need to be constraints on when the relational invariance circuit is invoked. One obvious constraint is that it should only be invoked when the property coded by a T1 unit is a value on a metric dimension. As discussed previously, additional constraints, for example, regarding which metric dimensions are most likely to invoke the circuit under what circumstances, are likely also important, but consideration of what those constraints are is beyond the scope of the current work (but see Spelke & Kinzler, 2007).

**Learning the Content and the Format of Relational Representations**

By integrating the relational invariance circuit with the DORA algorithm for learning structured representations of relations (Doumas et al., 2008), we have developed a single system that explains learning structured relational representations of similarity and relative magnitude from very simple (nonrelational) beginnings without assuming any structured representations, or even relational invariants, a priori. The model starts with flat feature vector representations of object properties. These vectors contain no relational features, just absolute information about properties and magnitudes along dimensions. As described above, when objects with these feature encodings are compared, invariant patterns emerge, which mark similarities and differences in featural encoding and absolute magnitudes. The relational invariance circuit exploits these patterns to identify relational instances and return invariant features identifying those relations.

The DORA learning algorithm identifies invariant features of compared objects and learns structured representations of those features in a format akin to a single-place predicate. The model then links systematically co-occurring predicate–argument bindings to form functional multiplace predicate representations. That is, over a series of progressive comparisons, the model isolates collections of object features, represents these as functional single-place predicates, links systematically co-occurring single-place predicates to form multiplace predicates, and produces increasingly more refined versions of these representations. When the representational content of these objects is relational, DORA will learn structured representations of this relational content.

**A Mechanism for Generalization**

We propose that operations on relational representations underlie human generalization and that generalization based on relations occurs in (at least) two ways. First, relational representations learned in one context are readily applicable to characterize new contexts. Relational representations are useful for characterizing multiple domains because the same relations apply across domains regardless of the objects involved. Second, theories and schemas learned from one domain allow us to make inferences about other domains using analogical inference.

Analogical inference—in this case, using a model of one domain to reason about another domain—follows directly from DORA’s mapping process. For example, suppose that DORA has learned about spatial relations (e.g., above, right-of, larger) and then learned that when playing the game Breakout—where the goal of the game is to hit a ball with a paddle moving horizontally—relations between the ball and paddle predict actions to take. Specifically, DORA has learned that the state right-of (ball, paddle1) supports moving right (i.e., right-of (paddle2, paddle1)); where paddle1 is the state of the paddle before the move, and paddle2 after the move), the state left-of (paddle1, ball) supports moving left, and the state same-x (ball, paddle1) supports making no move. When DORA encounters a game like Pong—where the goal of the game is to hit a ball with a paddle moving vertically—the moves available in Pong (up and down) might remind DORA of the moves available in Breakout (left and right). With the representation of the available Pong actions in the driver (e.g., above (paddle2, paddle1), and representations of the Breakout strategy retrieved into the recipient (e.g., right-of (ball, paddle) → right-of (paddle2, paddle)), DORA performs analogical mapping. Because of the shared relational similarity, corresponding moves between Pong and Breakout map—for example, above (paddle2, paddle1) in the driver will map to right-of (paddle2, paddle1) in the recipient (Figure 5a, mappings depicted as double-headed arrowed lines; Appendix A,—Update Mapping Hypotheses and Update Mapping Connections sections, for details of how such mappings are discovered). Generalization is performed on the basis of these mappings.

During analogical generalization, propositions with unmapped elements enter the driver, and any propositions to which they map enter the recipient (information is generalized from the driver to recipient; see Hummel & Holyoak, 2003). So, if DORA has mapped above (paddle2, paddle1) to right-of (paddle2, paddle1) in the driver, the representation of the rule from Breakout enters the driver, and the mapped representation of the move from Pong enters the recipient. Figure 5b depicts a case where the rule right-of (ball, paddle) → right-of (paddle2, paddle) is in the driver, and the mapped above (paddle, paddle2) is in the recipient. When a unit \( i \) in the driver learns an excitatory mapping condition to a given unit \( j \) in the recipient, it also learns a global inhibitory mapping connection to all other units, \( k \neq j \), in the recipient. Similarly, \( j \) learns a global inhibitory connection to all units \( i \neq l \) in the driver. These global inhibitory connections play a vital role in the inference process. Continuing the example, all units encoding the representation of Pong in the recipient have a positive mapping connection to a corresponding unit in the driver.
(i.e., right-of maps to above, paddle1 maps to paddle1, and paddle2 maps to paddle2; Figure 5). By consequence, all units in the recipient also have a global inhibitory mapping connection to all other units in the driver. Therefore, when more-x (ball) becomes active in the driver, some active units have nothing to map to in recipient (the units representing ball, more-x + ball, less-x + paddle1, and right-of (ball, paddle1)). DORA recruits and activates units to match the unmapped driver units (black units indicate recruited units; black double-arrowed lines indicate inference relationships). DORA learns connections between co-active token units in the recipient (heavy black lines). The result is a representation of the situation: above (ball, paddle1) and above (paddle2, paddle1) in the recipient. DORA = Discovery of Relations by Analogy.

This form of generalized inhibition occurs when all units in the recipient map to some unit in the driver, and no units in the recipient map to the currently active driver units. That is, the signal indicates that there are elements in the driver that map to nothing in the recipient. This occurrence signals DORA to initiate analogical inference. During analogical inference, DORA recruits and activates units in the recipient that corresponds to the unmapped (i.e., inhibitory) unit in the driver (e.g., DORA recruits a T1 unit in the recipient corresponding to an unmapped driver T1 unit in the driver). Newly recruited units are assigned positive mapping connections with the driver units that initiated their recruitment (black double-headed arrows in Figure 5b), and they learn connections to other recipient units by simple Hebbian learning (e.g., active T1 units learn connections to active T2 units and active feature units; thick black lines in Figure 5b). When more-x (ball) is active in the driver, the driver T1 unit representing more-x activates the recipient T1 unit more-y, as they map, however, the active ball T1 unit, more-x + ball T2 unit, and above (ball, paddle1) T3 unit will activate nothing and inhibit all recipient units (as they map to nothing in the recipient). In response to this generalized inhibition in the recipient, DORA recruits a T1 unit corresponding to the unmapped active driver T1 unit (representing ball), a T2 unit corresponding to the active unmapped driver T2 unit (representing more-x + ball), and a T3 unit is recruited corresponding to the active unmapped driver T3 unit (representing right-of (ball, paddle1); black units in Figure 5b). The recruited T1 unit learns connections to the active features of ball and to the recruited T2 unit (as they are all co-active; thick black connections in Figure 5b). The recruited T2 unit learns connections to the recruited T3 unit (as they are co-active), and then to the T1 unit representing more-y when it is active. As such, the recruited T1 unit becomes a representation of ball, and the recruited T2 links the representation of ball and more-y (more-y + ball). Similarly, when less-x (paddle1) becomes active in the driver (activating the less-y and paddle1 T1 units in the recipient), a T2 unit will be recruited to match the unmapped active driver T2 unit. That T2 unit will learn connections to the less-y and paddle1 T1 units, and to the active recruited T3 unit (which remains active as its corresponding driver T3 unit remains active). The result is a representation of above (ball, paddle1) in the recipient (Figure 5b).

The same process also accounts for how we might predicate (or explicitly represent) known relations about new situations. As a simple example, suppose DORA encounters two objects involved in a relation such as when object-1 is above object-2. Those objects will have properties of that relation (e.g., object-1 might have
features such as “more” and “y”, delivered by the relation invariance circuit). If DORA has learned explicit structured representations of the relation above (e.g., two linked predicates strongly connected to the features “more” and “y” and “less” and “y”, respectively), then it might retrieve a representation of that relation from LTM, say above (P, Q). The retrieved relational representation can then be projected on to object-1 and object-2 (Figure 6). Specifically, P, the higher item, will correspond to object-1, and Q, the lower item, will correspond to object-2. Based on these correspondences, DORA will infer the predicates bound to P and Q about object-1 and object-2 via the analogical inference algorithm. Shifting focus to the above (P, Q) proposition (above (P, Q) is in the driver; Figure 6), when the higher unit becomes active it inhibits all units in the recipient, signaling DORA to recruit a T1 unit in the recipient to match the active higher unit and a T2 in the recipient to match the active higher + P unit. The recruited T1 unit learns connections to the active feature units, becoming an explicit representation of higher, and the recruited T2 unit learns connections to the recruited T1 unit (representing higher) and the object-1 unit when it becomes active (Figure 6). Similarly, a representation of lower is represented about object-2. The result is the known relation above predicated about object-1 and object-2 (Figure 6).

In the following simulations, we demonstrate the efficacy of the computational account of relation learning and relation-based generalization that we have proposed. We show how DORA learns structured relational representations from simple visual nonstructured and nonrelational inputs, and how it then uses these representations to support human-level cross-domain generalization—by characterizing novel domains in terms of known relations, and then driving inferences about the new domain based on the systems of relations learned in previous domains. Additionally, we demonstrate that DORA captures several key properties of human representation learning and the development of generalization.

Simulations

As described previously, explaining cross-domain generalization as analogical inference entails explaining how a system: (a) detects (or learns to detect) relational content; (b) learns structured representations of that relational content; (c) uses these representations to characterize and behave in the domains it experiences (e.g., to build models of the domain to guide behavior); and (d) uses the representations learned from previously experienced domains to make analogies and subsequently inferences about new domains. In addition, there is a distinction between representing a domain and acting on those representations. Extrapolation from one domain to another relies on both using representations learned in one domain to characterize another (i.e., representational transfer), and adopting strategies from one domain for use in another (i.e., policy transfer).

Figure 6

Generalizing Known Relations to New Situations Using the Analogical Inference Algorithm in DORA

Note. The representations of object-P and object-Q in the driver map to the representations of object-1, object-2 in the recipient (gray double-arrowed lines). As the representations of higher (object-P) and lower (object-Q) become active in the driver, some active units have nothing to map to in recipient (e.g., the driver units representing higher and higher + object-P, and the units representing lower and lower + object-Q). DORA recruits and activates units to match the unmapped driver units (black units indicate recruited units; black double-headed arrows indicate inference relationships). DORA learns connections between co-active token units in the recipient (heavy black lines). The result is a representation of above (object-1, object-2) in the recipient. DORA = Discovery of Relations by Analogy.
Below we report a series of simulations evaluating these various capacities of the model. In Simulation 1, we show that the model learns structured relational representations—both their form and content—from nonstructured and nonrelational visual inputs without assuming a vocabulary of structured representations or relational features a priori. In Simulations 2–4, we show that the model can be integrated with methods for RL to use the representations that it learns to build more complex models (or policies) for behaving in the domain, and then use its representations to perform zero-shot (i.e., first trial) cross-domain generalization. Specifically, we show that after the model learns to play one video game (Breakout), it can generalize its knowledge to play a structurally similar but featurally very different game (Pong). Moreover, we show that generalization in the model relies exquisitely on the structured format of the representations that it learns. In Simulation 5, we evaluate whether the representations that the model learned in previous simulations generalize to more complex tasks like adult analogy problems, support generalization to completely novel stimuli (i.e., approximating universal generalization), and meet the hallmarks of human relational cognition. Finally, we are proposing an account of human generalization that includes relational representation learning. As such, it should be the case that our model mirrors the capacities of children as they learn relational representations. In Simulations 6 and 7, we use the learning trajectory the model underwent during Simulations 2–4 to simulate studies from the developmental literature on children’s magnitude reasoning (Simulation 6) and relational problem-solving (Simulation 7). Additionally, Simulations 6 and 7 provide further tests of the model’s capacity for cross-domain generalization: After learning representations in a domain like Breakout, the model extends these representations to reason in the domain of a psychology experiment. We show that the model not only generalizes the representations that it learns from one domain to reason about a new task (as children do when they enter the laboratory) but also goes through the same behavioral trajectory as children. Details of all simulations appear in Appendix B.

**Visual Front End**

Generalizing DORA to work with perceptual inputs, such as pixel images necessitated supplying the model with a basic perceptual front end capable of segmenting simple objects (e.g., paddles and balls) from visual displays. We endeavored to keep this extension as simple as possible, importing existing solutions.

We used a visual preprocessor that delivers object outlines using edge detection (via local contrast) with a built-in bias such that enclosed edges are treated as a single object. In brief, the preprocessor identifies “objects” (enclosed edges) and represents them in terms of their location on the “retina,” size, and color. This information roughly corresponds to the total retinal area of the object and the enervation of the superior, inferior, lateral, and medial rectus muscles in reaching the (rough) center of the object from a reference point (see Demer, 2002). The information is encoded as the raw pixels and direction (specific muscle) between the rough object center and the reference point, and the red-green-blue (RGB) encoding of the pixels composing the object. One consequence of this encoding is that the model shows the same bias to classify along the cardinal directions observed in humans (Girshick et al., 2011).

This preprocessor is clearly a vast oversimplification of human perception. However, we chose it because it is adequate for our current purposes, it is computationally inexpensive, and the representations it generates are at least broadly consistent with what is known about human vision. For example, the visual system detects edges by local contrast (e.g., Marr & Hildreth, 1980), represents objects and their spatial dimensions (e.g., Wandell et al., 2007), and these representations and the visual image are quasihomomorphic (e.g., Demer, 2002; Engel et al., 1994; Furmanski & Engel, 2000; Moore & Engel, 2001). We certainly do not claim the preprocessor is an accurate model of human vision; only that it is not grossly inconsistent with what is known about biological vision, and that it is adequate to our current goal, which is to model learning and generalization in the domain of simple visual images, with generalization to novel domains. The output of the visual preprocessor adds two assumptions to the model: (a) that the visual system can individuate objects and (b) that dimensional information is rate-coded. Finally, it is possible to use the preprocessor as a front-end to both DORA and to the comparison DNNs, allowing us to equate the inputs used by DORA with those used by the DNNs.

**Simulation 1: Unsupervised Discovery of Relations**

The goal of Simulation 1 was to evaluate the capacity of the model to learn, without supervision, structured representations of relational concepts from nonstructured representations of objects that include only absolute (nonrelational) information. We ran two Simulations, 1a and 1b, testing the model’s capacity to learn from both simpler and more featurally complex stimuli.

**Simulation 1a**

Simulation 1a served as a basic proof of concept. In this simulation, we tested whether DORA would learn structured relational concepts when presented with simple visual displays.

**Visual Displays.** We started with 150 two-dimensional images, each differing in shape, contrast, size, width, and height (see Figure 7 for examples of the images). Each of the 150 shapes was then...
randomly grouped with between 1 and 4 other shapes to create a total of 150 multi-object displays. The displays were processed by the visual preprocessor. As described above, the visual preprocessor identified an object as any item with a continuous and connected edge and represented that object as a T1 unit connected to a collection of features corresponding to the pixels composing its absolute width and height (x- and y-extent), size, and vertical and horizontal deviation from the edge of the screen (x- and y-deviation). As the images were grayscale, we left out the RGB information in this simulation. To add extraneous noise to the objects, each T1 unit encoding an object was also randomly connected to 10 “noise” features from a set of 1,000. The result was 150 “scenes” containing between 2 and 5 objects, with each object represented as a set of absolute rate-coded spatial dimensions and noise features.

Learning Structured Relational Representations. We designed this simulation to mimic a child noticing a visual display (e.g., a scene) and attempting to use their memory of previous experiences to understand and learn about that display. DORA started with no representations (i.e., all weights set to 0). The representations of the 150 multi-object scenes were placed in DORA’s LTM. DORA attempted to learn from these stimuli, but it did not otherwise have any “task” to perform, and it received no feedback on its performance during the simulation. Rather, DORA performed 3,000 “learning trials.” On each learning trial, DORA randomly selected one collection of objects from LTM and placed that collection in the driver, thus simulating the perception of a visual display. DORA ran the driver representations through the relational invariance circuit, and then performed memory retrieval, analogical mapping, and representation learning (as described previously and in Appendix A). For the current simulations, we constrained DORA’s retrieval algorithm to favor more recently experienced displays (such recency effects are common in the memory literature; e.g., Logie et al., 2020). With probability .2, DORA attempted to retrieve from the last 100 analogs that it had learned, otherwise it attempted to retrieve from LTM generally.

In evaluating DORA’s learning, what we wanted to know was whether the model learned structured representations of relation content. That is, we wanted to know first, if the model had learned T1 units connected strongly to features defining a specific relational role concept (and only weakly to other features), and second, whether the model linked representations of complementary relational roles into multiplace relational structures. For example, if DORA learned a representation of a T1 unit connected strongly to the features encoding “more” and “x-extent” and weakly to all other features, then it had learned a relative (relational) representation of more-x-extent. Similarly, if the model learned a representation of a T1 unit connected strongly to the features encoding for “less” and “x-extent” and weakly to all other features, then it had learned a relative (relational) representation of less-x-extent. Finally, if the model learned a full LISAese structure wherein these T1 units (one representing more-x-extent and another representing less-x-extent) were bound to objects via T2 units that were linked via a T3 unit (as in Figure 1), then it had learned a structured multiplace relational representation.

To this end, we first defined a set of meaningful relational roles that the model could learn given the input images. This list comprised the set of relative encodings of the absolute dimensional information returned by the visual preprocessor: That is, the features encoding “more,” “less,” and “same” paired with the encoding of x-extent [(“more,” “x-extent”), (“less,” “x-extent”), (“same,” “x-extent”)], y-extent [(“more,” “y-extent”), (“less,” “y-extent”), and (“same,” “y-extent”)], size [(“more,” “size”), (“less,” “size”), and (“same,” “size”)], x-deviation (henceforth x; [(“more,” “x”), (“less,” “x”), and (“same,” “x”)], and y-deviation (henceforth y; [(“more,” “y”), (“less,” “y”), and (“same,” “y”)].

In order to evaluate DORA’s learning of the relations in the displays, we defined the relational selectivity metric, $Q_r$, for a T1 unit $I$ as:

$$r' = \arg\max_r \frac{1}{n} \sum_{j \in r} I_{ij}$$

$$Q_r = \frac{\frac{1}{2} \sum_{j \in r} I_{ij}}{1 + \frac{1}{2} \sum_{i} I_{ij}}$$

where $r'$ is the relational role that maximizes the mean weight of unit $i$ to the features, $j = 1 \ldots n$, that make up the role’s content, and $k = 1 \ldots m$ are all other features. $Q_r$ scales with the degree to which unit $i$ codes selectively for a relational role, where $Q_r = 1.0$ indicates that the unit responds exclusively to the features of a single relational role, $r'$. We measured the relational specificity of the T1 units in DORA’s LTM over the course of 3,000 learning trials (Figure 8). As Figure 8 illustrates, DORA learned representations (T1 units) encoding meaningful relational roles. That is, DORA learned T1 units encoding roles like more-x-extent (strongly connected only to features for “more” and “x-extent”), less-y (strongly connected only to “less” and “y”), or same-size (strongly connected only to “same” and “size”). The results indicate that DORA’s learning algorithm produces representations that encode invariant relational content.

Next, we checked whether the representations DORA learned were composed into meaningful relational structures (i.e., whether representations of complementary roles were linked into multiplace structures). For example, if DORA links more-y (obj1) and less-y (obj2) to form the relation above (obj1, obj2), or links more-x-extent to

Figure 8
Mean Relational Selectivity of T1 Units (as Defined in Text) as a Function of Number of Training Examples, Simulation 1a
(obj2) and less-x-extent (obj1) to form the relation wider (obj2, obj1), then it has learned representations of meaningful relations. To this end, we checked the number of representations in LTM representing single-place predicates (a learned T1 unit linked to another T1 unit via a T2 unit but not connected to a T3 unit), meaningful multiplex relations (T1 units representing complementary relational roles, each linked to an object T1 unit via T2 units that are also linked via a single T3 unit), and meaningless multiplex relations (T1 units not representing complementary relational roles, each linked to a single T1 unit via T2 units that are linked via a single T3 unit). As presented in Figure 9, DORA learns representations of meaningful relations with experience. By the 1000th learning trial, DORA had learned representations of all possible meaningful relations (e.g., above, below, same-vertical, right-of, left-of, same-horizontal, wider, thinner, same-width, taller, shorter, same-height, larger, smaller, same-size), and it learned progressively more refined representations of these relations with additional learning trials.

As described above, DORA learns multiplex relations by comparing sets of single-place predicates. During learning, this process runs in parallel with the discovery of the single-place predicates that will form the roles of these relations. However, because the linking operation depends on having a vocabulary of single-place predicates to combine, DORA necessarily follows a developmental trajectory in which it acquires single-place predicates before it acquires multiplex relations. As shown in Figure 9 roughly the first 300 learning trials are dominated by the discovery of single-place predicates like more-x, less-x-extent, and their complements. After that initial period, learning is dominated by the discovery of multiplex relations like above.

In contrast to error-correction learning (such as back propagation), DORA’s learning algorithm does not replace old knowledge (e.g., predicates discovered early in learning) with new knowledge (predicates learned later), but rather adds new knowledge to its existing knowledge. For example, the multiplex relations it learns do not replace the single-place predicates from which they were composed, and refined predicates and relations do not replace their less refined predecessors. However, as a consequence of DORA’s retrieval algorithm, less refined predicates become less likely to be retrieved (and thus used as the basis of new comparisons) than their (increasingly common) more-refined counterparts (the retrieval algorithm is biased in favor of retrieving the simplest pattern that fits the retrieval cue; Hummel & Holyoak, 1997). Effectively, the less refined predicates tend to fall out of service as they become obsolete. As a result, there is a difference between the predicates DORA knows (i.e., has stored in memory) and those it frequently uses.

Moreover, DORA not only learns specific relations such as above and below, but it also discovers more abstract relations such as greater-than and same-as as a natural consequence. For example, when DORA compares two different instances of wider (x, y), then it will learn a more refined representation of wider (x, y) (as described previously). But if it compares an instance of wider (x, y) to an instance of taller (z, w), then it will learn a representation that retains what wider has in common with taller, or a generic greater (a, b) relation. This result mirrors the development of abstract magnitude representations in children (e.g., Sophian, 2008).

In total, these results indicate that DORA learns structured representations of relative magnitude and similarity relations from unstructured (i.e., flat feature vector) representations of objects that include only absolute values on dimensions and extraneous noise features. However, a potential criticism of this simulation is that the starting representations are quite simple. Perhaps DORA only learns useful relational representations because each object only has five dimensions, whereas real objects have many more. Simulation 1b addresses this limitation.

**Simulation 1b: Scaling up**

In this simulation, we tested DORA’s capacity to learn from messier examples containing more competing and extraneous information. We created 150 scenes each containing between 2 and 5 objects (using the same procedure as in Simulation 1a). We then altered the objects in two important ways. First, we added 1,000 distractor features from a pool of 100,000. Second, we added absolute encodings from 45 additional dimensions. That is, while in Simulation 1a, each object was connected to rate-coded features encoding an absolute value on five dimensions (as delivered by the visual preprocessor), in this simulation each object was connected to rate-coded features encoding an absolute value on five dimensions (the five from Simulation 1a, and 45 additional “dimensions”). As a consequence, each object was now much messier, containing not only more noise features, but also encoding more dimensions (that DORA could potentially learn explicit representations of). If DORA’s learning algorithm is indeed robust, then we would expect it to (a) learn relational representations of all dimensions (there should be nothing special about the five used in Simulation 1a) and (b) learn these representations in a number of learning trials that scales proportionally with the number of items to be learned (i.e., the model should take roughly 10 times as long to learn comparably refined structured relational representations of 50 dimensions as it took to learn five).
Simulation 1b proceeded like Simulation 1a. A no-representations version of DORA was created. The representations of the 150 multi-object scenes were placed in DORA’s LTM. As in Simulation 1a, DORA attempted to learn from these stimuli, but it did not otherwise have any “task” to perform, and it received no feedback on its performance during the simulation. In this simulation, DORA performed 10,000 learning trials.

Figure 10 shows the progression of the relational selectivity of DORA’s T1 units over the course of training. Just as in Simulation 1a, DORA learned progressively more refined representations of relational content. Vitally, DORA learned structured relational (relative; more/less/same) representations of all 50 dimensions, as well as of general greater, lesser, and same. In addition, as seen in Figure 10, the number of learning trials required to learn refined representations of structured representations scales linearly. While DORA learned representations of all meaningful relations from five dimensions in roughly 1,000 learning trials (Simulation 1a), DORA learned meaningful relational representations of 50 dimensions in 10,000 trials. In addition, with more learning trials, the representations that DORA learned became progressively more refined. Just as in Simulation 1a, after 3,000 learning trials the relational selectivity in the model was just below 0.7, though with additional learning trials in Simulation 1b, relational selectivity continued to increase. Finally, Figure 11 shows that just as in Simulation 1a, DORA learned progressively more meaningful structured representations of relational representations with more learning trials and follows the same trajectory of learning single-place predicates first, followed by multiplace relational representations.

The results of Simulation 1b show that DORA’s learning algorithm scales well with the complexity of the learning environment. Finally, it is worth noting that although DORA’s learning in Simulations 1a and 1b was unsurprisingly slightly slow (recall that DORA received no feedback or guidance), Sandhofer and Doumas (2008) showed that DORA’s learning accelerates (to the same rate as human learners) when the model receives the kind of general guidance children routinely receive from adults during the normal course of cognitive development (e.g., being guided to make specific comparisons in specific sequences; Sandhofer & Smith, 1999; Sandhofer et al., 2000).

Simulation 2: Cross-Domain Transfer in Simple Video Games

Our second simulation was designed as a test of cross-domain generalization. We wanted to evaluate whether the model could learn representations from a domain, use those representations to perform intelligently in that domain, and then transfer that knowledge to a new domain in a single shot (i.e., without any additional training). We used transfer between different video games as a case study. In this simulation, after DORA learned to play Breakout, we tested its capacity to generalize, without additional training, to Pong\(^5\) (which is a structurally analogous to Breakout but featurally quite different—among other differences, the player moves the paddle up and down in Pong but left and right in Breakout), and then tested its capacity to return to playing Breakout.

For the purposes of comparison, we also trained four statistical learning systems, including (a) a deep Q-learning network (DQN; Mnih et al., 2015) with the standard convolutional neural network front end; (b) a DQN with the same visual front end as DORA; (c) a supervised DNN with the same visual front end as DORA; and (d) a graph network (e.g., Battaglia et al., 2018) with the same front end as DORA. These controls allowed us to compare DORA to systems that do not have structured relational representations, and to control for the visual front-end and its assumptions: Networks 2–4 also had

---

\(^5\) In this simulation, we discuss DORA’s performance transferring from Breakout to Pong for expositional clarity. As shown in Supplemental Simulation 1, the generalization results were the same when DORA learned to play Pong and attempted to transfer to Breakout.
Learning to Play a Game in DORA

In this simulation, as in Simulation 1, DORA started with no knowledge. To begin, DORA learned representations from Breakout game screens. Learning in this simulation proceeded as described in Simulation 1, with the difference that we used game screens from Breakout rather than images of collections of 2D shapes. We allowed the system to play 250 games of Breakout making completely random responses, which produced game screens. These game screens ran through the visual preprocessor described above generating representations of scenes composed of simple objects (e.g., the paddle, the ball, the rows of bricks), which were stored in LTM. DORA attempted to learn from these stimuli performing 2,500 learning trials. It did not have any “task” to perform, and it received no feedback during this part of the simulation. DORA successfully learned structured representations of above, below, same-vertical, right-of, left-of, same-horizontal, wider, thinner, same-width, taller, shorter, same-height, larger, smaller, same-size, same-color, and different-color (screen images were colored).

The next step was for DORA to use the representations that it learned from the game environment to engage intelligently with it. Several accounts of how relational representations, once available, may be used to characterize particular domains have been proposed (e.g., Kemp, 2012; Lake et al., 2015; Nye et al., 2020; Tenenbaum et al., 2011). However, these probabilistic program induction approaches do not directly address the problem of building a relational model of the environment from a reward signal (these models are supervised). As learning to play video games entails learning to associate actions with states of the game based on reward signals (e.g., points), RL methods (Sutton & Barto, 2018) are a natural starting point to solve this problem. RL has been widely applied to account for several aspects of human learning and exploratory behavior (Gershman, 2017; Otto et al., 2010; Rich & Gureckis, 2018). In tabular RL, which is the version of RL that we use in this simulation, the state-action space is represented as a table where the rows are defined by the individual states and the columns are defined by the actions. A known problem with tabular RL is that as the size of the table increases, learning becomes intractable. As relational representations can be combined, the size of the table grows exponentially on the number of relations considered when describing the state. Therefore, in our simulations, we make the simplifying assumption that the agent knows what the relevant relations to build a model of the domain are (from the relations that the model had learned from game screens previously). By definition, a full account of how to build a relational model of the domain from the reward signal would need to solve the problem of selecting the relevant relations from a potentially very large set of relations. We return to this point in the general discussion.

As mentioned above in our simulations, states were represented as the relevant relations to learn to play a game. In Breakout, the relations considered were right-of, left-of, and same-horizontal applied over the paddle and the ball and the same relations applied over the ball at times $t - 1$ and $t$. For example, one state in Breakout could be \([\text{right-of (ball1, paddle1)}]\). On the other hand, actions were represented as a relation between the object that the action was performed over at time $t$, and the same object at time $t + 1$. For example, in Breakout the action \(\text{move-right}\) was represented as \(\text{right-of (paddle2, paddle1)}\), where paddle1 is the paddle before acting, and paddle2 is the paddle after. To associate actions with states of the game, we augmented DORA with the capacity for RL. Specifically, we used tabular Q-learning (Watkins, 1989). RL algorithms seek to maximize the expected discounted cumulative reward, or return, by interacting with the environment. In each iteration of this process, the environment produces a state $S_t$ and a reward $R_t$ and the agent takes an action $A_t$ in response. The goal of RL is to find the optimal policy $\pi^*$ that maximizes the return. To do so, Q-learning utilizes action values as the basis for this search. The action value of a state-action pair $Q(s, a)$ is the return when the agent is in state $s$ at time $t$, $S_t = s$, and takes action $a$, $A_t = a$. Q-learning follows an epsilon-greedy policy, where most of the time the action is selected greedily regarding the current action values and with a small probability the action is selected randomly, while updating the action values according to the equation:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)),$$

where $\gamma$ is a discount factor (in all simulation we set this value to .99).

Applied iteratively, this algorithm approximates the true action values and, therefore, the output policy (greedy regarding these values) will approximate the optimal policy. We trained DORA for 1,000 games using tabular Q-learning. The model learning to associate relational states with relational representations of the available actions. Importantly for our purposes, because the states are relational the resulting policy corresponds to a set of relational rules that can be used as a basis for analogical inference (see below).

Generalizing to a New Game in DORA

As described above, analogical inference occurs when a system uses analogical correspondences between two situations to flesh out one situation based on knowledge of the other. This method is precisely how DORA infers how to play a game like Pong based on its experience with a game like Breakout. While learning to play Breakout, DORA had learned that relations between the ball and paddle predicted actions. Specifically, DORA learned that the state \(\text{right-of (ball, paddle1)}\) supported moving \(\text{right}\) (i.e., \(\text{right-of (paddle2, paddle1)}\)), that the state \(\text{left-of (paddle1, ball)}\) supported moving \(\text{left}\), and that the state \(\text{same-x (ball, paddle1)}\) supported making \(\text{no move}\). With the
representation of the available Pong actions in the driver, these Breakout representations could be retrieved into the recipient. DORA then performed analogical mapping. Because of the shared relational similarity, corresponding moves between Pong and Breakout mapped—for example, above (paddle2, paddle1) in the driver mapped to right-of (paddle2, paddle1) in the recipient. DORA then performed analogical generalization on the basis of these mappings.

As described previously and illustrated in Figure 5, after mapping the moves in Breakout and Pong, DORA infers the relational configurations that might reward specific moves in Pong based on the relational configurations that reward specific moves in Breakout. For example, given that right-of (ball, paddle) tends to reward a right response (right-of (paddle2, paddle1)) in Breakout, and the mapping between the right response and the up response (above (paddle2, paddle1)) in Pong, DORA inferred that above (ball, paddle) tends to reward an up response in Pong. The same process allowed DORA to generalize other learned rules (e.g., below (paddle2, paddle1) then move down).

Importantly, like DORA’s representation learning algorithm, mapping and analogical inference are completely unsupervised processes: The model discovers the correspondences between the games on its own, and based on those correspondences, makes inferences about what kinds of moves are likely to succeed in the new situation.

**Simulation Results**

The DQN, the DQN with the same visual front end as DORA, and the graph network were all trained for 31,003, 20,739, and 10,000 games, respectively. The DNN was trained via back-propagation for 4,002 games. Models that use structured representations often require far fewer training examples than networks trained with traditional feature-based statistical learning algorithms (see, e.g., Bowers, 2017; Hummel, 2010). It is, therefore, unsurprising that DORA learned to play Breakout much faster than the other networks.

Figure 12a shows the mean score over the last 100 games of Breakout for all five networks. As expected, all the networks performed well.

We then had the networks play a new game, Pong, for 100 games. Figure 12b shows the models’ zero-shot (i.e., immediate) transfer from Breakout to Pong. DORA performed well on the very first game of Pong (left columns) and over its first 100 games (beyond making an analogical inference, DORA did not engage in any additional learning during these test games). That is, DORA demonstrated zero-shot transfer between the games: Having learned to play Breakout, generalized to how to play Pong. In contrast, the statistical learning algorithms did not transfer from Breakout to Pong: Having learned to play Breakout, the statistical algorithms knew nothing at all about Pong. (The bars for these other networks are not missing from Figure 12c, they are simply at zero.) This result is largely unsurprising: One does not expect a lookup table for addition to do subtraction, and one does not expect a lookup table for Breakout to play Pong.

The reason for DORA’s zero-shot transfer from Breakout to Pong is straightforward. As described above, during its first game of Pong, DORA represented the game state using the relations it had learned playing Breakout. Armed with these relations, the model used analogical mapping to discover the correspondences between the two games, and based on those correspondences, made inferences about what kinds of moves were likely to succeed in the new situation. The model’s prior experience with Breakout thus allowed DORA to play Pong immediately without any additional learning.

---

**Figure 12**

*Results of Game Play Simulations With DORA, the Deep Q-learning Networks (DQNs) and the Deep Neural Networks (DNNs)*

![Graphs showing performance](image)

**Note.** Error bars represent two std errors. (a) Performance humans and networks on Breakout as an average of 100 test games. (b) Results of networks playing Pong after training on Breakout as score on the first game played and mean score over the first 100 games played. (c) Results of networks when returning to play Breakout after playing or learning to play Pong as an average of the first 100 games played. DORA = Discovery of Relations by Analogy.
it to play its first game of Pong like a good rookie rather than a rank
novice.

As a final test, we trained the DNNs (but not DORA) to play Pong
until they could play with competence, and then retested the DNNs
and DORA for their ability to play Breakout. Of interest in this
simulation was whether the various systems, upon learning to play
Pong, would still know how to play Breakout. Figure 12c shows the
performance of the networks on the first 100 games of Breakout after
learning Pong. DORA returned to Breakout with little difficulty,
t(198) = 1.26, p > .05; again DORA engaged in no learning during
these test games). By contrast, the deep DNNs showed extremely poor
performance, indicating that learning to play Pong had completely
overwritten their ability to play Breakout (i.e., the networks suffered
interference from Pong to Breakout; see, e.g., French, 1999).6

It is important to stress that the supervised DNN, one of the
DQNs, and the graph net used the same visual preprocessor as
DORA, so the differences in generalization performance cannot be
attributed to differences in inputs (e.g., individuated objects and
rate-coded dimensions). Rather, the DNNs’ generalization failure
reflects the purely statistical nature of their representations. For a
DNN screens from Breakout and screens from Pong are simply from
different distributions, and therefore, it has no reason to generalize
between them. By contrast, relation-based learning—like a variabil-
ized algorithm—naturally generalizes to novel values (arguments)
bound to the variables (relational roles) composing in the algorithm
(model of the task).

Simulation 3: Cross-Domain Transfer From Shape
Images to Video Games

A serious limitation of Simulation 2 is that all the models
embarked on learning Breakout as blank slates, with no prior
knowledge of any kind. We adopted this practice to remain consist-
tent with the tradition in the prior literature on neural networks for
game play (e.g., Mnih et al., 2015), which is to start with completely
untrained networks. This convention no doubt reflects the fact that
DNNs do not profit from cross-domain transfer, so there is no point
in training them on any other kind of task (e.g., to teach them basic
spatial relations) before training them to play video games.

People, by contrast, learn very differently than DNNs. Rather than
approaching each new task as a tabula rasa, people bring their prior
knowledge to the learning of new tasks. By the time a person plays
their first video game, they have no doubt had extensive experience
with such basic spatial relations as above and left-of, not to mention
years of experience in numerous other domains. This difference
between how people and DNNs learn is important as it speaks
directly to the importance of cross-domain transfer: Whereas purely
associative systems such as DNNs suffer from retraining on a new
task, people rely on it.

Accordingly, Simulation 3 explored a slightly more realistic
course of learning in DORA. Instead of learning the relations
relevant to playing the game from the game itself, DORA first
learned representations from a different domain, namely the first 300
images from the CLEVR (Compositional Language and Elementary
Visual Reasoning) data set (pictures consisting of multiple objects
on a screen; Johnson et al., 2017). We started with a version of
DORA with no knowledge. CLEVR images were run through the
preprocessor and the results were encoded into DORA’s LTM. We
then ran DORA for 2,500 unsupervised learning trials (as in
Simulation 1). As in previous simulations, DORA successfully
learned structured representations of the relations present in the
stimuli—here above, below, same-vertical, right-of, left-of, same-
horizontal, wider, thinner, same-width, taller, shorter, same-height,
larger, smaller, same-size, same-color, and different-color. We do not
claim that this pretraining with the CLEVR images provides
DORA with a realistic approximation of a person’s prevideo game
experience. On the contrary, DORA’s pretraining is a pale imitation
of the rich experiences people bring to their first video game
experience. But the crucial question in Simulation 3 is not whether
we can endow DORA with all the advantages a person’s prior
experiences bring to their ability to learn video games, but merely
whether DORA, like people, is capable of profiting from prior
exposure to relevant spatial relations even if that exposure comes
from a completely different domain.

Following representation learning from the CLEVR images,
DORA learned to play Breakout via Q-learning for 800 games.
Again, the key difference from the previous simulation was that
DORA used the representations that it had learned from the CLEVR
images to encode game screens. No additional representation learn-
ing occurred from experience with Breakout. Only associations
between the previously learned representations and successful
moves were updated via Q-learning. As in Simulation 2, after
training with Breakout, we tested the model’s ability to generalize
to playing Pong, and then return to playing Breakout.

Using representations learned from CLEVR, DORA learned to
play Breakout and transferred learning from Breakout to Pong and
back to Breakout in a manner very similar to the results of
Simulation 2 (Figure 13a–13c, black and dark gray bars). However,
DORA learned to play Breakout in fewer games when it started with
the representations learned from the CLEVR images than it did
starting with a blank slate in Simulation 1 (800 vs. 1,250 games,
respectively; as it did not need to learn representations, only a policy
for associating representational states with actions). This simulation
demonstrates that DORA—like a human learner—exploits cross-
domain transfer rather than suffering from it. DORA’s capacity to do
so is a direct reflection of its ability to represent the domain-relevant
relations explicitly, bind them to arguments, and map them onto
corresponding elements between the familiar and novel games.

Simulation 4: The Centrality of Binding and
Structured Representations in Cross-Domain
Transfer in DORA

DORA relies on neural oscillations to dynamically bind distrib-
uted representations of objects and relational roles into relational
structures. According to our account, these oscillations play a central
role in learning and generalization because, without them, DORA’s
representations would be nonstructured feature lists—akin to the
representations used by DNNs and other associative learning
algorithms—and its generalization ability would be correspondingly
limited. To explore the role of neural oscillations—that is, structured
relational representations—in DORA’s performance, we reran

6 Catastrophic forgetting can be avoided by interleaved training (i.e.,
training on to-be-learned tasks simultaneously, with “batch” updating of
connection weights; e.g., Kirkpatrick et al., 2017). Sequential training of the
type people routinely encounter continues to produce catastrophic forgetting
in DNNs.
Simulation 2 (allowing the model to learn to play Breakout, and then attempting to generalize to Pong), but with two different ablated versions of the model. In both ablated versions, we disrupted the lateral inhibition between token units (specifically, we reduced the weight of the inhibitory lateral connections between tokens from $-1$ to $-0.1$), disrupting the model’s ability to maintain systematic oscillatory behavior. In the first ablated model (A1), we ablated the inhibitory connections from the onset of the simulation. As a result, neural oscillations were disrupted both during predicate learning and thereafter. In the second ablated model (A2), we ablated the inhibitory connections after the model had learned to play Breakout: Although the model was intact when it learned to play Breakout, the neural oscillations, and thus role-argument bindings, were disrupted during generalization to Pong.

The current simulations were otherwise identical to Simulations 2. As expected, Model A1 failed to learn any useful predicate representations. Disrupting the model’s neural oscillations eliminated its capacity to learn predicates, and thus greatly reduced its capacity to learn Breakout. The model resorted to learning based on the absolute features of the stimuli, and thus learned much like a less sophisticated DQN. Based on these representations, the model struggled with Breakout even after 20,000 training games and failed to generalize to Pong (Figure 13, light gray bars). Model A2, which was intact during predicate learning and Breakout training, learned predicate representations and achieving good performance on Breakout within 1,000 games (Figure 13a, white bar). However, when the oscillations were disabled after training, the model failed to generalize to Pong (Figure 13b, white bar). This result demonstrates the centrality of systematic oscillations in the model’s capacity to learn relational representations (Model A1) and of structured relational representations to its ability to perform generalization (Models A1 and A2).

**Simulation 5: Transfer From Games to More Complex Tasks**

This simulation was designed to further challenge the capacity of the representations that DORA learns. In this simulation, we investigated whether the representations DORA learned playing video games and from CLEVR (i.e., Simulations 2 and 3) would allow the model to generalize to the very different domain of analogical reasoning. To this end, we used the same model and representations from Simulations 2 and 3 and set to it tasks representing characteristics of human-level analogical reasoning (Bassok & Olseth, 1995; Gick & Holyoak, 1983; Holyoak, 2012; Holyoak & Thagard, 1996). Specifically, after learning representations from Breakout and CLEVR, we tested whether it could immediately (i.e., with no additional experience) use those representations to: (a) solve analogical cross mappings; (b) analogically map similar, but nonidentical predicates; (c) analogically map objects with no featural overlap—including completely novel objects—that play similar roles; and (d) map the arguments of an $n$-place relation onto those of an $m$-place relation even when $n$ and $m$ are unequal (i.e., called violation the $n$-ary restriction; Hummel & Holyoak, 1997). As such, the simulation had two purposes: (a) to evaluate the capacity of the representations the model learns to support human-level analogical reasoning and (b) to provide a further test of the model’s capacity for cross-domain generalization: Just like humans do, the model had to learn representations in one domain, and use these representations to reason in a novel (laboratory) task.
During a cross-mapping, an object (object1) is mapped to a featurally less similar object (object2) rather than a featurally more similar object (object3) because it (object1) plays the same role as the less similar object. For example, if cat1 chases mouse1 and mouse2 chases cat2, then the structural cross-mapping places cat1 into correspondence with mouse2 because both are bound to the chaser role. The ability to find such mappings is a key property of human relational (i.e., as opposed to feature-based) reasoning (e.g., Bassok & Olseth, 1995; Gick & Holyoak, 1980, 1983; Holyoak, 2012; Richland et al., 2006). Cross-mappings serve as a stringent test of a computational system’s structure sensitivity as they require the system to discover mappings based on relational similarity in the face of competing featural or statistical similarity.

We tested the representations DORA learned in Simulations 2 and 3 for their ability to support cross-mapping. DORA randomly selected two of the predicates (T1 units) it had learned during Simulations 2 and 3, such that both predicates coded for the same relation (e.g., both coded for above or both coded for same-width). DORA bound the relations to new objects, to form two new propositions, P1 (e.g., above (object1, object2)) and P2 (e.g., above (object3, object4)). We manipulated the objects such that the agent of P1 (object1) was featurally identical to the patient of P2 (object 4) and the patient of P1 (object 2) was featurally identical to the agent of P2 (object 3). Based on the objects’ featural similarity, DORA would therefore map object1 to object4 and object2 to object3, but based on their roles in the relational structures, it should map object1 to object3 and object2 to object4. We repeated this procedure 100 times (each time with different randomly chosen T1 units) using representations from Simulations 2 and 3. In every case, DORA successfully mapped object1 to object3 and object2 to object4 (the structurally consistent mappings) rather than object1 to object4 and object2 to object3. This result demonstrates that the relations DORA learned in Simulations 2 and 3 immediately transfer to analogy tasks and support relational cross-mapping.

We then tested whether DORA’s relational representations support mapping similar but nonidentical relations (such as mapping above to greater-than and mapping objects with no featural overlap based only on their bindings to similar roles). DORA randomly selected two of the relations, R1 and R2 (e.g., above (x, y) or wider (x, y)), that it had learned during Simulations 2 and 3 such that each role of R1 shared roughly half of its features with the corresponding role of R2 (e.g., the role more-y has half of its features in common with the role more-length). The objects serving as arguments of the relations had no featural overlap at all. We repeated this process 100 times and each time, DORA mapped the agent role of R1 to the agent role of R2 and the patient role of R1 to the patient role of R2.

Even though the objects had no features in common, and even though the relations to which they were bound were not identical, DORA found the structurally correct object and role mappings.

Next, we tested whether the representations DORA learned can violate the n-ary restriction, mapping the arguments of an n-place predicate onto those of an m-place predicate when n ≠ m. In each of these simulations, DORA randomly selected a relation, R1, that it had learned in Simulations 2 and 3, and we created a single-place predicate (r2) that shared 50% of its features with the agent role of R1 and none of its features with the patient role. DORA then bound two objects to the roles of R1 to form the Proposition R1 (object1, object2), and bound a third object to r2, to form the Proposition R2 (object3). Object3 shared half its features with object1 and the other half with object2 (i.e., it was equally similar to both object1 and object2). DORA attempted to map r2 (object3) onto R1 (object1, object2). If the model can violate the n-ary restriction, then it should consistently map object3 to object1 based on the similarity of r2 to the first (agent) role of R1 (recall that R1 is represented as a linked set of roles). This process was repeated 100 times using a different randomly chosen R1 each time. Each time DORA successfully mapped object3 to object1, along with corresponding relational roles (i.e., DORA maps the predicate representing one of the roles of R1 and the predicate representing the single-place predicate r2). We then ran 100 simulations in which r2 shared half its features with the second (patient) role of R1 rather than the first (agent) role. In these 100 additional simulations, DORA successfully mapped the patient role of R1 to r2 (along with their arguments).

Finally, we tested whether the representations that DORA learns support generalization to completely novel (i.e., never before experienced) stimuli. The ability to make generalizations about completely novel items is the hallmark of the capacity for universal generalization (see, e.g., Marcus, 2001). In this simulation, we created six objects with completely novel features (features units grafted onto DORA). DORA randomly selected two instances of the same relation that it had learned in Simulations 2 and 3 (e.g., two instances of bigger). We bound one instance of the relation to three of the objects—object1, object2, and object3—to create three propositions R1–R3 that instantiated a transitive relation such that object2 served as both the patient of one proposition and the agent of the other. For example, if the relation DORA had selected was bigger, then it bound that relation to object1 and object2 to make the Proposition R1, bigger (object1, object2), bound the same relation to object2 and object3 to make the Proposition R2, bigger (object2, object3), and finally bound the same relation to object1 and object3 to make the Proposition R3, bigger (object1, object3) (recall that within a single analog, or story, token units are shared between proposition, so the same T1 unit instantiating, say, more-size was bound to both object1 and object2). From the remaining three objects—object4, object5, object6—we used the second instance of the relation to make the propositions R4 and R5 such that object5 served as both the patient in one proposition and the agent in the other. For example, we bound object4 and object5 to bigger to make R4, bigger (object4, object5), and bound object5 and object6 to bigger to make R5, bigger (object5, object6). Importantly, none of the objects had any features (including metric features) that DORA has previously experienced. As such the simulation was akin to telling DORA that a transitive relation held between three novel objects and that some of the same relations held between a second three objects. We placed R1 and R2 and R3 in the driver, and R4 and R5 in the recipient. We wanted to see whether (a) DORA could integrate relations such that it would map R1 to R4 and R2 to R5 (i.e., it would map the instances where the same object was a patient and an agent) and (b) whether it would use these mappings (if discovered) to generalize the transitive relation in R3 to the objects in the recipient (i.e., would it complete the transitive set of relations about object4–6). If DORA found a mapping, it then attempted to perform analogical inference. We ran the Simulation 100 times. In each simulation, DORA mapped the representation of R1 to R4, and R2 to R5. Moreover, DORA generalized R3 to the recipient to create a transitive Proposition R6—for example, to continue the
above example, DORA inferred that bigger (object4, object6). That is, armed with the knowledge that some objects it had never before experienced played particular relational roles, DORA generalized that a proximal object (that it had also never experienced before) might play a complementary role. The inability to reason about completely novel features (i.e., features not part of the training space) is a well-known limitation of traditional neural networks (e.g., Bowers, 2017). However, this limitation does not apply to DORA. Not only can DORA represent that novel objects can play certain roles (because it can dynamically bind roles to fillers), but it can also use its representations to make inferences about other completely novel objects.

In total, Simulation 5 demonstrates that the relational representations DORA learned during Simulations 2 and 3 immediately support performance of an unrelated task (analogical reasoning) even with completely novel objects. After learning representations in one domain (game play and images of shapes), DORA, with no additional experience (zero-shot), used these representations to solve a set of analogical reasoning tasks representing several hallmarks of human analogical thinking, and thus used these representations to generalize to completely novel objects. The results provide further evidence that DORA’s representations support cross-domain transfer and highlight the generality of the DORA framework.

Simulation 6: Development of Representations of Relative Magnitude

We have previously shown that DORA’s format learning algorithm provides a good account of several developmental phenomena in representational development (e.g., Doumas et al., 2008). The purpose of Simulations 6 and 7 was to examine whether the representations that DORA learned, and the trajectory of the representation learning mirror human development when DORA is learning both relational content and relational format. Additionally, these simulations provided another opportunity to evaluate generalization of representations across domains: Learning representations in one domain and deploying those representations to reason about a new domain (as human often do when they engage in laboratory experiments).

Children develop the ability to reason about similarity and relative magnitude on a variety of dimensions (e.g., Smith, 1984). The development of children’s capacity to reason about basic magnitudes is well demonstrated in a classic study by Nelson and Benedict (1974). In their experiment, children aged 3–6 years old were given a simple identification task. An experimenter presented the child with two pictures of similar objects that differed in some dimensions. The experimenter then asked the child to identify the object with a greater or lesser value on some dimension. For example, the child might be shown pictures of two fences that differed in their height, their size, and their color, and then asked which of the two fences was taller or shorter. The developmental trajectory was clear: Children between 3-years-10-months and 4-years-4 months (mean age ~48 months) made errors on 34% of trials, children between 4-years-7-months and 5-years-5-months (mean age ~60 months) made errors on 18% of trials, and children aged 5-years-6-months and 6-years-6-months (mean age ~73 months) made errors on only 5% of trials. In short, as children got older, they developed a mastery of simple magnitude comparisons on a range of dimensions.

If DORA is a good model of human representational development, then it should be the case that DORA’s representations follow a similar developmental trajectory. To test this claim, we used the representations that DORA had learned during Simulation 2. If DORA develops like a human child, then early in the learning process, DORA’s performance on the Nelson and Benedict task should mirror 3–4-year-old children, later in the learning process DORA’s performance should mirror 4–5-year-old children, and later in the learning process DORA’s performance should mirror 5–6-year-old children.

To simulate children of different ages, we stopped DORA at different points during learning and used the representations that it had learned to that point (i.e., the state of DORA’s LTM) to perform the magnitude reasoning task. To simulate each trial, we created two objects instantiated as T1 units attached to features. These features included 100 random features selected from the pool of 10,000, along with features encoding height, width, and size (dimensions used in Nelson and Benedict) in a pixel format (as in the simulations above; e.g., for an object 109 pixels wide, one feature unit describing “width10” and 109 features encoded “109 pixels wide”). A dimension was selected at random as the question dimension for that trial. DORA sampled at random a representation from its LTM that was strongly connected to that dimension (with a weight of .95 or higher). If the sampled item was a relation or a single-place predicate, DORA applied it to the objects and placed that representation in the driver. For example, if the key dimension was size, the two objects (obj1 and obj2) were then run through the relational invariance circuit on the dimension of size, marking one (assume obj1) as relatively larger and the other (assume obj2) as relatively smaller. If DORA had sampled a representation of the relation larger (x, y), then the more-size T1 unit was bound to the obj1 T1 unit and the less-size T1 unit was bound to the obj2 T1 unit. To simulate a dimensional question, DORA randomly sampled a representation of the question dimension from LTM and placed that in the recipient. For example, if the question was, “which is bigger?” DORA sampled a representation of a T1 unit encoding more-size from LTM. DORA then attempted to map the driver and recipient representation. If DORA mapped a representation in the driver to a representation in the recipient, the mapped driver item was taken as DORA’s response on the task. If DORA failed to find a mapping, then an item was chosen from the driver at random and taken as DORA’s response for that trial (implying that DORA was guessing on that trial). The probability of guessing the correct item by chance was 0.5.

To simulate 4-year-olds, we used the representations in DORA’s LTM after 1,000 total training trials, to simulate 5-year-olds we used the representations in DORA’s LTM learned after 1,000 additional training trials (2,000 total trials), and to simulate 6-year-olds we used the representations in DORA’s LTM learned after 1,000 additional training trials (3,000 total trials). We ran 50 simulations each with 20 trials at each age level (each simulation corresponding to a single child). The results of the simulation and the original results of Nelson and Benedict are presented in Figure 14.

The qualitative fit between DORA’s performance and the performance of the children in Nelson and Benedict’s study is close. Just like the children in the original study, DORA is better than chance, but still quite error-prone early during learning, but gradually comes to learn representations that support very successful classification of dimensional magnitudes. These simulation results provide evidence
that the trajectory of the development of DORA’s representations of relative dimensional magnitude mirrors that of humans.

Simulation 7: The Relational Shift

One of the key findings from work on the development of analogical reasoning in children is that children go through a relational shift (e.g., Gentner et al., 1995). The relational shift describes a qualitative change in children’s reasoning wherein they progress from making analogies based on the literal features of things, to making analogies based on the relations that objects are involved in (e.g., Richland et al., 2006). With development, children learn progressively more powerful representations of similarity and relative magnitude relations that support more proficient relational generalization (Smith, 1984). In addition, children develop the capacity to integrate multiple relations in the service of reasoning (e.g., Halford & Wilson, 1980), and their relational representations grow more robust with learning, and allow them to overcome ever more excessive featural distraction (e.g., Halford & Wilson, 1980; Rattermann & Gentner, 1998).

One of the classic examples of the relational shift and the associated phenomena is given in Rattermann and Gentner (1998). In their experiment, Rattermann and Gentner had 3-, 4-, and 5-year-old children participate in a relational matching task. Children were presented with two arrays, one for the child and one for the experimenter. Each array consisted of three items that varied on some relative dimension. For example, the three items in each array might increase in size from left to right or decrease in width from left to right. The dimensional relation in both presented arrays was the same (e.g., if the items in one array increased in size from left to right, the items in the other array also increased in size from left to right). The items in each array were either sparse (simple shapes of the same color) or rich (different shapes of different colors). The child watched the experimenter hide a sticker under one of the items in the experimenter’s array. The child was then tasked to look for a sticker under the item from the child’s array that matched the item selected by the experimenter. The correct item was always the relational match—for example, if the experimenter hid a sticker under the largest item, the sticker was under the largest item in the child’s array. Critically, at least one item from the child’s array matched one of the items in the experimenter’s array exactly except for its relation to the other items in its array. To illustrate, the experimenter might have an array with three squares increasing in size from left to right (Figure 15a). The child might have an array of three squares also increasing in size from left to right, but with the smallest item in the child’s array identical in all featural properties to the middle item in the experimenter’s array (Figure 15b). Thus, each trial created a cross-mapping situation, where the relational choice (same relative size in the triad) was at odds with the featural choice (exact object match). The child was rewarded with the sticker if she chose correctly.

Rattermann and Gentner found a relational shift. Children between 3 and 4 years old were very drawn by featural matches and had trouble systematically making relational matches (making relational matches 32% of the time in the rich condition and 54% of the time in the sparse condition). Children between 4 and 5 years old were quite good at making relational matches with sparse objects—making relational matches 62% of the time—but still had trouble with rich objects when featural matches were more salient—making relational matches 38% of the time. Children between 5 and 6 years old were quite good at making relational matches in both the rich and the sparse conditions, with the rich condition providing more trouble than the sparse condition—making relational matches 68% for rich and 95% of the time for sparse stimuli.

We simulated the results of Rattermann and Gentner (1998) as in the simulation above, using the representations learned during Simulation 2. Again, to simulate children of different ages we stopped DORA at different points during learning and used the representations that it had learned to that point. To simulate 3-year-olds we used the representations in DORA’s LTM after 850 training trials, to simulate 4.5-year-olds we used the representations in DORA’s LTM after 1,500 training trials, and to simulate 5.5-year-olds we used the representations in DORA’s LTM after 2,500 training trials. To simulate each trial, we created two arrays...
of three objects, each object instantiated as a T1 unit connected to features. For the sparse trials, each object was connected to feature units such that some features encoded absolute size, height, width, x-position, y-position, color, 10 features describing shape, and four features were chosen at random from a pool of 1,000. The identical objects from both arrays matched on all features. For the rich trials, each object was attached to additional features: some features encoding absolute size, height, width, x-position, y-position, color, 10 features describing shape, four features describing object kind (e.g., “shoe,” “train,” “bucket”), and 40 features chosen at random from a pool of 1,000. The identical objects from both arrays matched on all features.

We ordered the objects in both arrays according to some relation (e.g., increasing size, decreasing width). DORA sampled four representations from its LTM that were strongly connected to that dimension (with a weight of .95 or higher) and applied two of the sampled representations to each of the two arrays. If the sampled representation was a relation or a single-place predicate, it applied to the objects. For example, if the key dimension was size, and DORA sampled a representation of the relation larger (x, y), it applied that representation to the objects, binding the larger object to the more-size role and the smaller object to the smaller role (as described in Simulation 6). If the sampled representation was a single-place predicate like more-size (x), then it was bound to the larger object. As each array consisted of two instances of the key relation (e.g., larger (object1, object2), and larger (object2, object3)), DORA applied one of the two sampled items to one of the relations in the array, chosen at random, and the other sampled item to the other relation in the array. For simplicity, the model only considered relations between adjacent objects.

The representation of the child’s array entered the driver, and the experimenter’s array the recipient. An item from the recipient was chosen at random as the “sticker” item (i.e., the item under which the sticker was hidden). The capacity to ignore features is a function of the salience of those features, and so richer objects with more features are harder to ignore (see, e.g., Goldstone & Son, 2012).

To simulate the effect of the rich versus the sparse stimuli, on each rich trial, DORA made a simple similarity comparison before relational processing started. It randomly selected one of the items in the driver and computed the similarity between that item and the “sticker” item in the recipient using the equation:

\[
\text{sim}_{ij} = \frac{1}{1 + \sum_{i} (1 - s_i)},
\]

where sim is the 0–1 normalized similarity of PO unit i and PO unit j, and \(s_i\) is the activation of feature unit i. If the computed similarity was above .8, then DORA learned a mapping connection between the two items. Finally, DORA attempted to map the items in the driver to the items in the recipient. If any driver representation was mapped to the “sticker” item in the recipient, the mapped item was taken as DORA’s response on the task. If DORA failed to find a mapping, then it selected an item from the recipient at random as a response for that trial (implying that DORA was guessing on that trial). The probability of guessing the correct item by chance was 0.33.

We ran 50 simulations each consisting of 20 trials at each age level. The results of the simulation, as well as those from the original Rattermann and Gentner experiment with both sparse and rich trials, are presented in Figure 16a and 16b respectively.

As Figure 16 shows, there is a close qualitative fit between DORA’s performance and the performance of the children in Rattermann and Gentner (1998). Initially, DORA, just as the 3-year-old children in the original study, had some trouble correctly mapping the items in the driver and the recipient, and struggled to solve the cross-mapping. As DORA learned more refined representations (after more training), like the 4-year-old children in the original study, DORA began to solve the sparse problems more successfully, while still struggling with the rich problems. Finally, like the 5-year-old children in the original study, after even more learning, DORA was quite successful at both rich and sparse trials, reaching ceiling-level performance on
the sparse problems. These simulation results indicate that, like humans, the trajectory of the development of DORA’s representations of relative dimensional magnitude undergoes a relational shift with learning. Additionally, the representations that DORA learns during its development support the same kind of performance on relational matching tasks that is evidenced by human children during their development.

General Discussion

Summary and Overview

We have presented a theory of human cross-domain generalization instantiated in the DORA computational framework. Our proposal is that people represent knowledge domains as models consisting of structured representations of the relations among the elements of those domains. These representations specify the invariant content of the relations and their arguments, and representations of relational roles are dynamically bound to their fillers while maintaining role-filler independence. DORA learns both the content and structure of these relations from nonrelational inputs, such as visual displays, without supervision. DORA uses comparison to bootstrap learning both the content and structure of relational representations. By integrating these representations with a capacity for RL, DORA learns which relations to use in what contexts in the service of problem-solving (e.g., game play). The resulting representations can be applied to new domains, including completely novel ones, by a process of analogical inference. That is, the model generalizes across domains as a natural consequence of its ability to represent relations in a manner that is invariant with both the arguments of the relations and the specific circumstances in which those relations arise.

A series of simulations demonstrated that this approach to learning and knowledge representation greatly facilitates cross-domain generalization. Simulation 1 showed that the model is capable of learning structured representations of relations from unstructured, nonrelational visual inputs. Simulation 2 showed that, as a result of learning to play one video game, DORA learns representations that support immediate (zero-shot) transfer to a different game (Pong), and the capacity to move between games successfully. By contrast, four different associative networks (two DQNs, a DNN, and a GNN) both (a) failed to transfer knowledge from one game to another and (b) lost their ability to play the first game after training on the second. Simulation 3 demonstrated that the representations DORA learns in one domain (images of 3D shapes) support learning to play one video game play and immediately generalize to a new game. Simulation 4 demonstrated the essential role of structured representations in the model’s learning and generalization. Simulation 5 showed that the representations DORA learns from domains like video games and pictures also support successful zero-shot transfer to unrelated reasoning tasks (cross-mapping, mapping nonidentical predicates, mapping novel objects, and violating the n-ary restriction), and, importantly, support generalization to completely novel (i.e., never previously experienced) stimuli. Finally, Simulations 6 and 7 showed that DORA follows the same developmental trajectory as children as it learns representations. That is, DORA accounts for results from the literature on children’s reasoning as it learns to play video games.

LISAese, Relational Databases, and Generalization

We have argued that the reason that people can learn relations and apply them to new domains is because we learn representations of those relations that specify relational invariants in a form that permits the binding of relational roles to arguments without changing the representation of either (see also Doumas et al., 2008; Halford, Bain, et al., 1998; Halford, Wilson, & Phillips, 1998; Hummel & Biederman, 1992; Hummel & Holyoak, 1997, 2003; Phillips, 2018, 2021; Phillips et al., 1995). It turns out that our proposal has an analog in computer science in the form of relational databases.7

In mathematical logic, a relation is defined as a subset of the Cartesian product of two or more potentially infinite sets. For example, the relation larger-than (), defined over the integers, can be represented as a matrix (a Cartesian product), with integers in the rows and columns, and 1s and 0s in the cells, such that a 1 appears in every cell whose row is larger than its column. More generally, a binary relation between sets A and B is a subset of the Cartesian product A × B for each pair (a, b) over which the relation holds. A relation is thus represented by a characteristic function, \( \chi_R(a, b) \), which maps to 1 if the relation is true for \((a, b)\) and 0 otherwise.

The characteristic function captures the same information as the relational invariants we described in the Introduction. In other words, the characteristic function specifies the content of the relation, so learning the invariant that defines a relation is a matter of executing the characteristic function. The circuit DORA uses to discover relational invariants is nothing more than an implementation of the characteristic function of, for example, larger-than () over a rate-coded neural representation of magnitude.

However, we have argued that simply expressing an invariant (the output of the characteristic function) is not sufficient to support cross-domain generalization. It is also necessary to somehow represent the dynamic binding of arguments to roles of the relation in a way that preserves the identity of both the relational roles and their arguments. The representational format we have employed for this purpose, LISAese, is isomorphic to a relational schema (or relational database) developed in computer science.

A relational schema (see, e.g., Phillips, 2018) corresponds to the headings of a table that describe the roles of the so-related elements. The table consists of a set of rows, representing instances of the relation, and columns, corresponding to the roles of the relation. For example, the relational schema for the relation larger-than (x, y) includes two columns, one specifying the larger item and the other specifying the smaller, with each row of the table an instance of the larger-than relation. This representational format has the property that relations (tables) are represented explicitly, as are their roles (columns), and arguments (cells), while simultaneously expressing the bindings of arguments to roles without altering the meaning of either.

Halford, Bain, et al. (1998; Halford, Wilson, & Phillips, 1998, Phillips, 2018, 2021; Phillips et al., 1995) have argued that relational schemas are a good model of human mental representations. Specifically, (a) they identify a relation symbolically, (b) the roles (or argument slots) of the relation are represented independently of the

7 We are indebted to Reviewer 3 for their help in making the discussion of correspondence between LISAese and relational schemas much stronger.
fillers of those roles, (c) binding of roles to fillers is explicit, (d) the format supports representing higher-order relations (i.e., relations between relations), and (e) the resulting representations have the property of systematicity, meaning that they permit simultaneous expression of the meaning of (a) the relation, (b) its roles, and (c) their composition into a larger expression (see Halfford, Bain, et al., 1998; Halfford, Wilson, & Phillips, 1998, Phillips et al., 1995). Phillips (2018, 2021) has observed that LISAsese is a representational format akin to a relational schema. And indeed, the representations that DORA learns (i.e., LISAsese) satisfy all the properties of a relational database. In DORA relational representations are specified as sets of linked single-place predicates (columns) composing a header, and values bound to those predicates instantiating (rows) of the specific relation.

In a set of experiments, Halfford, Bain, et al. (1998) demonstrated that people’s inferences in a complex learning task are better captured by representations based on relational schemas than simple associations. In their Experiment 2, participants learned to associate a shape and trigram pair with another trigram. The stimuli were composed from three shapes (say, square, circle, triangle) and three trigrams (say, BEJ, FAH, PUW). The shape acted like an operator, and the mapping from shape-trigram pair to the output (i.e., the other trigram) was given by a simple rotation rule. For example, one shape when linked with a trigram mapped to the identical trigram. The second shape when linked to a trigram, mapped to the next trigram from the list (trigram-1 mapped to trigram-2, trigram-2 mapped to trigram-3, and trigram-3 mapped to trigram-1). The third shape when linked to a trigram mapped to the trigram two jumps away (trigram-1 mapped to trigram-3, trigram-2 mapped to trigram-1, and trigram-3 mapped to trigram-2). As expected, participants learned to perform the task after several exposures. The experimenters reasoned that if participants had learned the mappings by association, then when given new trigrams and new shapes that followed the same rotation rule, the participants would need roughly the same number of exposures to learn them. However, if participants had learned the relation (i.e., rotation) between shape-trigram pair and trigram output, then they should be able to apply the relation between two new shapes and trigram sets within a few exposures. Participants did apply the rule to new shapes and trigram within a few trials, indicating that they had learned the relational table.

More recently, Phillips (2018, 2021), using tools from Category Theory, showed that performance on tasks like the relational schema inference task requires structured relational representations (like a relational database) and cannot be accounted for by association alone. Our simulation results resonate with this claim and suggest the capacity extends to cross-domain transfer. As demonstrated in Simulations 2–4, after learning structured relational representations, DORA can learn to play a video game and then immediately generalize to a relational similar (but featurally different) game.

Other Classes of Relations

In the work reported here, we have focused on transitive relations, those that can be defined by differences on a single dimension, such as larger-than () and left-of (). It is natural to ask whether these same principles apply to other, nontransitive relations such as chases (), mother-of (), and loves (). The short answer is yes. Starting with whatever regularities it is given or can calculate from the environment, DORA’s learning algorithm will isolate those invariants, learn structured representations (i.e., functional predicates) that take arguments, and, where appropriate, compose them into relational structures. In short, given a set of invariants (or a means to calculate them), DORA’s learning mechanisms will produce explicit predicate and relational representations of those invariants. DORA will learn structured representations of concepts based on their invariant properties, whether the invariants the system detects are instances of stimulus magnitude or romantic love (see Doumas et al., 2008). The hard part is finding the invariants. And for this problem, the human visual system may give us a leg up.

Nontransitive relations like chase, support, or love fall on a gradient in terms of how spatial they are. A relation like chase is comparatively easy to reduce to spatial properties: There are two objects, a and b, such that the vector characterizing the movement of a is in the direction of the vector characterizing the location of b, and this configuration is maintained through time. The representations necessary to induce this relation are delivered by the visual system. Michotte (1963) showed that participants would overwhelmingly interpret chasing occurring in a situation where two dots moved across a screen (or, in Michotte’s original version, two lights moved on a grid of lights) such that one stayed in front of the other as they moved. Similarly, relations such as support (one object above and in contact with another object) or lift (one object supports and raises another object), are definable in spatial terms. Even a relation like loves might reduce, at least in part, to spatial relations, though. In a study by Richardson et al. (2003), participants were asked to use configurations of objects to represent a relation produced overwhelmingly similar spatial arrangements for relations like love, admire, and hate.

We do not claim that all relations are spatial in origin, or that there are no invariants (e.g., characterizing social relations such as love, hate, friend, adversary, etc.) that have nonspatial origins. On the contrary, we are completely agnostic about the number and nature of the psychological dimensions over which relational invariants might be computed. What we do claim is that any psychologically privileged dimension, whether it be spatial, auditory, social, or what have you, is subject to the kind of invariant isolation and structure-inducing processes embodied in DORA: If there is an invariant, wherever it originates, intersection discovery can find it and DORA can predicate it and use it for inference and cross-domain generalization.

Doumas (2005) and Shurkova and Doumas (2021) proposed a compression mechanism, complementary to DORA’s refinement algorithm, which is a form of chunking. During compression, multiple roles attached to the same object fire together, and a unit learns to respond to that new conjunction as a unitary predicate. Compression allows DORA to combine multiple representations of the same object. For example, if DORA encounters situations in which one element is both larger and occludes some second object, DORA can compress the roles larger and occluder and the roles
smaller and occluded to form a representation like cover \((a, b)\). Such a procedure might serve as a basis for combining primitive transitive relations into more complex relations.

A second question is whether the role-filler representational system DORA uses is sufficient to represent all the relations people learn. Again, the short answer is, at least in principle, yes (as pointed out originally by Leibniz). Formally, any multiplace predicate is representable as a linked set of single-place predicates (Mints, 2001). Therefore, a role-filler system can, at least in principle, be used to represent higher-arity predicates (or relations; recall the distinction, noted above, between a relation qua a relational schema and a function). Models based on role-filler representations account for a large number of phenomena in analogy making, relational learning, cognitive development, perception, and learning (e.g., Doumas & Hummel, 2010; Doumas et al., 2008; Hummel & Biederman, 1992; Hummel & Holyoak, 1997, 2003; Livins & Doumas, 2015; Livins et al., 2015; Martin & Doumas, 2017; Morrison et al., 2004; Sandhofer & Doumas, 2008; Son et al., 2010). Moreover, access to and use of role-based semantic information is quite automatic in human cognition, including during memory retrieval (e.g., Gentner et al., 1993; Ross, 1989), and analogical mapping and inference (Bassok & Olseth, 1995; Kubose et al., 2002; Krawczyk et al., 2004; Ross, 1987). Indeed, the meanings of relational roles influence relational thinking even when they are irrelevant or misleading (e.g., Bassok & Olseth, 1995; Ross, 1989). Role information appears to be an integral part of the mental representation of relations, and role-filler representations provide a direct account for why. Moreover, role-filler systems appear uniquely capable of accounting for peoples’ abilities to violate the “\(n\)-ary restriction,” mapping \(n\)-place predicates to \(m\)-place predicates, where \(n\) is not equal to \(m\) (e.g., mapping “\(x\) murdered \(y\)” onto “\(a\) caused \(b\) to die”; Hummel & Holyoak, 2003).8

Livins et al., 2016 showed that we can affect the direction of a relation by manipulating which item one looks at first, for both obvious similarity and magnitude relations, and the other kinds of relations. Livins et al. showed participants images depicting a relation that could be interpreted in different forms (e.g., chased pursued-by, lifithang). Before the image appeared on screen, a dot appeared on the screen drawing the participant’s attention to a location that one of the objects involved in the relation would appear. For example, the image might show a monkey hanging from a man’s arm, and the participant might be cued to the location where the monkey would appear. The relation that the participant used to describe the image was strongly influenced by the object that they attended to first. That is, if the participant saw the image of the monkey hanging from the man’s arm, and she was cued to the monkey, they would describe the scene using a hanging relation. However, if the participant was cued to the man, she would describe the scene using a lifting relation. This result follows directly from a system based on role-filler representations wherein complementary relations are represented by a similar set of roles, but the predicate, or role, that fires first, defines the subject of the relation.

Limitations and Future Directions

People routinely learn structured representations from experience, an ability we argue is fundamental to our understanding of the world and our ability to use the knowledge we have gained in one context to inform our understanding of another. We offer an account of this process that is based on minimal assumptions, assumptions that, with the exception of the capacity for dynamic role-filler binding, are standard in neural networks such as DNNs. Our account is, of course, limited in several ways. In the following, we outline some of the limitations of our model and suggest ways to address these limitations.

First, the constraints on learning in DORA are underdetermined. DORA learns when it can and stores all the results of its learning. We have implemented a crude form of recency bias in our simulations (biasing retrieval of the most recently learned representations during learning; see Simulation 1), but future work should focus on the development of more principled mechanisms for constraining learning and storage. Such mechanisms might focus on either constraining when learning takes place, or when the results of learning are stored for future processing. Most likely, though, it will be necessary to account for both.

Constraining when DORA learns amounts to constraining when it performs comparison. We have previously proposed several possible constraints on comparison such as language (e.g., shared labels) and object salience, and have shown how direction to compare (i.e., instruction) serves as a very powerful constraint on learning (see Doumas et al., 2008; Doumas & Hummel, 2013). These constraints may also serve to limit when the results of learning are stored in memory. DORA might be extended or integrated with existing accounts of language or perceptual (feature) processing to implement such constraints (see, e.g., Martin & Doumas, 2017).

Both these limitations might be addressed by refining DORA’s control structure. The quality of comparisons DORA makes and the representations it learns may serve as important constraints on the control process it uses. RL provides a useful tool for implementing these constraints.

Second, as pointed out in Simulation 2, we do not have yet a complete solution for the problem of how to select the right representations to build a relational model of a domain from the reward signal when the domain of potential relations is large. In artificial intelligence, the problem of learning relational (a.k.a., first-order) policies has been studied under the name of relational RL (Driessens & Dzeroski, 2004; Driessens & Ramon, 2003; Dzeroski et al., 2001), but these early models do not scale well to large problems involving multiple relations. However, recent models based on differentiable versions of inductive logic programming (Evans & Grefenstette, 2018; Jiang & Luo, 2019) may be a promising approach to this problem. These systems have shown

\[ \text{antonym}(x, y) \text{ is symmetrical, and both arguments play the same role, then in a LISAsm representation of that relation, there will only be a single token for the role (recall T1 tokens are not repeated within an analog). For example, if antonym (x, y) is symmetrical, and both arguments play the same role—like opposite-of—then a single T1 token unit will represent that role in the proposition and both x and y will be bound to that role by distinct T2 role-binding units in LTM and by asynchrony of firing in WM. That is, the relation does not need two distinct roles, but rather is representable (and learnable) as a single role involved in two (or more) role-bindings, with those role-bindings linked to form a higher-arity proposition.} \]
that it is possible under some circumstances to use gradient descent methods to prune a prebuilt large set of rules to obtain a program (i.e., a refined set of rules) that allows the agent to interact effectively with the environment. We are currently working toward incorporating this kind of error-correction learning with DORA.

The discovery of invariance has relevance beyond the few problems presented here. For example, detecting invariants in speech and language is a defining and unsolved problem in language acquisition and adult speech processing, including in automatic speech recognition by machines. Similarly, whether the generalization of grammatical rules can be fully accounted for in systems that rely on statistical learning alone remains contentious. The account of learning invariance from experience offered here, combined with principles like the compression of role information (Doumas, 2005), may present new computational vistas on these classic problems in the language sciences (see Martin, 2016; Martin & Doumas, 2017, for further discussion). Systems with the properties of DORA may offer an inroad to representational sufficiency across multiple domains, built from the same mechanisms and computational primitives.

Conclusion

A cognitive architecture that is prepared to learn structured representations of relations is prepared to generalize broadly based on those relations. This kind of generalization includes cross-domain transfer as a special case. In fact, it is a mundane consequence of the way people conceptualize the world.

Purely statistical learning systems will most likely continue to outperform people at any single task on which we choose to train them. But people, and cognitive architectures capable of learning relations in an open-ended fashion, will continue to outperform any finite set of purely statistical systems as generalists. And general intelligence, we argue, is not the capacity to be optimal at one task, but is instead the capacity to excel, albeit imperfectly, at many.

References

Anderson, J. R. (2007). How can the human mind occur in the physical universe? Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195224259.001.0001

Bassok, M., & Olseth, K. L. (1995). Judging a book by its cover: Interpretative effects of content on problem-solving transfer. Memory & Cognition, 23(3), 354–367. https://doi.org/10.3758/BF03197236

Battaglia, P. W., Hamrick, J. B., Bastp, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Guclelhe, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., ... Pascual, R. (2018). Relational inductive biases, deep learning, and graph networks. ArXiv.

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. Psychological Review, 94(2), 115–147. https://doi.org/10.1037/0033-295X.94.2.115

Biederman, I. (2013). Human object recognition: Appearance vs. shape. In S. J. Dickinson & Z. Pizlo (Eds.), Shape perception in human and computer vision (pp. 387–397). Springer. https://doi.org/10.1007/978-1-4471-5195-1_26

Bonds, A. B. (1989). Role of inhibition in the specification of orientation selectivity of cells in the cat striate cortex. Visual Neuroscience, 2(1), 41–55. https://doi.org/10.1017/S0952523800004314

Bowers, J. S. (2017). Parallel distributed processing theory in the age of deep networks. Trends in Cognitive Sciences, 21(12), 950–961. https://doi.org/10.1016/j.tics.2017.09.013

Cadene, R., Ben-Younes, H., Cord, M., & Thome, N. (2019). Murel: Multimodal relational reasoning for visual question answering. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 1989–1998). Institute of Electrical and Electronics Engineers.

Carey, S. (2000). The origin of concepts. Journal of Cognition and Development, 1(1), 37–41. https://doi.org/10.1207/s15326747jcd0101n3_3

Carey, S. (2009). Oxford series in cognitive development: Vol. 3. The origin of concepts. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195367638.001.0001

Carpenter, G. A., & Grossberg, S. (1990). ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. Neural Networks, 3(2), 129–152. https://doi.org/10.1016/0893-6080(90)90085-Y

Choplin, J. M., & Hummel, J. E. (2002). Magnitude comparisons distort mental representations of magnitude. Journal of Experimental Psychology: General, 131(2), 270. https://doi.org/10.1037.0096-3451.131.2.270

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. Behavioral and Brain Sciences, 24(1), 87–114. https://doi.org/10.1017/s0140525x0003922

Demer, J. L. (2002). The orbital pulley system: A revolution in concepts of orbital anatomy. Annals of the New York Academy of Sciences, 956(1), 17–32. https://doi.org/10.1111/j.1479-6632.2002.tb08205.x

Doumas, L. A. A. (2005). A neural-network model for discovering relational concepts and learning structured representations. Learning from delayed rewards [PhD thesis]. University of California, Los Angeles.

Doumas, L. A. A., & Hummel, J. E. (2013). Comparison and mapping facilitate relation discovery and predication. PLOS ONE, 8(6), Article e63889.

Doumas, L. A. A., & Hummel, J. E. (2005). Approaches to modeling human mental representations: What works, what doesn’t and why. In K. J. Holyoak & R. G. Morrison (Eds.), The Cambridge handbook of thinking and reasoning (pp. 73–94). Cambridge University Press.

Doumas, L. A. A., & Hummel, J. E. (2010). A computational account of the development of the generalization of shape information. Cognitive Science, 34(4), 698–712. https://doi.org/10.1111/j.1551-6709.2010.01103.x

Doumas, L. A. A., & Hummel, J. E. (2012). Computational models of higher cognition. In K. J. Holyoak & R. G. Morrison (Eds.), The Oxford handbook of thinking and reasoning (pp. 52–66). Oxford University Press.

Doumas, L. A. A., Hummel, J. E., & Sandhofer, C. M. (2008). A theory of the discovery and predication of relational concepts. Psychological Review, 115(1), 1–43. https://doi.org/10.10370033-295X.115.1.1

Doumas, L. A. A., & Martin, A. E. (2018). Learning structured representations from experience. Psychology of Learning and Motivation, 69, 165–203. https://doi.org/10.1016/bsl.pml.2018.10.002

Driessens, K., & Džeroski, S. (2004). Integrating guidance into relational reinforcement learning. Machine Learning, 57(3), 271–304. https://doi.org/10.1023/B:MACI.0000039779.47329.3a

Džeroski, S., De Raedt, L., & Driessens, K. (2001). Relational reinforcement learning. Machine learning, 43(1), 7–52. https://doi.org/10.1023/A:1007694015589

Driessens, K., & Ramon, J. (2003). Relational instance based regression for relational reinforcement learning. Proceedings of the 20th International Conference on Machine Learning (ICML-03) (pp. 123–130).

Engel, S. A., Murmelhart, D. E., Wandell, B. A., Lee, A. T., Glover, G. H., Chichilnisky, E. J., & Shadlen, M. N. (1994). FMRI of human visual cortex. Nature, 369, Article 525. https://doi.org/10.1038/369525a0

Evans, R., & Grefenstette, E. (2018). Learning explanatory rules from noisy data. Journal of Artificial Intelligence Research, 61, 1–64. https://doi.org/10.1613/jair.5714

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. Artificial Intelligence, 41(1), 1–63. https://doi.org/10.2307/423470

Foley, J. M. (1994). Human luminance pattern–vision mechanisms: Masking experiments require a new model. Journal of the Optical Society of
Morrison, R. G., Doumas, L. A., & Richland, L. E. (2011). A computational account of children’s analogical reasoning: Balancing inhibitory control in working memory and relational representation. *Developmental Science, 14*(3), 516–529. https://doi.org/10.1111/j.1467-6877.2010.00999.x

Morrison, R. G., Krawczyk, D. C., Holyoak, K. J., Hummel, J. E., Chow, T. W., Miller, B. L., & Knowlton, B. J. (2004). A neurocomputational model of analogical reasoning and its breakdown in frontotemporal lobar degeneration. *Journal of Cognitive Neuroscience, 16*(2), 260–271. https://doi.org/10.1162/089892904322984553

Murphy, G. L., & Medin, D. L. (1985). The role of theories in conceptual coherence. *Psychological Review, 92*(3), 289–316. https://doi.org/10.1037/0033-295X.92.3.289

Nelson, K., & Benedict, H. (1974). The comprehension of relative, absolute, and contrastive adjectives by young children. *Journal of Psycholinguistic Research, 3*(4), 333–342. https://doi.org/10.1007/BF01068168

Phillips, S. (2018). Going beyond the data as the patching (sheaving) of local knowledge. *Frontiers in Psychology, 9*, Article 1926. https://doi.org/10.3389/fpsyg.2018.01926

Phillips, S. (2021). A reconstruction theory of relational schema induction. *PLOS Computational Biology, 17*(1), Article e1008641. https://doi.org/10.1371/journal.pcbi.1008641

Phillips, S., Halford, G. S., & Wilson, W. H. (1995). The processing of association versus the processing of relations and symbols: A systematic comparison. In J. D. Moore & J. F. Lehman (Eds.), *Proceedings of the seventeenth annual conference of the cognitive science society* (pp. 688–691). Lawrence Erlbaum.

Piaget, J. (1954). *The construction of reality in the child*. Basic Books.

Rabagliati, H., Doumas, L. A., & Bemis, D. K. (2017). Representing composed meanings through temporal binding. *Cognition, 162*, 61–72. https://doi.org/10.1016/j.cognition.2017.01.013

Rao, A., & Cecchi, G. (2016). The effects of feedback and lateral connections on perceptual processing: A study using oscillatory networks. In *International Joint Conference on Neural Networks (IJCNN)* (pp. 1177–1184). Institute of Electrical and Electronics Engineers.

Rao, A. R., & Cecchi, G. A. (2010). An objective function utilizing complex sparsity for efficient segmentation in multi-layer oscillatory networks. *International Journal of Intelligent Computing and Cybernetics, 3*(2), 173–206. https://doi.org/10.1108/1756381011049160

Rattermann, M. J., & Gentner, D. (1998). The effect of language on similarity: The use of relational labels improves young children’s performance in a mapping task. In K. Holyoak, D. Gentner & B. Kokinov (Eds.), *Advances in analogy research: Integration of theory and data from the cognitive, computational, and neural sciences* (pp. 274–282). New Bulgarian University.

Reichert, D., & Serre, T. (2014). Neuronal synchrony in complex-valued deep networks. In *International conference on learning representations (ICLR-2014)*.

Rich, A. S., & Gureckis, T. M. (2018). The limits of learning: Exploration, generalization, and the development of learning traps. *Journal of Experimental Psychology: General, 147*(11), 1553. https://doi.org/10.1037/xge0000466

Ricci, M., Cadène, R., & Serre, T. (2021). Same-Different conceptualization: A machine vision perspective. *Current Opinion in Behavioral Sciences, 37*, 47–55.

Richardson, D. C., Spivey, M. J., Barsalou, L. W., & McRae, K. (2003). Spatial representations activated during real-time comprehension of verbs. *Cognitive Science, 27*(5), 767–780. https://doi.org/10.1016/S0364-0213(03)00064-8

Richland, L. E., Morrison, R. G., & Holyoak, K. J. (2006). Children’s development of analogical reasoning: Insights from scene analogy problems. *Journal of Experimental Child Psychology, 94*(3), 249–273. https://doi.org/10.1016/j.jecp.2006.02.002

Ross, B. H. (1987). This is like that: The use of earlier problems and the separation of similarity effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 13*(4), 629–639. https://doi.org/10.1037/0278-7393.13.4.629

Ross, B. H. (1989). Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*(3), 456–468. https://doi.org/10.1037/0278-7393.15.3.456

Sandhofer, C. M., & Doumas, L. A. (2008). Order of presentation effects in learning color categories. *Journal of Cognition and Development, 9*(2), 194–221. https://doi.org/10.1080/15248370802022639

Sandhofer, C. M., & Smith, L. B. (1999). Learning color words involves learning a system of mappings. *Developmental Psychology, 35*(3), 668–679. https://doi.org/10.1037/0012-1649.35.3.668

Sandhofer, C. M., Smith, L. B., & Luo, J. (2000). Counting nouns and verbs in the input: Differential frequencies, different kinds of learning? *Journal of Child Language, 27*(3), 561–585. https://doi.org/10.1017/S0305000900004256

Sanorto, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascana, R., Battaglia, P., & Lillicrap, T. (2017). A simple neural network module for relational reasoning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Eds.), *Advances in neural information processing systems* (pp. 4967–4976). Curran.

Sengupta, B., & Friston, K. J. (2018). How robust are deep neural networks? *ArXiv:1804.11313.*
Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Taylor, E. G., & Hummel, J. E. (2009). Finding similarity in a model of relational reasoning. *Cognitive Systems Research, 10*(3), 229–239. https://doi.org/10.1016/j.cogsys.2008.09.004

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *Science, 331*(6022), 1279–1285. https://doi.org/10.1126/science.1192788

Thomas, J. P., & Olzak, L. A. (1997). Contrast gain control and fine spatial discriminations. *Journal of the Optical Society of America. A, Optics, Image Science, and Vision, 14*(9), 2392–2405. https://doi.org/10.1364/JOSAA.14.002392

Usher, M., & Niebur, E. (1996). Modeling the temporal dynamics of IT neurons in visual search: Mechanism for top-down selective attention. *Journal of Cognitive Neuroscience, 8*(4), 311–327. https://doi.org/10.1162/jocn.1996.8.4.311

Viskontas, I. V., Morrison, R. G., Holyoak, K. J., Hummel, J. E., & Knowlton, B. J. (2004). Relational integration, inhibition, and analogical reasoning in older adults. *Psychology and Aging, 19*(4), 581–591. https://doi.org/10.1037/0892-7974.19.4.581

Appendix A

**Computational Details**

For completeness, we provide full implementational details of DORA’s operation below. Code for the model is available online (see Author note).

**Parts of Discovery of Relations by Analogy (DORA)**

As described in the main text, DORA consists of an LTM composed of three bidirectionally connected layers of units. Units in LTM are referred to as token units (or tokens). Units in the lowest layer of LTM are connected to a common pool of feature units. Token units are yoked to integrative inhibitors that integrate input from their yoked unit and token units in higher layers.

DORA learns representations of a form we call LISAsese via unsupervised learning. Propositions in LISAsese are coded by a hierarchy of units in layers of a neural network (see main text). At the bottom of the hierarchy, feature (or semantic) nodes code for the featural properties of represented instances in a distributed manner. At the next layer, localist predicate and object units (T1) conjointively code collections of feature units into representations of objects and roles. At the next layer localist role-binding units (T2) conjointively bind object and role T1 units into linked role-filler pairs. Finally, proposition units (T3) link T2 units to form whole relational structures.

Sets, groups of potentiated units, correspond to attention or working memory (WM) within a cognitive framework. The driver corresponds to DORA’s current focus of attention. The recipient corresponds to active memory. Token units are laterally inhibitive (units in the same layer inhibit one another) within, but not across, sets.

Each layer of token units is negatively connected to a local inhibitor, and all token unit are connected to a global inhibitor. Active token units in a layer inhibit the local inhibitor to inactivity. When no token units in a given layer are active, the local inhibitor becomes active and sends a refresh signal to all tokens in that layer and below across LTM (see below). When no token units in the driver are active, the global inhibitor becomes active, and sends a refresh signal to all tokens across LTM (see below). Each layer of token units is connected to a clamping unit (C), which is excited by unclamped units in the layer below and inhibited by unclamped units in the same layer and the layer above (see below). C units play a role in recruiting and activating token units during learning.

We use the term analog to refer to a complete story, event, or situation (e.g., from a single object in isolation to a full proposition in LISAsese). Analogs are represented by a collection of token units (T1–T3). Token units are not duplicated within an analog (e.g., within an analog, each proposition that refers to Don connects to the same “Don” unit). Separate analogs do have nonidentical token units (e.g., Don will be represented by one T1 unit in one analog and by a different T1 in another analog). The feature units thus represent general type information and token units represent instantiations (or tokens) of those types in specific analogs.

**Functional Overview of Processing in DORA**

In this section, we describe DORA’s operation in strictly functional terms. How these operations are instantiated in the neural network using traditional connectionist computing principles is detailed in the section directly below.

**Retrieval**

When there are items in the driver (i.e., DORA is attending to something), but nothing in the rest of AM, then DORA performs retrieval. In short, some representation b is retrieved into the recipient to the extent that it is similar to a in the driver and prevalent in memory, and other representations c in memory are not similar to a and are not prevalent in memory. Functionally, retrieval works as follows:

\[
\text{Ret}(B) \leftrightarrow f_{\text{Rel}}(\text{sim}(B, A^D), \text{sim}(C \neq B, A^D), p(B), p(C)), \quad (A1)
\]

(Appendices continue)
where \( \text{Ret}(B) \) is a retrieved representation \( B \), \( A^D \) is a driver representation (i.e., a collection of connected token units instantiating a LISAes representation), \( p(B) \) is the prevalence of representation \( B \) in LTM, and \( f_{\text{Ret}} \) is the retrieval function.

### Mapping

When there are items in the driver (i.e., DORA is attending to something), and in the recipient, then DORA performs mapping. In short, representation \( a \) in the driver will map to representation \( b \) in the recipient to the extent that there are correspondences between \( a \) and \( b \), and there are no correspondences between \( a \) and any other items \( c \) in the recipient. Functionally, mapping works as follows:

\[
M(A^D, B^R) \leftarrow f_M(\text{sim}(A^D, B^R), \text{sim}(A^D, C \neq B^R)),
\]

(A2)

where \( M(A, B) \) is a mapping between \( A \) and \( B \) (instantiated as learned bidirectional weighted connections), \( A^D \) is a driver representation, \( B^R \) is a recipient representation, \( C \) are other representations in the recipient that are not \( B^R \), \( \text{sim}(A, B) \) is a similarity function, and \( f_M \) is the mapping function.

### Relation Learning

If DORA has learned mapping connections between representations in driver and recipient, then it can learn from the matching. During learning, there are two possibilities. In the first case, if two objects (\( a \) and \( b \)) that are not already bound to predicates (i.e., no \( T_2 \) units are active) are mapped, then DORA learns a single-place predicate composed of the featural intersection of \( a \) and \( b \). In the second case, when sets of role-filler pairs are mapped—for example, \( P_i(a) \) and \( P_i(b) \) are mapped to \( P_i(c) \) and \( P_i(d) \)—and are not already linked into multiplace relational structures (i.e., \( \text{no } T_3 \text{ units are active} \)), then DORA links one of the mapped pairs (via a \( T_3 \) unit) forming a functional multiplace relation. Functionally, learning can be defined as follows:

\[
E_i \leftarrow \begin{cases} 
\forall \mathcal{P}_i(a), \mathcal{P}_i(b) \leftarrow f_i(M(\mathcal{P}_i(a), \mathcal{P}_i(b))), & \exists T_3 \\
\forall \mathcal{P}_i(a), \mathcal{P}_i(b), \mathcal{P}_i(c), \mathcal{P}_i(d) \leftarrow f_M(M(\mathcal{P}_i(a), \mathcal{P}_i(b)), M(\mathcal{P}_i(c), \mathcal{P}_i(d))), & \exists T_3
\end{cases}
\]

(A3)

where \( E_i \) is a learned representation, \( \mathcal{P}_i(a) \) is a single-place predicate \( i \), in set \( \mathcal{J} \), \( D \) ( \( D \) = driver, \( R \) = recipient, \( M \) = LTM) that takes the argument \( a \), lowercase letters (\( a, b, c, d \)) indicate objects, \( a \cap b \) indicates the intersection of the features of \( a \) and \( b \), \( R_{i,a}^I \) is a relational structure \( i \) (consisting of linked predicate–argument pairs; see Equation A4, directly below) of arity \( n \) in set \( J \), \( T_k \) is an active \( T_k \) token unit, and \( f_M \) is the learning function.

The relational structure \( R_{i,a}^I \) is instantiated in DORA functionally as:

\[
R_{i,a}^I = [\mathcal{P}_i(a) \Leftrightarrow \mathcal{P}_i(c)],
\]

(A4)

where \( \Leftrightarrow \) is a linking operator. For any \( R_k \) a single \( T_3 \) unit instantiates all \( n - 1 \) instances of the \( \Leftrightarrow \) operator in \( R_k \), linking (i.e., conjuncting) predicate–argument pairs \( \mathcal{P}_i(a) \ldots \mathcal{P}_i(c) \).

### Refinement

If DORA has learned mapping connections between representations in driver and recipient, then it can learn a refined (or schematized) representation consisting of the featural intersection of the mapped representations. Refinement is defined as follows:

\[
R_{i,a}^M \leftarrow f_R(M(R_{i,a}^I, R_{i,a}^R)),
\]

(A5)

where \( R_{i,a}^M \) (defined directly below) is a refined relational structure \( i \) of arity \( n \) in set \( J \), and \( f_R \) is the refinement function. The refined structure \( R_{i,a}^M \) is then:

\[
R_{i,a}^M = \left[ p_M^1(a) \Leftrightarrow \ldots p_M^k(b) \right],
\]

(A6)

where \( p_M \) is a single-place predicate composed of the featural intersection of mapped predicates \( p_1 \) and \( p_{\mathcal{J}} \).

### Relational Generalization

If DORA has learned mapping connections between representations in driver and recipient, then it can perform relational generalization, inferring structure from the driver about items in the recipient. Generalization in DORA follows the standard copy-with-substitution-and-generalization format (Holyoak et al., 1994), and can be defined as follows:

\[
G_k^R \leftarrow f_G(M(R_k^I, R_k^R) \& \sim M(R_k^I)),
\]

(A7)

where \( G_k^R \) is a generalized structure (see Equation A8, directly below), \( \sim M(R_k^I) \) is an unmapped structure \( A \), and \( f_G \) is the generalization function. The generalized structure \( G_k^R \) is then:

\[
G_k^R = [R_k^I \& R_k^R],
\]

(A8)

where \( R_k^I \) is the mapped relational structure from Equation A7, and \( R_k^R \) is generalized relational information in the recipient that matches the unmapped \( R_k^I \) from Equation A7.

### Processing in DORA

DORA’s operation is outlined in pseudocode in Figure A1. The details of each step, along with the relevant equations and parameter values, are provided in the subsections that follow. DORA is very robust to the values of the parameters (see Doumas et al., 2008). For equations in this section, we use the variable \( a \) to denote a unit’s activation, \( n \) its (net) input, and \( w_{ij} \) to denote the connection from unit \( i \) to unit \( j \).

An analog, \( F \) (selected at random, or based on the current game screen), enters the driver. Network activations are initialized to 0. Either (a) the firing order of propositions in \( F \) is random (however, see Hummel & Holyoak, 2003, for a detailed description of how a system like DORA can set its own firing order according to the constraints of pragmatic centrality and text coherence), or (b) a roughly random firing order is instantiated by passing a top-down input signal to all units in the highest layer of \( D \) sampled from a uniform distribution with values between 0 and 0.4. DORA performs similarity and relative magnitude calculation through the relational invariance circuit, then runs retrieval from LTM, analogical mapping, and comparison-based unsupervised learning (predicate learning, refinement, and (relational) generalization). Currently, the order

(Appendices continue)
of operations of these routines is set to the order: retrieval, mapping, learning (predicate learning, refinement, and generalization).

Relational Invariance Circuit

The operation of the relational invariance circuit is described in the main text. The relational invariance circuit runs when two or more items (T1 units) are present in the driver and those items are connected to similar magnitude representations (e.g., pixels, etc.). It is inhibited to inactivity when two or more circuit output features are active above threshold (\(= .9\)) when both T1 units are active. As noted in the main text, we presently make no strong commitment to an account of dimensional salience (as discussed in, e.g., Spelke & Kinzler, 2007). As such, two T1 units in the driver, both encoding an object or both encoding a predicate, are selected at random, and when those items are connected to multiple dimensions, a dimensional encoding becomes active at random (activation of features encoding that magnitude for the two T1 units are clamped to 1). The choice of two T1 units follows work on the WM capacity of children (e.g., Halford, Bain, et al., 1998). As described in the main text, proxy units connected to the active driver T1 units update their activation and input by Equations 4 and 5, respectively. $E$ unit activation is updated by Equation 7. Feature units connected to $E$ units update their input by Equation 8, and their activation by Equation 9. Weights between feature units and active T1 units are updated by Equation 10.

![Figure A1](image)

**Pseudocode of Processing in Discovery of Relations by Analogy (DORA)**

| Processing step | Description |
|-----------------|-------------|
| Items $F$ in $D$ | < Potentiated units, $F$, in driver, $D$. |
| $w(\tau_i, \tau_j, a_k) \leftarrow LE(\tau_i, \tau_j)$ | < Run relational invariance circuit. |
| For operation in \{retrieving, mapping, predicate learning, refining, generalizing\} | < Perform DORA operations in sequence. |
| For $i \in F$ : | < $F$ is the set of units at the highest token layer for any set of connected units. |
| Until $y_i > \theta_l$ : | < While unit $i$’s yoked inhibitor has not fired. |
| \{\(a_j, n_j, y_j, l\) \leftarrow f(a_j, a_k, w_{jk}, y_j, l)\} | < Update inputs and activation of units in the network. |
| When mapping : | < During mapping. |
| \(\Delta h_{jk} \leftarrow a_j a_k\) | < If there are mapping hypotheses, $h$, between units $j$ and $k$. |
| If \(\exists (m_{d,R})\) : | < During predicate learning. |
| When predicate learning : | < Activation of recruited token $\tau_j$ in $R$ layer $L$ is clamped to 1 as a function of the activation of unclamped $\tau_j$ and $\tau_{xz}$ at $L-1$ and $L$ respectively. |
| \(a_{\tau_l,R,L} = 1\) \leftarrow \left(\Sigma a_{\tau_l,R,L} - \Sigma a_{\tau_{xz},R,L}\right) > \theta_p\) | < During refinement. |
| When generalizing : | < Activation of recruited token unit, $\tau_z$, in long term memory, $LTM$, layer $L$, clamped to 1 to match active $\tau_z$ in $D$ layer $L$ with mapping connections. |
| \(a_{\tau_x,R,L} = 1\) \leftarrow a_{\tau_y,D,L} \cap \exists m_{\tau_y,R}\) | < During generalization. |
| $\Delta w_{ij} \leftarrow a_i(a_j - w_{ij})\gamma$ | < (2.3.3) Activation of recruited $\tau_z$ in $R$ layer $L$ clamped to 1 to match active $\tau_z$ in $D$ layer $L$ with no mapping connections. |
| End If | < Update weights via Hebbian learning. |
| End For | |
| When retrieving : | < During retrieval. |
| $g_R \leftarrow p\left(\frac{A(g)}{\sum A(h)}\right), h \neq g$ | < Activated, $A$, units $g$, retrieved into recipient, $R$. |
| When mapping : | < During mapping. |
| $\Delta m_{jk} \leftarrow \eta(1.1 - h_{jk} \eta_{jk})^2$ | < Activated, $A$, units $g$, retrieved into recipient, $R$, via the Luce choice rule. |
| End For | |

(Appendices continue)
Main DORA Operations

Repeat the following until each token unit $i$ in the highest layer of $F$ has fired three times if mapping, or once, otherwise (each token unit at the highest layer firing is referred to as the phase set). If a firing order has been set, select the current unit $i$ in the firing order and set $a_i$ to 1.0. Otherwise, pass a top-down input $(n = \text{unif}(0, 0.4))$ to token units in the highest layer of $F$.

Update Inputs and Activations of Network Units

**Update Mode of All T3 Units in Driver and Recipient**

T3 units in all propositions operate in one of three modes: Parent, child, and neutral (Hummel & Holyoak, 1997, 2003). T3 mode is important for representing higher-order relations (e.g., $R1(x, R2(y, z))$; Hummel & Holyoak, 1997). As detailed below, higher-order relations are represented in DORA such that if one proposition takes another as an argument, the T3 unit of the lower-order proposition serves as the object of an T2 unit for the higher-order proposition. By contrast, when a T3 unit is not acting as the argument of another proposition, it operates in parent mode. The mode $m_i$ of T3 unit $i$ is updated by the equation:

$$m_i = \begin{cases} \text{Parent}(1), & T2_{\text{above}} < T2_{\text{below}} \\ \text{Child}(-1), & T2_{\text{above}} > T2_{\text{below}}, \\ \text{Neutral}(0), & \text{otherwise} \end{cases}$$

where $T2_{\text{above}}$ is the summed input from all T2 units to which $i$ is upwardly connected (i.e., relative to which $i$ serves as an argument), and $T2_{\text{below}}$ is the summed input from all T2 units to which $i$ is downwardly connected. In the current simulations, T3 mode did not have to change their mode. We include this step here solely for the purposes of completeness (see Hummel & Holyoak, 1997, 2003, for details).

**Update Input to All Units in the Network**

**Update Input to All Token Units in Driver.** Token units in the driver update their input by the equation:

$$n_i = \sum_j a_j w_{ij} + \text{SEM}_i \Gamma + M_i - \sum_k a_k - s \sum_m 3a_m - \sum_n a_n - \Gamma_G - \Gamma_L,$$

where $j$ is any unit above token unit $i$ (i.e., T3 units for T2 units, T2 units for T1 units; input from $j$ is only included on phase sets beyond the first), $\text{SEM}_i$ is the feature input to unit $i$ if unit $i$ is a PO, and 0 otherwise, $M_i$ is the mapping input to unit $i$, $k$ is all units in either recipient (if unit $i$ is in recipient) or LTM (if unit $i$ is in LTM) in the same layer as $i$ (for T1 units, $k$ is only those T1 units and T3 units currently in child mode units not connected to the same T2 as unit $i$); see the Update Mode of All T3 Units in Driver and Recipient section), $m$ is T1 units connected to the same T2 as $i$ (or 0 for non-T1 units), $n$ is units above unit $i$ to which unit $i$ is not connected, $\Gamma_G$ is the activation of the global inhibitor (see below), and $\Gamma_L$ is the activation of the local inhibitor in the same layer as or any layer above $i$. When DORA is operating in binding-by-asynchrony mode, $s = 1$; when it is operating in binding-by-synchrony mode (i.e., like LISA), $s = 0$. $\text{SEM}_i$, the feature input to $i$, is calculated as:

$$\text{SEM}_i = \frac{\sum_j a_j w_{ij}}{1 + \text{num}(j)},$$

where $j$ is feature units, $w_{ij}$ is the weight between feature unit $j$ and T1 unit $i$, and $\text{num}(j)$ is the total number of feature units $i$ is connected to with a weight above 0 ($\approx 0.1$). $M_i$ is the mapping input to $i$:

$$M_i = \sum_j a_j (3w_{ij} - \text{Max(Map}(i)) - \text{Max(Map}(j))),$$

where $j$ is token units of the same type as $i$ in driver (e.g., if $i$ is a T2 unit, $j$ is all T2 units in driver), Max$(Map(i))$ is the highest of all unit $i$’s mapping connections, and Max$(Map(j))$ is the highest of all unit $j$’s mapping connections. As a result of Equations A14, active token units in driver will excite any recipient units of the same type to which they map and inhibit all recipient units of the same type to which they do not map.

(Appendices continue)
**Update Input to the Yoked Inhibitors**

Every token unit is yoked to an inhibitor unit \( i \). T2 and T3 inhibitors are yoked only to their corresponding T2. T1 inhibitors are yoked both to their corresponding T1 and all T2 units in the same analog. Inhibitors integrate input over time as:

\[
\Delta n_i^{(t+1)} = \Delta n_i^{(t)} + \alpha_i + \sum_k \alpha_k,
\]

(A15)

where \( t \) refers to the current iteration, \( j \) is the token unit yoked to inhibitor unit \( i \), and \( k \) is any T2 units if \( j \) is a T1, and 0 otherwise. Inhibitor units become active \((a_i = 1)\) when \( n_i \) is greater than the activation threshold \((\Gamma_i = 0)\) by any driver unit in its layer with activation above \( \Theta_{IN} \) or \( \Theta_{IN} = 0.5 \), and becomes active \((\Gamma_i = 10)\) when no token unit in its layer has an activity above \( \Theta_{IN} \). A firing local inhibitor sets the activation and potentiation of all other local inhibitors below and including itself to 0. The global inhibitor, \( \Gamma_G \), is potentiated \((P(\Gamma_G) = 1)\) when any driver unit in \( \Gamma_G \)’s layer is active, is inhibited to inactivity \((\Gamma_G = 0)\) by any driver unit in its layer with activation above \( \Theta_G \) \((= 0.5)\), and becomes active \((\Gamma_G = 10)\) when no token unit in its layer has an activity above \( \Theta_G \). A firing local inhibitor sets activation and potentiation of all other local inhibitors to 0.

**Update the Local and Global Inhibitors**

The local inhibitors, \( \Gamma_L \), are inhibitory units connected to all units in a single layer of LTM (i.e., there is a local inhibitor for T1 units, another for T2 units). The local inhibitor is potentiated \((P(\Gamma_L) = 1)\) when a driver unit in \( \Gamma_L \)’s layer is active, is inhibited to inactivity \((\Gamma_L = 0)\) by any driver unit in its layer with activation above \( \Theta_L \) \((= 0.5)\), and becomes active \((\Gamma_L = 10)\) when no token unit in its layer has an activity above \( \Theta_L \). A firing local inhibitor sets the activation and potentiation of all other local inhibitors below and including itself to 0. The global inhibitor, \( \Gamma_G \), is potentiated \((P(\Gamma_G) = 1)\) when any driver unit in its layer is active, and is inhibited to inactivity \((\Gamma_G = 0)\) by any driver unit in its layer with activation above \( \Theta_G \) \((= 0.5)\), and becomes active \((\Gamma_G = 10)\) when no T1 in its layer has an activity above \( \Theta_G \). The global inhibitor sets activation and potentiation of all other local inhibitors to 0.

**Update Activations of All Units in the Network**

All token units in DORA update their activation by the leaky integrator function:

\[
\Delta a_i = \gamma n_i (1.1 - a_i) - \delta a_i^0,
\]

(A16)

where \( \Delta a_i \) is the change in activation of unit \( i \), \( \gamma \) \((= 0.3)\) is a growth parameter, \( n_i \) is the net input to unit \( i \), and \( \delta \) \((= 0.1)\) is a decay parameter. Activation of all token units \( i \), is hard limited to between 0 and 1 inclusive.

Feature units update their activation by the equation:

\[
a_i = \frac{n_i}{\max(n_j)},
\]

(A17)

where \( n_i \) is the activation of feature unit \( i \), and \( n_j \) is the net input to feature unit \( i \), and \( \max(n_j) \) is the maximum input to any feature unit. There is physiological evidence for divisive normalization in the feline visual system (e.g., Bonds, 1989; Heeger, 1992), and psychophysical evidence for divisive normalization in human vision (e.g., Foley, 1994; Thomas & Olzak, 1997).

Token unit inhibitors, \( i \), update their activations according to a threshold function:

\[
a_i = \begin{cases} 1, & n_i > \Theta_{IN} \\ 0, & \text{otherwise} \end{cases},
\]

(A18)

where \( \Theta_{IN} = 220 \) for T1 and T2 units and \( 220 \times n \) for T3 units (where \( n \) is the number of T2 units to which that T3 unit is connected).

**Update Mapping Hypotheses**

If mapping is licensed, DORA learns mapping hypotheses between all token units in driver and token units of the same type in recipient, that is, between T3 units, between T2 units and between T1 units in the same mode (described below). Mapping hypotheses initialize to zero at the beginning of a phase set. The mapping hypothesis between a driver unit and a recipient unit of the same type is updated by the equation:

\[
\Delta h^i_j = a^i a'^i_j,
\]

(A19)

where \( a^i_j \) is the activation of driver unit \( i \) at time \( t \).

**Comparison-Based Unsupervised Learning**

If licensed, DORA will perform comparison-based learning (CBL). CBL is unsupervised. In the current version of the model, learning is licensed whenever 70% of driver token units map to recipient items (this 70% criterion is arbitrary, and in practice either 0% or 100% of the units nearly always map).

**Predicate and Relation Learning**

During predicate and relation learning, DORA recruits (and clamps the activation of) token units in the recipient to respond to patterns in firing in adjacent layers. The recruitment procedure is a simplified version of ART (Carpenter & Grossberg, 1990). Each layer of token units \( i \) is connected to a clamping unit \( C_i \), which 10 iterations after any inhibitor unit has fired, is activated by the equation:

\[
C_j = \begin{cases} 1, & \sum \max(a_i) - \sum \max(a_j) \geq \Theta_{C}, \\ 0, & \text{otherwise} \end{cases},
\]

(A20)

where \( a_i \) is the activation of unclamped token units in the layer below \( i \) for T2 and T3 units, \( a_j \) is the activation of unclamped token units in layer \( j \), \( \max(a_j) \) is the maximum activation of a unit in layer \( j \), and \( \Theta_{C} \) is a threshold \((=0.6)\). \( C_i \) for the T1 layer is equal to \( C_j \) for the T2 layer.

An active \( C_i \) in the recipient sends an input, \((p_i = 1.0)\), to a randomly selected token unit, \( j \) (where \( j \) is not connected to units in other layers), in layer \( i \) \((p_k = 0 \text{ for all units } k \neq j)\). Token units are clamped by the equation:

\[
c_j = \begin{cases} 1, & p_j - 3 \sum_a a_k > 0, \\ 0, & \text{otherwise} \end{cases},
\]

(A21)

where \( c_j \) is the clamped activation of unit \( j \) in layer \( i \), and \( a_k \) is the activation of all clamped token units in the same layer as \( j \), where \( k \neq j \), if \( j \) is in T1, and all token units in the same layer as \( j \),
where \( k \neq j \), otherwise. Unit \( j \) remains clamped until \( \Gamma_j \) fires, and \( j \) is inhibited to inactivity. If the recruited token is in T1 its mode is set to 1 (marking it as a learned representation; although the idea of units firing in modes sounds nonneural, Hummel et al. (1994) described how it can be accomplished with two or more auxiliary nodes with multiplicative synapses) and connections between the recruited token unit and all active features update by the equation:

\[
\Delta w_{ij} = a_i(a_j - w_{ij})/\gamma,
\]

(A22)

where \( \Delta w_{ij} \) is the change in weight between the new T1 unit \( i \), and feature unit \( j \), \( a_i \) and \( a_j \) are the activations of \( i \) and \( j \), respectively, and \( \gamma \) is a growth rate parameter. Additionally, connections between corresponding token units (i.e., between T3 and T2, or T2 and T1 units) are also updated by Equation A22, where \( i \) is recipient token units in layers adjacent to recruited unit \( j \). When the phase set ends, connection weights between a T2 or T3 unit \( i \) and any token unit in the adjacent lower layer \( j \) (i.e., \( j \) is a T2 unit when \( i \) is a T3 unit, and \( j \) is a T1 unit when \( i \) is a T2 unit), are updated by the equation:

\[
w_{ij} = \begin{cases} w_{ij}, & \sum_k w_{ik} \geq 2 \\ 0, & \text{otherwise} \end{cases},
\]

(A23)

where \( k \) is all other units, including \( j \), in the same layer as \( j \). This operation removes weights to redundant tokens that do not conjunct two or more units at a lower layer.

**Refinement Learning**

During refinement, DORA infers token units in the LTM that match active tokens in the driver. Specifically, DORA infers a token unit in the LTM in response to any mapped token unit in the driver. If unit \( j \) in the driver maps to nothing in the LTM, then when \( j \) fires, it will send a global inhibitory signal to all units in the LTM (Equation A14). This uniform inhibition, unaccompanied by any excitation in recipient, is a signal that DORA exploits to infer a unit of the same type (i.e., T1, T2, T3) in LTM. Inferring T1 units in the LTM have the same mode as the active T1 in driver. The activation of each inferred unit in the LTM is set to 1. DORA learns connections between corresponding active tokens in the LTM (i.e., between T3 and T2 units, and between T2 and T1 units) by the simple Hebbian learning rule in Equation A22 (where unit \( j \) is the newly active token unit, and unit \( i \) is the other active token unit). Connections between T1 units and feature units are updated by Equation A22. When the phase set ends, connection weights between any T2 or T3 unit \( i \) and any token units in an adjacent lower layer \( j \) to which \( i \) has connections are updated by Equation A23.

**Retrieval**

DORA uses a variant of the retrieval routine described in Hummel and Holyoak (1997). During retrieval units in the driver fire as described above for one phase set. Units in the LTM become active as in Step 2.1. After one phase set representations are retrieved from LTM into the recipient probabilistically using the Luce choice axiom:

\[
L_i = \frac{\gamma_i}{\sum_j \gamma_j},
\]

(A24)

where \( L_i \) is the probability that T3 unit \( i \) will be retrieved into working memory, \( \gamma_i \) is the maximum activation T3 unit \( i \) reached during the retrieval phase set and \( j \) is all other T3 units in LTM. If a T3 unit is retrieved from LTM, the entire structure of tokens (i.e., connected T1 \( \ldots \) T3 units) are retrieved into recipient.

**Update Mapping Connections**

If DORA is mapping, mapping connections are updated at the end of each phase set. First, all mapping hypotheses are normalized by the equation:

\[
h_{ij} = \left( \frac{h_{ij}}{\text{MAX}(h_{ij}, h_{ij}^\prime)} \right) - \text{MAX}(h_{ij}),
\]

(A25)

where \( h_{ij} \) is the mapping hypothesis between units \( i \) and \( j \), \( \text{MAX}(h_{ij}, h_{ij}^\prime) \) is the largest hypothesis involving either unit \( i \) or unit \( j \), and \( \text{MAX}(h_{ij}) \) is the largest mapping hypothesis where either \( k = i \) and \( l \neq j \), or \( l = j \) and \( k \neq i \). That is, each mapping hypothesis is normalized divisive: Each mapping hypothesis, \( h_{ij} \), between units \( i \) and \( j \), is divided by the largest hypothesis involving either unit \( i \) or \( j \). Next, each mapping hypothesis is normalized subtractively: The value of the largest hypothesis involving either \( i \) or \( j \) (not including \( h_{ij} \) itself) is subtracted from \( h_{ij} \). The divisive normalization keeps the mapping hypotheses bounded between zero and one, and the subtractive normalization implements the one-to-one mapping constraint by forcing mapping hypotheses involving the same \( i \) or \( j \) to compete with one another. Finally, the mapping weights between each unit in driver and the token units in recipient of the same type are updated by the equation:

\[
\Delta w_{ij} = \eta(1 - w_{ij})h_{ij}^{0},
\]

(A26)

where \( \Delta w_{ij} \) is the change in the mapping connection weight between driver unit \( i \) and recipient unit \( j \), \( h_{ij} \) is the mapping hypothesis between unit \( i \) and unit \( j \), \( \eta \) is a growth parameter, and \( \Delta w_{ij} \) is truncated for values below 0 and above 1. After each phase set, mapping hypotheses are reset to 0. The mapping process continues for three phase sets.

(Appendices continue)
Learning the Relational Invariance Circuit

As described in the main text, the relational invariance circuit consists of three layers of nodes. At the top layer, proxy units are connected to individual T1 units in the driver. The next layer, E, consists of four nodes and takes input from any active proxy units. At the bottom are feature units, initially randomly connected to nodes in E. Weights between units in E and feature units are initialized to random numbers between 0 and 0.9, and lateral connections for E are set to −1. Connections between units in E and feature units are updated by Equations 10 in the main text.

Higher-Order Relations

Although they are not necessary for the current simulations, for the purposes of completeness it is important to note that DORA easily represents higher-order relations (i.e., relations between relations; see Doumas et al., 2008; Hummel & Holyoak, 1997). In short, when a proposition takes another proposition as an argument, the T3 unit of the lower-order proposition serves as the object of a T2 unit for the higher-order proposition. For example, in the higher-order relation greater (distance (a, b), distance (c, d)), the T3 unit of the proposition distance (a, b) serves as the argument of the more role of the higher-order greater relation, and the T3 unit of the proposition distance (c, d) serves as the argument of the less role of the higher-order greater relation. When a T3 unit serves as the object of a T2 unit, it operates in child mode (see above). The modes of T3 units change as a function of whether they are receiving top-down input. A T3 unit receiving top-down input from a T2 unit (i.e., when the T3 unit is serving as the argument of that T2 unit) will operate in child mode, while a T3 unit not receiving input from any T2 units will operate in parent mode.

Appendix B
Simulation Details

DORA: Learning Representations From Screens

We used DORA to simulate learning structured representations from screenshots from the game Breakout. This simulation aims to mirror what happens when a child (or adult) learns from experience in an unsupervised manner (without a teacher or guide). While we describe the results in terms of DORA learning to play Breakout and generalizing to Pong, but results were the same when run in the other direction (i.e., train on Pong and generalize to Breakout; Supplementary Results, Figure S1).

For Simulation 2, screens were generated from Breakout during 250 games with random move selection. Each screen from each game was processed with the visual preprocessor that identified objects and returned the raw pixel values as features of those objects. When learning in the world, objects have several extraneous properties. To mirror this point, after visual preprocessing, each object was also attached to a set of 100 additional features selected randomly from a set of 10,000 features. These additional features were included to act as noise, and to make learning more realistic. (Without these noise features, DORA learned exactly as described here, only more quickly.)

DORA learned from object representations in an unsupervised manner. On each learning trial, DORA selected one pair of objects from a screen at random. DORA attempted to characterize any relations that existed between the objects using any relations it has previously learned (initially, it had learned nothing, and so nothing was returned) by selecting a dimension at random and running the two objects through the relational invariance circuit (described above) over that dimension. If the features returned matched anything in LTM (e.g., “more” and “less” “x”), then DORA used that representation from LTM to characterize the current objects. DORA then ran (or attempted to run) retrieval from LTM, the relational invariance circuit, mapping, and representation learning (see above). Learned representations were stored in LTM. We placed the constraint on DORA’s retrieval algorithm such that more recently learned items were favored for retrieval. Specifically, with probability 0.6, DORA attempted to retrieve from the last 100 representations that it had learned. This constraint followed our assumption that items learned more recently are more salient and more likely to be available for retrieval.

The process was identical for Simulation 3, except that instead of screens from Breakout, we used the first 300 images from the CLVR data set for representation learning. In Simulation 4, we had two ablated versions of the model: In the first ablated model (A1), we ablated the inhibitory connections from the onset of the simulation; in the second ablated model (A2), we ablated the inhibitory connections after the model had learned to play Breakout. Representation learning for both models was as in Simulation 1.

DORA: Q-Learning for Game Play

For Simulations 2–4, for a given screen, DORA used the representations it had previously learned to represent the relations between objects on that screen and the previous screen. That is, for any pair of objects, if DORA had learned a representation that characterized the relation between the two objects (in LTM and as measured by the relational invariance circuit), DORA used that representation to characterize the objects.

The relations were then used to form a table of encountered relational states, and Q-learning (Sutton & Barto, 1998) was used to learn the approximate action-value function for Breakout. We used a rule length constraint of two relations per state, reflecting the simplicity of the game and the WM capacity exhibited by humans (Logie et al., 2020). We trained DORA decreasing the learning rate linearly from 0.1 to 0.05 and the exploration rate linearly from 0.1 to 0.01 throughout the training session. We saved the version of the table that yielded the maximum score during the session.

Deep Q-Network

A deep Q-network (DQN; Mnih et al., 2015) was trained to play Breakout and Pong. The raw 210 × 160 frames were preprocessed by first converting their RGB representation to grayscale and downsampling it to a 105 × 80 image. We stacked the last four consecutive frames to form the input of each state.

(Appendices continue)
The input to the neural network was the \(105 \times 80 \times 4\) preprocessed state. The first hidden convolutional layer applied 16 filters of size \(8 \times 8\) with stride \(4\) with a relu activation function. The second hidden convolutional layer applied 32 filters of size \(4 \times 4\) with stride \(2\) with a relu activation function. The third hidden layer was fully connected of size \(256\) with a relu activation function. The output layer was fully connected with size \(6\) and a linear activation function.

We implemented all the procedures of the DQN to improve training stability, in particular: (a) We used memory replay of size \(1,000,000\); (b) We used a target network which was updated every 10,000 learning iterations; (c) We fixed all positive rewards to be \(1\) and all negative rewards to be \(-1\), leaving \(0\) rewards unchanged; and (d) We clipped the error term for the update through the Huber loss.

We also ran the same network using the input from the visual preprocessor described above.

**Supervised DNN**

We trained a DNN in a supervised manner to play Breakout and Pong and tested generalization between games. One network was trained using random frame skipping and the other with fixed frame skipping.

The inputs to the network were the output of the visual preprocessor described above. Specifically, the network took as input the \(x\) and \(y\) positions of the ball and player-controlled paddle, as well as the left paddle for Pong (left as zeros when playing Breakout). The input to the neural network was a vector of size \(24\) corresponding to the preprocessed last seen four frames. This was fed to three fully connected layers of size \(100\) each with a relu activation function.

The output layer was fully connected with size \(6\) and a softmax activation function.

The criteria for training were the correct action to take in order to keep the agent-controlled paddle aligned with the ball. In Breakout if the ball was to the left of the paddle the correct action was “LEFT,” if the ball was to the right of the paddle the correct action was “RIGHT” and if the ball and the paddle were at the same level on the \(x\)-axis the correct action was “NOOP.” In Pong if the ball was higher than paddle the correct action was “RIGHT,” if the ball was lower than paddle the correct action was “LEFT” and if the ball and the paddle were at the same level on the \(y\)-axis the correct action was “NOOP.” This action was encoded as a one-hot vector (i.e., activation of \(1\) for the correct action and \(0\) for all other actions).

**Graph Network**

Graph networks (see Battaglia et al., 2018, for a review) are neural network models designed to approximate functions on graphs. A graph is a set of nodes, edges, and a global feature. The representation of the nodes, edges, and the global attribute encode feature information. A graph network takes as input a graph and returns a graph of the same size and shape, but with updated attributes.

Our graph net agent used an encode-process-decode architecture (Battaglia et al., 2018) where three different graph networks are arranged in series. The first graph net encodes the nodes, edges and global attributes independently, the second graph net performs three recurrent steps of “message passing” and the third graph net decodes the nodes, edges, and global attributes independently.

The graph agent takes in a graph-structured representation of the screen where each object corresponds to a node in the graph. In our simulations, the node representation corresponds to the position, area, color, and velocity of the objects on the screen. In order to use the graph network as a reinforcement learning agent, we set the number of edge attributes to the number of possible actions. In this way, our agent produces a vector of Q-values for each edge, corresponding to the valid actions in each game. To choose actions, the agent takes an argmax across all edges’ Q-values.

To train our agent, we used a replay memory of size \(50,000\). Before training, we feed the replay memory with \(1,600\) memories (i.e., tuples containing a state graph, action, edge, reward, next state graph, and a “done” variable). At each time step, we saved the current memory and sample a batch of 32 memories from the replay memory to train the agent. We used the Adam learning algorithm with a learning rate of \(0.01\) and default learning parameters.