Extracting and Utilizing Abstract, Structured Representations for Analogy

Steven M. Frankland (steven.frankland@princeton.edu)
Princeton University

Taylor W. Webb
Princeton University

Alexander A. Petrov
The Ohio State University

Randall C. O'Reilly
University of California, Davis

Jonathan D. Cohen
Princeton University

Abstract

Human analogical ability involves the re-use of abstract, structured representations within and across domains. Here, we present a generative neural network that completes analogies in a 1D metric space, without explicit training on analogy. Our model integrates two key ideas. First, it operates over representations inspired by properties of the mammalian Entorhinal Cortex (EC), believed to extract low-dimensional representations of the environment from the transition probabilities between states. Second, we show that a neural network equipped with a simple predictive objective and highly general inductive bias can learn to utilize these EC-like codes to compute explicit, abstract relations between pairs of objects. The proposed inductive bias favors a latent code that consists of anti-correlated representations. The relational representations learned by the model can then be used to complete analogies involving the signed distance between novel input pairs (1:3 :: 5:? (7)), and extrapolate outside of the network's training domain. As a proof of principle, we extend the same architecture to more richly structured tree representations. We suggest that this combination of predictive, error-driven learning and simple inductive biases offers promise for deriving and utilizing the representations necessary for high-level cognitive functions, such as analogy.

Keywords: abstract structured representations; analogy; neural networks; predictive learning; relational reasoning;

Introduction

Analogy requires the flexible, yet orderly, transfer of abstract knowledge within and between domains. Although this transfer occasionally enables new theoretical insights (e.g., Rutherford's planetary model of atomic structure or the hydraulic model of blood circulation (Gentner, 1983)), it also provides critical support for basic cognitive functions, such as memory retrieval, categorization, and schema induction (Gick & Holyoak, 1983; Doumas, Hummel, & Sandhofer, 2008; Gentner & Forbus, 2011; Holyoak, 2012). Here, we test the idea that representations of abstract, structural relationships can be derived using simple forms of training. We evaluate whether these representations can, in turn, support higher-level functions such as analogical inference, without any explicit training on analogy itself.

Specifically, we test the idea that abstract, structured representations arise from learning systems that (a) exploit the vast amounts of observational data available to natural agents ((Rao & Ballard, 1999), O'Reilly, Wyatte, and Rohrlich, 2017) coupled with (b) particular inductive biases in the learning algorithm and network architecture.

We explore what particular input representations and model architectures enable the extraction and utilization of this relational knowledge. To do so, we focus on modeling signed distance relations in a 1-dimensional (1D) domain (e.g., analogies such as 1:3 :: 5:7), and conclude by extending the principles to analogies involving a simple family tree structure.

Throughout, we take inspiration from the representational properties of the mammalian Entorhinal Cortex (EC), believed to extract low-dimensional representations of the metric structure common across environments from the transition probabilities between states (Dayan, 1993; Gustafson & Daw, 2011; Stachenfeld, Botvinick, & Gershman, 2017). We then propose a learning procedure that combines error-driven learning and an inductive bias that naturally extracts explicit relational representations from pre-structured, EC-like representations of the current domain. We show that this combination of representation and model architecture can support analogical inference, and extrapolate well outside the network's training domain.

Methods

Input Representations We focus first on the 1D domain. What input representations enable relation learning that can support analogy (e.g., Figure 1A)? Here, we compare place codes, a successor representation (SR) of states, and lower dimensional representations generated by eigendecompositions of this SR matrix, truncated to 10, 5, or 2 components.

For the place code representation, locations are represented as 1xN one-hot vectors, each orthogonal to the others. This set of representations forms an nXn identity matrix (see Figure 1C, upper left panel), and thus lacks intrinsic structural

information about the domain. It is therefore, a priori, a poor candidate for discovering relational structure that could support analogy, which requires knowledge about structural equivalences (e.g., that the difference between 1-3 is the same as 5-7).

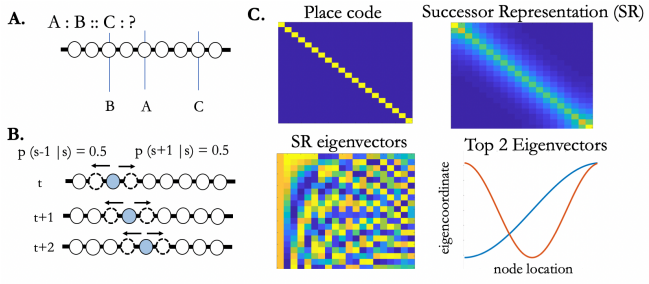

Figure 1: Schematic representation of the 1D domain. (A) depicts the task the network will perform: a signed distance analogy in 1D (here, 5:3 :: 8:?). (B). Following Stachenfeld et al. (2017), we use a random-walk transition matrix to compute the Successor Representation (SR), and derive a low-dimensional representation of the domain by computing the eigendecomposition of the SR. (C) The three forms of representations are compared as inputs to the network: place codes, the SR, and various number of eigenvectors of the SR.

We therefore extend the representation to carry information about the similarity structure of the dimension. To do so, we use the *successor representation* (SR) (Dayan, 1993). The SR is a predictive representation of possible future states, and thus carries information about the dimension's intrinsic structure. To illustrate, imagine, an animal navigating a linear track, with equal probability of moving to spatially adjacent states at successive time points (See Figure 1B). Experience along the track allows learning state-state transition probabilities. The SR encodes the likelihood of visiting a particular location s_i, given a trajectory initiated in a particular state s. The resulting nXn SR matrix is a collection of these conditional probabilities (See Figure 1C, upper right panel). The probability of visitation is high for neighboring states, but low for distant states. Although the SR has primarily been used in learning spatial structure, we take this approach to be representative of continuous dimensions, more broadly (e.g., *brightness*, *loudness*, *pitch*, etc., as well as semantic dimensions such as *animacy*, *size*, or *agreeableness*). The SR can be acquired using a temporal difference learning algorithm (Dayan, 1993). Here, however, we simply stipulate a transition matrix with equal probability of transitioning to adjacent states and a discounted value of future states of 0.9 (results are robust to this parameter). For the SR, the resulting input representation is an nXn matrix in which nearby states have similar representations.

Representing this similarity structure makes the SR a better candidate for analogical inference than a pure place code. However, it is still a high-dimensional representation that does not compactly encode variation along the dimension of interest. Critically, recent work has shown that a spectral representation of the SR provides an abstract, implicitly structured representation of the domain (Stachenfeld et al., 2017). Intriguingly, this spectral representation of the SR resembles the representational properties of grid-like cells in medial entorhinal cortex. Entorhinal cortex (EC) is believed to encode low-dimensional representations of the metric structure of the environment (Hafting et al., 2005; Gustafson and Daw, 2011; Stachenfeld et al., 2017). Here, we take grid-cells in EC to reflect a more general idea: the decomposition of transition probabilities into low-dimensional embeddings provides an abstract, implicitly structured representation of a domain. When domains share an underlying transition structure, these representations can be used to support analogies. To implement our 1D case, we follow Stachenfeld et al. (2017) and compute the eigendecomposition of the square (nXn) SR matrix. The eigenvalues (λ) are obtained solving $det(M-\lambda I)=0$, where M is the SR matrix, and I the identity matrix. We can obtain the corresponding eigenvector (U_i) for a particular eigenvalue (λ_i) by solving $MU_i = U_i\lambda_i$ for U_i. Finally, we encode each location in the native space using its eigencoordinates ($U\lambda$) (Figure 1C, lower left panel). We compare models trained on the top 10, top 5, top 2 highest eigenvalue eigenvectors (e.g., Figure 1C, lower right panel), and the neural network learns over these eigencoordinate representations, which implicitly encode the ordering and neighborhood relations among the locations. Finally, we include a scrambled version of the top 2 eigenvectors. This scrambled version removes the true similarity structure in the native space, and serves as a control to ensure that the neural network is not simply memorizing its inputs.

Model Architectures. We trained and tested networks on each of the input representations described above. Each network was constructed to take a pair of inputs (A and B — we refer to objects and relations, themselves, using non-italicized capitalized letters, and a model's representations of them using italicized lower-case (e.g., a and b)), and trained to predict each member of the pair from the other. We evaluated five competing model architectures, that we describe in detail below. In all models, representations of A converge on a common learned internal layer, which can then encode their relationship R. The ability to form a systematic, abstract relational encoding (r) is a primary focus of this paper. We evaluate the ability of r to support simple metric analogies by completing problems of the form A:B as C:? (e.g., 1:3 is to 5:?, answer: 7).

r is computed as,

$$r = f_\Theta(a,b)$$

where a and b are representations of the two input objects to be related, and Θ are the learned encoder parameters (by default, a one layer multiple layer perceptron (MLP) with 100 rectified linear units (RELU)) and (r) is the activation state across two linear nodes.

To train the model, the latent state r is composed with the object A representation (a) to predict B ($\hat{b}$), and composed with the object B representation (b) to predict A ($\hat{a}$).

$$\hat{a} = g_{\Phi}(r,b) \quad \hat{b} = g_{\Phi}(r,a)$$

where, $\hat{a}$ is the predicted version of a, g is the decoder function (a one layer MLP with 100 RELU units), with learned parameters Φ. r is $r = f_{\Theta}(a,b)$, as above, and b is the input representation of object B. Likewise for $\hat{b}$, *mutatis mutandis*.

The representations r , a , and b are thus arguments to functions parameterized by the encoder and decoder. Here, we control the selection and application of these arguments by hand, but not their values, therein focusing on the problem of representation learning.

Intuitively, the objective of the model is to learn a representation that enables transformation of the representation a to match representation b, and vice versa. The loss is the mean squared error of the reconstruction across both objects, and weights of the encoder (Θ) and decoder (Φ) are jointly updated using standard backpropagation (Rumelhart, Hinton, & Williams, 1985). We emphasize that the network is trained to minimize reconstruction error ($(a-\hat{a})^2 + (b-\hat{b})^2$), not to complete analogies.

Model architectures are shown in Figure 2. All 5 models consist of two linear nodes in the latent space (r), but differ in how they constrain the nodes in r to be used. For models 1 and 2, the same 2D vector of r is used in both $\hat{a} = g_{\Phi}(r,b)$ and $\hat{b} = g_{\Phi}(r,a)$. Model 2 differs from Model 1 only in allowing separate decoder parameters to be learned for $\hat{a} = g_{\Phi_a}(r,b)$ and $\hat{b} = g_{\Phi_b}(r,a)$. These models are free to learn how to use these nodes (r) to predict across a and b. By contrast, models 3, 4, and 5 specifically commit each of these two nodes to separately predicting a and b, which we refer to as r_a and r_b, respectively. That is, $\hat{a} = g_{\Phi}(r_a,b)$, and $\hat{b} = g_{\Phi}(r_b,a)$. Models 3, 4, and 5 all share decoder parameters across $\hat{a}$. and $\hat{b}$.

Models 4 and 5 involve chaining the two latent nodes in r (r_a and r_b), such that one of the components of the relational representation is a function of the other. The motivation for this chaining of latent nodes is as follows. Recall that activation in r_a encodes information necessary to transform b to a when passed through the decoder, and vice versa for r_b. Although the input representation may contain *implicit information* about the relationship between A and B, the encoder must extract this information and explicitly represent relational information in a low-dimensional form. Logically, in a metric space, how A relates to B cannot change without affecting how B relates to A – the two predictive tasks imposed on network. The chain weights in models 4 and 5 force such a bi-directional dependence in r (See Figure 2). While it is possible that an unconstrained neural network could learn the dependence between r_a and r_b, we find that, using the current objective, this does not occur reliably without imposing the constraint that $r_a = f_{\Theta}(a,b)$, and $r_b = r_a W + bias$.

Model 4 imposes the weakest version of this constraint, by allowing the weight W (that links r_a and r_b) to be randomly initialized. However, randomly initializing the chain weight fails to exploit all the world-structure that may be easily incorporated. Critically, the general relation between the components r_a and r_b should be anti-correlated (e.g., the "bigger" A is than B, the "smaller" B is than A). That is, r_a, r_b can be thought of as conjugate pairs (e.g., +-2). A prior that biases the network toward the discovery of this relational structure can easily be implemented by initializing W to -1. We refer to this architecture as the *conjugate symmetry prior*, as it favors extracting a representation in r that treats r_a and r_b as a conjugate pair, reflected about zero. Note that this was strictly an initialization, and that W was free to vary over training. Empirically, we find that it tends not to vary over the course of training when initialized to -1.

Analogy Evaluation. We address whether the trained networks can perform analogy as follows. First, two object representations (a and b) are passed to the network, and the resulting latent state (r) is computed, and manually clamped to use for analogical completion. Next, a third object representation (c) is passed directly to the decoder without modifying r. The decoder then generates the expected d ($\hat{d}$), given c and the latent state r that was previously computed from the ab pair, as $\hat{d} = g_{\Phi}(r,c)$. To quantify a network's analogical ability, we compute the cosine similarity between $\hat{d}$ and d, as well as the cosine similarity between d and three randomly chosen foils. If the correct mapping is more similar than all three incorrect mappings, the network is determined to have succeeded on that trial. Chance performance on this metric is thus 25%.

Our primary results involve a 1x80 space (Figure 3). We also explore 1x20, 1x40, and 1x160 spaces, and the ability to generalize between them (Figure 5). In all cases, we trained the model on 500 ab pairs from the same space, and held 100 unique pairs from that space out of training for model testing. The number of possible pairs varies by the size of the native space. For example, in the smallest space (1x20), there are 380 possible ab pairs, entailing that some were necessarily repeated in training. At the other extreme (1x160), there are 25,540 possible pairs, and only 500 were selected for training, meaning that the network only experienced 2% of the possible space of relation-tokens. We trained the network using batch sizes of 64 samples and the ADAM optimizer (with a constant learning rate of $3x10^{-4}$), implemented in Tensorflow. We trained all networks for a fixed number of 1000 batches. This was sufficient to observe reliable asymptotes in training loss. Unless otherwise indicated, we ran 50 replications of each network with random encoder and decoder weights and random samples from the training set, and plot the mean and 95% CI for each class of networks and input representations.

Results

Analogy Results Across Input Representations and Architectures. In the models explored here, we find that success on these analogies requires both a particular input representa-

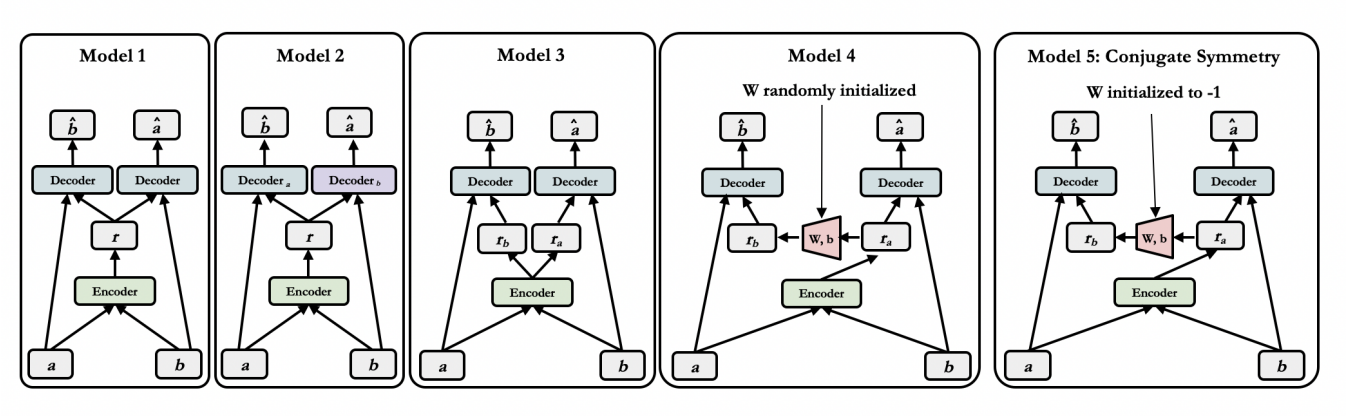

Figure 2: Schematic of model architectures. In each model, the encoder infers the relation r between a and b. The decoder predicts a given b and r, and predicts b given a and r. Gray boxes indicate activity vectors (inputs, outputs, or hidden states). Colored boxes indicate parameterized functions (linear layers or multilayer neural networks); boxes with the same name and color indicate shared parameters.

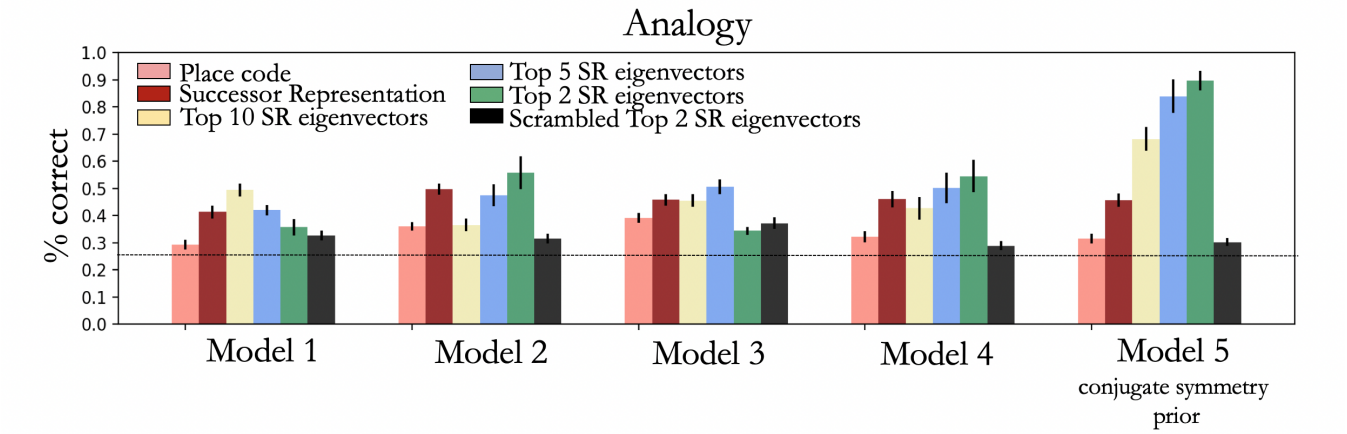

Figure 3: Analogy results for different model architectures. We see that only Model 5 (conjugate symmetry prior) trained on a low dimensional eigenrepresentation (e.g., 2 and 5) reliably learns relations enabling analogical completion.

tion and a particular architecture (See Figure 3.) Only models that integrate a low-dimensional eigenrepresentation of the SR (such as the top 2 or top 5 eigenvectors), and learn over this representation using a conjugate symmetry prior on the relational representation (Model 5) reliably complete analogies involving novel tokens (e.g., 1:3 :: 5: ? (7)). In the 1x80 space, this combination of architecture and top-2 eigenvectors averaged analogical performance of 90.5% across 50 iterations, well above a priori chance levels of 25%. Place codes, the full SR matrix, and scrambled eigenvectors never extracted relational representations that support analogical completion, regardless of architecture. Likewise, Models 1-4 did not reliably extract the appropriate relational structure, regardless of the input representation employed (i.e., even when the input representation was the top 2 or 5 eigenvectors).

Why is this particular combination of input representations and architectural biases important? The highest-eigenvalue eigenvectors carry implicit structural information about the dimension. In this 1D space, the 2nd eigenvector (Fiedler vector) here monotonically increases over the range of the domain, and is thus the closest approximation to a linear representation of the native space in the set of eigenvectors. (See Figure 1). In learning over these representations, Model 5 biases the network to represent the explicit bidirectional relation between a and b using anti-correlated conjugate pairs, reflecting the structure of the components of the relational representation (See Figure 4). This bias appears necessary for backpropagation to reliably learn suitable encoder and decoder parameters to support analogy, when using the transformational objective we employ, here (i.e., $\hat{a} = g_\Phi(R_a, b)$, and $\hat{b} = g_\Phi(R_b, a)$).

Figure 4 shows the latent representations in the two nodes of r for single, randomly chosen runs of all 5 models when trained on the top-2 eigenvectors. For Models 3-5, these

two nodes are functionally restricted, corresponding to r_a and r_b. Notably, we see that in Model 5, the latent activations (r_a, r_b) approximate anti-correlated linear representation of the signed distance. Moreover, for a given signed distance, r_a and r_b are in conjugate symmetric states $(0 + r_a, 0 - r_b)$, reflected about the X axis. Note also that many different A,B pairs will produce the same signed distance. For example, 10-5, 23-18, and 76-71, would all be at the same position along the X axis in Figure 4, given that they are all equal in signed distance. Only Model 5, with the conjugate symmetry prior, extracts similar representations in r, encoding the abstract relational structure as stationary over the range of inputs.

It is worth noting how this relates to the parallelogram model (Rumelhart & Abrahamson, 1973), which captures the abstract logic of linear analogies in vector space. The parallelogram model completes the analogy using fixed vector addition and subtraction operations $D = (B\text{-}A) + C$. We note that we do not see our model as standing in competition with an algorithmic parallelogram computation, a la (Rumelhart & Abrahamson, 1973). Instead, the focus of our model as deriving useful representations from experience, using only observation and standard gradient-based learning, coupled with particular inductive biases. Our model can be thought of as learning an encoding from the input (canonical) space to a latent space in which a linear parallelogram-like computation (B-A+C, (Rumelhart & Abrahamson, 1973; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013)) of the analogy can be performed. Here, we use the conjugate symmetry prior on the chained weight connection to encourage r_a to encode the trajectory from b to a, and r_b to encode the reverse trajectory from a to b.

Generalization across 1D native spaces of various size. Here, we show that the extraction of relational information not only allows the network to complete analogies within the domain over which it has been trained, but also to generalize out of the domain of its support (i.e., pure extrapolation). To do so, we train the network on one range of magnitudes (1x20,1x40, 1x80, or 1x160), and test its analogical performance on magnitudes outside of that range (see Figure 5). For example, one network is trained on inputs that differ in magnitude between 1 and 20 units (1x20 space), and then tested on inputs that differ by different ranges of magnitudes (e.g., 1x40, 1x80, 1x160). We repeat this procedure by training on all four ranges, and testing on the others. Note that these different ranges have the same underlying, local transition structure (See Figure 1), but different numbers of states. In all cases, the network can generate analogies in ranges of magnitudes that are beyond the scope of training without further weight updates, including extrapolating from learning in the 1x20 range and testing in 1x160. Thus, given the assumption of suitable procedures for re-computing and normalizing the eigendecomposition in novel domains, this algorithm can naturally generalize to similarly structured environments without retraining.

Extracting and utilizing tree structured representations. Thus far, we have focused on analogy in 1D linear spaces. However, it is clear that human reasoning also exploits other forms of relational structure present in the environment (e.g., (Kemp & Tenenbaum, 2008)), such as trees, rings, and radial geometries. To explore whether the model presented above can be extended to learn, and make use of other, non-linear structure in generalization and analogical inference, we apply it to a simple hierarchical graph.

Specifically we use a tree composed of two identically structured "families" with connected root nodes (See Figure 6). Both families have 4 generations (levels), with equivalent number of individuals per generation. Edges in the graph exist only between parents and children (results are similar when edges between siblings are included). To generate the transition matrix, we assume that the probability of moving from one node to another is 1/node degree, and compute the eigendecomposition over this random-walk transition matrix. Based on the results of the 1D case, we use the top 2 eigenvectors, and compare the performance of Models 1 (standard) and Model 5 (conjugate symmetry prior), including a version of Model 5 with scrambled eigenvectors.

For this tree, the top 2 eigenvectors are highly structured and interpretable (See Figure 6): the highest eigenvalue eigenvector carries information about family identity, and the second highest-eigenvalue eigenvector about generation, invariant to family. We imposed two further constraints on the analogy test used for the 1D space. First, (a, b) training pairs were constrained to come from the same family. Second, the siblings of the target node were prevented from being foils (though, of course, other close relatives such as cousins, parents, or children could be included as foils). Notably, we again see that Model 5, with a conjugate symmetry prior, learning over the top-2 eigenrepresentations is able to successfully complete the multiple choice analogy tasks, despite no experience with the particular (a, b) pairs, and no direct training on analogy. See Figure 6. Although this particular tree structure is a simplification of more complex structures observed in the world, our results suggest that the model can be applied usefully to more complex structures than the simple linear 1D metric focused on above.

Discussion

Here, we considered the possibility that abstract relation learning can derive from the combination of (a) error-driven learning using the vast amounts of perceptually observational data available ((Rao & Ballard, 1999; O'Reilly, Wyatte, & Rohrlich, 2017), with (b) particular inductive biases in the learning algorithm and network architecture. We compared a variety of input representations and model architectures, testing their ability to extract relational information and use that to complete novel analogies in simple 1D and tree structure domains, without explicit training on analogy. We found that two properties of the models are critical for exhibiting this ability. First, the inputs to the model must be mapped into a canonical representational space that carries implicit struc-

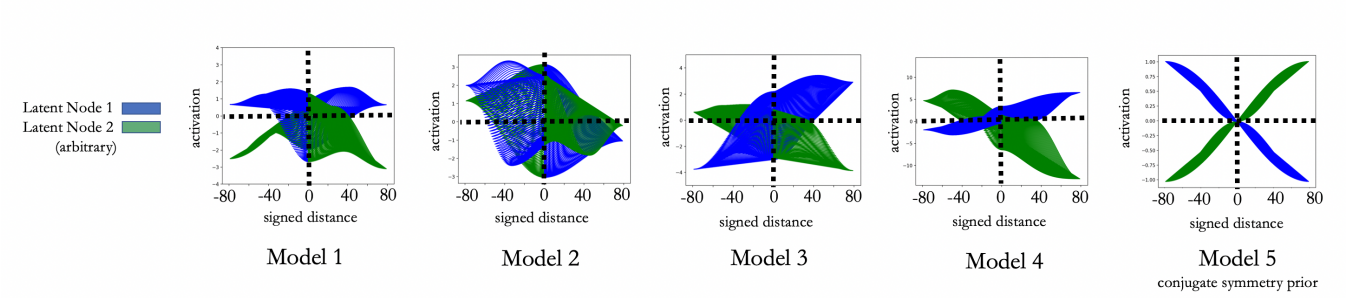

Figure 4: Latent node activations for two nodes as a function of signed distance when trained on top-2 eigenvectors. Only Model 5 (conjugate symmetry prior) learns representations that reliably track the ground truth signed distance. Note that different input pairs produce the same signed distance (e.g., 55-51, 11-7). Every possible pair was plotted here, resulting in the visible, thin individual lines. Notably, only Model 5 reliably produces similar activations in r for different tokens of the same signed distance. Note also that the two latent nodes are anti-correlated.

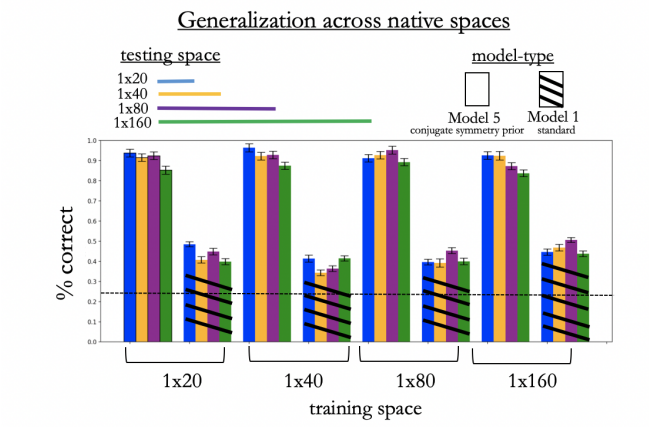

Figure 5: Analogy results when training and testing on native spaces of different size, but the same transition structure.

tural information about the domain (i.e., ordering and neighborhood relations). Learning over these low-dimensional, structured representations (Figure 1) enables the extraction of explicit relational representations, that can be used for analogy. Mapping to a canonical space also allows generalization across environments of different sizes that share structure (Figure 5).

The particular canonical representation we employ is inspired by the relationship between hippocampal place cells and grid cells in medial entorhinal cortex. These grid-cells have been suggested to reflect a decomposition of the predictive map reflecting the transition structure common to spatial environments (Stachenfeld, Botvinick, Gershman, 2017). Evidence for grid-cells has been found in the human brain beyond EC, in medial PFC, posterior cingulate, and lateral inferior parietal cortex (Doeller et al, 2010; Jacobs et al., 2013; Constantinescu et al., 2016), and in conceptual (non-spatial) tasks (Constantinescu et al., 2016). We thus take grid-cells to be one example of the abstract, structured representations that may be shared across domains and exploited for high-level relational reasoning tasks, such as analogy. More broadly, we suggest that similar mechanisms for extracting low-dimensional structure may support other common representational forms. Kemp and Tenenbaum (2008) provide a small inventory of representational forms that recur in human cognition (rings, chains, grids, hierarchies, trees, orders), presenting a hierarchical Bayesian model that identifies the best structural form for a dataset. Understanding how biologically plausible predictive learning mechanisms (coupled with inductive biases) may extract other representational structures is a topic of ongoing work.

Second, we show that if a simple neural network architecture is trained on pairwise relationships among these dimensionally-reduced encodings, and is imbued with a simple, local inductive bias that favors the extraction of conjugate bidirectional relationships among those pairs, it can learn representations that allow it to carry out analogical completions, and generalize this ability well out of the range of its training domain. Intuitively, we can think of the network as asking: having seen a and b, how would I transform a to make it b, and vice versa? We combine this objective with a prior on the relational representation that favors dedicated, but systematically anti-correlated, nodes in the latent space (r_a ,r_b). These nodes may be thought of as a conjugate pair of component relational representations that can be used to reconstruct $a(r_a, b)$ and $b(r_b, a)$. Notably, the learned relations (or trajectories) in the latent space are abstract and approximately stationary with respect to the input domain (i.e., 1-3 = 15-17) (See Figure 4), and can be composed with other object representations to generate analogies.

Though these results are encouraging, they rely on a simplified version of the problem that humans face in generating analogies. One of the most notable features of human analogical ability is the ability to *select* the relevant dimension of variation from the indefinite number of possible relations that might obtain. Indeed, some of the limits in applying the parallelogram procedure (B-A +C) on semantic embeddings as a model of human ability stem from problems handling se-

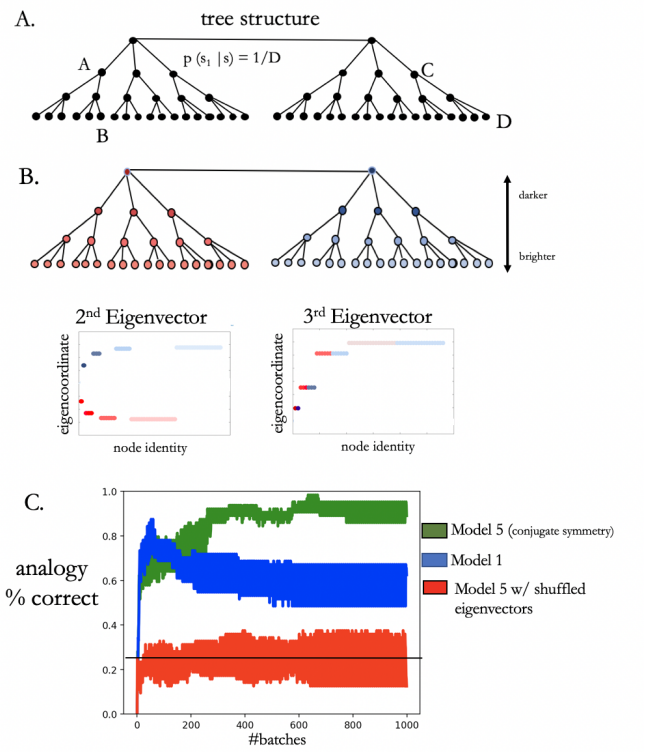

Figure 6: Extraction and utilization of structured representations for the particular tree shown in (A). (B) visualizes the 2nd and 3rd largest eigenvalue eigenvectors. Here, locations in the tree (shown in the upper portion of B) are linked to the points in the eigenplots (lower portion) that share color and brightness. The 2nd EV can be seen to encode family identity (one red, one blue), and the third encodes generation invariant to family (varying in brightness). (C) shows analogy results for this domain.

quencing and context-sensitivity (Chen, Peterson, & Griffiths, 2017). Here, we have pre-selected the relevant dimension for the analogy, focusing instead on general mechanisms for acquiring and utilizing these structured representations. However, a proper understanding of analogy, and human intelligence more broadly, requires directly addressing the relevant search and attentional selection problems, which we see as a critical target for future work.

References

Chen, D., Peterson, J. C., & Griffiths, T. L. (2017). Evaluating vector-space models of analogy. *arXiv preprint arXiv:1705.04416*.

Constantinescu, A. O., O'Reilly, J. X., & Behrens, T. E. (2016). Organizing conceptual knowledge in humans with a gridlike code. *Science*, *352*(6292), 1464–1468.

Dayan, P. (1993). Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, *5*(4), 613–624.

Doeller, C. F., Barry, C., & Burgess, N. (2010). Evidence for grid cells in a human memory network. *Nature*, *463*(7281), 657.

Doumas, L. A., Hummel, J. E., & Sandhofer, C. M. (2008). A theory of the discovery and predication of relational concepts. *Psychological review*, *115*(1), 1.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive science*, *7*(2), 155–170.

Gentner, D., & Forbus, K. D. (2011). Computational models of analogy. *Wiley interdisciplinary reviews: cognitive science*, *2*(3), 266–276.

Gick, M. L., & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive psychology*, *15*(1), 1–38.

Gustafson, N. J., & Daw, N. D. (2011). Grid cells, place cells, and geodesic generalization for spatial reinforcement learning. *PLoS computational biology*, *7*(10), e1002235.

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., & Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, *436*(7052), 801.

Holyoak, K. J. (2012). Analogy and relational reasoning. *The Oxford handbook of thinking and reasoning*, 234–259.

Jacobs, J., Weidemann, C. T., Miller, J. F., Solway, A., Burke, J. F., Wei, X.-X., ... others (2013). Direct recordings of grid-like neuronal activity in human spatial navigation. *Nature neuroscience*, *16*(9), 1188.

Kemp, C., & Tenenbaum, J. B. (2008). The discovery of structural form. *Proceedings of the National Academy of Sciences*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

O'Reilly, R. C., Wyatte, D. R., & Rohrlich, J. (2017). Deep predictive learning: A comprehensive model of three visual streams. *arXiv preprint arXiv:1709.04654*.

Rao, R. P., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, *2*(1), 79.

Rumelhart, D. E., & Abrahamson, A. A. (1973). A model for analogical reasoning. *Cognitive Psychology*, *5*(1), 1–28.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation* (Tech. Rep.). California Univ San Diego La Jolla Inst for Cognitive Science.

Stachenfeld, K. L., Botvinick, M. M., & Gershman, S. J. (2017). The hippocampus as a predictive map. *Nature neuroscience*, *20*(11), 1643.

Acknowledgements

This project / publication was made possible through the support of a grant from the John Templeton Foundation. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the John Templeton Foundation.