COGNITIVE SCIENCE
A Multidisciplinary Journal

Cognitive Science 44 (2020) e12904

ISSN: 1551-6709 online
DOI: 10.1111/cogs.12904

Holographic Declarative Memory: Distributional Semantics as the Architecture of Memory

M. A. Kelly,[a,b] Nipun Arora,[c] Robert L. West,[c] David Reitter[b,d]

[a]*Department of Computer Science, Bucknell University*
[b]*College of Information Sciences and Computing, The Pennsylvania State University*
[c]*Department of Cognitive Science, Carleton University*
[d]*Google Research*

Received 27 September 2019; received in revised form 30 March 2020; accepted 31 August 2020

Abstract

We demonstrate that the key components of cognitive architectures (declarative and procedural memory) and their key capabilities (learning, memory retrieval, probability judgment, and utility estimation) can be implemented as algebraic operations on vectors and tensors in a high-dimensional space using a distributional semantics model. High-dimensional vector spaces underlie the success of modern machine learning techniques based on deep learning. However, while neural networks have an impressive ability to process data to find patterns, they do not typically model high-level cognition, and it is often unclear how they work. Symbolic cognitive architectures can capture the complexities of high-level cognition and provide human-readable, explainable models, but scale poorly to naturalistic, non-symbolic, or big data. Vector-symbolic architectures, where symbols are represented as vectors, bridge the gap between the two approaches. We posit that cognitive architectures, if implemented in a vector-space model, represent a useful, explanatory model of the internal representations of otherwise opaque neural architectures. Our proposed model, Holographic Declarative Memory (HDM), is a vector-space model based on distributional semantics. HDM accounts for primacy and recency effects in free recall, the fan effect in recognition, probability judgments, and human performance on an iterated decision task. HDM provides a flexible, scalable alternative to symbolic cognitive architectures at a level of description that bridges symbolic, quantum, and neural models of cognition.

Keywords: Cognitive architectures; Vector symbolic architectures; Common model of cognition; Distributional semantics; Embeddings; Holographic memory; Declarative memory; Fan effect

Correspondence should be sent to M. A. Kelly, Department of Computer Science, Bucknell University, 336 Dana Engineering Building, One Dent Drive, Lewisburg, PA 17837. E-mail: m.alex.kelly@bucknell.edu

1. Introduction

Modern machine learning techniques based on neural networks and deep learning are implemented through algebraic manipulations of vectors, matrices, and tensors in high-dimensional spaces. Neural networks have an impressive ability to process data to find patterns, but they do not typically model high-level cognition and it is often unclear how they work. Symbolic cognitive architectures, such as the widely used ACT-R (Anderson, 2009; Ritter, Tehranchi, & Oury, 2019), can capture the complexities of high-level cognition and provide human-readable, explainable models of behavioral phenomena. However, symbolic models scale poorly to naturalistic, non-symbolic data (such as images) or big data (e.g., corpora with hundreds of millions of words).

Are symbolic and machine learning approaches compatible? Can they be unified? Symbolic and neural models can be understood as theories of cognition operating at different levels of description or analysis (see Kersten, West, & Brook, 2016, for a discussion of levels in computational cognitive models). Is it possible to provide a theory that bridges these two levels, a reduction of the symbolic to the neural, while retaining the strengths and capabilities of each?

Distributional semantics models, such as word embeddings, represent concepts as points in a high-dimensional space. Similarity between concepts is distance in that space. Distributional models can process millions of data points to infer semantic similarities from language data (e.g., Burgess & Lund, 1997; Griffiths, Steyvers, & Tenenbaum, 2007; Jones & Mewhort, 2007; Landauer & Dumais, 1997; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013; Pennington, Socher, & Manning, 2014), to infer product recommendations from patterns of user preferences (e.g., Rutledge-Taylor, Vellino, & West, 2008), to predict human probability judgments (Bhatia, 2017), or to predict racial and gender biases (Caliskan, Bryson, & Narayanan, 2017). Thus, it has been argued that distributional semantics models are a potentially appropriate basis for knowledge representation in general-purpose cognitive models (Bhatia, Richie, & Zou, 2019).

We posit that cognitive architectures, if implemented in a vector-space model, represent a useful, explanatory model of the internal representations of otherwise opaque neural architectures. We demonstrate that a distributional semantics model can be integrated into a cognitive architecture. Specifically, we substitute the ACT-R declarative memory (DM) for our model, holographic declarative memory (HDM). HDM is a variant of dynamically structured holographic memory (DSHM; Rutledge-Taylor, Kelly, West, & Pyke, 2014). DSHM, in turn, is a variant of the BEAGLE model of distributional semantics (Jones & Mewhort, 2007) generalized to non-linguistic tasks.

HDM is a model of learning for both declarative and procedural tasks. In ACT-R's DM, the association strengths between items in memory are typically set by the modeler by hand. Conversely, in HDM, the representations in memory are learned estimates of the association strengths and conditional probabilities of stimuli in the environment. DM scales poorly to large datasets (as discussed in Section 4.1), whereas HDM can scale from learning small, experimental datasets up to corpora representative of a lifetime's worth of experiences. In what follows, we illustrate how memory retrieval response time,

interference between memories, probability estimation, motivation, and surprise can be implemented by simple mechanisms in a vector space. Using these mechanisms, we demonstrate that HDM can account for primacy and recency effects in free recall, the fan effect in recognition, human probability judgments, and human performance on learning an iterated decision task.

2. Cognitive architectures and the common model of cognition

Since Newell (1973) first argued that good empirical work and piecemeal theoretical work are insufficient to achieve the goal of understanding the mind, researchers in cognitive science have sought to develop functional, testable theories of cognition as a whole. Cognitive architectures serve as both unified theories of cognition and as computational frameworks for implementing models of specific experimental tasks. Hundreds of cognitive architectures have been developed over the past 40 years and many have strong similarities to each other (Kotseruba & Tsotsos, 2018). The similarities suggest an emerging consensus on the basic principles of cognition. Laird, Lebiere, and Rosenbloom (2017) find commonalities between three cognitive architectures, namely ACT-R (Anderson & Lebiere, 1998), Soar (Laird, 2012), and SIGMA (Rosenbloom, Demski, & Ustun, 2016). On the basis of the commonalities, Laird et al. (2017) propose a Common Model of Cognition. The Common Model of Cognition is a high-level theory of the modules of the mind and how these modules interact (see Fig. 1).

The Common Model of Cognition consists of perceptual and motor modules that interact with the agent's environment, working memory buffers which hold the active data in the agent's mind, a declarative or long-term memory module that holds the agent's world knowledge, and a procedural memory module that controls the flow of information and evaluates possible actions (Laird et al., 2017). A large-scale evaluation of fMRI data, collected from over 1,000 participants across diverse tasks, found correlations in patterns of activity across brain areas consistent with the Common Model of Cognition's description of modules and their interactions (Steine-Hanson, Koh, & Stocco, 2018).

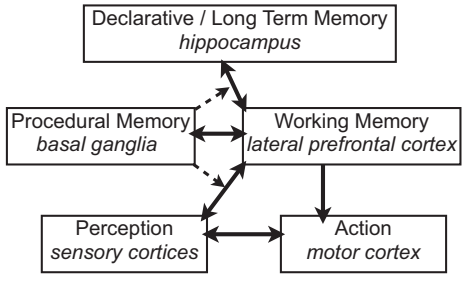


Fig. 1. The Common Model of Cognition (Laird et al., 2017) and associated brain areas (Steine-Hanson et al., 2018; Stocco, Laird, Lebiere, & Rosenbloom, 2018). Solid arrows indicate connections that pass data between modules. Dashed arrows indicate modulation of connections.

To evaluate our model, HDM, we compare to the ACT-R cognitive architecture in particular. However, HDM could be used as a model of long-term memory in any cognitive architecture described by the Common Model of Cognition.

3. ACT-R declarative memory

ACT-R's declarative memory (DM) consists of items of knowledge, called *chunks*, weighted by an estimate of the probability that chunk is useful in the agent's current context. Each chunk is either an unordered list of *slot:value* pairs (e.g., "name:tiger type:animal has:stripes") or an ordered list of values (e.g., *tiger animal striped*).

In ACT-R, the agent's current context serves as a cue to the memory system. Chunks in memory are activated according to a combination of the chunk's base-level activation and spreading activation. Base-level activation reflects how frequently and recently that chunk has been accessed. Spreading activation reflects the strength of the association between the cue and the chunk. More active memories are retrieved more easily and quickly.

For a chunk i, the activation of that chunk, A_i, is:

$$A_i = B_i + \sum_{j=1}^{n} W_j S_{ji}, \tag{1}$$

where B_i is the baseline activation of the chunk, n is the number of slot–value pairs in the cue, W_j is the attention paid to slot–value pair j of the cue, and each S_{ji} is an association strength: A measure of the probability that chunk i is relevant given that the cue contains slot–value pair j.

DM can be understood by analogy to a hydraulic system. Activation flows like water through connections between concepts like pipes. Activation spreads from the cue to the chunks in DM. Chunks that have stronger associations to the cue (wider pipes) receive more activation. The chunk that receives the most activation floats to the surface of consciousness and is selected and retrieved from memory. The time, T, to retrieve a chunk, i, is a function of the chunk's activation, A_i, and two fitting parameters, I and F,

$$T = I + Fe^{-A_i}. \tag{2}$$

The higher the activation, the shorter the retrieval time.

ACT-R's equations for activation (Eq. 1) and retrieval time (Eq. 2) form the backbone of ACT-R's declarative memory. They can successfully model a wide range of behavioral phenomena, including practice and forgetting effects in learning new words (Pavlik & Anderson, 2005), the cognitive availability of words (Cole & Reitter, 2018) and syntactic structures (Reitter, Keller, & Moore, 2011) when producing language, and the spread of

neologisms through online communities (Cole, Ghafurian, & Reitter, 2017). ACT-R's activation function has been related to the neurophysiology of long-term potentiation in the hippocampus (Pavlik & Anderson, 2005) and the firing patterns of neurons that control saccadic eye movements (Anderson, 2009, pp. 131–134).

4. Representing declarative memory in a vector space

HDM is a model of long-term memory that can be integrated within the ACT-R cognitive architecture.[1] HDM uses holographic reduced representations (Plate, 1995), a method of representing arbitrarily complex concepts using high-dimensional vectors. Holographic reduced representations belong to a family of methods know as vector-symbolic architectures (Gayler, 2003) or hyperdimensional computing (Kanerva, 2009) and are closely related to the (low-dimensional) conceptual spaces (Lieto, Chella, & Frixione, 2017). As such, HDM operates at a level of description that is the lingua franca of cognitive modeling (Lieto et al., 2017), bridging symbolic (e.g., ACT-R), quantum (e.g., Bruza, Wang, & Busemeyer, 2015), and neural (e.g., Eliasmith, 2013) models.

Unlike DM, HDM does not store chunks as discrete data structures. Instead, HDM stores a pair of vectors for each unique *value* associated with a feature (e.g., *black* or

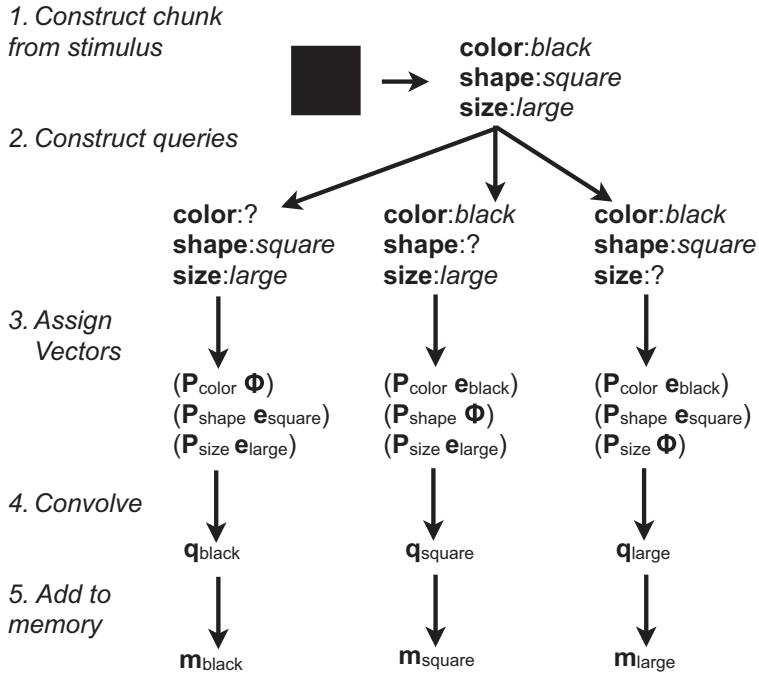


Fig. 2. Example of the process of encoding a stimulus in HDM.

square, see Fig. 2). As chunks are added to HDM, the information from each successive chunk is distributed across the relevant vectors and superimposed onto the information from prior chunks, such that the vectors in the high-dimensional space shift to better represent the relationships described by all chunks stored.

In what follows, we describe first HDM's ability to scale to big data, then the elementary operations of HDM, and finally how those processes can be combined to account for human memory and learning.

4.1. Scalability

HDM is more tractable than DM for scaling to large datasets in three ways: (a) ease of training, (b) space complexity, and (c) time complexity. HDM gains scalability by sacrificing DM's perfect storage of each and every chunk. Given that human memory also sacrifices precision for efficiency, we believe the tradeoff made by HDM is reasonable as a model of human memory. However, HDM may be more appropriate for modeling semantic or cortical memory, as opposed to episodic or hippocampal memory, as discussed in Section 9.2.

4.1.1. Ease of training

When using DM, association strengths between chunks are typically set by hand, which is not feasible for large datasets. Conversely, HDM automatically acquires the associations between all chunks stored in memory by representing the relationships between values as distance in the high-dimensional space.

4.1.2. Space complexity

HDM is based on distributional semantics, and hence, the original application is language. When representing the information in a corpus of hundreds of millions of sentences, HDM needs only to store a representation for each of the tens of thousands of unique words in the corpus's vocabulary. More generally, the size of HDM scales with the number of atomic components of experience (i.e., the number of unique *values*) rather than with the number of experiences stored. Conversely, when using DM with spreading activation, it is necessary to track each of the association strengths S_{ji} between chunks and values. As such, the size of DM scales with the *product* of the number of chunks and the number of unique values. It may even be possible to implement HDM as a model that is invariant in size, as discussed in Section 11.

4.1.3. Time complexity

The compute time to add a chunk to memory in HDM scales linearly with the number of slot–value pairs in the chunk, as discussed in Section 4.2. The compute time for retrieval from HDM (not to be confused with predicted response times) scales linearly with the number of unique *values* in memory. Conversely, the compute time for retrieval from DM scales as a function of the number of *chunks*. If the number of unique *values* (i.e., atomic elements of experience) is *less* than the number of *chunks* (i.e., experiences

stored), as is likely to be the case for big data applications, HDM will be the more efficient model in terms of compute time.

4.2. Add a chunk with slots

HDM represents each *slot* by a permutation $\mathbf{P}_{\text{slot}}$. Each *value* is represented by an environment vector $\mathbf{e}_{\text{value}}$ and a memory vector $\mathbf{m}_{\text{value}}$. The process of encoding is summarized in Fig. 2.

The permutation for a given slot, $\mathbf{P}_{\text{slot}}$, is initially generated randomly and then used consistently thereafter. A permutation can be represented as a permutation matrix, a $k \times k$ matrix of zeros with a single, randomly placed one in each row and column. Multiplying $\mathbf{e}_{\text{value}}$ by $\mathbf{P}_{\text{slot}}$ reorders the elements of $\mathbf{e}_{\text{value}}$ to create a unique representation of the slot–-value pair.

An environment vector $\mathbf{e}_{\text{value}}$ stands for the perceptual features of a stimulus. We do not simulate the details of the perceptual features (but see Cox, Kachergis, Recchia, & Jones, 2011; Kelly, Blostein, & Mewhort, 2013; Kievit-Kylar & Jones, 2011, for models that do). Instead, environment vectors are generated by randomly sampling from a Gaussian distribution with a mean of zero and a variance of $1/k$, where k is the dimensionality. In HDM, the dimensions are meaningless; only the relationships between vectors are meaningful. The number of dimensions, k, determines the fidelity with which HDM stores information, such that smaller k yields poorer encoding.

A memory vector $\mathbf{m}_{\text{value}}$ represents the associations a stimulus has with other stimuli in the environment. Memory vectors are constructed as HDM learns about the world. Memory vectors are holographic in that they use circular convolution (denoted by *) to compactly encode associations between values (Plate, 1995).

Each $\mathbf{m}_{\text{value}}$ is a sum of questions to which the given *value* is an answer. For example, as illustrated in Fig. 2, when a chunk representing a large black square, "color:black shape:square size:large," is added to HDM, $\mathbf{m}_{\text{black}}$, $\mathbf{m}_{\text{square}}$, and $\mathbf{m}_{\text{large}}$ are updated. To update $\mathbf{m}_{\text{black}}$, the four questions "What color is it?," "What color is the square?," "What color is the large thing?," and "What color is the large square?" are summed together:

$$\mathbf{q}_{\text{black}} = \mathbf{q}_{\text{color:?}} + \mathbf{q}_{\text{color:?shape:square}} + \mathbf{q}_{\text{color:?size:large}} + \mathbf{q}_{\text{color:?shape:square size:large}} \tag{3}$$

Then, we add the queries $\mathbf{q}_{\text{black}}$ to the memory $\mathbf{m}_{\text{black}}$:

$$\mathbf{m}_{\text{black},t} = \alpha \mathbf{m}_{\text{black},t-1} + \mathbf{q}_{\text{black}}, \tag{4}$$

where t is the current time step and α is the forgetting rate.

Each question is represented by a cue vector, $\mathbf{q}$, constructed by permuting each $\mathbf{e}_{\text{value}}$ by the corresponding $\mathbf{P}_{\text{slot}}$ and then convolving. We use "?" to denote the placeholder, as it functions much like a question mark. The placeholder vector is generated randomly like an environment vector. Using the placeholder vector, $\mathbf{\Phi}$, the question "What color is the large square?" is constructed as:

$$\mathbf{q}_{\text{color:?shape:square size:large}} = (\mathbf{P}_{\text{color}}\mathbf{\Phi}) * (\mathbf{P}_{\text{shape}}\mathbf{e}_{\text{square}}) * (\mathbf{P}_{\text{size}}\mathbf{e}_{\text{large}}). \quad (5)$$

The process of computing $\mathbf{q}_{\text{black}}$ (Step 4 in Fig. 2) may seem complex, but the cue is efficiently constructed by HDM in $O(nk\log(k))$ time, scaling linearly with the number of slot–value pairs n and scaling in $O(k\log(k))$ time with respect to the vector dimensionality k. Linear computation with respect to the number of slot–value pairs is achieved by taking advantage of the fact that convolution distributes over addition (i.e., $\mathbf{a}*(\mathbf{b}+\mathbf{c}+\mathbf{d}) = \mathbf{a}*\mathbf{b}+\mathbf{a}*\mathbf{c}+\mathbf{a}*\mathbf{d}$), such that all four of the questions that make up $\mathbf{q}_{\text{black}}$ are computed in a single pass.

The question-based encoding used by HDM allows the model to be structured around the atomic items of experience (*values* or concepts) rather than the experiences (chunks or episodes) themselves. The encoding technique used by HDM has proven effective as a method of modeling the semantic (Jones & Mewhort, 2007) and syntactic (Kelly, Ghafurian, West, & Reitter, 2020) knowledge stored in the mental lexion, but here we explore its utility as a general purpose scheme for declarative memory.

4.3. Add a chunk without slots

To encode chunks without slots, that is, as ordered sequences of values (e.g., "large black square"), HDM uses the permutation $\mathbf{P}_{\text{before}}$ recursively to create nested permutations. The cue vector for "What came after *large* and before *square*?" or "large ? square" is:

$$\mathbf{q}_{\text{large?square}} = \left(\mathbf{P}_{\text{before}}\left(\left(\mathbf{P}_{\text{before}}\mathbf{e}_{\text{large}}\right) * \mathbf{\Phi}\right)\right) * \mathbf{e}_{\text{square}}. \quad (6)$$

HDM uses skip-grams such that values in a chunk do not need to be consecutive to be associated with each other. For example, in "large black square," "? square" (i.e., "What came before square?") is added to both $\mathbf{m}_{\text{large}}$ and $\mathbf{m}_{\text{black}}$.

4.4. Recall: Exact matching and partial matching

To recall something, the request to HDM must provide a chunk with exactly one unknown (e.g., "color:? shape:square size:large"). In *exact matching*, the chunk is represented by a cue vector corresponding to a single question (e.g., "What color is the large square?"). In *partial matching*, the chunk is decomposed into all subquestions (e.g., "What color is it?," "What color is the square?," "What color is the large thing?," and "What color is the large square?") and represented as the sum of those questions. The cue vector can be constructed in the same amount of time irrespective of the number of questions, such that exact and partial matching is equally efficient.

The memory vector that has the highest similarity to the cue vector is selected as the response. Similarity between vectors is measured by the vector cosine, the cosine of the angle between vectors, which is equivalent to a normalized dot product:

$$\text{cosine}(\mathbf{q}, \mathbf{m}_{\text{value}}) = \frac{\sum_{i=1}^{k} q_i m_i}{\sqrt{\sum_{i=1}^{k} q_i^2}\sqrt{\sum_{i=1}^{k} m_i^2}}, \tag{7}$$

where $\mathbf{q}$ is the cue and $\mathbf{m}_{\text{value}}$ is a memory vector, each of k dimensions. HDM measures the cosine similarity between the cue and the memory vector for each possible *value*. Once the most similar memory vector is selected, the placeholder is replaced with the corresponding *value* and the modified chunk is returned to procedural memory.

4.5. Recognition: Request with no unknowns

To recognize something, the request to HDM must provide a chunk with no unknowns. HDM then computes the *coherence* of the chunk. *Coherence* is calculated as the mean cosine between the memory vector for each value in the chunk and the cue vectors for a chunk with that value substituted for an unknown. For example, the coherence of "color: black shape:square" is:

$$\frac{(\text{cosine}(\mathbf{q}_{\text{color:?}} + \mathbf{q}_{\text{color:?shape:square}}, \mathbf{m}_{\text{black}}) + \text{cosine}(\mathbf{q}_{\text{shape:?}} + \mathbf{q}_{\text{color:black shape:?}}, \mathbf{m}_{\text{square}}))}{2}. \tag{8}$$

5. Decay and the serial position curve

In DM, each chunk i in memory has a base-level activation B_i that decays as a power function of the time since the chunk was last added to memory,

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right), \tag{9}$$

where n is the number of times chunk i is presented and t_j is the time since the jth presentation. The rate of decay, d, is such that with $d = 0$, the activation of a chunk does not decay with time, whereas higher d yields faster decay.

Conversely, HDM has association strengths, but no base-level activation. The activation of a value is a function of the distance in the high-dimensional space between that value and a given cue. The structure of HDM commits us to representing the availability of information in memory as entirely contingent on associations.

The serial position effect (Ebbinghaus, 1885) is the finding that when people study a list, the items at the beginning and end of the list are remembered best. The serial position effect is an oft-studied key finding that has shaped the development of models and theories of human memory. The recall advantage for items at the beginning of the list (the *primacy effect*) is generally attributed to participants having longer to process and

rehearse those items. There are, however, several competing theories of the recall advantage for items at the end of list (the *recency effect*).

In DM, the recency effect is due to decay. In distributed processing models, such as neural networks or holographic memory models, a recency effect can be modeled as interference from more recently acquired information partially overwriting older information (retroactive interference). Older information also interferes with the encoding and retrieval of newer information (proactive interference). To account for the recency effect, it is necessary to postulate a mechanism such that retroactive interference is stronger than proactive interference.

The holographic memory model *TODAM* (Murdock, 1982) uses a forgetting coefficient α to update memory,

$$\mathbf{m}_i = \alpha \mathbf{m}_{i-1} + \mathbf{v}_i, \tag{10}$$

where $\mathbf{v}_i$ is the vector for a *memory trace*, which is equivalent to a *chunk* in ACT-R, and $\mathbf{m}_{i-1}$ and $\mathbf{m}_i$ are the memory vector before and after storage. The forgetting coefficient α ranges from 0 to 1. Multiplying the memory store by α privileges new information over old information, allowing retroactive interference to be stronger than proactive interference, which produces a recency effect. If $\alpha = 0$, the model has complete amnesia, and if $\alpha = 1$, the model does not privilege more recent information over older information.

In comparison to ACT-R, the forgetting coefficient α is inverse to the decay rate d, such that $\alpha = 0$ is equivalent to $d = \infty$ and $\alpha = 1$ is equivalent to $d = 0$. However, the decay of activation over time t in ACT-R is a power function, t^{-d}, whereas the decay of activation in TODAM is an exponential function, α^t. While on average, human learning tends to mimic a power function (Ritter & Schooler, 2001), this may be an artifact of aggregation. A survey by Heathcote, Brown, and Mewhort (2000) finds that individual learning curves tend to more closely resemble exponential functions than power functions. Additionally, while decay in ACT-R is over time in milliseconds, decay in TODAM is, properly speaking, the result of interference from the addition of new information to memory.

We use α to control forgetting in HDM. D. R. J. Franklin and Mewhort's (2015) holographic model accounts for both primacy and recency effects in terms of rehearsal and interference without needing a time-based memory decay function. D. R. J. Franklin and Mehwort's model is a single holographic memory vector that stores all list items. As such, all new items stored interfere with all previous items.

Conversely, HDM has one memory vector per item, such that interference occurs only between different associations for a given item. As a result, there is less interference in HDM than in a single vector model. Less interference allows HDM to better handle big data, such as modeling language learning, but at the cost of making HDM lessaccurate at modeling small-scale memory tasks, such as list learning.

To compensate, we add a time-based decay function to HDM. Whether memories decay with time (Ricker, Spiegel, & Cowan, 2014) or strictly due to interference (Oberauer & Lewandowsky, 2013) is controversial. The time-based decay function in HDM

may be merely a proxy for sources of interference that have not been explicitly modeled. Decay is implemented by adding noise over time to all memory vectors,

$$\mathbf{m}_t = \mathbf{m}_{t-1} + \eta\mathbf{n}, \tag{11}$$

where $\mathbf{m}$ is a memory vector, t is the time in seconds, $\mathbf{n}$ is a random vector, and η is the noise coefficient. If η is zero, no noise is added and there is no decay over time. For larger η, more noise is added per second and decay is steeper.

To compare DM and HDM, we treat the *coherence* of a chunk in HDM as analogous to base-level activation in DM. In Fig. 3, we compare activation of a chunk over 30 s in DM and HDM. The DM model has a decay rate of $d = 0.5$. The random noise component of DM's activation function is set to zero for ease of comparison with HDM, as turning DM's random noise on would introduce an additional uncorrelated source of variation. HDM has dimensionality $k = 64$, forgetting $\alpha = 0.7$, and noise $\eta = 3.0$. The chunk is repeatedly stored in memory at random intervals indicated in Fig. 3 by vertical lines.

Activation in DM and vector cosine in HDM are both estimates of the relevance of a given chunk in memory to the current situation. Specifically, in DM, activation is an estimate of the *log-odds* of a chunk's relevance:

$$A \approx \ln\left(\frac{p}{1-p}\right), \tag{12}$$

where A is the activation of the given chunk and p is the probability that the chunk is relevant. Similarly, vector cosine in HDM is an estimate of the square root of p (see

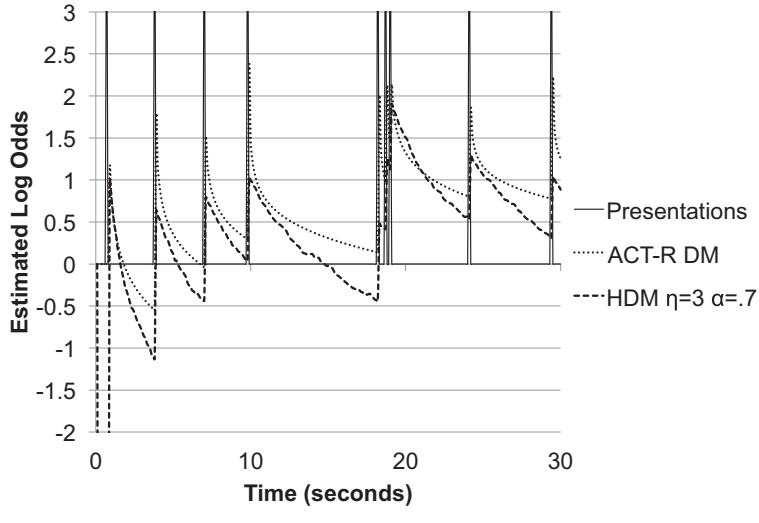


Fig. 3. Example activation of a chunk over time in DM (with noise turned off) and HDM (averaged over 10 runs).

Section 6 and Eqs. 17 and 18):

$$C \approx \sqrt{p}, \tag{13}$$

where C is the vector cosine between the given chunk and the cue. Accordingly, for Fig. 3, we convert vector cosine to activation as follows:

$$A = \ln\left(\frac{C^2}{1 - C^2}\right). \tag{14}$$

We fit the HDM model to the DM model using a systematic grid search of the parameters η and α in the ranges to η = 1–100 and α = 0.0–1.0. We average over 10 runs of the HDM model. At η = 3 and α = 0.7, HDM and DM produce comparable activation values for the chunk over time ($r = .59$). The decay over time in HDM is more linear than in DM because noise is added to the vectors as a linear function of time (Eq. 11). A better fit to DM could likely be achieved by adding noise non-linearly in a manner that mimics the DM decay function. The simplest approach to adding noise non-linearly would be to incorporate the α parameter from Eq. 10 into the memory update in Eq. 11, but we leave this modification to HDM for future work.

To demonstrate HDM's forgetting parameters on human data, we model the serial position effect. In Fig. 4, we compare two ACT-R models of the serial position effect to human data from Murdock (1962). The two models are identical except that one uses DM and the other uses HDM. Participants and models were presented 20 words at a rate of one word every 2 s. After, participants reported back the list in any order (free recall).

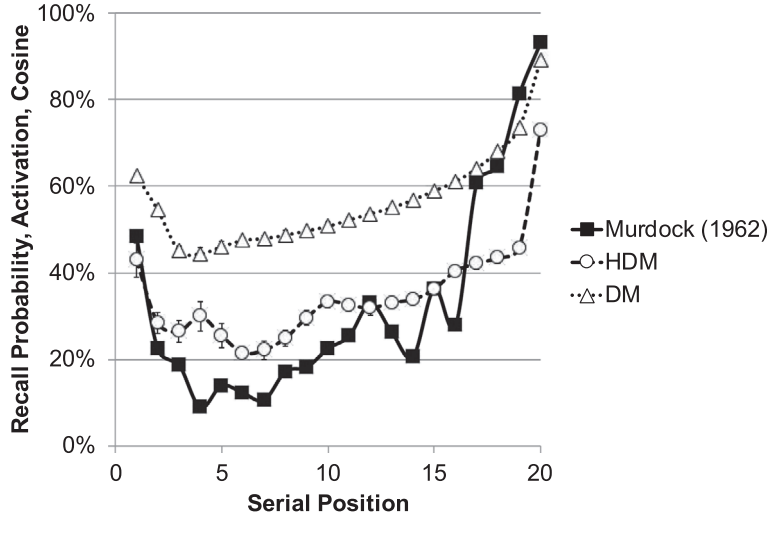


Fig. 4. Performance as a function of list position. Error bars indicate standard error for models. Results averaged across 10 runs of each model.

For simplicity, the models did not report back the list, and instead, we use the state of memory as a proxy for recall probability.

To study the list, the models store chunks that contain pairs of items: the current item and the previous item. There are 22 items in total: the 20 words of the list, the start list cue, and the end list cue. There are 21 chunks, one for each pair of items (*START 1*, *1 2*, ... *9 10*, *10 END*). In Fig. 4, the activation of a given item is calculated as the mean of the activations of the two chunks that the item appears in (e.g., for word *6*, the activation is the mean of the chunks *5 6* and *6 7*). We divide DM activation by 4 to convert to predicted recall probability. Error bars indicate standard deviation.

To capture the primacy effect, the models use the following rehearsal strategy: Rehearse the chunk for the current and previous item of the list, then return to the start of the list and rehearse forward as far as can be recalled.

DM and HDM use the same equation to determine time to recall a chunk. Retrieval time T is an exponential function of A_i, the activation of chunk i,

$$T = Fe^{-A_i}, \tag{15}$$

where F is the latency factor. For the free recall task, to prevent the models from rehearsing through the entire 20-item list every 2 s during the study phase, we use long retrieval latencies for both models.

We fit the DM model to human data by adjusting only the latency factor, while keeping the decay rate fixed to the standard value of $d = 0.5$ (Anderson, 2009, p. 110), achieving a best fit with latency $F = 8.0$. For HDM, we fix the dimensionality to $k = 64$, as dimensionality does not change average performance (more dimensions increase information storage capacity, which is necessary for scaling to bigger tasks). We use a grid search to fit the parameters α, η, and F, achieving a best fit with $\alpha = 0.9$, $\eta = 1$, and $F = 0.5$. While α and η serve a similar role in HDM as DM's decay rate d, determining if there is appropriate standard values for α and/or η, as there is for d, is a matter for future work.

Both models strongly correlate with the human data ($r = .95$ for DM, $r = .90$ for HDM). The results in Fig. 4 demonstrate that HDM is able to account for primacy and recency effects, providing approximately as good a fit as DM to the serial position curve.

6. Interference and the fan effect

In the fan effect task (Anderson, 1974), participants study word pairs, such as person–-location pairs (e.g., *hippy-park* or *lawyer-bank*). At test, participants are presented pairs that are either studied (targets, e.g., *hippy-park*) or novel (foils, e.g., *lawyer-park*) and must quickly discriminate.

The *fan* of a word is the number of pairs in the study set that contain that word. For example, if there are three pairs in the study set that contain *hippy* (*hippy-park*, *hippy-bank*, and *hippy-store*), then hippy has a fan of three. The *fan effect* is the finding that, at

test, participants are slower to make judgments about words with a higher fan. If *lawyer* has a fan of 1 (i.e., is only in the pair *lawyer-bank*), then participants are faster to make judgments about pairs that contain *lawyer* than they are about pairs that contain *hippy* (fan of 3).

The fan effect illustrates a fundamental principle of human memory: The availability of information in memory is an estimate of the probability that the information is useful in the current situation. If a participant has studied three pairs with the word *hippy*, each pair has only one in three chances of being useful for judging a test pair that contains *hippy*. Conversely, if the participant has studied only one pair with the word *lawyer*, that pair has a 100% chance of being useful for judging a test pair that contains *lawyer*.

DM models the fan effect by setting the association strengths between the words in the cue and the pairs in memory to a function of the fan of each word (Anderson & Reder, 1999). Simplifying the ACT-R equations, we find that the DM model produces a response time T in seconds that is a function of the fans f_{person} and f_{place},

$$T(f_{\text{person}}, f_{\text{place}}) = 0.233(f_{\text{person}} f_{\text{place}})^{1/3} + 0.845, \tag{16}$$

which correlates well with human data ($r = .95$, see Fig. 5).

We model the fan effect task using HDM. Without changing *any* parameters in Anderson and Reder's (1999) model of the fan effect, the HDM model provides a good fit to the data ($r = .91$, Fig. 5). Both the DM and HDM models use a latency of $F = 0.63$ and no decay (i.e., for HDM, $\alpha = 1$ and $\eta = 0$). HDM uses a dimensionality of $k = 256$. As demonstrated by Rutledge-Taylor et al. (2014, p. 18, Fig. 4), the fan effect is robust across variations in vector dimensionality, but reaction times are more stochastic when using vectors with fewer dimensions (we choose $k = 256$ for stability).

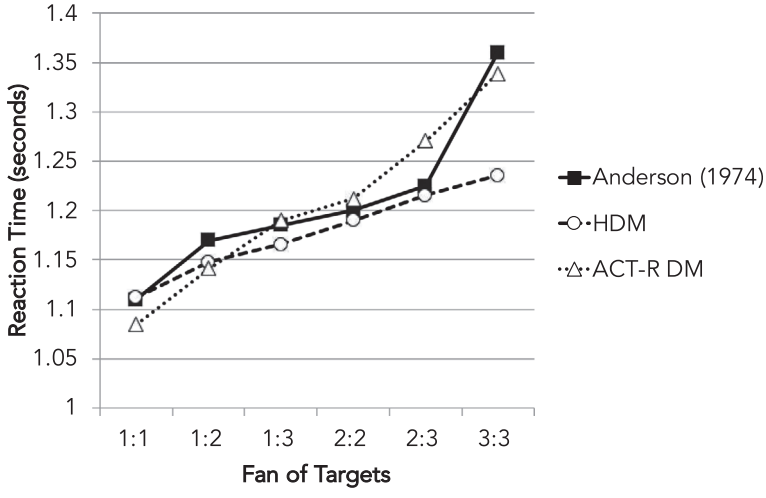


Fig. 5. Response time for targets in the fan effect task.

The fan effect arises from the geometry of the vector space, as is illustrated in Fig. 6.

The memory vector for *hippy*, $\mathbf{m}_{\text{hippy}}$, is constructed as a sum of cues. For a fan of 2, those cues are "Who is in the park?" and "Who is in the bank?," respectively, represented by the chunks "? park" and "? bank" and the corresponding vectors $\mathbf{q}_{\text{? park}}$ and $\mathbf{q}_{\text{? bank}}$. If these two questions are weighted equally, $\mathbf{m}_{\text{hippy}}$ will be equidistant from $\mathbf{q}_{\text{? park}}$ and $\mathbf{q}_{\text{? bank}}$. HDM uses randomly generated vectors that are orthogonal in expectation. If we assume the vectors are perfectly orthogonal, $\mathbf{m}_{\text{hippy}}$ will be at a 45° angle from $\mathbf{q}_{\text{? park}}$ and $\mathbf{q}_{\text{? bank}}$ with a cosine of 0.71. At a fan of 3, $\mathbf{m}_{\text{hippy}}$ is equidistant from $\mathbf{q}_{\text{? park}}$, $\mathbf{q}_{\text{? bank}}$, and $\mathbf{q}_{\text{? store}}$ with a 55 ° angle and a cosine of 0.58.

As the fan increases, the angle between memory and the cue increases. For a fan of f and perfectly orthogonal vectors, the cosine is $f^{-1/2}$, that is, the square root of the probability of the item conditional on the cue. In the fan effect task, all pairs of words in the study set are equiprobable. For events with unequal probabilities, we note that for n events with frequencies v_1 to v_n, the cosine of event i is:

$$\text{cosine} = \frac{v_i}{\sqrt{v_1^2 + \ldots + v_i^2 + \ldots + v_n^2}}. \tag{17}$$

Whereas the probability of event i is:

$$\text{probability} = \frac{v_i}{v_1 + \ldots + v_i + \ldots + v_n}. \tag{18}$$

As a result, the cosine underestimates the probability of low-frequency events. When making decisions from experience, people tend to underestimate the probability of rare events (Hertwig, Barron, Weber, & Erev, 2004). Thus, the cosine's biased estimate of probability may support HDM's validity as a cognitive model.

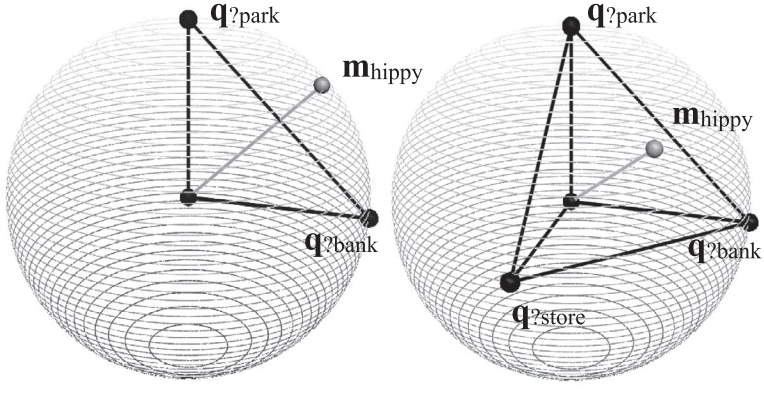


Fig. 6. $\mathbf{m}_{\text{hippy}}$ with a fan of 2 (left) or 3 (right). With a fan of 2, $\mathbf{m}_{\text{hippy}}$ is at a 45° angle from the cues, but with a fan of 3, $\mathbf{m}_{\text{hippy}}$ is at a 55° angle.

Like HDM, quantum probability models (Busemeyer, Pothos, Franco, & Trueblood, 2011) use the cosine between vectors to model human probability judgments. Thus, HDM can be understood as a realization of the quantum probability model of human judgment within a cognitive architecture, as we discuss in the next section.

7. Quantum models and probability judgment

The conjunction rule of classical probability theory dictates that the probability of the conjunction of two events A and B is always less than, or equal to, the probability P of the individual events:

$$\begin{aligned} P(A \wedge B) &\leq P(A) \\ P(A \wedge B) &\leq P(B). \end{aligned} \tag{19}$$

Famously, Tversky and Kahneman (1983) showed that human reasoning around probability often violates the conjunction rule. In an experiment, Tversky and Kahneman (1983) presented participants with a story about a hypothetical person, Linda, whose characteristics aligned her toward certain interests (like feminism) more than others (like financial systems), such that:

$$P(\text{Linda is a bank teller}) < P(\text{Linda is a feminist}). \tag{20}$$

Participants were then asked to evaluate the probability of various statements about Linda. The critical comparison concerned the probability estimates of statements "Linda is a bank teller" and "Linda is a bank teller and a feminist." Most participants judged the latter as more probable than the former, implying that for them:

$$P(\text{Linda is a bank teller}) < P(\text{Linda is a bank teller} \wedge \text{Linda is a feminist}). \tag{21}$$

Tversky and Kahneman's experiment illustrates the *conjunction fallacy*, the belief that a conjunction is more probable than the constituent events. While the conjunction fallacy is not compatible with the classical probability theory, Busemeyer et al. (2011) argue that the fallacy is explicable through quantum probability theory. The primary difference between the two theories is that while classical probability specifies outcomes in set-theoretic relationships, quantum probability theory is a geometric theory which specifies outcomes as subspaces of a vector space (for a general introduction to quantum probability theory, see Busemeyer et al., 2011).

Using quantum probability theory, a person's knowledge of Linda can be expressed as a vector, $\mathbf{m}_{\text{Linda}}$, and the probability of Linda being in a particular state can be understood as the size of the projection of $\mathbf{m}_{\text{Linda}}$ onto the subspace representing a particular state. The formulation of probability in terms of vectors allows for the conjunction fallacy to emerge.

For example (Fig. 7), if we take a projection of $\mathbf{m}_{\text{Linda}}$ onto the vector for *bank teller*, $\mathbf{m}_{\text{bankteller}}$, we would expect to get a very small shadow, the vector projection $\mathbf{p}_{\text{Linda}\rightarrow\text{bankteller}}$, indicating a low probability of Linda being a bank teller. The projection of $\mathbf{m}_{\text{Linda}}$ onto the vector representing *feminist*, $\mathbf{m}_{\text{feminist}}$, though is much bigger ($\mathbf{p}_{\text{Linda}\rightarrow\text{feminist}}$). To get a vector for "Linda is a feminist and a bank teller," we take a projection $\mathbf{p}_{\text{Linda}\rightarrow\text{feminist}}$ onto $\mathbf{m}_{\text{bankteller}}$ to get $\mathbf{p}_{\text{Linda}\rightarrow\text{feminist}\rightarrow\text{bankteller}}$. As we can see in the diagram, $\mathbf{p}_{\text{Linda}\rightarrow\text{feminist}\rightarrow\text{bankteller}}$ is greater than $\mathbf{p}_{\text{Linda}\rightarrow\text{bankteller}}$ indicating a higher probability of Linda being a bank teller *and* a feminist compared to Linda being a bank teller.

Because of the geometric nature of quantum probability, it is trivial to implement in vector-symbolic architectures, allowing models such as HDM to account for the conjunction fallacy. However, HDM differs from Busemeyer et al. (2011)'s quantum probability models in two important ways:

1. HDM can learn the strengths of associations between items from experience, whereas in quantum models, the vector similarities are set by hand;
2. Quantum probability models use the square root of the frequencies and the square of the cosine to compute the exact probability, whereas HDM computes an approximate probability (cf. Eqs. 17 and 18).

HDM is unable to use the square root (2) because it learns from experience (1). HDM learns the frequencies of events as an accumulating sum. To incrementally accumulate the square root of the frequency as a sum, rather than the frequency itself, would require dividing the weight of each event as it is added to memory by the square root of the frequency, and thus would require knowing the frequencies a priori.

To demonstrate the conjunction fallacy in HDM, we construct two different models. The first model is based on Busemeyer et al.'s (2011) model of the conjunction fallacy, and like in Busemeyer et al.'s (2011) model, the vectors are set by hand. The second model is similar to Bhatia's (2017) judgment model of the conjunction fallacy, and like in Bhatia's (2017) model, the vectors are learned from experience.

For the first model, we conduct 100 simulations, each with a different set of randomly generated environment vectors. The vectors $\mathbf{m}_{\text{Linda}}$, $\mathbf{m}_{\text{feminist}}$, and $\mathbf{m}_{\text{bankteller}}$ are each

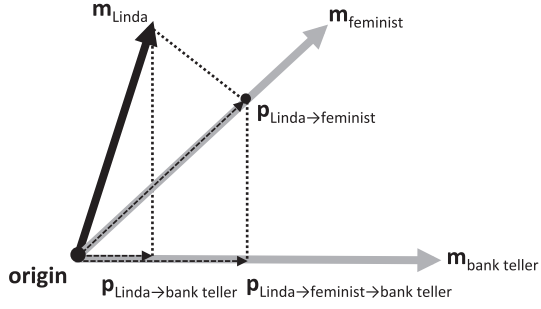


Fig. 7. Computing geometric probabilities using projections $\mathbf{p}$ of memory vectors $\mathbf{m}$ onto other memory vectors. Human probability judgments are the squared magnitudes of the projections, for example, $|\mathbf{p}_{\text{Linda}\rightarrow\text{feminist}\rightarrow\text{bankteller}}|^2 = P(\text{feminist}\wedge\text{bankteller}|\text{Linda})$.

constructed as a sum of environment vectors. So, if S_L, S_F, and S_B are the sets of environment vectors summed to construct $\mathbf{m}_{\text{Linda}}$, $\mathbf{m}_{\text{feminist}}$, and $\mathbf{m}_{\text{bankteller}}$, respectively, the sets need to satisfy the following conditions to conform to the geometric arrangement in Fig. 7:

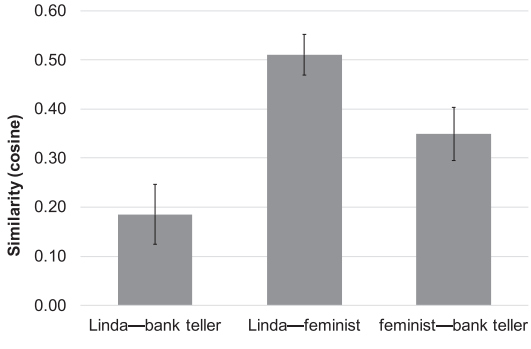
1. $|S_L \cap S_F| > |S_L \cap S_B|$
2. $|S_B \cap S_F| > |S_B \cap S_L|$

To realize these inequalities, the vectors are summed as illustrated in Table 1. For example, $\mathbf{m}_{\text{Linda}}$ is a sum of four unique environment vectors as well as 10 environment vectors that $\mathbf{m}_{\text{Linda}}$ shares with $\mathbf{m}_{\text{feminist}}$ and three environment vectors that it shares with $\mathbf{m}_{\text{bankteller}}$, such that $\mathbf{m}_{\text{Linda}}$ is more similar to $\mathbf{m}_{\text{feminist}}$ than $\mathbf{m}_{\text{bankteller}}$ and non-identical to either. We validated the relationships by measuring the cosines of the generated vectors (Fig. 8a).

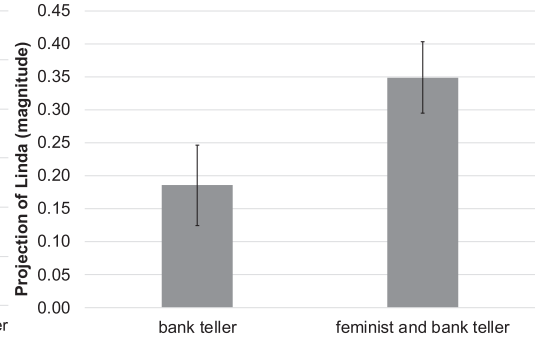
Finally, projections were computed to reflect the probability of *Linda* being a *bank teller* and *Linda* being both a *feminist* and a *bank teller* (Fig. 8b). We reproduce the conjunction fallacy reported by Tversky and Kahneman (1983). The magnitude of the projection of *Linda* onto *bank teller* is less than the magnitude of the projection onto *feminist and bank teller*.

Table 1
Unique and shared environment vectors summed into $\mathbf{m}_{\text{Linda}}$, $\mathbf{m}_{\text{feminist}}$, and $\mathbf{m}_{\text{bankteller}}$

	Linda	Feminist	Bank Teller
Linda	4	—	—
Feminist	10	4	—
Bank teller	3	6	4



(a) Similarity between vectors for *Linda*, *feminist*, and *bank teller*



(b) Magnitude of the projection of *Linda* onto *bank teller* and both *feminist and bank teller*

Fig. 8. HDM model of the conjunction fallacy using vectors artificially constructed according to Table 1. Error bars indicate standard error over 100 runs.

However, this model of the conjunction fallacy has two major problems: (1) It uses arbitrary rather than learned vector representations for the concepts, and (2) the model assumes that the conjunction fallacy emerges from question order effects.

When modeling question order effects, there is a clear order of operations: One question is asked first and how the participant answers that question affects how they answer the second question. For example, participants may be asked about honesty and trustworthiness of the American politicians *Bill Clinton* and *Al Gore*. If asked about Bill Clinton first, people rate Al Gore as less trustworthy than if asked about Al Gore first (Wang, Solloway, Shiffrin, & Busemeyer, 2014). This question order effect can be modeled by projecting from the participant's initial belief state onto the vector representing *Bill Clinton* and then onto *Al Gore*, or vice versa (Wang et al., 2014).

If the projection model is an appropriate model of the conjunction fallacy, we would expect that the conjunction fallacy to be contingent on the order of the questions, "Is Linda a feminist?," "Is Linda a bank teller?," and "Is Linda a feminist and a bank teller?." Boyer-Kassem, Duchêne, and Guerci (2016) experimentally manipulate the question order for the Linda story, and other scenarios that elicit a conjunction or disjunction fallacy, and while Boyer-Kassem et al. replicate the fallacy, they find no question order effects.

Thus, while the quantum projection model may be an appropriate model for question order effects, another model is needed to account for the conjunction fallacy. Aerts et al. (2017) propose an alternative quantum model of the conjunction fallacy that relies on the "emergence of new meanings when concepts are combined" rather than on question order effects.

Similarly, Bhatia (2017) proposes a model where judgments about the probability of *Linda* being both a *bank teller* and a *feminist* are computed as the similarity of *Linda* to the sum of *bank teller* and *feminist*. By training distributional semantics models such as *word2vec* (Mikolov et al., 2013) and *GloVe* (Pennington et al., 2014) on a large corpus and then constructing the *Linda* vector using the description of Linda given by Tversky and Kahneman (1983), Bhatia (2017) finds that the vectors for *Linda*, *feminist*, and *bank teller* have the necessary arrangement for the conjunction fallacy to emerge.

Our second model of the conjunction fallacy is similar to Bhatia's (2017) model. We train HDM, or more specifically, HDM's language-based precursor, BEAGLE (Jones & Mewhort, 2007), on either the novels corpus (145 million words; Johns, Jones, & Mewhort, 2016) or the British National Corpus (100 million words; The British National Corpus, 2007). We construct a vector for *Linda* as a sum of the memory vectors for the words in the description of *Linda* and a vector for *bank teller* as the sum of the memory vectors for *bank* and *teller*. For *feminist*, we use the memory vector for feminist. Representing a conjunction as a sum of vectors, as Bhatia's (2017) judgment does, we find that that for both corpora:

$$\text{cosine}(\mathbf{m}_{\text{Linda}}, \mathbf{m}_{\text{feminist}} + \mathbf{m}_{\text{bankteller}}) > \text{cosine}(\mathbf{m}_{\text{Linda}}, \mathbf{m}_{\text{bankteller}}). \tag{22}$$

Thus, a conjunction fallacy is predicted by the model (see Table 2 for cosines).

Table 2
Cosine similarity between vector representations generated by BEAGLE, on either the novels corpus or the British National Corpus (BNC), or by GloVe

Vector Cosines	BEAGLE (novels)	BEAGLE (BNC)	GloVe
$\text{cosine}(\mathbf{m}_{\text{Linda}}, \mathbf{m}_{\text{feminist}} + \mathbf{m}_{\text{bankteller}})$	0.6795	0.7248	0.3333
$\text{cosine}(\mathbf{m}_{\text{Linda}}, \mathbf{m}_{\text{bankteller}})$	0.6220	0.6418	0.2396
$\text{cosine}(\mathbf{m}_{\text{feminist}}, \mathbf{m}_{\text{bankteller}})$	0.4083	0.4586	−0.0094

Conversely, to predict a conjunction fallacy using projection, it is necessary for the projection of feminist onto bank teller be larger than the projection of Linda onto bank teller, such that $\text{cosine}(\mathbf{m}_{\text{feminist}}, \mathbf{m}_{\text{bankteller}}) > \text{cosine}(\mathbf{m}_{\text{Linda}}, \mathbf{m}_{\text{bankteller}})$. For BEAGLE using either corpora, as well as for GloVe trained on English Wikipedia and the Gigaword corpus (six billion words)[2], we find the reverse: Linda is more similar to bank teller than feminist is to bank teller, and thus, no conjunction fallacy is predicted by the projection model (see Table 2).

We thus see that HDM can model the specific ways in which human probability judgments depart from classical probability theory, both in question order effects, which can be modeled using vector projection, and conjunction fallacies, which can be modeled using vector addition. Furthermore, distributional semantics models are able to model human judgments in a range of paradigms, including predicting gender and racial bias in the implicit association task (Caliskan et al., 2017), and answering trivia questions (Bhatia, 2017).

8. Procedural learning and an iterated decision task

Procedural and declarative memory are often characterized as memory of *how* and *what*, respectively. Procedural memory consists of "if *condition* then *action*" production rules weighted by utilities that estimate how good it is to do *action*. Declarative memory consists of information weighted by how useful it is to remember that information given a cue.

At a high level of description, these two systems are the same: "If *cue* then *information*" is not much different from "if *condition* then *action*." Chunks and production rules are both weighted by probability estimates. Activation estimates the probability that a chunk is useful to know, and utility estimates the probability that a production rule is useful to do.

The usefulness of *knowing* and *doing* is distinct. For example, *knowing* that touching a sharp object will hurt you is useful. *Touching* a sharp object is not. Nevertheless, the functional similarity between procedural and declarative memory suggests that the same model of memory could be used to implement both systems.

While ACT-R traditionally uses the procedural memory system to model decision-making, ACT-R can, instead, rely on declarative memory. For example, ACT-R's declarative

memory has been used to model how humans learn to make sequential decisions when playing simple games, such as backgammon (Sanner, Anderson, Lebiere, & Lovett, 2000), rock–paper–scissors (Lebiere & West, 1999), and prisoner's dilemma (Ben-Asher, Dutt, & Gonzalez, 2013; Gonzalez & Ben-Asher, 2014; Lebiere, Wallach, & West, 2000). One approach to using ACT-R DM to make decisions is codified in instance-based learning theory.

Instance-based learning theory (Gonzalez, Lerch, & Lebiere, 2003; Lejarraga, Dutt, & Gonzalez, 2012) is a theory of choice based on ACT-R. In instance-based learning theory, *instances* are stored in declarative memory. *Instances* are specialized chunks consisting of a choice made, the context of the choice, and the outcome. To make decisions, an instance-based learning theory model evaluates each possible choice by recalling from declarative memory an aggregate of the outcomes of that choice in contexts similar to the current situation. Instance-based learning theory departs from standard ACT-R DM in two ways:

1. Instance-based learning theory uses graded, partial matching of remembered contexts to the current situation, and
2. Instance-based learning theory retrieves an aggregate across stored chunks rather than an exact stored chunk.

In this manner, instance-based learning theory is able to use declarative memory to estimate expected utility in a wide range of choice tasks. We note that both (1) graded similarity and (2) aggregation across instances arise inherently from the architecture of vector-space models, such as HDM.

To illustrate how HDM can implement instance-based learning, we model a task from Walsh and Anderson (2011). Walsh and Anderson have human participants perform an iterated binary decision task with initially unknown payoffs (see Fig. 9). The tasks consist of a first choice, made by pressing one of two letter keys (represented by *R* and *J* in Fig. 9). Then, an abstract cue is displayed (one of two geometric shapes, represented by ▮ and ◆ in Fig. 9). After the cue, a second choice is made by pressing one of two

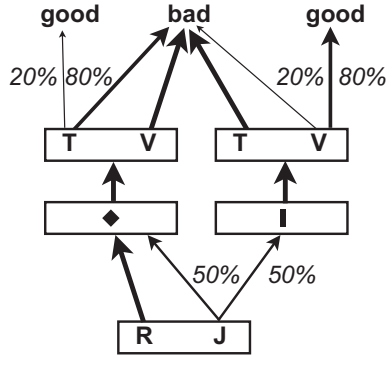


Fig. 9. Iterated decision task. Figure adapted from Walsh and Anderson (2011).

different letter keys (represented by *T* and *V* in Fig. 9). After the second choice, participants receive either positive or negative feedback (represented in the experiment by an asterisk * or hash # symbol, respectively). This completes a single trial of the task. The probability of positive feedback is contingent on the first and second choice as well as the cue. The task is difficult to learn as optimal choices yield only a 50% chance of positive feedback.

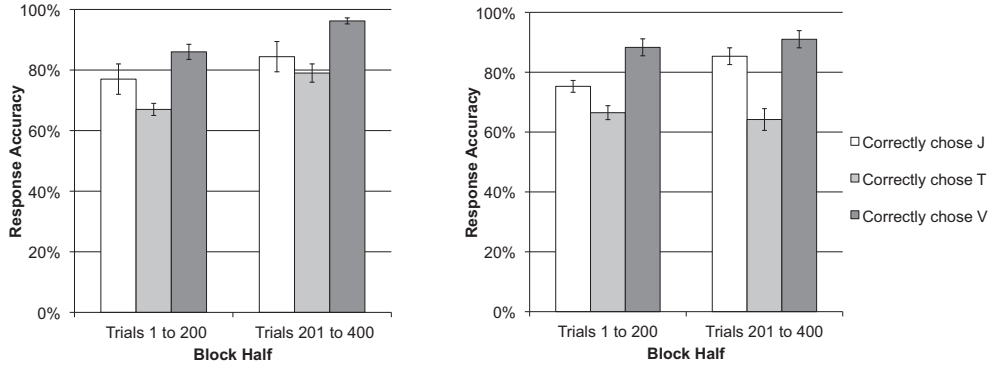
As shown in Fig. 10a, over 400 trials, participants gradually learn to perform the task well. Results are from Walsh and Anderson (2011). Data are averaged over 26 participants. Error bars indicate standard error.

Rutledge-Taylor et al. (2014) apply HDM to Walsh and Anderson's task. HDM is initialized to a state of *optimism*, the belief that all choices are potentially rewarding. Optimism is standard in instance-based learning theory models (Lejarraga et al., 2012, p. 3). Each possible decision in the task, $decision_i$, is initially associated with positive feedback, *good*, by adding the chunk "$decision_i$ *good*" to memory 30 times.

Rutledge-Taylor et al. (2014) find that optimism motivates the model to explore the decision space. This is consistent with the *broaden-and-build* (Fredrickson, 2001) theory of positive emotions, which holds that positive emotions broaden the repertoire of actions considered when making decisions.

Each completed trial is represented sequentially as a chunk of the form "*start decision1 cue decision2 feedback*" and added to memory. The first decision is made by querying memory with "*start ? good*" and the second is made by querying with "*start decision1 cue ? good*." Both decisions use a recall with partial matching. The HDM model learns to perform the task well at a rate similar to humans (see Fig. 10b). We replicate Rutledge-Taylor et al.'s model here. The HDM model uses 256-dimensional vectors. Results are averaged across 100 runs of the model.

HDM captures the general pattern of results from human participants in this task: Learning to correctly choose *V* is the easiest ($p<.0001$, repeated measures permutation



(a) Humans (b) HDM

Fig. 10. Rate at which humans and HDM make optimal decisions. (a) Adapted from Walsh and Anderson (2011). Error bars indicate standard error.

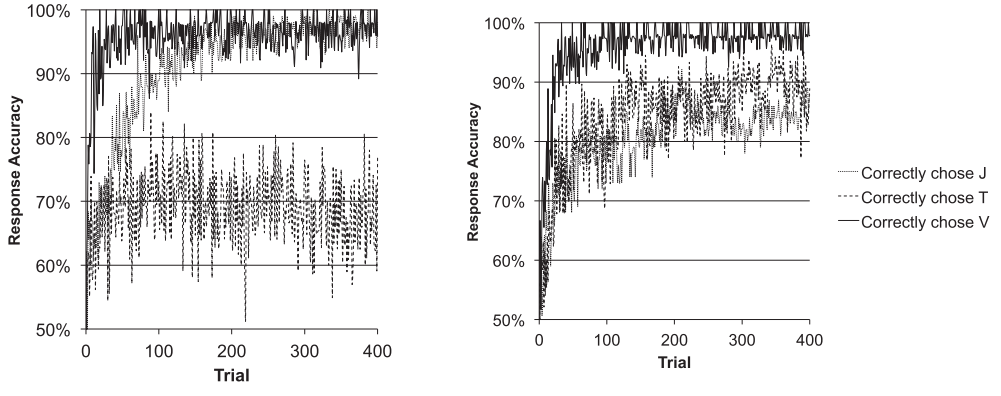
test), learning to correctly choose T is the hardest ($p<.0001$), and overall performance improves from the first block of 200 trials to the second ($p<.0001$).

However, while HDM learns to correctly choose T at greater than chance ($p<.0001$), HDM correctly chooses T in the second block of trials much less often than human participants (64% for HDM vs. 79% for humans). Fig. 11 shows the rate at which HDM makes optimal decisions over the 400 trials, averaged across 100 runs of the model. Fig. 11a illustrates that after the first 50 trials, HDM's ability to correctly choose T does not improve further, despite performance well below 100% correct.

HDM struggles to learn to correctly choose T because, in absolute terms, given the choice between T and V, V is more likely to yield positive feedback. Choosing T is only correct conditional on having seen a particular cue (◆, see Fig. 9). As discussed in Section 6, and unlike an instance-based learning model implemented using DM, HDM underestimates low-probability events. As such, the probability of receiving positive feedback from T will be underestimated by the HDM model.

A critical difference between procedural memory and declarative memory is that procedural memory uses reinforcement learning. In reinforcement learning, an association is learned as a function of how surprising it is. How surprising an observation is can be measured by the magnitude of the difference between prediction and observation. Walsh and Anderson (2011) find that human performance on the task is consistent with temporal difference reinforcement learning models. We hypothesize that the performance of Rutledge-Taylor et al.'s (2014) model could be improved by implementing surprise-driven reinforcement learning in HDM.

To implement surprise in HDM, we borrow from MINERVA-AL (Jamieson, Crump, & Hannah, 2012), an instance-based model of associative learning. MINERVA-AL implements surprise as *discrepancy encoding*: The model learns the difference between the observed and expected outcome. When using discrepancy encoding, HDM predicts the next symbol at each step of the task, conditional on the symbols seen so far. At the end



(a) Normal encoding

(b) Discrepancy encoding

Fig. 11. HDM response accuracy across trials, without (a) and with (b) discrepancy encoding.

of each round, the difference between the observed sequence and the predicted sequence is added to memory. For example, if the predicted sequence is *start J* ▮ *V good* but the observed sequence is *start J* ◆ *V bad*, then the sequence of vectors added to memory is:

$$\mathbf{e}_{\text{start}}, \mathbf{e}_J, (\mathbf{e}_{\blacklozenge} - \mathbf{e}_{\blacksquare}), \mathbf{e}_V, (\mathbf{e}_{\text{bad}} - \mathbf{e}_{\text{good}}). \tag{23}$$

With discrepancy encoding, HDM's ability to correctly choose *T* improves over the 400 trials (see Fig. 11b compared to Fig. 11a) such that by the end, HDM correctly chooses *T* as often as humans (80% for HDM vs. 79% for humans; see Fig. 12).

9. Related work

In the preceding discussion, we demonstrate that HDM can account for interference effects in declarative memory, probability judgments, and learning an iterated decision task. HDM is, however, part of a wider family of related models, with broader applications to modeling language and math cognition, game playing, and knowledge representation and inference. HDM is also not the only approach to integrating modern machine learning techniques with cognitive architectures. In what follows, we discuss models closely related to HDM and other approaches to re-expressing cognitive architectures in a vector space.

9.1. Related models

HDM is closely related to the following models, and as such, can replicate human performance on the same tasks.

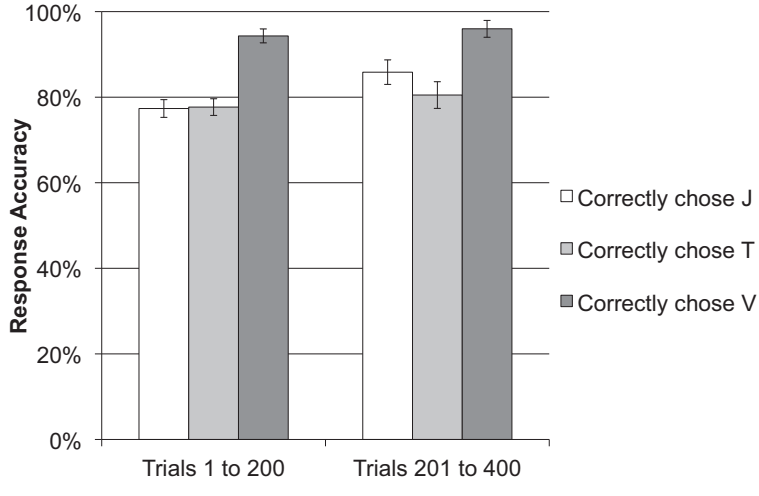


Fig. 12. Rate at which HDM with discrepancy encoding makes optimal decisions. Error bars indicate standard error. As in Fig. 10b, HDM uses 256-dimensional vectors and results are averaged across 100 runs of the model.

9.1.1. BEAGLE

BEAGLE can capture semantic similarity between words (Jones & Mewhort, 2007) and semantic priming effects in lexical decision tasks (Jones, Kintsch, & Mewhort, 2006). For example, people are faster to recognize the word *pepper* when primed with *salt*. Semantic priming is predicted by the distance between BEAGLE's memory vectors in the high-dimensional space. BEAGLE has also been used for early detection of Alzheimer's disease by identifying abnormal language deficits (Johns et al., 2013).

9.1.2. Hierarchical holographic model

The hierarchical holographic model, a recursive variant of BEAGLE with multiple levels of representations, is able to learn arbitrarily abstract relationships (Kelly et al., 2020). Sensitivity to abstract relationships is useful for capturing syntactic similarity between words, for ordering words into grammatical sentences, and for being able to distinguish between grammatical and ungrammatical word orderings, even in the case of nonsensical sentences that lack semantics. For example, both humans and the hierarchical holographic model show a preference for "Colorless green ideas sleep furiously" over "Furiously sleep ideas green colorless."

9.1.3. Dynamically structured holographic memory

Dynamically structured holographic memory (Rutledge-Taylor et al., 2014) can account for the problem size effect: the finding that smaller sums (e.g., $2+3=5$) are easier to remember than larger sums (e.g., $5+7=12$). DSHM is also able to account for human decisions when playing rock–paper–scissors, demonstrating the model's ability to adapt quickly to rapidly changing patterns in behavior.

9.1.4. K-HDM

K-HDM (Arora, West, Brook, & Kelly, 2018) is a variant of HDM that incorporates an ontology proposed by Kant (1781) as a method of representing knowledge of the world and making inferences using that knowledge. K-HDM uses an ontology in order to better support modeling general intelligence.

Taken together with our findings, the aforementioned research demonstrates that HDM's algorithm, and, more broadly, distributional semantics models and vector-symbolic architectures, are capable and versatile techniques for modeling human learning, memory, and knowledge.

9.2. Episodic memory

Declarative memory in ACT-R is understood as comprising both episodic and semantic memory. Episodic memory is typically associated with the hippocampus and semantic memory with cortical regions, such as the temporal lobe. If there is a computational distinction to be made between semantic and episodic memories, it is the degree of aggregation. Recalling a specific episode or event relies on narrow focused retrieval from a

relatively small set of memories (i.e., the memories of that event), whereas recalling a fact or meaning relies on broad focused retrieval across a large set of memories (i.e., all memories in the agent's lifetime related to that fact).

HDM is a modified model of the mental lexicon, though, as we demonstrate in this paper, the approach used by HDM has broad applicability to modeling memory in general. However, HDM may be more appropriately understood as a model of semantic memory. A good candidate for an episode memory model is the MINERVA class of memory models (e.g., Hintzman, 1986; Jamieson et al., 2018), a vector-based model of human memory that stores one vector for each memory trace (i.e., ACT-R chunk) and has strong commonalities with ACT-R DM (Dimov, 2016).

To account for the full capabilities of human memory across all timescales of learning, it may be necessary to adopt a content-addressable auto-associative memory model to represent episodic memory (such as MINERVA) and a distributional semantics model to represent semantic memory (such as HDM). The practical difference between these two types of memory model is primarily the granularity of the stored representations: An episodic memory system stores episodes, specific events that occur in the life of the agent, whereas a semantic memory system stores concepts, which can be understood as collections of events interrelated by a shared pattern.

9.3. Cognitive architectures

While modern machine learning techniques have, for the most part, not yet been integrated into cognitive architectures (Kotseruba & Tsotsos, 2018), there are some notable exceptions. The cognitive architectures LIDA and SPAUN both take an approach similar to HDM to model cognitive functions.

9.3.1. LIDA

The LIDA cognitive architecture (Learning Intelligent Distribution Agent; S. Franklin et al., 2016) uses a high-dimensional vector-symbolic model of long-term memory in order to capture the dynamics of interference and forgetting. Specifically, LIDA uses Sparse Distributed Memory (SDM; Kanerva, 1988), which, like HDM, uses paired environment and memory vectors, but the vectors in SDM are referred to as *addresses* and *words*, respectively, by analogy to the Random Access Memory. However, unlike HDM, there is not a one-to-one correspondence in SDM between *addresses* and the items in the environment. Instead, items are represented distributed across multiple *addresses* and the corresponding *words*. The advantage of this is that SDM can in principle handle environments where the items are initially unknown, but the disadvantage is that SDM is less well optimized than HDM for a given set of items.

SDM also encodes information quite differently from HDM. While SDM simply stores the item itself in the *word* vector, HDM stores the item's associations in the memory vector. As a result, HDM can compactly encode a network of associations between items, allowing it to model, for example, interference in the fan effect and the semantics of words in natural language. That said, we believe that HDM could be re-implemented as

an SDM and that SDM is a sufficiently versatile architecture that it could be used to account for many of the same effects as HDM.

LIDA is a symbolic architecture and thus has to translate between symbols and vectors when storing information in SDM. Re-implementing the entirety of LIDA as a vector-symbolic architecture (e.g., using holographic vectors) has been proposed (Snaider & Franklin, 2014) but remains a matter of future work.

9.3.2. SPAUN

The SPAUN cognitive architecture (Semantic Pointer Architecture Unified Network; Eliasmith, 2013), like HDM, uses holographic vectors, but is implemented as a biologically realistic spiking neural model using the NENGO neural modeling platform. SPAUN is a lower level cognitive architecture, concerned with how, exactly, cognitive functions are realized by the brain. As such, SPAUN is computationally intensive to simulate on conventional computers and is best run on neuromorphic hardware.

While SPAUN has a robust model of procedural memory (i.e., the basal ganglia; Stewart, Bekolay, & Eliasmith, 2012), SPAUN has yet to standardize on a model of declarative memory. HDM could potentially be implemented in NENGO and integrated into SPAUN as a declarative memory system. Re-implementing HDM in NENGO would likely require re-expressing HDM in terms of a lower level model (see Section 11 for discussion), such as a Sparse Distributed Memory (SDM; Kanerva, 1988) or a memory tesseract (Kelly, Mewhort, & West, 2017).

10. Conclusion

Our model, HDM, realizes a fundamental principle of human memory identified by the rational analysis of cognition. Namely, the availability of information in memory is an estimate of the probability that the information is useful in the current situation (Anderson & Schooler, 1991; Chater & Oaksford, 1999). HDM realizes this principle through the geometries of the high-dimensional vector space. This geometric property allows HDM to model both the interference effects between related stimuli in the fan effect task and quantum probability effects in probability judgments, such as the conjunction fallacy.

HDM is able to learn to perform an iterated decision-making task with initially unknown payoffs at a rate comparable to human learners. To do so effectively, HDM makes use of two mechanisms: sensitivity to prediction error and a motivation to explore the decision space. When making choices, HDM selects the option that it estimates as having the highest probability of positive feedback. To motivate HDM to try unexplored options, each option is initialized to estimate a high probability of positive feedback (i.e., HDM is *optimistic*). Sensitivity to prediction error (i.e., surprise) is implemented in HDM by weighting events more strongly in memory when the model fails to predict them. The weighting helps HDM learn and make decisions about low-probability events, improving the model's ability to do the decision task and improving the fit to human data.

Furthermore, HDM is able to approximate the primacy and recency effect in free recall, demonstrating an ability to simulate forgetting caused by interference effects from the temporal order of stimuli.

HDM's mechanisms of encoding, retrieval, surprise, and forgetting are intended as architectural features of the model. Conversely, motivation (e.g., optimism) exists outside of HDM and is implemented by the procedural memory system.

HDM is highly scalable, as the memory system is an $N \times 2k$ matrix, where N is the total number of unique slots and values, and k is the vector dimensionality. When training HDM on very large datasets (i.e., millions of chunks), we recommend using thousands of dimensions to maintain encoding fidelity (e.g., $k = 1{,}024$ or $2{,}048$). However, for domains where the number of possible slots and values N that an input can take is very large or even infinite (such as in vision), it may be necessary to adopt a different implementation of HDM, as discussed in Section 11.

An integrated theory of cognition should have both symbolic (e.g., a description in terms of features and values) and sub-symbolic (e.g., a description in terms of vector algebra) components to provide satisfying explanations. We find that distributional semantic representations instantiated using a vector-symbolic architecture are a natural candidate for an account of declarative memory and its learning processes, as well as aspects of what is commonly considered procedural memory and learning.

Our intent is to advance toward a cognitive architecture that is capable of modeling human performance at all scales of learning, from the half-hour lab experiment to skills acquired gradually over a lifetime. By re-implementing ACT-R's declarative memory using distributional semantics, we create a system that can be integrated with modern machine learning techniques in deep learning while retaining long-term memory, single-trial learning, judgment, and other cognitive capacities associated with high-level cognition.

11. Future work

Since Newell (1973) first argued for the necessity of unified theories of cognition, there has been a great deal of work in developing computational, cognitive architectures. However, few of these architectures yet make use of modern, powerful, machine learning techniques. The Common Model of Cognition (Laird et al., 2017) describes a blueprint for realizing a cognitive architecture, consisting of declarative memory, procedural memory, working memory, perceptual systems, and a motor module. HDM is a candidate model for realizing declarative memory, and aspects of procedural memory, in a manner that can be integrated with other vector-based approaches, such as deep-learning models of perception and action.

HDM, however, has two limitations as a model of cognition, both arising from the fact that HDM stores information across a large number of discrete addresses (i.e., memory vectors):

1. HDM is not a good model of proactive and retroactive interference effects, as it cannot exhibit such interference between memories that are stored in separate memory vectors. We approximate such interference effects using a decay mechanism, as ACT-R does, in Section 5. However, there is good reason to believe that decay is merely an approximation, as experiments have shown that the magnitude of forgetting is a function of interference rather than time (see D. R. J. Franklin & Mewhort, 2015, for discussion and an interference-based model of the serial position curve).
2. HDM assumes that the perceptual environment can be tokenized into a finite set of items, each of which is assigned a memory vector. While this assumption works well for language, which can be tokenized into words without difficulty, the requirement limits the ability of HDM to model other, more complex environments with continuous-valued inputs.

Given these limitations, HDM is best understood as a high-level model of how declarative memory is implemented in the brain. HDM could be realized in a lower level model that does not make the simplifying assumption of one memory vector per item, such as a Sparse Distributed Memory (SDM; Kanerva, 1988) or a memory tesseract (Kelly et al., 2017). Such a lower level re-implementation of HDM would allow proactive and retroactive interference to emerge between unrelated items, causing temporal order and serial position effects. Furthermore, relaxing the assumption of one item per memory vector would allow HDM to operate in complex environments that cannot be easily tokenized into items. Thus, both of the limitations of HDM can potentially be addressed using a more complex, lower level architecture.

Acknowledgments

The authors gratefully acknowledge funding from the National Science Foundation (NSF), the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Ontario Ministry of Training, Colleges and Universities. This research was funded by an Ontario Graduate Scholarship and NSERC Post-Doctoral Fellowship to M. A. Kelly, a National Science Foundation grant (BCS-1734304) to David Reitter and M. A. Kelly, and a grant from NSERC to Robert L. West.

Open Research badges

This article has earned Open Data and Open Materials badges. Data and materials are available at https://github.com/ecphory/HDM.

Notes

1. Python ACT-R (Stewart & West, 2007) can be installed from https://github.com/tcstewar/ccmsuite. Python ACT-R with HDM can be downloaded from https://github.com/ecphory/ccmsuite. Example HDM models can be downloaded from https://github.com/ecphory/HDM.
2. Pre-trained GloVe downloaded from: https://nlp.stanford.edu/projects/glove/.

References

Aerts, D., Arguëlles, J. A., Beltran, L., Beltran, L., de Bianchi, M. S., Sozzo, S., & Veloz, T. (2017). Testing quantum models of conjunction fallacy on the World Wide Web. *International Journal of Theoretical Physics*, *56*(12), 3744–3756. https://doi.org/10.1007/s10773-017-3288-8

Anderson, J. R. (1974). Retrieval of propositional information from long-term memory. *Cognitive Psychology*, *6*, 451–474. https://doi.org/10.1016/0010-0285(74)90021-8

Anderson, J. R. (2009). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., & Reder, L. M. (1999). The fan effect: New results and new theories. *Journal of Experimental Psychology: General*, *128*, 186–197. https://doi.org/10.1037/0096-3445.128.2.186

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, *2*, 396–408. https://doi.org/10.1111/j.1467-9280.1991.tb00174.x

Arora, N., West, R., Brook, A., & Kelly, M. A. (2018). Why the common model of the mind needs holographic a-priori categories. *Procedia Computer Science*, *145*, 680–690. (Papers on the Common Model of Cognition) https://doi.org/10.1016/j.procs.2018.11.060

Ben-Asher, N., Dutt, V., & Gonzalez, C. (2013). Accounting for the integration of descriptive and experiential information in a repeated prisoner's dilemma using an instance-based learning model. In B. Kennedy, D. Reitter, & R. St. Amant (Eds.), *Proceedings of the 22nd Annual Conference on Behavior Representation in Modeling and Simulation*. Ottawa, Canada: BRIMS Society. Available at: http://cc.ist.psu.edu/BRIMS/archives/2013/BRIMS2013-135.pdf Accessed September 30, 2020.

Bhatia, S. (2017). Associative judgment and vector space semantics. *Psychological Review*, *124*(1), 1–20. https://doi.org/10.1037/rev0000047

Bhatia, S., Richie, R., & Zou, W. (2019). Distributed semantic representations for modeling human judgment. *Current Opinion in Behavioral Sciences*, *29*, 31–36. https://doi.org/10.1016/j.cobeha.2019.01.020

Boyer-Kassem, T., Duchêne, S., & Guerci, E. (2016). Quantum-like models cannot account for the conjunction fallacy. *Theory and Decision*, *81*(4), 479–510. https://doi.org/10.1007/s11238-016-9549-9

Bruza, P. D., Wang, Z., & Busemeyer, J. R. (2015). Quantum cognition: A new theoretical approach to psychology. *Trends in Cognitive Sciences*, *19*(7), 383–393. https://doi.org/10.1016/j.tics.2015.05.001

Burgess, C., & Lund, K. (1997). Modelling parsing constraints with high-dimensional context space. *Language and Cognitive Processes*, *12*, 177–210. https://doi.org/10.1080/016909697386844

Busemeyer, J. R., Pothos, E. M., Franco, R., & Trueblood, J. (2011). A quantum theoretical explanation for probability judgement errors. *Psychological Review*, *118*, 193–218. https://doi.org/10.1037/a0022542

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, *356*(6334), 183–186. https://doi.org/10.1126/science.aal4230

Chater, N., & Oaksford, M. (1999). Ten years of the rational analysis of cognition. *Trends in Cognitive Sciences*, *3*(2), 57–65. https://doi.org/10.1016/S1364-6613(98)01273-X

Cole, J. R., Ghafurian, M., & Reitter, D. (2017). Linking memory activation and word adoption in social language use via rational analysis. In M. K. van Vugt, A. P. Banks, & W. G. Kennedy (Eds.), *Proceedings of the 15th International Conference on Cognitive Modeling* (pp. 205–210). Coventry, United Kingdom: University of Warwick. Available at: https://iccm-conference.neocities.org/2017/ICCMprogram_files/paper_49.pdf Accessed September 30, 2020

Cole, J. R., & Reitter, D. (2018). The timing of lexical memory retrievals in language production. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Vol. 1, pp. 2017–2027). Available at: http://www.aclweb.org/anthology/N18-1183 Accessed September 30, 2020.

Cox, G. E., Kachergis, G., Recchia, G., & Jones, M. N. (2011). Towards a scalable holographic representation of word form. *Behavior Research Methods*, *43*, 602–615. https://doi.org/10.3758/s13428-011-0125-5

Dimov, C. M. (2016). Connections between act-r's declarative memory system and minerva2. In A. Papafragou, D. Grodner, D. Mirman, & J. C. Trueswell (Eds.), *Proceedings of the 38th annual meeting of the Cognitive Science Society* (pp. 492–495). Austin, TX: Cognitive Science Society. Available at: https://cogsci.mindmodeling.org/2016/papers/0096/paper0096.pdf Accessed September 30, 2020

Ebbinghaus, H. (1885). *Memory: A contribution to experimental psychology*. New York: Dover.

Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition*. New York: Oxford University Press.

Franklin, D. R. J., & Mewhort, D. J. K. (2015). Memory as a hologram: An analysis of learning and recall. *Canadian Journal of Experimental Psychology*, *69*, 115–135. https://doi.org/10.1037/cep0000035

Franklin, S., Madl, T., Strain, S., Faghihi, U., Dong, D., Kugele, S., Snaider, J., Agrawal, P., & Chen, S. (2016). A LIDA cognitive model tutorial. *Biologically Inspired Cognitive Architectures*, *16*, 105–130. https://doi.org/10.1016/j.bica.2016.04.003

Fredrickson, B. L. (2001). The role of positive emotions in positive psychology: The broaden-and-build theory of positive psychology. *American Psychologist*, *56*, 218–226. https://doi.org/10.1037/0003-066X.56.3.218

Gayler, R. W. (2003). Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. In P. Slezak (Ed.), *Proceedings of the Joint International Conference on Cognitive Science* (pp. 133–138). Sydney, Australia: University of New South Wales. Available at: http://cogprints.org/3983/ Accessed September 30, 2020

Gonzalez, C., & Ben-Asher, N. (2014). Learning to cooperate in the prisoner's dilemma: Robustness of predictions of an instance-based learning model. In P. Bello, M. Guarini, M. McShane, & B. Scassellati (Eds.), *Proceedings of the 36th annual conference of the Cognitive Science Society* (pp. 2287–2292). Austin, TX: Cognitive Science Society. Available at: https://escholarship.org/uc/item/8n6437tp Accessed September 30, 2020

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, *27*(4), 591–635. https://doi.org/10.1016/S0364-0213(03)00031-4

Griffiths, T. L., Steyvers, M., & Tenenbaum, J. B. (2007). Topics in semantic representation. *Psychological Review*, *114*, 211–244. https://doi.org/10.1037/0033-295X.114.2.211

Heathcote, A., Brown, S., & Mewhort, D. J. K. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, *7*(2), 185–207. https://doi.org/10.3758/BF03212979

Hertwig, R., Barron, G., Weber, E. U., & Erev, I. (2004). Decisions from experience and the effect of rare events in risky choice. *Psychological Science*, *15*(8), 534–539. https://doi.org/10.1111/j.0956-7976.2004.00715.x

Hintzman, D. L. (1986). "Schema abstraction" in multiple-trace memory models. *Psychological Review*, *93*, 411–428. https://doi.org/10.1037/0033-295X.95.4.528

Jamieson, R. K., Avery, J. E., Johns, B. T., & Jones, M. N. (2018). An instance theory of semantic memory. *Computational Brain & Behavior*, *1*(2), 119–136. https://doi.org/10.1007/s42113-018-0008-2

Jamieson, R. K., Crump, M. J. C., & Hannah, S. D. (2012). An instance theory of associative learning. *Learning & Behavior*, *40*, 61–82. https://doi.org/10.3758/s13420-011-0046-2

Johns, B. T., Jones, M. N., & Mewhort, D. J. K. (2016). Experience as a free parameter in the cognitive modeling of language. In A. Papafragou, D. Grodner, D. Mirman, & J. C. Trueswell (Eds.), *Proceedings of the 38th annual meeting of the Cognitive Science Society* (pp. 1325–1330). Austin, TX: Cognitive Science Society. https://cogsci.mindmodeling.org/2016/papers/0397/ Accessed September 30, 2020

Johns, B. T., Taler, V., Pisoni, D. B., Farlow, M. R., Hake, A. M., Kareken, D. A., Unverzagt, F. W., & Jones, M. N. (2013). Using cognitive models to investigate the temporal dynamics of semantic memory impairments in the development of Alzheimer's disease. In R. West & T. Stewart (Eds.), In *Proceedings of the 12th International Conference on Cognitive Modeling* (pp. 23–28). Ottawa, Canada: Carleton University. Available at: http://iccm-conference.org/2013-proceedings/papers/0004/index.html Accessed September 30, 2020

Jones, M. N., Kintsch, W., & Mewhort, D. J. K. (2006). High-dimensional semantic space accounts of priming. *Journal of Memory and Language*, *55*, 534–552. https://doi.org/10.1016/j.jml.2006.07.003

Jones, M. N., & Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, *114*, 1–37. https://doi.org/10.1037/0033-295X.114.1.1

Kanerva, P. (1988). *Sparse distributed memory*. Cambridge, MA: The MIT Press.

Kanerva, P. (2009). Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, *1*, 139–159. https://doi.org/10.1007/s12559-009-9009-8

Kant, I. (1781). *Critique of pure reason*. Indianapolis, IN: Hackett . (Translated by Werner S. Pluhar, 1987).

Kelly, M. A., Blostein, D., & Mewhort, D. J. K. (2013). Encoding structure in holographic reduced representations. *Canadian Journal of Experimental Psychology*, *67*, 79–93. https://doi.org/10.1037/a0030301

Kelly, M. A., Ghafurian, M., West, R. L., & Reitter, D. (2020). Indirect associations in learning semantic and syntactic lexical relationships. *Journal of Memory and Language*, *115*, 104153. https://doi.org/10.1016/j.jml.2020.104153

Kelly, M. A., Mewhort, D. J. K., & West, R. L. (2017). The memory tesseract: Mathematical equivalence between composite and separate storage memory models. *Journal of Mathematical Psychology*, *77*, 142–155. https://doi.org/10.1016/j.jmp.2016.10.006

Kersten, L., West, R. L., & Brook, A. (2016). Leveling the field: Talking levels in cognitive science. In A. Papafragou, D. Grodner, D. Mirman, & J. C. Trueswell (Eds.), *Proceedings of the 38th annual meeting of the Cognitive Science Society* (pp. 2399–2404). Austin, TX: Cognitive Science Society. Available at: https://cogsci.mindmodeling.org/2016/papers/0415/paper0415.pdf Accessed September 30, 2020

Kievit-Kylar, B., & Jones, M. (2011). The semantic pictionary project. In L. Carlson, C. Hoelscher, & T. Shipley (Eds.), *Proceedings of the 33rd annual conference of the Cognitive Science Society* (pp. 2229–2234). Austin, TX: Cognitive Science Society. Available at: https://mindmodeling.org/cogsci2011/papers/0520 Accessed September 30, 2020

Kotseruba, I., & Tsotsos, J. K. (2020). 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, *53*(1), 17–94. https://doi.org/10.1007/s10462-018-9646-y Accessed September 30, 2020

Laird, J. E. (2012). *The Soar cognitive architecture*. Cambridge, MA: MIT Press.

Laird, J. E., Lebiere, C., & Rosenbloom, P. S. (2017). A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *AI Magazine*, *38*(4), 13–26. https://doi.org/10.1609/aimag.v38i4.2744

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211–240. https://doi.org/10.1037/0033-295X.104.2.211

Lebiere, C., Wallach, D., & West, R. L. (2000). A memory-based account of the prisoner's dilemma and other 2x2 games. In N. Taatgen & J. Aasman (Eds.), *Proceedings of the third international conference on cognitive modeling* (pp. 185–193). Groningen, The Netherlands: Universal Press.

Lebiere, C., & West, R. L. (1999). A dynamic ACT-R model of simple games. In M. Hahn & S. C. Stoness (Eds.), *Proceedings of the twenty first annual meeting of the Cognitive Science Society* (pp. 296–301). Mahwah, NJ: Lawrence Erlbaum Associates. Available at: https://cognitivesciencesociety.org/wp-content/uploads/2019/01/cogsci_21_1999_Vancouver.pdf Accessed September 30, 2020

Lejarraga, T., Dutt, V., & Gonzalez, C. (2012). Instance-based learning: A general model of repeated binary choice. *Journal of Behavioral Decision Making*, *25*(2), 143–153. https://doi.org/10.1002/bdm.722

Lieto, A., Chella, A., & Frixione, M. (2017). Conceptual spaces for cognitive architectures: A lingua franca for different levels of representation. *Biologically Inspired Cognitive Architectures*, *19*, 1–9. https://doi.org/10.1016/j.bica.2016.10.005

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 26* (pp. 3111–3119). Curran Associates, Inc. Available at: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf Accessed September 30, 2020

Murdock, B. B. (1962). The serial position effect of free recall. *Journal of Experimental Psychology*, *64*(5), 482–488. https://doi.org/10.1037/h0045106

Murdock, B. B. (1982). A theory for the storage and retrieval of item and associative information. *Psychological Review*, *89*, 609–626. https://doi.org/10.1037/0033-295X.89.6.609

Newell, A. (1973). You can't play 20 questions with nature and win. In W. G. Chase (Ed.), *Visual information processing*, (pp. 283–308). New York: Academic Press.

Oberauer, K., & Lewandowsky, S. (2013). Evidence against decay in verbal working memory. *Journal of Experimental Psychology: General*, *142*(2), 380–411. https://doi.org/10.1037/a0029588

Pavlik Jr., P. I., & Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive Science*, *29*(4), 559–586. https://doi.org/10.1207/s15516709cog0000_14

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1532–1543). Available at: http://www.aclweb.org/anthology/D14-1162 Accessed September 30, 2020

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, *6*, 623–641. https://doi.org/10.1109/72.377968

Reitter, D., Keller, F., & Moore, J. D. (2011). A computational cognitive model of syntactic priming. *Cognitive Science*, *35*(4), 587–637. https://doi.org/10.1111/j.1551-6709.2010.01165.x

Ricker, T. J., Spiegel, L. R., & Cowan, N. (2014). Time-based loss in visual short-term memory is from trace decay, not temporal distinctiveness. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *40*(6), 1510–1523. https://doi.org/10.1037/xlm0000018

Ritter, F. E., & Schooler, L. J. (2001). Learning curve. In N. J. Smelser & P. B. Baltes (Eds.), *International encyclopedia of the social and behavioral sciences* (pp. 8602–8605). Oxford: Pergamon. https://doi.org/10.1016/B0-08-043076-7/01480-7

Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, *10*(3), e1488. https://doi.org/10.1002/wcs.1488

Rosenbloom, P. S., Demski, A., & Ustun, V. (2016). The Sigma cognitive architecture and system: Towards functionally elegant grand unification. *Journal of Artificial General Intelligence*, *7*(1), 1–103. https://doi.org/10.1515/jagi-2016-0001

Rutledge-Taylor, M. F., Kelly, M. A., West, R. L., & Pyke, A. A. (2014). Dynamically structured holographic memory. *Biologically Inspired Cognitive Architectures*, *9*, 9–32. https://doi.org/10.1016/j.bica.2014.06.001

Rutledge-Taylor, M. F., Vellino, A., & West, R. L. (2008). A holographic associative memory recommender system. In *Proceedings of the 3rd international conference on digital information management* (pp. 87–92). London, UK. https://doi.org/10.1109/ICDIM.2008.4746700

Sanner, S., Anderson, J. R., Lebiere, C., & Lovett, M. (2000). Achieving efficient and cognitively plausible learning in backgammon. In *Proceedings of the seventeenth international conference on machine learning (ICML-2000)* (pp. 823–830). Stanford, CA. https://doi.org/10.1184/R1/6613298.v1

Snaider, J., & Franklin, S. (2014). Vector LIDA. *Procedia Computer Science*, *41*, 188–203. https://doi.org/10.1016/j.procs.2014.11.103

Steine-Hanson, Z., Koh, N., & Stocco, A. (2018). Refining the common model of cognition through large neuroscience data. *Procedia Computer Science*, *145*, 813–820. https://doi.org/10.1016/j.procs.2018.11.026

Stewart, T. C., Bekolay, T., & Eliasmith, C. (2012). Learning to select actions with spiking neurons in the basal ganglia. *Frontiers in Neuroscience*, *6*, 1–14. https://doi.org/10.3389/fnins.2012.00002

Stewart, T. C., & West, R. L. (2007). Deconstructing and reconstructing ACT-R: Exploring the architectural space. *Cognitive Systems Research*, *8*(3), 227–236. https://doi.org/10.1016/j.cogsys.2007.06.006

Stocco, A., Laird, J., Lebiere, C., & Rosenbloom, P. (2018). Empirical evidence from neuroimaging data for a standard model of the mind. In T. T. Rogers, M. Rau, X. Zhu, & C. W. Kalish (Eds.). *Proceedings of the 40th annual conference of the Cognitive Science Society*, 1094–1099). Austin, TX: Cognitive Science Society. Available at: https://mindmodeling.org/cogsci2018/papers/0216/ Accessed September 30, 2020

The British National Corpus (version 3, BNC XML ed.). (2007). Bodleian Libraries. University of Oxford, on behalf of the BNC Consortium. Available at: http://www.natcorp.ox.ac.uk/ Accessed September 30, 2020

Tversky, A., & Kahneman, D. (1983). Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment. *Psychological Review*, *90*(4), 293–315. https://doi.org/10.1037/0033-295X.90.4.293

Walsh, M. M., & Anderson, J. R. (2011). Learning from delayed feedback: Neural responses in temporal credit assignment. *Cognitive, Affective, and Behavioral Neuroscience*, *11*, 131–143. https://doi.org/10.3758/s13415-011-0027-0

Wang, Z., Solloway, T., Shiffrin, R. M., & Busemeyer, J. R. (2014). Context effects produced by question orders reveal quantum nature of human judgments. *Proceedings of the National Academy of Sciences*, *111* (26), 9431–9436. https://doi.org/10.1073/pnas.1407756111