Bridging Cognitive Architectures and Generative Models with Vector Symbolic Algebras

P. Michael Furlong, Chris Eliasmith
Centre for Theoretical Neuroscience, University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada, N2L 3G1
michael.furlong@uwaterloo.ca, celiasmith@uwaterloo.com

Abstract

Recent developments in generative models have demonstrated that with the right data set, techniques, computational infrastructure, and network architectures, it is possible to generate seemingly intelligent outputs, without explicitly reckoning with underlying cognitive processes. The ability to generate novel, plausible behaviour could be a boon to cognitive modellers. However, insights for cognition are limited, given that generative models' blackbox nature does not provide readily interpretable hypotheses about underlying cognitive mechanisms. On the other hand, cognitive architectures make very strong hypotheses about the nature of cognition, explicitly describing the subjects and processes of reasoning. Unfortunately, the formal framings of cognitive architectures can make it difficult to generate novel or creative outputs. We propose to show that cognitive architectures that rely on certain Vector Symbolic Algebras (VSAs) are, in fact, naturally understood as generative models. We discuss how memories of VSA representations of data form distributions, which are necessary for constructing distributions used in generative models. Finally, we discuss the strengths, challenges, and future directions for this line of work.

Introduction

Recent developments in generative models have demonstrated that with the right data set, techniques, computational infrastructure, and network architectures, it is possible to learn distributions over complex data and processes, like images, sound, and language (*e.g.*, Ramesh et al. 2021; Mittal et al. 2021; Ramesh et al. 2022; Brown et al. 2020; Kojima et al. 2022). Samples drawn from these distributions can generate seemingly intelligent outputs without explicitly reckoning with underlying cognitive processes. The ability to generate novel, plausible behaviour could be a boon to cognitive modellers. However, the insights to cognition they provide are limited, given that their blackbox nature does not provide readily human interpretable hypotheses about the representations and functional manipulations that generative models employ.

Cognitive architectures, on the other hand, make very strong hypotheses about the nature of cognition in organisms. Where generative models lack explicit statements about the underlying computation, cognitive architectures are explicit about modularity, manipulated symbols, abstractions, and about the mechanism of reasoning (*e.g.*, the common cognitive architecture; Laird, Lebiere, and Rosenbloom 2017), often explicitly codifying reasoning in rule-based production systems (Anderson et al. 1995). Unfortunately, the structural form adopted by many cognitive architectures that is imposed by programming language abstractions can make it difficult to generate novel outputs. Perturbations to symbolic representations need to be sensitive to the specific imposed structure, while in generative models, simple perturbations of locally smooth representations can result in sensible outputs. The creativity of generative models and the legibility of cognitive architectures are compelling reasons to investigate a unification of these two approaches in the hopes of generating hypotheses about cognition that can embody both creativity and explainability.

Vector Symbolic Algebras (VSAs)[1] may be able to bridge this gap. VSAs are a family modelling frameworks that unify symbolic and non-symbolic data (Smolensky et al. 2022), and can be used to construct functional cognitive architectures that can be translated directly into populations of neurons, as in the case of SPAUN (Eliasmith et al. 2012). In these frameworks data are represented as high-dimensional vectors, and then manipulated using a defined set of operations. Representations of data structures can be composed using the operators and existing data, and then manipulated in order to conduct analogical reasoning, construct content-addressable memories, or integrate with connectionist models, enabling learning.

Previous work has shown that certain VSAs can be used to represent probability distributions (Joshi, Halseth, and Kanerva 2017; Frady et al. 2021; Furlong and Eliasmith 2022). For the right choice of representation and operators, certain VSAs have a mathematical relationship to probability, and memories of these vector representations approximate distributions. We propose to show that cognitive architectures that rely on these methods are, in fact, naturally understood as generative models.

To support this argument, we first briefly outline the VSA that we work in, and show how it relates to probability. Next,

[1]More commonly *vector symbolic architectures*, but we prefer *algebras* to distinguish from cognitive architectures.

we discuss how representations of individual data points are inherently distributions, via a relationship to kernel approximations. Then we show how memories combine with representations to construct distributions over input data. Given the centrality of memory to cognitive architectures, the relationship between memory and distribution modelling makes for a fundamental connection to generative models. Finally, we discuss the implications of these results, how they relate to modern generative models, challenges in using these particular representations, and outline avenues of future work.

Preliminaries

The connection between VSAs and probability is illustrated by considering Kernel Density Estmimators (KDEs). In KDEs the estimated probability of a query point, $\mathbf{x}$, is the average similarity between $\mathbf{x}$ and the n elements of the training set $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Similarity is measured using a kernel function, $k(\cdot, \cdot)$, which is typically a valid probability density function. We define a KDE as $f_X(\mathbf{x}) = \frac{1}{nh}\sum_{i=1}^{n} k_h\left(\mathbf{x}, \mathbf{x}_i\right)$ for bandwidth $h \in \mathbb{R}^+$.

The memory resources of KDEs grow with the size of the data set, $\mathcal{D}$, as does the time to compute a query. This efficiency can be improved by using the kernel trick – representing a kernel function between two data points as the dot product of two vector embeddings of those data points (Rahimi, Recht et al. 2007). That is $k(\mathbf{x}, \mathbf{x}') \approx \phi(\mathbf{x}) \cdot \phi(\mathbf{x}')$. With this trick we can rewrite the KDE as $f_X(\mathbf{x}) = \phi(\mathbf{x}/h) \cdot \left(\frac{1}{nh}\sum_{i=1}^{n}\phi(\mathbf{x}_i)\right)$, with the sum being a vector that is fixed at training time.

Rahimi, Recht et al. (2007) demonstrated a method for generating the embedding, $\phi(\cdot)$, through randomly selecting frequency components from the power spectral density of the kernel being approximated. VSAs have been said to generalize the kernel trick (Frady et al. 2021), allowing one to construct kernels over more complex data structures. Next we briefly review the Holographic Reduced Representation (HRR) VSA and the operators it uses to represent and manipulate data.

VSAs and Holographic Reduced Representations

VSAs are a family of algebras that can be used to implement cognitive models that can be translated into neural networks (Smolensky, Legendre, and Miyata 1992; Kanerva 1988, 2009; Plate 1995; Eliasmith 2013), a characteristic that was employed by the Semantic Pointer Architecture (SPA) in constructing SPAUN (Eliasmith et al. 2012), a cognitive architecture implemented in spiking neurons. While the SPA can be implemented using different algebras (Eliasmith 2013), here we discuss probabilistic modelling using the HRRs of Plate (1995), primarily focusing on representations of continuous data, using representations we refer to as Spatial Semantic Pointers (SSPs) (Komer et al. 2019; Komer 2020; Dumont and Eliasmith 2020). Below we briefly describe the operations that we use from the HRR VSA and tie them to representing probabilities.

Vector space We restrict ourselves to *unitary vectors*, vectors whose Fourier components all have magnitude one. We write this

$$\mathbf{v} = \mathcal{F}^{-1}\left\{\mathrm{e}^{\mathrm{i}\mathbf{a}}\right\} \tag{1}$$

where $\mathbf{a} = (a_1, \ldots, a_d)^T$ are a collection of frequencies, $a_i \in [0, 2\pi]$, and d is the dimensionality of the vector. We also enforce conjugate symmetry in the vector $\mathbf{a}$ to ensure the inverse Fourier transform ($\mathcal{F}^{-1}$) is entirely real-valued.

The choice of $\mathbf{a}$ can be random, like the method of Random Fourier Featuress (RFFs), or we can apply designed representations to model observed biological phenomena (Dumont and Eliasmith 2020). Ultimately, the choice of frequency components shapes how the similarity between represented data changes as the content of the data changes. In this paper we do not consider optimal choices of generating distributions, but we do observe that selecting different distributions can result in more efficient representations, depending on the task at hand.

Operations: **Binding**, $\circledast$, implemented with circular convolution, is at the core of our approach — in VSAs, binding is used to combine two symbols or state representations together to produce slot-filler pairs, *e.g.*, combining a sensing modality type with a sensor value, or an edge in a graph with the edge's traversal cost. In the context of probability, binding two vectors induces a kernel product.

We employ an extension of binding, called *fractional* binding (Plate 1992; Komer et al. 2019), to represent data in a continuous domain, $\mathcal{X} \subseteq \mathbb{R}^m$, into a high-dimensional vector representation (eq. (2)).

$$\phi_{\mathrm{X}}(\mathbf{x}/h) = \mathcal{F}^{-1}\left\{\mathrm{e}^{\mathrm{i}A_X\mathbf{x}/h}\right\} \tag{2}$$

Where $\mathbf{x} \in \mathcal{X}$ and h is a length scale parameter, as in kernel density estimation, and A_X is the *phase matrix*, each column of which corresponds to an $\mathbf{a}$ vector selected for each dimension of the input domain. A_X and h define a "type" representation in the high-dimensional space for the low-dimensional space.

Similarity between two VSA-encoded objects is computed with the vector dot product, $\cdot$. For SSPs similarity has a strict mathematical meaning through the connection to RFFs, the dot product between two SSPs approximates a kernel function (Voelker 2020; Frady et al. 2021), expressed:

$$k(\mathbf{x}, \mathbf{x}') \approx \phi_{\mathrm{X}}(\mathbf{x}/h) \cdot \phi_{\mathrm{X}}(\mathbf{x}'/h). \tag{3}$$

The kernel induced by the dot product will depend on the choice of A_X.

Bundling is used in VSAs to represent sets of objects. Similarity between a vector and a bundle gives a measure of membership in the set. We use bundles of fractionally-bound objects to represent a distribution, and when we compute the similarity between a query point encoded as an SSP with a bundle of SSPs, we get a quantity that approximates the probability of the query point. In math:

$$\hat{f}(\mathbf{x} \mid \mathcal{D}) \approx \phi_{\mathrm{X}}(\mathbf{x}/h) \cdot \frac{1}{nh}\sum_{\mathbf{x}_i \in \mathcal{D}} \phi_{\mathrm{X}}(\mathbf{x}_i/h) \tag{4}$$

where we can replace the normalized sum $\frac{1}{nh}\sum_{\mathbf{x}_i \in \mathcal{D}} \phi_{\mathrm{X}}(\mathbf{x}_i/h)$ with a memory vector that represents the dataset, $M_{\mathrm{X},n}$. This memory can be updated online, allowing for changes in the distribution that reflect the experience of an agent. As will be discussed below, the choice of A_{X} may cause the induced kernel to take on negative values, making it a *quasi*-probability. However, conversions to probability are possible.

Unbinding is the inverse of binding, and is implemented by binding with the pseudoinverse of the argument, $\phi_{\mathrm{X}}(x) \circledast \phi_{\mathrm{Y}}(y) \circledast \phi_{\mathrm{Y}}^{-1}(y) \approx \phi_{\mathrm{X}}(x)$. In cognitive modelling, unbinding can be used to select from bundles a subset where the querying vector matches. We have found that unbinding can be used to condition memory vectors, $M_{\mathrm{XY}} = \sum_{(x_i,y_i)\in\mathcal{D}} \phi_{\mathrm{X}}(x_i) \circledast \phi_{\mathrm{Y}}(y_i)$, selecting only those elements where $y_i \approx y$, that is:

$$f(x \mid y, \mathcal{D}) \propto \phi_{\mathrm{X}} \cdot \left(M_{\mathrm{XY}} \circledast \phi_{\mathrm{Y}}^{-1}(y)\right). \qquad (5)$$

In prior work we provided an in-depth treatment of probabilistic modelling using SSPs (Furlong and Eliasmith 2022), which permits us to construct distributions over vector spaces. Here we suggest that it may be extended to mixed discrete and continuous representations, although we do not provide rigorous proofs to that end.

Cognitive Architectures Using Appropriate VSAs are Generative Models

The argument we wish to put forward in this paper is that cognitive architectures that are implemented using certain classes of vector symbolic algebras are already inherently generative. We draw this inference from two observations: first, that data represented using vector symbolic architectures induce (quasi-)kernels, and second that memories (typically vector sums) of these representations are vector objects that have a strong mathematical relationship to probability distributions.

HRR Representations Imply Kernel Functions

The vector dot product, the measure of similarity proposed for analogical reasoning with symbolic representations (Plate 1993; Eliasmith and Thagard 2001), also provides a meaningful measure of similarity between continuous-valued data. Specifically, the dot product between real-valued vector data represented using SSPs is a product of sinc functions (*e.g.*, Komer et al. 2019; Voelker 2020; Dumont and Eliasmith 2020; Furlong, Stewart, and Eliasmith 2022). That is to say, if we have a problem domain $\mathcal{X} \subseteq \mathbb{R}^m$ and a projection $\phi_{\mathcal{X}} : \mathbb{R}^m \rightarrow \mathbb{R}^d$, then for two points $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$

$$\phi_{\mathcal{X}}(\mathbf{x}_1) \cdot \phi_{\mathcal{X}}(\mathbf{x}_2) \approx \prod_{i=1}^{m} \mathrm{sinc}(|x_{1,i} - x_{2,i}|). \qquad (6)$$

This relationship, given in the context of SSPs by Voelker (2020), follows naturally from the theory of Random Fourier Features (Rahimi, Recht et al. 2007). While it is not commonly used, the sinc function is admissible for use in kernel methods for estimating probability (Tsybakov 2009).

Similarly, for representing symbolic data, the dot product induces an admissible kernel. We represent atomic concepts by randomly selecting points on the surface of an d-dimensional hypersphere. The kernel function induced by

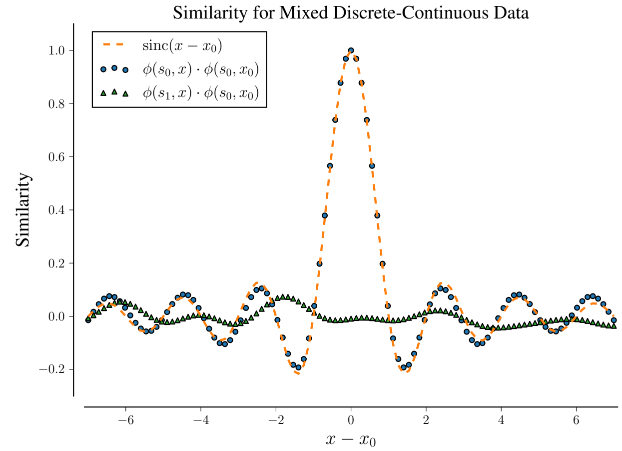

Figure 1: Similarity between points $(s, x) \in \{s_0, s_1\} \times \mathbb{R}$, and an origin point, (s_0, x_0) encoded as an HRR. Circles indicate the similarity between the origin and points where $s = s_0$, triangles indicate $s = s_1$. When the discrete component is equal to the origin, the similarity over x follows the normalized sinc function (solid line). Because sinc can take on negative values it is a *quasi-kernel*, suggesting that these representations induce quasi-probability models.

this representation is the cosine kernel. However, because the points are randomly generated, the likelihood that two vectors are close to each other is relatively low. In fact, for two symbols in a set $s_1, s_2 \in \mathcal{S}$, and a projection $\phi_{\mathcal{S}} : \mathcal{S} \rightarrow \mathbb{R}^d$, we can write the kernel function induced by the dot product as

$$\phi_{\mathcal{S}}(s_i) \cdot \phi_{\mathcal{S}}(s_j) = \begin{cases} 1 & \text{if } s_i = s_j \\ \varepsilon_{ij} & \text{else} \end{cases} \qquad (7)$$

where ε is a random number with $\mathbb{E}[\varepsilon_{ij}] = 0$ and $Var(\varepsilon_{ij}) = 1/d$ (Gosmann and Eliasmith 2016; Voelker, Gosmann, and Stewart 2017). Hence, for arbitrarily large vectors, the likelihood of high similarity becomes arbitrarily small. Furthermore, if the number of symbols that will be represented in the cognitive system is known *a priori*, then one can select a dimensionality and develop a set of phase vectors, $(\mathbf{a}_1, \ldots, \mathbf{a}_{\|\mathcal{S}\|})$, such that the extreme values of ε_{ij} are minimized.

The above two results show that the dot product between representations of atomic data – vectors of numbers or individual symbols – induce kernels that are admissible in probabilistic models. However, it is also the case that by using VSA operations like binding and bundling, we can produce representations of greater complexity, suggesting that such a formulation is not only a language for representing data in cognitive models, but also is a language for composing kernel functions between these data. We appeal to the notion that the product and sum of valid kernel functions are themselves valid kernel functions (Bishop 2006, p296), and that bundling and binding operations represent sums and products of kernels, respectively. To illustrate this point, fig. 1 shows the similarity between mixed

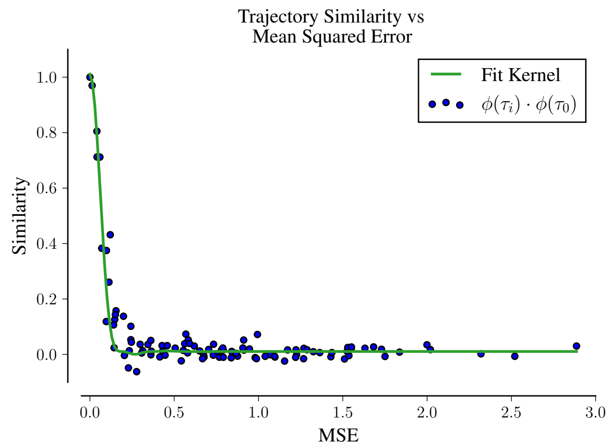

Figure 2: Similarities between 100 random trajectories. As the mean L_2 increases, the similarity decreases.

discrete-continuous points in the space $\{s_0, s_1\} \times \mathbb{R}$, encoded $\phi(s, x) = \phi_{\mathrm{S}}(s) \circledast \phi_{\mathrm{X}}(x)$. When the symbolic elements are equal, we see the kernel follows the sinc function, otherwise the similarity looks like noise. Figure 2 shows the similarity for trajectories encoded using SSPs. We use a modification of the standard method for representations of trajectories using HRRs (Plate 1992; Voelker et al. 2021).

For a trajectory, $\tau = \{(t_1, \mathbf{x}_1), \ldots, (t_n, \mathbf{x}_n)\}$, we define the projection into a vector space:

$$\tilde{\phi}(\tau) = \sum_{i=1}^{n} \phi(t_i) \circledast \phi_{\mathcal{X}}(x_i/h). \tag{8}$$

Which we normalize to be a unit vector, $\phi(\tau) = \tilde{\phi}(\tau)/\|\tilde{\phi}(\tau)\|$. Note, that for this representation the induced kernel should be of the form:

$$k(\tau, \tau') \propto \sum_{i=1}^{n} \mathrm{sinc}(|t_i - t_i'|) \prod_{j=1}^{m} \mathrm{sinc}(|x_{ij} - x_{ij}'|/h) + \eta \tag{9}$$

where η is a noise term due to cross-talk. Since the trajectories we present here cover a two-dimensional space, we approximate the kernel as a function of the mean squared error, ϵ, of the trajectories as $\mathrm{sinc}(k_1\epsilon/h)^3 + k_2$. This function will not be a perfect fit, because the true kernel is a function of the difference in $\mathbf{x}(t)$ at each time point, not the MSE of the trajectories.

What follows from these observations is that when we engage in analogical reasoning about data encoded using these methods, we are also computing a value that has a meaningful relationship to probability. What we show next is that memories constructed from these representations are, inherently, probability distributions.

VSA Memories are Distributions

When compared under the dot product, individual data points are (quasi-)distributions in and of themselves. However, when we aggregate these data into memories we find that they become objects that can represent data sets. We understand memories in the context of a VSA to be a weighted superposition (bundle in the VSA terminology) of one or more vectors encoding data.

In the most basic case, we can assume that we are given a dataset, $\mathcal{D}$ of observations $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ that are drawn from some generating distribution. We create a memory by simply averaging the VSA-encoded representation of these data

$$M_{\mathcal{D}} = \frac{1}{n} \sum_{i=1}^{n} \phi(\mathbf{x}_i). \tag{10}$$

When one computes the dot product between a query point $\phi(\mathbf{x})$ and the memory $M_{\mathcal{D}}$, we see that this approximates a kernel density estimator:

$$\phi(\mathbf{x}) \cdot M_{\mathcal{D}} = \phi(\mathbf{x}) \cdot \left(\frac{1}{n} \sum_{i=1}^{n} \phi(\mathbf{x}_i) \right) \tag{11}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i) \tag{12}$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} k(\mathbf{x}, \mathbf{x}_i). \tag{13}$$

Granted, because the induced kernels can take on negative values, similarities are not strictly probabilities. However, the sinc function, without correction, can be used in density estimation (Davis 1977; Glad, Hjort, and Ushakov 2007), and can be a more efficient kernel than the "optimal" Epanechnikov kernel (Tsybakov 2009, §1.3). However, if strictly non-negative values are required one can employ corrections like squaring the quantity (*i.e.*, Born's rule (Born 1926))

$$p(\mathbf{x}) = (\phi(\mathbf{x}) \cdot M_{\mathcal{D}})^2 \tag{14}$$

or using the biased rectification,

$$p(\mathbf{x}) = \max\{0, \phi(\mathbf{x}) \cdot M_{\mathcal{D}} - b\}, \tag{15}$$

of Glad, Hjort, and Ushakov (2003). Here, b is a bias selected to ensure that $\int_{\mathcal{X}} p(\mathbf{x}) d\mathbf{x} = 1$. Most notably, this last correction takes on the form of a rectified linear neuron, leading to the observation that while using VSAs can provide quasi-probabilistic representations, neurons can transform quasi-probability into exact probability.

With this memory representation and other VSA operations, one can manipulate the memory, like conditioning (through the unbinding operation) or marginalization (through simple linear operations), and can construct networks to implement other information theoretic functions over distributions (Furlong and Eliasmith 2022).

A valid criticism of the above approach is that it does not take into account the fact that cognitive agents are embedded in time, and must necessarily make observations, and hence learn, sequentially. We can imagine a temporal memory that is defined by the difference equation for a low-pass filter:

$$M(t) = (1 - \gamma)\phi(\mathbf{x}(t)) + \gamma M(t - \Delta t) \tag{16}$$

where $\gamma \in [0, 1[$ is a temporal discount factor. This kind of memory is sequentially updated, making it more plausible for agents embedded in time, and since it uses a decay factor instead of an average, it does not require knowledge of the size of the entire data set ahead of time. Further, it gives a distribution over observations that has a temporal aspect. At any given time, T this system implies a memory defined as:

$$M(T) = (1-\gamma)\sum_{t=0}^{T}\gamma^{T-t}\phi(\mathbf{x}(t)). \quad (17)$$

When we compute the probability of a given query point with this new dynamical system, what we see is a kernel density estimator with an exponential decay over time:

$$p(\mathbf{x}) \approx \phi(\mathbf{x})\cdot M(T) \quad (18)$$

$$\approx (1-\gamma)\sum_{t=0}^{T}\gamma^{T-t}k(\mathbf{x},\mathbf{x}(t)). \quad (19)$$

The implication of this is that more recent observations are considered more likely, and hence are more likely to be drawn, should samples be taken from the distribution. This structure, designed to increase the biological fidelity, introduces a recency bias, a phenomena observed in humans that cognitive models should explain.

We can also consider a single neuron attempting to learn the distribution of the data. Taking inspiration from Schölkopf et al. (2001), we trained the perceptron to predict 1 for all input observations. We trained it for 1 epoch with a learning rate set to $1/n$, where n is the number of samples in the dataset. This is a very contrived example, as it assumes that the number of of samples are known, and there is only one presentation of each data point. As the number of epochs grows, this will eventually learn to approximate a function that returns 1 if the sample has been part of the training set, or 0 otherwise. This is consistent with the network of Schölkopf *et al.* learning the support of the distribution, rather than the distribution itself.

Figure 3 shows the distribution learned by each of these methods for constructing memories. We sampled 1000 observations from a one-dimensional bi-modal Gaussian mixture model by sampling 700 observations from one mode, 300 from the other, and then randomly shuffling the observations. The atemporal memory described in eq. (10) provides a good match to the true underlying distribution. The temporal memory, described in eq. (16) was provided with a $\gamma = 0.93$. This value was selected to illustrate the how more recent observations can influence the learned distribution.

The chosen temporal memory over-predicts the more recent observations, forgetting earlier ones, however, it will approximate the atemporal memory as $\gamma \to 1$. After one epoch, the perceptron learns a reasonable approximation of the memory but requires a fine-tuned learning rate. Without engaging in a detailed exploration, we can conclude that common formulations of memory can approximate observed distributions, and the particular formulations have implications for the quality of the distribution that is learned, and hence resulting behaviours.

Provided that a cognitive architecture uses VSAs that permit a probabilistic interpretation, we can reframe the fundamental unit that such an architecture operates on as distributions. Consequently, when characterizing systems that work

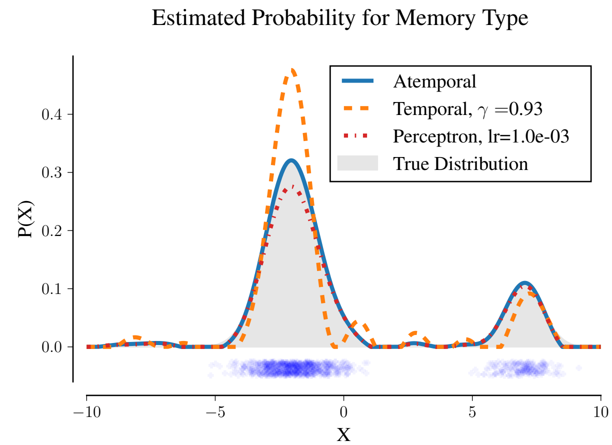

Figure 3: Distributions modelled with three different types of memory: atemporal, temporal, and a perceptron trained to predict membership in the dataset. Dots below the estimated distributions represent the sample points used to generate the memories, with the intensity of colour indicating the relative time of the samples, with darker values being most recent.

on these representations, like populations of neurons representing recent memory, or the synaptic weights of neurons learning distributions, they can be understood as learning generative models, *i.e.*, representations of the data's distribution. Further, if we consider the necessary limitations of biological systems, we begin to see the effects of being embedded in time that cause cognitive models to deviate from optimal models.

Discussion

The vector algebra we used in this paper provides a way of representing states that unify reasoning over symbolic and non-symbolic data, neural implementations, and probability models. This algebra can support the implementation of various cognitive models, but importantly, by representing memory as a superposition of high-dimensional vector representations, we see that memories are equivalent to probability distributions. In the context of the standard model of cognitive architectures, memories - working, episodic, and procedural - are a primary object of concern.

Strengths

Unification of Cognitive Architectures, Generative Models, and Neural Networks VSAs provide a formal algebra for implementing cognitive architectures that supports instantiating cognitive scientists' hypotheses about cognition, as documented by Eliasmith (2013) and Choo (2018). At the same time VSA's vector representations are readily interfaced with neural networks, unifying connectionist approaches with the symbolic (Smolensky, Legendre, and Miyata 1992; Smolensky et al. 2022; Eliasmith 2013). Furthermore, research using the SPA, shows that for any VSA statement there exists at least one neural network that implements that statement. Finally, cognitive architectures using appropriate VSAs that use either VSA memories or memories storied in synaptic weights learn distributions over their experiences, a necessary component of generative models. Using appropriate VSAs, one can construct a single model that simultaneously proposes hypotheses about functional cognition, brain activity and structure, and representations of uncertainty.

Data structures define kernels Defining kernels for complex data structures is a non-trivial problem, but HRR representations provide a simple approach to this problem. Data represented in this VSA support reasoning with analogical similarity through the dot product, which can be converted to a kernel function. Since every data structure implies a kernel, humans can easily construct kernels simply by describing the data they are representing. Ready access to kernels for complex data makes it easier to build the distributions necessary for generative models. Furthermore, because VSAs provide an algebra for designing kernels, one could apply techniques for kernel structure search (*e.g.*, Duvenaud et al. 2013) to find representations that more readily explain human behaviour.

Integration with Machine Learning Techniques Furthermore, probability distribution representations have proven beneficial in machine learning applications. Because VSAs generalize kernel embeddings, they can be used to implement efficient algorithms for exploration (Furlong, Stewart, and Eliasmith 2022). We have also had success using them as a representational basis for reinforcement learning tasks, finding solutions with less variability than Deep Q Networks on benchmark RL tasks (Bartlett et al. 2023).

Continuing in the vein of borrowing from machine learning literature – these representations are differentiable. Consequently, gradient based methods can be applied and even propagated through cognitive systems to learn improved projections from perception or to motor activity in a task constrained manner. Allowing that the biological plausibility of backpropagation is debatable, it is undeniably a useful tool, and adopting these techniques permits their integration into cognitive architectures.

Challenges

Resource-Accuracy Trade-Off High-dimensional representations necessarily require large numbers of elements. Representing these vectors in neural networks requires further resources, as there is not necessarily a one-to-one correspondence between vector dimensionality and the number of neurons in the representing population.

High-dimensional vectors are preferred because randomly selected vectors are only orthogonal in expectation – the higher the dimensionality, the lower the variance of the dot product between vectors. Algorithms can be defined optimally in VSAs, but there is always the risk of *cross-talk* – non-zero dot product values between different random vectors. Because the vectors are necessarily finite dimensional, the term ε_{ij} in eq. (7) will always be non-zero for some pairs of vector-symbols, although the magnitude is limited by the dimensionality of the vectors. Consequently, there is a drive to make vectors in VSAs to be as high dimensional as possible. Unfortunately, increases in the dimensionality of the vector come with an increase demand for resources to represent these vectors. Work is on-going to find more efficient representations. The Hexagonal SSP (Komer et al. 2019; Dumont and Eliasmith 2020), for example, often permits using lower-dimensional representations compared to purely random values. Additionally, work in using different neuron models can reduce the size of the population required to represent states (*e.g.*, Frady and Sommer 2019; Orchard and Jarvis 2023).

Conversely, the limiting of optimality due to finite representations suggests hypotheses about bounded rationality. Does the behaviour of a cognitive model converge to, and then away from, human behaviour as a function of the representation dimensionality? Choices of encoding schemes and representational resources can produce behaviour that deviates from optimal solutions, even while trying to solve problems formulated as optimization problems. Perhaps a shortcoming of VSAs, in terms of algorithmic performance, is a benefit, in terms of understanding bounded rationality.

Online Hyperparameter Estimation An outstanding problem that needs to be addressed is how to best fit model hyperparameters. Using the Glad-style conversion from quasi-probability to probability relies on a bias term that needs to be fit to a given dataset. Solving for this term exactly requires computing a non-linear integral over the entire domain, $\mathcal{X}$. If the conversion is to be learned sequentially then the bias must be updated as well. Similarily, the length scale parameter(s), h, is(are) dependent on the data and needs to be fit to the dataset. If this parameter is to be updated online, the encoding scheme and any learning rules will need to change accordingly. Conversely, if one views these parameters as being fixed after early development, one may attempt to explain errors in judgement due to improper fitting of hyperparameters.

Another consideration is that the bias parameter b, is an artefact of the particular choice of conversion from quasi-probability to probability. It may be that that conversion is not the most biologically plausible, or that a conversion is not necessary – the SPAUN model did not explicitly convert similarity values to probability, but it still was able to replicate observed data from the mammalian brain.

It is also worth considering the possibility that cognition may be generative, but not strictly generative in the space of probabilities defined by Kolmogorov's axioms. Other models of quasi-probability, like quantum probability, violate standard axioms of probability but are useful in describing physical systems. Perhaps cognition also does not respect formal definitions of probability, and conversions are unnecessary.

Future Research Directions

Automating Representation Design Good representations are fundamental to machine learning. To date, we work primarily with hand-engineered representations for projecting data into the VSA vector space. Feature engineering in the machine learning community was standard, until deep learning demonstrated that learned features – given sufficient data – can produce better results than hand-engineered features. In a similar vein, we may ask if there are learned representations, encoded as VSA expressions, that are superior to hand-engineered methods.

Higher-order representations are compositions of lower-order VSA objects. More importantly, they can be understood as algebraic statements about their compositional components. One can then frame the choice of encoding as a problem of finding either an optimal or a satisfactory representation for accomplishing particular tasks. One might consider using structural search techniques that permit probabilistic interpretations (*e.g.*, Duvenaud et al. 2013; Lake, Salakhutdinov, and Tenenbaum 2015) to find better VSA statements for encoding data. Learning VSA representations would remove the limitations of human-engineered features while preserving interpretability, as representations would still be algebraic statements in the VSA. To take this approach further into the neurally plausible domain, one should consider the possibility of Bayesian structural search (Kappel et al. 2015) as a mechanism for constructing encoding networks for complex representations.

Efficient Sampling One major area of investigation is on how to translate these probabilistic representations into actions – for cognitive models ultimately decisions have to be translated into motor plans. This requires translating the distribution into specific actions that an agent can take, this is the "generative" aspect of generative modeling. One could imagine computing the average action specified by any particular distribution, but this would not perform well in the case of bimodal distributions. An alternative would be to select actions by generating samples from the distribution.

A naive approach would be to sample a number of points in the action, $x_s \in \mathcal{X}$, encode them, compute the probabilities of the different actions and choose the maximum (with tie breaking for multimodal distributions) or perhaps perform Bayesian bootstrapping from these sample points. This is a valid sampling approach, but as the dimensionality of the action space increases, the memory requirements grow exponentially.

Another approach would be to use a Markov Chain Monte Carlo sampling approach, but here the VSA representation can have a wrinkle that is not present in standard generative approaches. Current generative models operate in locally smooth latent spaces where any vector is a potentially valid point in space, *i.e.*, small perturbations of a Variational Autoencoder (VAE)'s internal space or random vectors in a Generative Adversarial Network (GAN) can all be decoded to something potentially meaningful, even if of low probability or undesirable. This is not strictly the case for our approach: our representations are locally smooth, but they are not dense everywhere in $\mathbf{R}^d$. For VSA representations, and more specifically SSPs, valid points are only defined on a subset of the hypersphere. Thus perturbations of any given points can quickly move a representation off the manifold that defines valid points, resulting in something that is either (approximately) orthogonal, or is itself a weighted combination of VSA vectors, creating a new sampling problem. Hence, the use of Langevin dynamics, unconstrained by the manifold, may not result in meaningful samples. Cleanup memories may help in these kind of sampling processes, but they too require defining a codebook to cleanup against. This is a fundamental trade-off of using VSA-style generative models instead of more standard approaches – random sampling becomes more challenging, but we retain interpretability, in the sense that data can be analyzed structurally, using the VSA algebra, and networks can be interrogated more formally.

Other approaches, like normalizing flows, may result in more efficient sampling techniques, particularly flows on high-dimensional tori or spheres (*e.g.*, Falorsi et al. 2019; Rezende et al. 2020). Alternatively, one could elide sampling altogether. In the context of an operating organism, one could pose the problem of translating VSA-encoded distributions over actions into specific motor plans as an RL problem, learning to decode the actions that are most valuable depending on a state that is the distributions over actions. However, this is seems like an unusual formulation from a developmental perspective – learning to translate motor plans into actions occurs *after* cognition has been developed.

Choice of VSA There are a number of different VSAs to choose from, each with relevant strengths and weaknesses. In this work we exclusively employed the HRR algebra developed by Plate (1995). While this work would translate naturally to the Fourier Holographic Reduced Representation (FHRR) (*circular vectors* in Plate's original terminology (Plate 1995)), as they are linearly related through the Discrete Fourier Transform, it may not be the case that every VSA can be interpreted probabilistically. The choice of operators and basis vectors imply different models.

For example, binding with the outer product (Smolensky, Legendre, and Miyata 1992) or the vector-derived tensor binding of Gosmann and Eliasmith (2019) can support representation of integer data, but representing real-valued data is non-obvious. Furthermore, when binding with the outer product, the dimensionality of the representation grows as larger values are represented. However, algebras equipped with these binding operators can represent discrete distributions. While we do not engage in a full analysis here, further investigation into what kinds of representations of uncertainty are capable in other VSAs is warranted. Different VSAs may be desirable for different theorists, and their ability to model probability can impact how of cognitive architectures may relate to generative models.

Deeper Connections to Modern Generative Models One can possibly gain further benefit by merging this representation with an `IF-ELSE` structure, via the mechanism of heteroassociative memories. Production system-like formulations have been integrated with VSAs previously, proposing the Basal Ganglia as a model of rule-based action selection (Stewart and Eliasmith 2009; Stewart, Choo, and Eliasmith 2010).

Heteroassociative memories, like the modern Hopfield network (Krotov and Hopfield 2016), can take the form of a single hidden layer neural network with an input weight matrix, $W_{\text{in}} \in \mathbb{R}^{n \times d_{\text{in}}}$, and an output weight matrix, $W_{\text{out}} \in \mathbb{R}^{n \times d_{\text{out}}}$. It then transforms an input vector, $\mathbf{z}$, to an output vector, $\mathbf{y}$:

$$\mathbf{y} = W_{\text{out}}^T f(W_{\text{in}}\mathbf{z}), \tag{20}$$

where $f(\cdot)$ is the neuron activation function. If we permit $\mathbf{z} = \phi_{\mathcal{X}}(\mathbf{x})$ to be a VSA encoded vector, and similarly allow the rows of W_{in} and W_{out} to be VSA vectors sampling the input and output domains, then we can draw two important inferences. First, we can consider the rows of W_{in} as the condition for an `if` statement in a production system, and the rows of W_{out} can be the resultant actions, which due to the quasi-probabilistic nature of the chosen VSA encoding, can themselves be distributions over actions.

The second inference we can draw is that if the activation functions of the network's hidden layer perform a suitable conversion to probability, the output vector, $\mathbf{y}$ represents a distribution over outputs. The output vector $\mathbf{y}$ is a combination of distributions over actions, $\mathbf{a}$ in the action space, $\mathcal{A}$:

$$\phi_{\mathcal{A}}(\mathbf{a}) \cdot \mathbf{y} = \phi_{\mathcal{A}}(\mathbf{a}) \cdot \sum_i \phi_{\mathcal{A}}(\mathbf{a}_i) f(\phi_{\mathcal{X}}(\mathbf{x}_i) \cdot \mathbf{x}) \tag{21}$$

$$\approx \sum_i p(A_i = \mathbf{a} \mid X = \mathbf{x}_i) p(X = \mathbf{x}_i). \tag{22}$$

The utility of this formulation is that it provides a simple mechanism for integrating probabilistic rules. It also provides a distribution over actions that can be sampled from in a generative way.

Integrating VSA representations with associative memories provides us a generative approach to implementing rules systems, but the same network structure can be used to implement an autoassociative, clean-up memory, as previously implemented for VSA inputs by Stewart, Tang, and Eliasmith (2011). These are two components that are fundamental to many cognitive architectures, which we can now understand to be inherently probabilistic.

But the benefit goes one step further: modern Hopfield networks are intimately related to transformer networks (Ramsauer et al. 2020). Transformers have been highly influential in generative modeling of sequences and time series. That combining VSAs with associative memories produces a quasi-probabilistic network analogous to structures that support state-of-the-art generative models warrants further investigation.

Related Work

Considerable effort has been invested in developing frameworks where symbolic representations can be combined with probability distributions (see, *e.g.*, the Sigma cognitive architecture Rosenbloom, Demski, and Ustun 2016) or models of cognition that use probabilistic programming (*e.g.*, Goodman, Tenenbaum, and Contributors 2016). While it was not the original intent of these models, it remains notable that they lack an explanation for biologically plausible implementations of the proposed models.

Similar concerns hold for quantum approaches to probabilistic cognition (Pothos and Busemeyer 2013; Busemeyer, Wang, and Shiffrin 2015; Pothos and Busemeyer 2022). However, quantum probability is also a quasi-probability model, in that it does not adhere to all of Kolmogorov's axioms of probability. Like the methods discussed above, it relies on representing data as points in high- or infinite-dimensional Hilbert spaces, using a set of operators on these vectors to effect models of cognition, and relying on a transformation (Born's rule) to turn quasi-probabilities into probabilities. It has been previously suggested that neural VSA models could support the implementation of quantum probability models (Stewart and Eliasmith 2013). An alternative approach using neural oscillators was explored by Busemeyer, Fakhari, and Kvam (2017), which is reminiscent of the FHRR approach to VSA modelling, although that technique is not explicitly considered. Regardless, the utility of quantum probability in general does raise the question of whether or not strict Kolmogorov probabilities are the best framework for modelling cognition, or perhaps some other method would be best suited.

More in the machine learning tradition, *Kernel Probabilistic Programming* (KPP), surveyed in (Muandet et al. 2017), provides a framework for operating on distributions represented as Kernel Mean Embeddings, equivalent to our eq. (4). This approach assumes the existence of vector embeddings of continuous data, instead of constructing it from iterated, fractional binding, and relies on the outer product to represent variable binding, similar to Smolensky, Legendre, and Miyata (1992) and quantum probability. As discussed above, the choice of circular convolution as a binding function allows vector operations to preserve dimensionality. Furthermore, with circular convolution, embedded values can be updated post-encoding. This fact enabled the implementation of a VSA algorithm for simultaneous localization and mapping (SLAM) using HRRs and spiking neurons (Dumont, Orchard, and Eliasmith 2022; Dumont et al. 2023). These convergent lines of research show that there is utility in pursuing operations on Hilbert space representations as a mechanism for encoding probabilistic computation.

Conclusion

VSAs are a tool that can be used to represent symbolic reasoning while integrating with neural networks, and being inherently probabilistic. These representations have successfully provided the functional description of the SPAUN cognitive architecture. As described above, memories of VSAs representations inherently learn distributions over input, suggesting a connection between these formulations and generative models. The centrality of memory to cognitive architectures suggests that VSA implementations may be a pathway to unifying generative models with cognitive architectures.

Acknowledgements

This work was funded in part by CFI and OIT infrastructure funding, Canada Research Chairs program, NSERC Discovery grant 261453, AFOSR grant FA9550-17-1-0026. P.M. Furlong was funded in part by an Intel Neuromorphic Research Community Grant.

References

Anderson, J. R.; John, B. E.; Just, M. A.; Carpenter, P. A.; Kieras, D. E.; and Meyer, D. 1995. Production system models of complex cognition. In *Proceedings of the seventeenth annual conference of the cognitive science society*, 9–12.

Bartlett, M.; Simone, K.; Dumont, N. S.-Y.; Furlong, P. M.; Eliasmith, C.; Orchard, J.; and Stewart, T. C. 2023. Improving Reinforcement Learning with Biologically Motivated Continuous State Representations. In *21st International Conference on Cognitive Modelling (in press)*.

Bishop, C. M. 2006. *Pattern recognition and machine learning*. springer.

Born, M. 1926. Quantenmechanik der stoßvorgänge. *Zeitschrift für Physik*, 38(11): 803–827.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Busemeyer, J. R.; Fakhari, P.; and Kvam, P. 2017. Neural implementation of operations used in quantum cognition. *Progress in biophysics and molecular biology*, 130: 53–60.

Busemeyer, J. R.; Wang, Z.; and Shiffrin, R. M. 2015. Bayesian model comparison favors quantum over standard decision theory account of dynamic inconsistency. *Decision*, 2(1): 1.

Choo, X. 2018. *Spaun 2.0: Extending the World's Largest Functional Brain Model*. PhD thesis, University of Waterloo.

Davis, K. B. 1977. Mean integrated square error properties of density estimates. *The Annals of Statistics*, 530–535.

Dumont, N. S.-Y.; and Eliasmith, C. 2020. Accurate representation for spatial cognition using grid cells. In *42nd Annual Meeting of the Cognitive Science Society. Toronto, ON: Cognitive Science Society*, 2367–2373.

Dumont, N. S.-Y.; Furlong, P. M.; Orchard, J.; and Eliasmith, C. 2023. Exploiting semantic information in a spiking neural SLAM system. *Frontiers in Neuroscience*, 17.

Dumont, N. S.-Y.; Orchard, J.; and Eliasmith, C. 2022. A model of path integration that connects neural and symbolic representation. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Duvenaud, D.; Lloyd, J.; Grosse, R.; Tenenbaum, J.; and Zoubin, G. 2013. Structure discovery in nonparametric regression through compositional kernel search. In *International Conference on Machine Learning*, 1166–1174. PMLR.

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

Eliasmith, C.; Stewart, T. C.; Choo, X.; Bekolay, T.; DeWolf, T.; Tang, Y.; and Rasmussen, D. 2012. A large-scale model of the functioning brain. *science*, 338(6111): 1202–1205.

Eliasmith, C.; and Thagard, P. 2001. Integrating structure and meaning: a distributed model of analogical mapping. *Cognitive Science*, 25: 245–286.

Falorsi, L.; de Haan, P.; Davidson, T. R.; and Forré, P. 2019. Reparameterizing distributions on lie groups. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 3244–3253. PMLR.

Frady, E. P.; Kleyko, D.; Kymn, C. J.; Olshausen, B. A.; and Sommer, F. T. 2021. Computing on functions using randomized vector representations. *arXiv preprint arXiv:2109.03429*.

Frady, E. P.; and Sommer, F. T. 2019. Robust computation with rhythmic spike patterns. *Proceedings of the National Academy of Sciences*, 116(36): 18050–18059.

Furlong, M.; and Eliasmith, C. 2022. Fractional binding in vector symbolic architectures as quasi-probability statements. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Furlong, P. M.; Stewart, T. C.; and Eliasmith, C. 2022. Fractional Binding in Vector Symbolic Representations for Efficient Mutual Information Exploration. In *ICRA Workshop: Towards Curious Robots: Modern Approaches for Intrinsically-Motivated Intelligent Behavior. 1ś5*.

Glad, I. K.; Hjort, N. L.; and Ushakov, N. 2007. Density estimation using the sinc kernel. *Preprint Statistics*, 2: 2007.

Glad, I. K.; Hjort, N. L.; and Ushakov, N. G. 2003. Correction of density estimators that are not densities. *Scandinavian Journal of Statistics*, 30(2): 415–427.

Goodman, N. D.; Tenenbaum, J. B.; and Contributors, T. P. 2016. Probabilistic Models of Cognition. http://probmods.org/v2. Accessed: 2023-1-23.

Gosmann, J.; and Eliasmith, C. 2016. Optimizing semantic pointer representations for symbol-like processing in spiking neural networks. *PloS One*, 11(2): e0149928.

Gosmann, J.; and Eliasmith, C. 2019. Vector-derived transformation binding: An improved binding operation for deep symbol-like processing in neural networks. *Neural computation*, 31(5): 849–869.

Joshi, A.; Halseth, J. T.; and Kanerva, P. 2017. Language geometry using random indexing. In *Quantum Interaction: 10th International Conference, QI 2016, San Francisco, CA, USA, July 20-22, 2016, Revised Selected Papers 10*, 265–274. Springer.

Kanerva, P. 1988. *Sparse distributed memory*. MIT press.

Kanerva, P. 2009. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive computation*, 1(2): 139–159.

Kappel, D.; Habenschuss, S.; Legenstein, R.; and Maass, W. 2015. Network plasticity as Bayesian inference. *PLoS computational biology*, 11(11): e1004485.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213.

Komer, B. 2020. *Biologically Inspired Spatial Representation*. Ph.D. thesis, University of Waterloo.

Komer, B.; Stewart, T. C.; Voelker, A.; and Eliasmith, C. 2019. A neural representation of continuous space using fractional binding. In *CogSci*, 2038–2043.

Krotov, D.; and Hopfield, J. J. 2016. Dense associative memory for pattern recognition. *Advances in neural information processing systems*, 29.

Laird, J. E.; Lebiere, C.; and 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.

Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266): 1332–1338.

Mittal, G.; Engel, J.; Hawthorne, C.; and Simon, I. 2021. Symbolic music generation with diffusion models. *arXiv preprint arXiv:2103.16091*.

Muandet, K.; Fukumizu, K.; Sriperumbudur, B.; Schölkopf, B.; et al. 2017. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2): 1–141.

Orchard, J.; and Jarvis, R. 2023. Hyperdimensional Computing with Spiking-Phasor Neurons. arXiv:2303.00066.

Plate, T. A. 1992. Holographic recurrent networks. *Advances in neural information processing systems*, 5.

Plate, T. A. 1993. Estimating analogical similarity by dot-products of holographic reduced representations. *Advances in neural information processing systems*, 6.

Plate, T. A. 1995. Holographic reduced representations. *IEEE Transactions on Neural networks*, 6(3): 623–641.

Pothos, E. M.; and Busemeyer, J. R. 2013. Can quantum probability provide a new direction for cognitive modeling? *Behavioral and brain sciences*, 36(3): 255–274.

Pothos, E. M.; and Busemeyer, J. R. 2022. Quantum cognition. *Annual review of psychology*, 73: 749–778.

Rahimi, A.; Recht, B.; et al. 2007. Random Features for Large-Scale Kernel Machines. *NIPS*, 3(4): 5.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, 8821–8831. PMLR.

Ramsauer, H.; Schäfl, B.; Lehner, J.; Seidl, P.; Widrich, M.; Adler, T.; Gruber, L.; Holzleitner, M.; Pavlović, M.; Sandve, G. K.; et al. 2020. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*.

Rezende, D. J.; Papamakarios, G.; Racaniere, S.; Albergo, M.; Kanwar, G.; Shanahan, P.; and Cranmer, K. 2020. Normalizing flows on tori and spheres. In *International Conference on Machine Learning*, 8083–8092. PMLR.

Rosenbloom, P. S.; Demski, A.; and Ustun, V. 2016. The Sigma cognitive architecture and system: Towards functionally elegant grand unification. *Journal of Artificial General Intelligence*, 7(1): 1.

Schölkopf, B.; Platt, J. C.; Shawe-Taylor, J.; Smola, A. J.; and Williamson, R. C. 2001. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7): 1443–1471.

Smolensky, P.; Legendre, G.; and Miyata, Y. 1992. Integrating Connectionist and Symbolic Computation for the Theory of Language. Technical Report CU-CS-628-92, University of Colorado, Boulder.

Smolensky, P.; McCoy, R.; Fernandez, R.; Goldrick, M.; and Gao, J. 2022. Neurocompositional computing: From the Central Paradox of Cognition to a new generation of AI systems. *AI Magazine*, 43(3): 308–322.

Stewart, T.; Choo, X.; and Eliasmith, C. 2010. Symbolic reasoning in spiking neurons: A model of the cortex/basal ganglia/thalamus loop. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 32.

Stewart, T. C.; and Eliasmith, C. 2009. Spiking neurons and central executive control: The origin of the 50-millisecond cognitive cycle. In *9th International Conference on Cognitive Modelling*, volume 122, 130–131. Citeseer.

Stewart, T. C.; and Eliasmith, C. 2013. Realistic neurons can compute the operations needed by quantum probability theory and other vector symbolic architectures. *Behavioral & Brain Sciences*, 36(3).

Stewart, T. C.; Tang, Y.; and Eliasmith, C. 2011. A biologically realistic cleanup memory: Autoassociation in spiking neurons. *Cognitive Systems Research*, 12(2): 84–92.

Tsybakov, A. B. 2009. *Introduction to nonparametric estimation*. Springer.

Voelker, A. R. 2020. A short letter on the dot product between rotated Fourier transforms. *arXiv preprint arXiv:2007.13462*.

Voelker, A. R.; Blouw, P.; Choo, X.; Dumont, N. S.-Y.; Stewart, T. C.; and Eliasmith, C. 2021. Simulating and predicting dynamical systems with spatial semantic pointers. *Neural Computation*, 33(8): 2033–2067.

Voelker, A. R.; Gosmann, J.; and Stewart, T. C. 2017. Efficiently sampling vectors and coordinates from the n-sphere and n-ball. *Centre for Theoretical Neuroscience-Technical Report*, 1.