A Survey on Hyperdimensional Computing aka Vector Symbolic Architectures, Part II: Applications, Cognitive Models, and Challenges

DENIS KLEYKO, University of California at Berkeley, USA and Research Institutes of Sweden, Sweden
DMITRI RACHKOVSKIJ, International Research and Training Center for Information Technologies, Ukraine and Luleå University of Technology, Sweden
EVGENY OSIPOV, Luleå University of Technology, Sweden
ABBAS RAHIMI, IBM Research–Zurich, Switzerland

This is Part II of the two-part comprehensive survey devoted to a computing framework most commonly known under the names Hyperdimensional Computing and Vector Symbolic Architectures (HDC/VSA). Both names refer to a family of computational models that use high-dimensional distributed representations and rely on the algebraic properties of their key operations to incorporate the advantages of structured symbolic representations and vector distributed representations. Holographic Reduced Representations [321, 326] is an influential HDC/VSA model that is well known in the machine learning domain and often used to refer to the whole family. However, for the sake of consistency, we use HDC/VSA to refer to the field.

Part I of this survey [222] covered foundational aspects of the field, such as the historical context leading to the development of HDC/VSA, key elements of any HDC/VSA model, known HDC/VSA models, and the transformation of input data of various types into high-dimensional vectors suitable for HDC/VSA. This second part surveys existing applications, the role of HDC/VSA in cognitive computing and architectures, as well as directions for future work. Most of the applications lie within the Machine Learning/Artificial Intelligence domain; however, we also cover other applications to provide a complete picture. The survey is written to be useful for both newcomers and practitioners.

CCS Concepts: • **Theory of computation → Random projections and metric embeddings**; *Data structures design and analysis*; **Bloom filters and hashing**; • **Computing methodologies → Artificial**

It is worth noting that some of the literature referenced in this survey might be difficult to find; however, most of the literature can be traced using the publication list available online at https://www.hd-computing.com/publications.

Denis Kleyko and Dmitri Rachkovskij contributed equally to this work.

The work of DK was supported by the European Union's Horizon 2020 Programme under the Marie Skłodowska-Curie Individual Fellowship Grant (839179). The work of DK was also supported in part by AFOSR FA9550-19-1-0241 and Intel's THWAI program. The work of DAR was supported in part by the National Academy of Sciences of Ukraine (grant no. 0120U000122, 0121U000016, 0122U002151, and 0117U002286), the Ministry of Education and Science of Ukraine (grant no. 0121U000228 and 0122U000818), and the Swedish Foundation for Strategic Research (SSF, grant no. UKR22-0024).

Authors' addresses: D. Kleyko, University of California at Berkeley, Berkeley, CA, USA, 94720, and Research Institutes of Sweden, Kista, Sweden, 16440; email: denkle@berkeley.edu; D. Rachkovskij, International Research and Training Center for Information Technologies, Kiev, Ukraine, 03680, and Luleå University of Technology, Luleå, Sweden, 97187; email: dar@infrm.kiev.ua; E. Osipov, Luleå University of Technology, Luleå, Sweden, 97187; email: evgeny.osipov@ltu.se; A. Rahimi, IBM Research–Zurich, Zurich, Switzerland, 8803; email: abr@zurich.ibm.com.

0360-0300/2023/01-ART175 $15.00
https://doi.org/10.1145/3558000

intelligence; Natural language processing; Cognitive science; Computer vision tasks; *Supervised learning by classification*; *Neural networks*; *Knowledge representation and reasoning;*

Additional Key Words and Phrases: Artificial intelligence, machine learning, distributed representations, cognitive architectures, cognitive computing, applications, analogical reasoning, hyperdimensional computing, vector symbolic architectures, holographic reduced representations, tensor product representations, matrix binding of additive terms, binary spatter codes, multiply-add-permute, sparse binary distributed representations, sparse block codes, modular composite representations, geometric analogue of holographic reduced representations

ACM Reference format:
Denis Kleyko, Dmitri Rachkovskij, Evgeny Osipov, and Abbas Rahimi. 2023. A Survey on Hyperdimensional Computing aka Vector Symbolic Architectures, Part II: Applications, Cognitive Models, and Challenges. *ACM Comput. Surv.* 55, 9, Article 175 (January 2023), 52 pages.
https://doi.org/10.1145/3558000

1 INTRODUCTION

This article is Part II of the survey of a research field known under the names **Hyperdimensional Computing (HDC)** (the term was introduced in [179]) and **Vector Symbolic Architectures (VSA)** (the term was introduced in [108]). As in Part I [222], below we will consistently use the joint name HDC/VSA when referring to the field. HDC/VSA is an umbrella term for a family of computational models that rely on mathematical properties of high-dimensional random spaces and use high-dimensional distributed representations called **hypervectors (HVs)** for a structured ("symbolic") representation of data, while maintaining the advantages of traditional connectionist vector distributed representations.

First, let us briefly recapitulate the motivation for this survey. The main driving force behind the current interest in HDC/VSA is the global trend of searching for computing paradigms alternative to the conventional (von Neumann) ones. Examples of the new paradigms are neuromorphic and nanoscalable computing, where HDC/VSA is expected to play an important role (see [204] and references therein for perspective). Due to this surge of interest in HDC/VSA, the need for providing a broad overview of the field, which is currently missing, became evident. Therefore, this two-part survey extensively covers the state-of-the-art of the field in a form that is accessible to a wider audience.

There were no previous attempts to make a comprehensive survey of HDC/VSA, but there are articles that overview particular topics of HDC/VSA. Probably the first attempt to overview and unify different HDC/VSA models should be attributed to Plate [322]. The key idea for the unification was to consider the existing (at that time, four) HDC/VSA models as different schemes for implementing two key operations: binding and superposition (see Section 2.2.3 in [222]). However, since that time numerous HDC/VSA models have come to prominence. A more recent summary of the most frequently used models was provided in [343]. In [374], the HDC/VSA models were compared in terms of their realizations of the binding operation. Both articles, however, missed some of the models. These and other gaps have been filled in Part I of this survey [222].

As for applications of HDC/VSA—the topic covered in this article—in Table 1 we identified the following substantial application domains, which reflect the structure of Sections 2 and 3: deterministic behavior, similarity estimation, classification, cognitive computing, and cognitive architectures. The columns in Table 1 list more fine-grained application clusters within these larger domains.

There is no previous article that would account for all currently known applications, though there are recent works overviewing either a particular application area (as in [344], where the focus was on biomedical signals), or certain application types (as in [110, 131], where solving

Table 1. A Qualitative Assessment of Existing HDC/VSA Literature That Have Some Elements of Survey

	Deterministic behavior			Similarity estimation			Classification			Cognitive computing				Cognitive architectures				
	Automata & instructions	Data transmission	String processing	Context vectors	Biomedical signals	Images	Feature vectors	Images or their properties	Structured data	Holistic transformations	Analogical reasoning	Cognitive modeling	Computer vision and scene analysis	Semantic Pointer Architecture Unified Network	Associative-Projective Neural Networks	Hierarchical Temporal Memory	Learning Intelligent Distribution Agent	Memories for cognitive architectures
[322]	✗	✗	✗	✗	✗	✗	✗	✗	✗	±	±	✗	✗	✗	✗	✗	✗	✗
[179]	✗	✗	✗	±	✗	✗	✗	✗	✗	±	±	✗	✗	✗	✗	✗	✗	✗
[343]	✗	✗	✗	✗	✗	✗	±	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[344]	✗	✗	✗	✗	✗	✗	±	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[298]	✗	✗	✗	✗	✗	±	±	±	✗	✗	✗	✗	±	✗	✗	✗	✗	✗
[110]	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[374]	✗	✗	✗	✗	✗	±	✗	±	✗	✗	✗	✗	±	✗	✗	✗	✗	✗
[204]	±	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[131]	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	±	✗	✗	✗	✗	✗
This survey, Part II Section #	2.1.1	2.1.2	2.1.3	2.2.1	2.2.2	2.2.3	2.3.1	2.3.2	2.3.3	3.1.1	3.1.2	3.1.3	3.1.4	3.2.1	3.2.2	3.2.3	3.2.4	3.2.5

✓ means that the article overviewed the area rather comprehensively, ✗ means that the area was not covered at all while ± indicates that the article partially addressed a particular topic, but either new results were reported since then or not all related work was covered.

classification tasks with HDC/VSA was the main theme). The topic of machine learning is also omnipresent in this survey, and due to its ubiquity we dedicated Section 2.3 to classification tasks. However, the scope of the survey is much broader, as it touches on all currently known applications. Table 1 contrasts the coverage of Part II of this survey with the previous articles (ordered chronologically). We use ± to indicate that an article partially addressed a particular topic, but either new results were reported since then or not all related work was covered.

In Part I of this survey [222], we considered the motivation behind HDC/VSA and basic notions, summarized currently known HDC/VSA models, and presented the transformation of various data types into HVs. Part II of this survey covers existing applications (Section 2) and the use of HDC/VSA in cognitive modeling and architectures (Section 3). The discussion and challenges, as well as conclusions, are presented in Sections 4 and 5, respectively.

2 APPLICATION AREAS

HDC/VSA have been applied across different fields for various tasks. For this section, we aggregated the existing applications into several groups: deterministic behavior (Section 2.1), similarity estimation (Section 2.2), and classification (Section 2.3).

2.1 Deterministic Behavior with HDC/VSA

In this section, we consider several use-cases of HVs designed to produce some kind of deterministic behavior. Note that due to the capacity limitations of HVs (see Section 2.4 in [222]), achieving deterministic behavior depends on several design choices. These include the dimensionality of HVs as well as, e.g., the number of atomic HVs and the kind of rules used for constructing compositional HVs, such as the number of arguments in the superposition operation. Note also that, strictly speaking, not all application areas listed here are perfectly deterministic (in particular, communications in Section 2.1.2), but the determinism is a desirable property in all the areas collected in this section.

2.1.1 Automata, Instructions, and Schemas.

Finite-state automata and grammars. A deterministic finite-state automaton is specified by defining a finite set of states, a finite set of allowed input symbols, a transition function (defines all

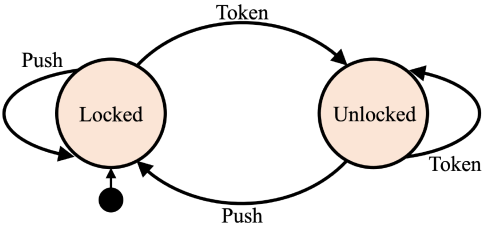

Fig. 1. An example of a state diagram of a finite-state automaton modeling the control logic of a turnstile.

transitions in the automaton), a start state, and a finite set of accepting states. The current state can change in response to an input. The joint current state and input symbol uniquely determine the next state of the automaton.

An intuitive example of an automaton controlling the logic of a turnstile is presented in Figure 1. The set of input symbols is { "Token", "Push" } and the set of states is { "Unlocked", "Locked" }. The state diagram in Figure 1 can be used to derive the transition function.

HDC/VSA-based implementations of finite-state automata were proposed in [307, 425]. Random HVs are assigned to represent states (**u** for "Unlocked"; **l** for "Locked") and input symbols (**t** for "Token"; **p** for "Push"). These HVs are used to form a compositional HV **a** for the transition function. The transformation is similar to the one used for the directed graphs in Section 3.5.1 in [222]. However, the HV representing the input symbol for the automaton is bound to the edge HV that corresponds to the binding of the HVs for the current and the next state. For instance, going from "Locked" to "Unlocked" upon receiving "Token" is represented as

$$\mathbf{t} \circ \mathbf{l} \circ \rho(\mathbf{u}). \tag{1}$$

Given the HVs of all transitions, the transition function **a** is represented as their superposition:

$$\mathbf{a} = \mathbf{p} \circ \mathbf{l} \circ \rho(\mathbf{l}) + \mathbf{t} \circ \mathbf{l} \circ \rho(\mathbf{u}) + \mathbf{p} \circ \mathbf{u} \circ \rho(\mathbf{l}) + \mathbf{t} \circ \mathbf{u} \circ \rho(\mathbf{g}). \tag{2}$$

The next state is obtained by querying **a** with the binding of HVs of the current state and of the input symbol, followed by the inverse permutation of the resultant HV that returns the noisy version of the next state's HV.[1] For example, if the current state is **l** and **p** is received, then

$$\rho^{-1}(\mathbf{a} \circ \mathbf{p} \circ \mathbf{l}) = \mathbf{l} + \text{noise}. \tag{3}$$

This noisy HV is used as the query for the item memory to obtain the noiseless atomic HV **l**.

Transformations of pushdown automata and context-free grammars into HVs have been presented in [12]. A proposal for implementing Turing machines and cellular automata was given in [204].

In [226], the **Holographic Reduced Representations (HRR)** model was used to represent Fluid Construction Grammars, which is a formalism that allows designing construction grammars and using them for language parsing and production. Another work related to parsing is [392], which presented an HDC/VSA implementation of a general-purpose left-corner parsing with simple grammars. An alternative approach to parsing with HVs using a constraint-based parser has been presented in [21].

[1] It is worth recalling that this and other applications use a common design pattern, relying on the unbinding operation (see Section 2.2.3 in [222]) that allows recovering one of the arguments. In the case of permutations, it is due to the fact that $\mathbf{a} = \rho^{-i}(\rho^{i}(\mathbf{a}))$, while in the case of multiplicative binding $\mathbf{a} = \mathbf{b} \oslash (\mathbf{a} \circ \mathbf{b})$, where for the implementations with self-inverse binding $\oslash = \circ$.

A related research direction was initiated by beim Graben and colleagues [11–13, 421]. It concerns establishing a mathematically rigorous formalism of Tensor Product Representations using the concept of a Fock space [83]. The studies focus largely on use-cases in computational linguistics, semantic processing, and quantum logic. The usage of the Fock space formalism for formulating minimalist grammars was presented in [11]. Syntactic language processing as part of phenomenological modeling was reported in [13]. An accessible and practical entry point to the area can be found in [421].

Controllers, instructions, schemas. In [249], using the **Multiply-Add-Permute** model **(MAP)**, it was demonstrated how to manually construct a compositional HV that implements a simple behavior strategy of a robot. Sensor inputs as well as actions were represented as atomic HVs. Combinations of sensor inputs as well as combinations of actions were represented as a superposition of the corresponding atomic HVs. HVs of particular sensor input combinations were bound to the corresponding HV of action combinations. The bound HVs of all possible sensor-action rules were superimposed to produce the compositional HV of the robot controller. Unbinding this HV with the current HV of sensor input combinations results in the noisy HV of the proper action combination. This idea was extended further in [297] through a proposed algorithm to "learn" a compositional HV representing a robot's controller using the sensor-actuator values obtained from successful navigation runs. This mode of robot operation is known as "learning by demonstration." It was realized as the superposition of the current "controller" HV with the HV corresponding to the binding of the sensor-actuator values—in case the current sensor HV was dissimilar to the ones already present in the "controller" HV. Another work studying robot navigation is [269], which investigated a number of ways to form compositional HVs representing sensory data and explored the integration of the resultant HVs with neural network instead of the "controller."

In [39], using HRR, instructions were represented as a sequence of rules and rules as a sequence of their antecedent and consequent elements. Multiplicative bindings with position HVs were used to represent the sequence. Antecedents and consequents, in turn, were represented as HVs of their elements using binding and superposition operations. This approach was used as a part of instruction parsing in a cognitive architecture [394]. In [245], a proposal was sketched for a HDC/VSA-based processor, where both data and instructions were represented as HVs.

In [295], HVs for the representation of "schemas" in the form <*context*, *action*, *result*> were used. A more general approach for modeling the behavior of intelligent agents as "functional acts" was considered in [337] for **Sparse Binary Distributed Representations (SBDR)** (see also Section 3.2.2). It is based on HVs representing triples <*current situation*, *action*, *resulting situation*> (which essentially correspond to "schemas"), with the associated evaluations and costs. Finally, it is worth recalling that, in general, data structures to be represented by HVs do not have to be limited to "schemas." For example, a recent proposal in [104] suggested that HVs are well suited for forming representations of the JSON format that can include several levels of hierarchy.

Membership query and frequency estimation. Section 3.1.2 in [222] presented the transformation of sets and multisets into HVs. When implemented with the SBDR model, it becomes evident that Bloom Filters [19] are a special case of HDC/VSA [223], which have been used in a myriad of applications involving the membership query. It is beyond the scope of this survey to overview them all; therefore, the interested readers are refereed to a survey in [400]). When implementing the transformation of multisets via the Sparse Block Codes model, a similar connection can be made to count-min sketch [50] that is used commonly for estimating frequency distributions in data streams (some applications are presented in [50]). The use of the HDC/VSA principles for constructing hash tables has been recently considered in [135].

2.1.2 Transmission of Data Structures.

Communications. The main motivation for using HDC/VSA in the communication context is their robustness to noise due to the distributed nature of HVs. Let us consider three similar but varying applications of HDC/VSA.

In [165], it was shown how to use **Binary Spatter Codes (BSC)** (Section 2.3.6 in [222]) in collective communication for sensing purposes. Multiple devices wirelessly sent their specific HVs representing some of their sensory data (the paper used temperature as a show-case). It was proposed to receive them in a manner that implements the superposition operation. This superposition HV was then analyzed by calculating the dist_{Man} of the normalized superposition and the atomic HVs and comparing with a threshold. For instance, the case that a particular temperature was transmitted could be detected. Another version of analysis allowed checking how many devices had been exposed to a particular temperature. The proposed communication scheme does not require a control mechanism for getting multiple access to the medium. So it can be useful in scenarios where there are multiple devices that have to report their states to some central node. Recently, it has been shown how such over-the-air superposition can be used for on-chip communications to scale up the architecture with multiple transmitters and receivers [126]. This has been done by carefully engineering modulation constellations, and it paves the way for a large number of *physically distributed* associative memories (as wireless-augmented receivers) to reliably perform the similarity search given a slightly different version of a query HV as their input.

In [211], BSC was used in the context of medium access control protocols for wireless sensor networks. A device was forming a compositional HV representing the device's sensory data (by the superposition of multiplicative bindings), which was then transmitted to the communication medium. It was assumed that the receiver knew the atomic HVs and, thus, could recover the information represented in the received compositional HV. The application scope of this approach is for scenarios where the communication medium is very harsh so that a high redundancy of HVs is useful for reliably transmitting the data.

In [199], it was proposed to combine forward error correction and modulation using Fourier HRR (Section 2.3.5 in [222]). The scheme represented individual pieces of data by using complex-valued HVs that were then combined into a compositional HV using permutation and superposition operations. Unnormalized complex values of the compositional HV were transmitted to the communication medium. The iterative decoding of the received compositional HV significantly increased the code rate. The application scope of this scheme is robust communication in a low signal-to-noise ratio regime. The scheme at a lower coding rate was compared to the low-density parity check and polar codes in terms of achieved bit error rates, while featuring lower decoding complexity. To improve its signal-to-noise ratio gain, a soft-feedback iterative decoding was proposed [140] to additionally take the estimation's confidence into account. That improved the signal-to-noise ratio gain by 0.2 dB at a bit-error-rate of 10^{-4}. In further works, the scheme has been applied to collision-tolerant narrowband communications [150], massive machine type communications [151], and near-channel classification [140].

Distributed orchestration. Another use-case of HDC/VSA in the context of transmission of data structures is distributed orchestration. The key idea presented in [382, 383] was to use BSC to communicate a workflow in a decentralized manner between the devices involved in the application described by the workflow. Workflows were represented and communicated as compositional HVs constructed using the primitives for representing sequences (Section 3.3 in [222]) and directed acyclic graphs (Section 3.5.2 in [222]). In [384], the approach was implemented in a particular workflow system: Node-RED. In [8], the approach was extended further to take into account the level of trust associated with various elements when selecting services.

2.1.3 String Processing. In [329], it was proposed to obtain HVs of a word using permutations (cyclic shifts) of HVs of its letters to associate a letter with its position in the word. The conjunction was used to bind together all the obtained letter-in-position HVs. HVs of words formed in such a way were then used to obtain *n*-grams of sequences of words using the same procedure. The obtained HVs were used to estimate the frequencies of various word sequences in texts to create a model of human text reader interests. Note that here the conjunction result is somewhat similar to all input HVs.

An interesting property of sequence representation using permutations of its element HVs (Section 3.3 in [222]) is that the HV of a shifted sequence can be obtained by the permutation of the sequence HV as a whole [206, 213, 284]. This property was leveraged in [213] for searching the best alignment (shift) of two sequences, i.e., the alignment that provides the maximum number of coinciding symbols. This can be used, e.g., for identifying common substrings. Such a representation, however, does not preserve the similarity of symbols' HVs in nearby positions, which would be useful for, e.g., spell checking. This can be addressed by, e.g., extending the permutation-based representations as in [215], where the resultant compositional HVs were evaluated on a permuted text, which was successfully reconstructed. An approach to transforming sequences into sparse HVs from [333], which preserves the similarity of symbols at nearby positions and is shift-equivariant, was applied to the spellchecking task.

An algorithm for searching a query string in the base string was proposed in [315] and modified in [204]. It is based on the idea of representing finite-state automata in an HV (see Section 2.1.1). The algorithm represented the base string as a non-deterministic finite-state automaton [328]. The symbols of the base string corresponded to the transitions between the states of the automaton. The automaton in turn was represented as a compositional HV. The automaton was initialized as a superposition of all atomic HVs corresponding to the states. The query substring was presented to the automaton symbol by symbol. If after the presentation of the whole substring the automaton appeared in one of the valid states, then this indicated the presence of the query substring in the base string.

In [200], the MAP model was used for DNA string matching. The key idea was as in [173]: the base DNA string was represented by one or several HVs containing a superposition of all *n*-grams of predefined size(s). HVs of *n*-grams were formed by multiplicative binding of appropriately permuted HVs of their symbols (Section 3.3.4 in [222]). A query string was considered present in the base DNA string if the similarity between its HV and the compositional HV(s) of the base DNA string was higher than the predefined threshold. The threshold value determined the balance between true and false positives, similarly to Bloom filters (see Section 3.1.2 in [222]). The approach was evaluated on two databases of DNA strings: escherichia coli and human chromosome 14. The main promise of the approach in [200] is the possibility to accelerate string matching with application-specific integrated circuits due to the simplicity and parallelizability of the HDC/VSA operations.

2.1.4 Factorization. The resonator networks [88, 197] were proposed as a way to solve factorization problems, which often emerge during the recovery procedure within HDC/VSA (Section 2.2.5 in [222]). It is expected that the resonator networks will be useful for solving various factorization problems, but this requires formulating the problem for the HDC/VSA domain. An initial attempt to decompose synthetic scenes was demonstrated in [87]. A more recent formulation for the integer factorization was presented in [203]. The transformation of numbers into HVs was based on the fractional power encoding [89, 90] (Section 3.2.1 in [222]) combined with log transformation, and the resonator network was used to solve the integer factorization problem. The approach was evaluated on the factorization of semiprimes.

2.2 Similarity Estimation with HVs

Transformations of original data into HVs allow constructing HVs in various application areas in a manner that preserves similarity relevant for a particular application. This provides a tool to use HDC/VSA for "similarity-based reasoning" that includes, e.g., similarity search and classification in its simplest form as well as the much more advanced analogical reasoning considered in Section 3.1.2. Due to the abundance of studies on classification tasks we devote another separate Section 2.3 to it. In this section, we primarily focus on "context" HVs, since for a long time these were the most influential application of HDC/VSA. We also present some existing efforts in similarity search.

2.2.1 Word Embeddings: Context HVs for Words and Texts. The key idea for constructing context vectors is usually referred to as the "distributional semantics hypothesis" [130], suggesting that linguistic items with similar distributions have similar meanings. The distributions are calculated as frequencies of item occurrence in particular contexts by using a document corpus. For items-words the contexts could be, e.g., documents, paragraphs, sentences, or sequences of words close to a focus word. For example, the generalized vector space model [423] used documents as contexts for words in information retrieval. In computational linguistics and machine learning, vectors are also known as embeddings.

In principle, context vectors can be obtained in any domain where objects and contexts could be defined. Below, we will only focus on the context vector methods that are commonly attributed to HDC/VSA. They usually transform the frequency distributions into HVs of particular formats, which we will call context HVs.

Historically, the first proposal to form context HVs for words was that of Gallant, e.g., [29, 102, 103, 105]. These studies, however, did not become known widely, but see [134]. In fact, two most influential HDC/VSA-based methods for context HVs are **Random Indexing (RI)** [181, 366] and **Bound Encoding of the Aggregate Language Environment (BEAGLE)** [172].

Random Indexing. The RI method [181, 366] was originally proposed in [181] as a simple alternative to **Latent Semantic Analysis (LSA)** [246]. Instead of the expensive **Singular Value Decomposition (SVD)** used in LSA for the dimensionality reduction of a word-document matrix, RI uses the multiplication by a random matrix, thereby performing random projection (Section 3.2.3 in [222]). The **Random Projection (RP)** matrix was ternary ($\{-1, 0, 1\}$) and sparse. Each row of the RP matrix is seen as a "random index" (hence RI) assigned to each context (in that case, the document). In implementation, the frequency matrix was not formed explicitly, and the resultant context HVs were formed by scanning the document corpus and adding the document's random index vector to the context HV of each word in the document. The sign function can be used to obtain binary context HVs. The similarity between unnormalized context HVs was measured by sim_{cos}. The synonymy part of TOEFL was used as the benchmark to demonstrate that a performance comparable to LSA could be achieved but at lower computational costs due to the lack of SVD. In [312], it was proposed to speed up LSA by using RP before SVD as a preprocessing step. In [185], similarly to [105, 256], RI was modified to use a narrow context window consisting of only few adjacent words on each side of the focus word.

In [367], permutations were used to represent the order information within the context window. Further extensions of RI included generalization to multidimensional arrays (N-way RI) [368] and inclusion of extra-linguistic features [183]. The RI was also extended to the case of training corpora that include information about relations between words. This model is called **Predication-based Semantic Indexing (PSI)** [43, 45, 46]. PSI has been mainly used in biomedical informatics for literature-based discovery, such as identification of links between pharmaceutical substances and diseases they treat (see [46] for more details). Later, PSI was extended to the **Embedding of Semantic Predications (ESP)** model that incorporates some aspects of "neural" word embeddings from [274] and similarity preserving HVs (Section 3.2 in [222]) for representing time periods [416].

It is worth mentioning that the optimization-based method [397] for obtaining similarity preserving HVs from co-occurrence statistics can be contrasted to RI. RI also uses co-occurrence statistics but implicitly (i.e., without constructing the co-occurrence matrix). The difference, however, is that the RI is optimization-free and it forms HVs through a single pass via the training data. Thus, it can be executed online in an incremental manner, while the optimization required by [397] calls for iterative processing, which might be more suitable for offline operations.

Bound Encoding of the Aggregate Language Environment. There is an alternative method for constructing context HVs with HDC/VSA, known as BEAGLE [172]. It was proposed independently of RI and used HRR to form the HVs of *n*-grams of words within a sentence, providing a representation of the order and context HVs of words. Words were initially represented with random atomic HVs as in HRR. The context HV of a word was obtained from two "parts." The first part included summing the atomic HVs of the words in the sentence other than the focus word for all corpus sentences. The second part was contributed by the word order HVs, which were formed as the superposition of the word *n*-gram HVs (with *n* between 2 and 7). The *n*-gram HVs were formed with the circular convolution-based binding of the special atomic HV in place of the focus word and the atomic HVs of the other word(s) in the *n*-gram. The word order in an *n*-gram was represented recursively, first by binding the HVs of the left and the right word permuted differently and then by binding the resultant HV with the next right word, again using permutations for the "left" and "right" relative positions. The total context HV was the superposition of the HVs for the two parts. The similarity measure was sim_{cos}.

Later, [357] presented a modified version of BEAGLE using random permutations. The authors found that their model was both more scalable to large corpora and gave better fits to semantic similarity than the circular convolution-based representation. A comprehensive treatment of methods for constructing context HVs of phrases and sentences as opposed to individual words was presented in [283]. In a similar spirit, in [257] the HRR model was used to construct compositional HVs that were able to discover language regularities resembling syntax and semantics. The comparison of different sentence embeddings (including BEAGLE) in terms of their ability to represent syntactic constructions was provided in [196]. It has been demonstrated that context HVs formed by BEAGLE can account for a variety of semantic category effects such as typicality, priming, and acquisition of semantic and lexical categories. The effect of using different lexical materials to form context HVs with BEAGLE was demonstrated in [170], while the use of negative information for context HVs was assessed in [171]. In [194], BEAGLE was extended to a Hierarchical Holographic Model by augmenting it with additional levels corresponding to higher-order associations, e.g., part-of-speech and syntactic relations. The Hierarchical Holographic Model was used further in [191] to investigate what grammatical information is available in context HVs produced by the model.

Due to high similarity between BEAGLE and RI, both methods were compared against each other in [357, 358]. It was shown that both methods demonstrate similar results on a set of semantic tasks using a Wikipedia corpus for training. The main difference was that RI is much faster as it does

Table 2. Experiments with Context Vectors

Ref.	Task	Dataset	Method	Baseline(s)
[181]	Synonymy test	TOEFL	RI with word–document matrix	LSA
[365]	Synonymy test	TOEFL	RI with word–word matrix	LSA
[281]	Synonymy test	TOEFL; ESL	RI with word–word matrix	LSA, RI
[281]	Semantic similarity of word pairs	[364]	RI with word–word matrix	LSA, RI
[281]	Word choice in Russian to English translation	Own data	RI with word–word matrix	LSA, RI
[282]	Semantic text search	MEDLARS; Cranfield; Time Magazine	RI with word–document matrix	(Generalized) Vector Space Model
[367]	Synonymy test	TOEFL	RI with permutations	BEAGLE
[48]	Retrieval of cancer therapies	A set of predications extracted from MEDLINE	FHRR-based PSI	BSC-based PSI
[47]	Identification of agents active against cancer cells	SemMedDB	PSI	Reflective RI
[358]	Synonymy test	TOEFL; ESL	BEAGLE; RI with permutations	BEAGLE
[358]	Semantic similarity of word pairs	From References [81, 275, 360, 364]	BEAGLE; RI with permutations	BEAGLE
[166]	Taxonomic organization	From References [172, 256]	ITS	BEAGLE; LSA
[166]	Meaning disambiguation in context	From [379]	ITS	BEAGLE; LSA
[170]	Synonymy test	TOEFL	BEAGLE with experiential optimization	BEAGLE
[24]	Prediction of side-effects for drug combinations	From Reference [435]	ESP	graph convolutional ANN
[170]	Semantic similarity of word pairs	From [81, 275, 364]	BEAGLE with experiential optimization	BEAGLE
[5]	Academic search engine for cognitive psychology	Own data	BEAGLE	RI with permutations
[168]	Influence of corpus effects on lexical behavior	English Lexicon Project; British Lexicon Project; etc.	BEAGLE	N/A
[378]	Contextual similarity among alphanumeric characters	Own data	RI with characters	Word2vec [274] & EARP [44]
[399]	Changes in verbal fluency	Canadian Longitudinal Study of Aging	BEAGLE	N/A

not use the circular convolution operation. For placing the methods in the general context of word embeddings, please refer to [419].

[271] aimed to address the representation of similarity, rather than relatedness represented in context HVs by BEAGLE and RI. To do so, for each word the authors represented its most relevant semantic features taken from a knowledge base ConceptNet. The context HV of a word was formed using BSC as a superposition of its semantic feature HVs formed by role-filler bindings. The results from measuring the semantic similarity between pairs of concepts were presented using the SimLex-999 dataset.

Table 2 provides a non-comprehensive summary of the studies that applied the BEAGLE and RI methods to various linguistic tasks. The interested readers are referred to [42], which is a survey describing the applications of RI, PSI, and related methods, including the biomedical domain. The range of applications described in [42] covers word-sense disambiguation, bilingual information extraction, visualization of relations between terms, and document retrieval. It is worth noting that there is a software package called "Semantic vectors" [414, 415, 418] that implements many of the methods mentioned above and provides the main building blocks for designing further modifications of the methods.

2.2.2 Similarity Estimation of Biomedical Signals. In [220, 221], BSC was applied to biomedical signals: heart rate and respiration. The need for comparing these signals emerged in the scope of a deep breathing test for assessing autonomic function. HDC/VSA was used to analyze cardiorespiratory synchronization by comparing the similarity between heart rate and respiration using feature-based analysis. Feature vectors were extracted from the signals and transformed into HVs by using role-filler bindings (Section 3.1.3 in [222]) and representations of scalars (Section 3.2 in [222]). These HVs were in turn classified into different degrees of cardiorespiratory synchronization/desynchronization. The signals were obtained from the healthy adult controls, patients with cardiac autonomic neuropathy, and patients with myocardial infarction. It was shown that, as expected, the similarity between different HVs were lower for patients with the cardiac autonomic neuropathy and myocardial infarction patients than for the healthy controls.

Another application of BSC was the identification of the ictogenic (i.e., seizure generating) brain regions from **intracranial electroencephalography (iEEG)** signals [28]. The algorithm first transformed iEEG time series from each electrode into a sequence of symbolic local binary pattern codes, from which a binary HV was obtained for each brain state (e.g., ictal or interictal). It then identified the ictogenic brain regions by measuring the relative distances between the learned HVs from different groups of electrodes. Such the identification was done by one-way ANOVA tests at two levels of spatial resolution, the cerebral hemispheres and lobes.

2.2.3 Similarity Estimation of Images. In [296], the HDC/VSA two-dimensional (2D) image representations (Section 3.4 in [222]) were applied for an aggregation of local descriptors extracted from images. Local image descriptors were real-valued vectors whose dimensionality was controlled by RP (Section 3.2.3 in [222]). To represent a position inside an interval, the authors concatenated parts of two basis HVs and used several intervals, as in [339, 341] but using MAP. Position HVs for x and y were bound to represent (x,y); see Section 3.4 in [222]. Subsequently, projected local image descriptors were bound with their position HVs, using component-wise multiplication, and the bound HVs were superimposed to represent the whole image. The image HVs obtained with different algorithms for extracting the descriptors could also be aggregated using the superposition operation. When compared to the standard aggregation methods in (mobile robotics) place recognition experiments, HVs of the aggregated descriptors exhibited an average performance better than alternative methods (except the exhaustive pair-wise comparison). A very similar concept was demonstrated in [285] using an image classification task, see also Table 15. One of the proposed ways of forming image HV used the superposition of three binary HVs obtained from three different hashing neural networks. The HVs representing the aggregated descriptors provided a higher classification accuracy. Finally, similarity-preserving shift-equivariant representation of images in HVs using permutations was proposed in [438].

2.3 Classification

Applying HDC/VSA to classification tasks is currently one of the most common application areas of HDC/VSA. This is due to the fact that similarity-based and other vector-based classifiers are widespread and machine learning research is on the rise in general. The recent survey [110] of classification with HDC/VSA was primarily devoted to the transformation of input data into HVs. Instead, here we focus first on the types of input data (in the second-level headings) and then on the domains where HDC/VSA have been applied (in the third-level headings). Moreover, we cover some of the studies not presented in [110]. The studies are summarized in the form of tables, where each table specifies a reference, type of task, dataset used, format of HVs, operations to form HVs from data,[2] the type of classifier, and baselines for comparison. For the sake of consistency, in this section we use a table even if there is only a single work in a particular domain.

2.3.1 Classification Based on Feature Vectors.

Language identification with the vector of n-gram statistics of letters. In [173], it was shown how to form a compositional HV corresponding to n-gram statistics (see Section 3.3.4 in [222]). The work also introduced a task of identifying a language among 21 European languages. Since then, the task was used in several studies summarized in Table 3.

Classification of texts. Table 4 summarizes the efforts of using HDC/VSA for text classification. The works in this domain dealt with different tasks such as text categorization, news identification, and intent classification. Most of the works [1, 332, 381] used HVs as a way of representing data for conventional machine learning classification algorithms.

[2] For the sake of generality, it was decided to avoid going to in-depth details of data transformations, so tables only specify the HDC/VSA operations used to construct HVs from data.

Table 3. HDC/VSA Studies Classifying Languages

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[173]	Language identification	Wortschatz Corpora & Europarl Corpus	Bipolar	Binding; permutation; superposition	Centroids	Vector centroids
[346]	Language identification	Wortschatz Corpora & Europarl Corpus	Dense binary	Binding; permutation; superposition	Binarized centroids	Localist centroids
[157]	Language identification	Wortschatz Corpora & Europarl Corpus	Sparse binary	Binding; permutation; superposition	Binarized centroids	Approach from Reference [346]
[219]	Language identification	Wortschatz Corpora & Europarl Corpus	Bipolar	Binding; permutation; superposition	Self-organizing map	Approach from Reference [173]
[369]	Language identification	Wortschatz Corpora & Europarl Corpus	Dense binary	Binding; permutation; superposition	Evolvable binarized centroids	FastText

Table 4. HDC/VSA Studies Classifying Texts

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[332]	Text classification	Reuters-21578	Sparse binary	Thresholded RP	SVM	SVM with frequency vectors
[82]	News classification	20 Newsgroups	Real valued	Binding; superposition	SVM with context and part of speech HVs	SVM with context HVs
[293]	News classification	Reuters newswire	Dense binary	Binding; permutation; superposition	Centroids	Bayes, *k*NN, and SVM without HVs
[381]	Intent classification	Chatbot; Ask Ubuntu; Web Applications	Dense binary	Binding; permutation; superposition	Binarized ANN	Classifiers without HVs
[1]	Intent classification	Chatbot; Ask Ubuntu; Web Applications	Bipolar	Binding; permutation; superposition	Machine learning algorithms	Machine learning algorithms without HVs
[402]	Text spam detection	Hotel reviews; SMS text; YouTube comments	Bipolar	Binding; permutation; superposition	Refined centroids	*k*NN; SVM; ANN; Random Forest

Table 5. HDC/VSA Studies Classifying Acoustic Signals

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[334]	Vowels recognition	Own data	Sparse binary	Binding; superposition	Stochastic perceptron; centroids in associative memory	N/A
[332]	Distinguish nasal and oral sounds	Phoneme dataset	Sparse binary	RSC or Prager	SVM	ANN; *k*NN; IRVQ
[351]	Recognition of spoken words	CAREGIVER Y2 UK	Real valued	RP; binding; permutation; superposition	Centroids	Gaussian mixture-based model
[158]	Recognition of spoken letters	Isolet	Dense binary	Binding; superposition	Binarized centroids; binarized centroids & ANN	3-layer ANN
[156]	Recognition of spoken letters	Isolet	Dense binary	Binding; permutation; superposition	Refined centroids	Approach from Reference [158]
[422]	Recognition of music genres	Own data	N/A	Superposition	Centroids	N/A
[160]	Recognition of spoken letters	Isolet	Dense binary	Binding; superposition	Binarized refined centroids	Binarized centroids
[163]	Recognition of spoken letters	Isolet	Dense binary	Binding; superposition	Multiple binarized refined centroids	*k*NN
[95]	Speaker recognition	Own data	Sparse binary	LIRA	Large margin perceptron	N/A
[136]	Recognition of spoken letters	Isolet	Bipolar	Binding; superposition	Conditioned centroids	ANN; SVM; AdaBoost
[437]	Recognition of spoken letters	Isolet	N/A	Trainable projection matrix	Refined centroids	ANN; SVM; AdaBoost
[148]	Recognition of spoken letters	Isolet	Bipolar	Trainable projection matrix	Binarized refined centroids	Binarized centroids
[189]	Recognition of spoken letters	Isolet	Bipolar	Binding; superposition	Quantized refined centroids	Non-quantized refined centroids
[32]	Recognition of spoken letters	Isolet	Bipolar	Binding; superposition	Binarized centroids	Approach from Reference [162]
[432]	Recognition of spoken letters	Isolet	Bipolar	Binding; superposition	Quantized refined centroids	N/A
[327]	Recognition of spoken letters	Isolet	Bipolar	Permutation; superposition	Binarized refined centroids	ANN; SVM; AdaBoost
[429]	Recognition of spoken letters	Isolet	Integer valued	Binding; superposition	Discretized stochastic gradient descent	Approach from Reference [159]
[294]	Recognition of spoken letters	Isolet	Integer valued	Compact code by ANN at low dimension	Centroids	ANN; other HDC/VSA solutions
[260]	Multimodal sentiment analysis	CMU MOSI; CMU MOSEI	Real valued	Binding; weighted superposition	Multimodal transformer	LSTM; multimodal transformer

Table 6. HDC/VSA Studies Classifying Faults

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[240]	Acoustic diagnostics of micro machine-tools	Own data	Sparse binary	RSC	Large margin perceptron	N/A
[217, 218]	Fault isolation	From Reference [313]	Dense binary	Binding; superposition	Average dist_{Ham} to the training data HVs	*k*NN
[69]	Ball bearing anomaly detection	IMS Bearing Dataset	Dense binary	Binding; permutation; superposition	Binarized centroids	N/A
[113]	Detection of wafer map defects	WM-811K	Dense binary	Binding; superposition	Binarized centroids	ANN; SVM

Classification of feature vectors extracted from acoustic signals. Classification of various acoustic signals using HVs is provided in Table 5. Tasks were mainly related to speech recognition, e.g., recognition of spoken letters or words.

Fault classification. Studies on applying HDC/VSA to fault classification are limited. We are only aware of two such use-cases (summarized in Table 6) applied to the problems of anomaly detection in a power plant and ball bearings. An earlier work on micro machine-tool acoustic diagnostics was presented in [240].

Automotive data. Table 7[3] presents studies where HDC/VSA was used with automotive data, mainly in autonomous driving scenarios.

Behavioral signals. Studies that used behavioral signals are summarized in Table 8. One of the most common applications was activity recognition, but other tasks were considered as well (see the table).

Biomedical data. Currently explored applications of HDC/VSA on biomedical data can be categorized into five types of data: **electromyography (EMG)** signals, **electroencephalography**

[3] It should be noted that [276, 280] were, strictly speaking, solving the regression problems while [279, 408] were concerned with anomaly detection. These studies are listed in this section for the sake of covering the applications within the automotive data.

Table 7. HDC/VSA Studies Classifying Automotive Data

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[212]	Identification of vehicle type	From Reference [207]	Dense binary	Binding; superposition	Centroids	N/A
[278]	Identification of driving context	Own data	Real valued	Binding; superposition	Spiking ANN	ANN
[276, 277]	Prediction of vehicle's trajectory	Own data; NGSIM US-101	Real valued	Binding; superposition	LSTM	LSTM without HVs
[280]	Prediction of vehicle's trajectory	From Reference [276]; NGSIM US-101	Real valued	Binding; superposition	LSTM	LSTM without HVs
[279]	Detection of abnormal driving situations	From Reference [276]; NGSIM US-101	Real valued	Binding; superposition	Autoencoder ANN	N/A
[373]	Identification of driving style	UAH-DriveSet	Complex valued	Binding; superposition; fractional power encoding	ANN; SNN; SVM; *k*NN	LSTM without HVs
[408]	Detection of automotive sensor attacks	AEGIS Big Data Project	Bipolar	Binding; superposition	Similarity between original and reconstructed samples	N/A

Table 8. HDC/VSA Studies Classifying Behavioral Signals

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[352]	Activity recognition	Palantir	Bipolar	Binding weighted superposition	Centroids	N/A
[201]	Activity recognition	UCIHAR; PAMAP2; EXTRA	Bipolar	Binding; superposition	Binarized refined centroids	Binarized ANN
[35]	Emotion Recognition	AMIGOS	Sparse ternary; binary	Binding; superposition	Binarized centroids	XGBoost
[353]	Next GPS location prediction	Nokia Lausanne	Sparse ternary	Weighted superposition	Centroids	Mixed-order Markov chain
[353]	Next mobile application prediction	Nokia Lausanne	Sparse ternary	Weighted superposition	Centroids	Mixed-order Markov chain
[353]	Next singer prediction	Nokia Lausanne	Sparse ternary	Weighted superposition	Centroids	Mixed-order Markov chain
[160]	Activity recognition	UCIHAR	Dense binary	Binding; superposition	Binarized refined centroids	binarized centroids
[163]	Activity recognition	UCIHAR	Binary	Binding; superposition	Multiple binarized refined centroids	*k*NN
[10]	Activity recognition	From Reference [17]	Bipolar	Superposition	Centroids	SVM
[10]	Detection of Parkinson's Disease	Parkinson's Disease digital biomarker	Bipolar	Superposition	Centroids	SVM
[136]	Activity recognition	UCIHAR; PAMAP2	Bipolar	Binding; superposition	Conditioned centroids	ANN; SVM; AdaBoost
[437]	Activity recognition	UCIHAR	N/A	Trainable projection matrix	Refined centroids	ANN; SVM; AdaBoost
[148]	Activity recognition	UCIHAR	Bipolar	Trainable projection matrix	Binarized refined centroids	Binarized centroids
[189]	Activity recognition	UCIHAR; PAMAP2	Bipolar	Binding; superposition	Quantized refined centroids	Non-quantized refined centroids
[32]	Activity recognition	UCIHAR	Bipolar	Binding; superposition	Binarized centroids	Approach from Reference [162]
[432]	Activity recognition	UCIHAR	Bipolar	Binding; superposition	Quantized refined centroids	N/A
[327]	Activity recognition	UCIHAR; PAMAP2	Bipolar	Permutation; superposition	Binarized refined centroids	ANN; SVM; AdaBoost
[429]	Activity recognition	UCIHAR	Integer valued	Binding; superposition	Discretized stochastic gradient descent	Approach from Reference [159]
[424]	Activity recognition	In-house based on LFMCW radar	Integer valued	Binding; superposition	Refined centroids with masking	10 different methods
[268]	Emotion recognition	AMIGOS DEAP	Dense binary	Binding; permutation; superposition	Binarized centroids	XGBoost; SVM
[270]	Emotion recognition	AMIGOS	Dense binary	Binding; permutation; superposition	Binarized centroids	SVM

Table 9. HDC/VSA Studies Classifying EMG Signals

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[342]	Hand gesture recognition	From Reference [14]	Bipolar	Binding; permutation; superposition	Conditioned centroids	SVM
[224]	Hand gesture recognition	From Reference [14]	Sparse & dense binary	Binding; permutation; superposition	Conditioned centroids	Approach from Reference [342]
[287]	Hand gesture recognition	Own data	Bipolar	Binding; permutation; superposition; scalar multiplication	Binarized centroids	N/A
[289]	Hand gesture recognition	From Reference [14]	Dense binary	Binding; permutation; superposition	Binarized centroids	SVM
[15]	Hand gesture recognition	Own data	Dense binary	Binding; permutation; superposition	Binarized centroids	SVM
[286]	Hand gesture recognition with contraction levels	Own data	Bipolar	Binding; permutation; superposition	Binarized centroids	N/A
[288]	Hand gesture recognition	Own data	Bipolar	Binding; permutation; weighted superposition	Binarized centroids	N/A
[434]	Adaptive hand gesture recognition	Own data	Bipolar	Binding; permutation; weighted superposition	Context-aware binarized centroids	SVM; LDA
[433]	Adaptive hand gesture recognition	Own data	Bipolar	Binding; permutation; weighted superposition	Context-aware binarized centroids	SVM; LDA
[69]	Hand gesture recognition	From Reference [287]	Dense binary	Binding; permutation; superposition	Binarized centroids	Approach from Reference [287]
[186]	Hand gesture recognition	From Reference [342]	Dense binary	Binding; permutation; superposition	Binarized centroids	Approach from Reference [289]

(EEG) signals, **cardiotocography (CTG)** signals, DNA sequences, and **surface-enhanced laser desorption/ionization time-of-flight (SELDI-TOF)** mass spectrometry. Most of the works so far have been conducted on EMG and EEG signals. In fact, there is a study [344], which provides an in-depth coverage of applying HDC/VSA to these modalities, so please refer to this article for a detailed overview of the area.

EMG signals. HDC/VSA was applied to EMG signals for the task of hand gesture recognition. This was done for several different transformations of data into HVs and on different datasets. Refer to Table 9 for the summary.

EEG and iEEG signals. EEG and iEEG signals were used with HDC/VSA for human–machine interfaces and epileptic seizure detection. These efforts are overviewed in Table 10. It is worth mentioning systematization efforts in [309, 310], which reported an assessment of several HDC/VSA models and transformations used for epileptic seizure detection. Another recent work [372] provides a tutorial on applying HDC/VSA for iEEG seizure detection.

CTG signals. So far, there is only one work [10] where CTG signals were used. It is summarized in Table 11.

DNA sequences. Table 12 presents two studies that used DNA sequences in classification tasks.

SELDI-TOF mass spectrometry. Table 13 summarizes a study that used SELDI-TOF mass spectrometry for classifying sensitivity of glioma to chemotherapy.

Multi-modal signals. Studies involding multi-modal signals are summarized in Table 14.

Table 10. HDC/VSA Studies Classifying EEG Signals

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[345, 348]	Subject’s intentions recognition	Monitoring error- related potentials	Bipolar	Binding; superposition; permutation	Conditioned centroids	Gaussian classifier
[10]	Subject’s intentions recognition	Monitoring error- related potentials	Bipolar	Superposition	Centroids	SVM
[344]	Multiclass subject’s intentions recognition	4-class EEG motor imagery signals	Bipolar	Binding; superposition	Binarized centroids	CNN
[141]	Multiclass subject’s intentions recognition	4-class and 3-class EEG motor imagery	Dense binary	Random and trainable projection; superposition	Multiple centroids	SVM
[27]	Epileptic seizure detection	Short-term SWEC-ETHZ iEEG	Dense binary	Binding; superposition	Binarized centroids	ANN
[25]	Epileptic seizure detection	Short-term SWEC-ETHZ iEEG	Dense binary	Binding; superposition each operating on a different feature set	Ensemble of binarized centroids combined via a linear layer	ANN; SVM; CNN
[28]	Epileptic seizure detection & Identification of Ictogenic Brain Regions	Short-term SWEC-ETHZ iEEG	Dense binary	Binding; superposition	Binarized centroids	ANN, LSTM, SVM, RF
[26]	Epileptic seizure detection	Long-term SWEC-ETHZ iEEG	Dense binary	Binding; superposition	Binarized centroids	LSTM, CNN, SVM
[4]	Epileptic seizure detection	CHB-MIT Scalp EEG	Bipolar	Binding; superposition	Centroids	CNN
[311]	Epileptic seizure detection	CHB-MIT Scalp EEG	Bipolar	Binding; superposition	Multi-centroids for sub-classes	Single centroid HDC/VSA solution
[111]	Epileptic seizure detection	UPenn and Mayo Clinic’s Seizure Detection	Dense binary	Binding; superposition	Binarized centroids	SVM
[309]	Epileptic seizure detection	Short-term SWEC-ETHZ iEEG; CHB-MIT Scalp EEG	Dense binary	Binding; superposition	Binarized centroids	SVM
[112]	Epileptic seizure detection	UPenn and Mayo Clinic’s Seizure Detection	Dense binary	Binding; superposition	Binarized centroids	N/A

Table 11. HDC/VSA Studies Classifying Cardiotocography Signals

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[10]	Classification of fetal state	Cardiotocography	Bipolar	superposition	Centroids	SVM

Table 12. HDC/VSA Studies Classifying DNA Sequences

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[161]	DNA classification	Empirical; Molecular Biology	Dense binary	Binding; permutation; superposition	Binarized centroids	*k*NN; SVM
[54]	Detection of tumor	BRCA; KIRP; THCA	Bipolar	Permutation; superposition	Refined centroids	SVM
[333]	Recognition of of splice junctions	Splice-junction Gene Sequences	Sparse binary	Permutation; superposition	Centroids, SVM, *k*NN	CNN, *k*NN
[333]	Prediction of protein’s secondary structure	Protein Secondary Structure	Sparse binary	Permutation; superposition	Centroids, SVM, *k*NN	ANN

Table 13. HDC/VSA Studies Classifying the Sensitivity of Glioma to Chemotherapy using SELDI-TOF

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[340]	Glioma sensitivity classification	Cancer-Glioma	Binary	RSC	SVM	MLP; Probabilistic NN; Associative memory

Table 14. HDC/VSA Studies Classifying Multi-modal Signals

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[35]	Emotion Recognition	AMIGOS	Sparse ternary; binary	Binding; superposition	Binarized centroids	XGBoost
[268]	Emotion recognition	AMIGOS DEAP	Sense binary	Binding; permutation; superposition	Binarized centroids	XGBoost; SVM
[270]	Emotion recognition	AMIGOS	Dense binary	Binding; permutation; superposition	Binarized centroids	SVM
[410]	Septic shock detection	eICU	Dense binary	Bermutation; binding; superposition	Nearest neighbor	N/A

2.3.2 *Classification of Images or Their Properties.* Table 15 provides an overview of the efforts involving images. Since using raw pixels directly would rarely result in a good performance, HVs were produced either from features extracted from images or using HVs obtained from neural networks (see Section 3.4.3 in [222]), which took images as an input.

2.3.3 *Classification of Structured Data.* Classification of structured data can be tricky with conventional machine learning algorithms, since local representations of structured data might not be convenient to use with vector classifiers, especially when the data involve some sorts of hierarchies. HDC/VSA should be well suited for structured data, since they allow representing various structures (including hierarchies) as HVs. To the best of our knowledge, however, the number of such studies is very limited (see Table 16). One such study in [385] used SBDR to predict the properties of chemical compounds and provided the state-of-the-art performance. A more recent example was demonstrated in [258], where 2D molecular structures were transformed into HVs that were used to construct classifiers for drug discovery problems. The approach outperformed the baseline methods on a collection of 30 tasks. Finally, in [302] it was proposed to classify graphs using HDC/VSA. A graph was represented as a superposition of HVs corresponding to vertices and edges. The proposed approach was evaluated on six graph classification datasets; when compared to the baseline approaches, it demonstrated comparable accuracy on four datasets and much shorter training time on all of them.

Table 15. HDC/VSA Studies Classifying Visual Images and Their Properties

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[243]	Texture classification	Own data	Sparse binary	Binding; superposition	Perceptron-like algorithm	N/A
[239]	Micro-object shape recognition	Own data	Sparse binary	RLD HV permutations; superposition	Large margin perceptron	N/A
[209]	Modality classification	IMAGE CLEF2012	Dense binary	Cellular automata; superposition	Centroids	SVM; *k*NN
[18]	Biological gender classification	fMRI from HCP	Bipolar	Binding; superposition	Non-quantized refined centroids	Random Forest; PCA; etc.
[298]	Visual place recognition	Nordland dataset	Bipolar	Permutation; binding; superposition	Centroids	SeqSLAM
[299]	Visual place recognition	OxfordRobotCar; StLucia; CMU Visual Localization	Complex valued	Permutation; binding; superposition	Centroids	AlexNet; HybridNet; NetVLAD; DenseVLAD
[284]	Ego-motion estimation	MVSEC	Dense binary	Permutation; binding; superposition	Centroids	N/A
[142]	Ego-motion estimation	MVSEC	Sparse binary	Random sparse projection; CDT	Incremental centroids	Dense binary/integer; ANN; various regressions
[411]	Detection of pneumonia	SARS-CoV-2 CT-scan [133] & [349]	Bipolar	Binding; superposition	Binarized centroids	ANN
[106]	Object classification	CIFAR-10; Artificial Face	Dense binary	Positional binding; superposition	Ridge regression	CNN
[285]	Object classification	CIFAR-10; NUSWIDE_81	Dense binary	Permutation; binding; superposition	Centroids	N/A
[429]	Object classification	Fashion-MNIST	Integer valued	Binding; superposition	Discretized stochastic gradient descent	Approach from Reference [159]
[235]	Character recognition	MNIST	Sparse binary	LIRA	Large margin perceptron	Conventional classifiers
[239]	Character recognition	MNIST	Sparse binary	RLD HV permutations; superposition	Large margin perceptron	Conventional classifiers
[332]	Character recognition	MNIST	Sparse binary	LIRA HV; superposition	Large margin perceptron	Feature selection
[261]	Character recognition	MNIST	Dense binary	Permutation; binding; superposition	Centroids	N/A
[188]	Character recognition	MNIST	N/A	Cellular automata-based	Random centroids	Naïve Bayes
[40]	Character recognition	MNIST	Binary	Binding; superposition	Non-quantized refined centroids	Refined centroids
[136]	Character recognition	MNIST	Bipolar	Binding; superposition	Conditioned centroids	ANN; SVM; AdaBoost
[189]	Character recognition	MNIST	Bipolar	Binding; superposition	Quantized refined centroids	Non-quantized refined centroids
[437]	Character recognition	MNIST	N/A	Trainable projection matrix	Refined centroids	ANN; SVM; AdaBoost
[32]	Character recognition	MNIST	Bipolar	Binding; superposition	Binarized centroids	Approach from Reference [162]
[327]	Character recognition	MNIST	Bipolar	Permutation; superposition	Binarized refined centroids	ANN; SVM; AdaBoost
[429]	Character recognition	MNIST	Integer valued	Binding; superposition	Discretized stochastic gradient descent	Approach from Reference [159]
[187]	Few-shot character recognition	Omniglot	Dense binary	5-layer CNNs	Weighted *k*NN	Various CNNs
[208]	Few-shot character recognition	Omniglot	Dense binary	5-layer CNNs	Outer product-based associative memory	Approach from [187]
[139]	Few-shot continual learning for image classification	CIFAR-100; miniImageNet; Omniglot	Real valued	Pre-trained CNN and a retrainable linear layer	(Bipolar) centroids; loss-optimized nudged centroids	Various deep ANNs
[239]	Face recognition	ORL	Sparse binary	RLD HV permutations; superposition	Large margin perceptron	Conventional classifiers
[163]	Face recognition	FACE	Binary	Binding; superposition	Multiple binarized refined centroids	*k*NN
[136]	Face recognition	FACE	Bipolar	Binding; superposition	Conditioned centroids	ANN; SVM; AdaBoost
[55]	Face recognition	ORL; FRAV3D; FEI	Sparse binary	RLD; permutation; superposition	Large margin perceptron	SVM; Iterative Closest Point
[189]	Face recognition	FACE	Bipolar	Binding; superposition	Quantized refined centroids	Non-quantized refined centroids
[327]	Face recognition	FACE	Bipolar	Permutation; superposition	Binarized refined centroids	ANN; SVM; AdaBoost
[438]	Character recognition	MNIST; MNIST-C	Sparse binary	local binary pattern; permutation; superposition	Large margin perceptron	Conventional classifiers

Table 16. HDC/VSA Studies Classifying Structured Data

Ref.	Task	Dataset	HV format	Primitives used in data transformation	Classifier	Baseline(s)
[385]	Prediction of chemical compound properties	INTAS00-397	sparse binary	binding; superposition;	SVM	DISCOVERY; ANALOGY
[258]	Drug discovery	Clintox; BBBP; SIDER	Bipolar	Permutation; superposition;	Refined centroids	Logistic Regression; SVM; Random Forest; etc.

Table 17. Example Database Records (Adapted from Reference [176])

Record	Name	Gender	Age
PF3	Pat	female	33
PM6	Pat	male	66
LF6	Lee	female	66
LM3	Lee	male	33
LM6	Lee	male	66

3 COGNITIVE COMPUTING AND ARCHITECTURES

In this section, we overview the use of HDC/VSA in cognitive computing (Section 3.1) and cognitive architectures (Section 3.2). Note that, strictly speaking, cognitive computing as well as cognitive architectures can also be considered to be application areas but we decided to separate them into a distinct section due to the rather different nature of tasks being pursued.

3.1 Cognitive Computing

3.1.1 Holistic Transformations.

Holistic transformations in database processing with HVs. In this section, we consider simple examples of HDC/VSA processing of a small database. This can be treated as analogical reasoning (though analogy researchers might disagree) in the form of query answering using simple analogs (we refer to them as records) without explicitly taking into account the constraints on analogical reasoning mentioned in the following sections.

[176] introduced the transformation of database records into HVs as an alternative to their symbolic representations. Each record is a set of role-filler bindings and is represented by HVs using transformations for role-filler bindings (Section 3.1.3 in [222]) and sets (Section 3.1.2 in [222]). We might be interested in querying the whole base for some property or in processing a pair of records. For example, knowing a filler of one of the roles in one record (the role is not known) can enable one to get the filler of that role in another record. In [176], the records were persons with attributes (i.e., roles) such as "name," "gender," and "age" (see Table 17).

Several ways were proposed to use HDC/VSA operations for querying the records by HVs. An example query to the database could be "What is the age of Lee who is a female?" The correct record for this query will be **LF6** and the answer is 66. Depending on the prior knowledge, different cases were considered. All the cases below assume that there is an item memory with the HVs of all the roles, fillers, and database records.

Case 1. We know the role-filler bindings name:Lee, gender:female, and only the role "age" from the third role-filler binding whose filler we want to find. *Solution* 1. The query is represented as $\mathbf{name} \circ \mathbf{Lee} + \mathbf{gender} \circ \mathbf{female}$. The base memory will return **LF6** as the closest match using the similarity of HVs. Unbinding $\mathbf{age} \oslash \mathbf{LF6}$ results in the noisy version of HV for **66**, and the clean-up procedure returns the value associated with the nearest HV in the item memory, i.e., **66**, which is the answer.

Case 2. The following HVs are available: record **PM6** and the fillers **Pat**, **male**, and **Lee**, as well as the roles **female** and **age**. *Solution* 2a. First, we find **name** by the clean-up procedure on $\mathbf{Pat} \oslash \mathbf{PM6}$, and **gender** by the clean-up procedure on $\mathbf{male} \oslash \mathbf{PM6}$. Then we apply the previous solution (*Solution* 1). *Solution* 2b. This solution[4] uses the correspondences $\mathbf{Pat} \leftrightarrow \mathbf{Lee}$ and $\mathbf{male} \leftrightarrow \mathbf{female}$ by forming the transformation HV $\mathbf{T} = \mathbf{Pat} \circ \mathbf{Lee} + \mathbf{male} \circ \mathbf{female}$. The transformation $\mathbf{T} \oslash \mathbf{PM6}$ returns approximate **LF6** (see [176] for a detailed explanation), and its clean-up provides exact **LF6**. Then, as above, $\mathbf{age} \oslash \mathbf{LF6}$ after the clean-up procedure returns **66**. Note that such a transformation is intended for HDC/VSA models where the binding operation is self-inverse (e.g., BSC).

Case 3. Only **33** and **PF3** are known, and the task is to find an analog of **33** in **LF6**, that is, **66**. A trivial solution would be to get **age** as the result of the clean-up procedure for $\mathbf{33} \oslash \mathbf{PF3}$, and then get **33** by $\mathbf{age} \oslash \mathbf{PF3}$ and the clean-up procedure. But there is a more interesting way, which can be considered as an analogical solution. *Solution* 3. One step solution is $\mathbf{LF6} \oslash (\mathbf{33} \oslash \mathbf{PF3})$. This exemplifies a possibility of processing without the intermediate use of a clean-up procedure.

In some HDC/VSA models (e.g., BSC), however, the answer for this solution will be ambiguous, being equally similar to **33** and **66**. This is due to the self-inverse property of the binding operation in BSC. Note that both **LF6** and **PF3** include $\mathbf{gender} \circ \mathbf{female}$ as the part of their records. Unbinding **33** with **PF3** creates $\mathbf{33} \circ \mathbf{gender} \circ \mathbf{female}$ (since $\oslash = \circ$) among other bindings. When unbinding the result with **LF6**, the HV $\mathbf{gender} \circ \mathbf{female}$ will cancel out, thus releasing **33**, which will interfere with the correct answer, which is **66**. This effect would not appear if the records would had different fillers in their role-filler bindings. For example, if instead of **LF6** we consider **LM6**, then $\mathbf{33} \circ \mathbf{PF3} \circ \mathbf{LM6}$ produces the correct answer.

In models with self-inverse binding, the result of $\mathbf{PF3} \oslash \mathbf{LM6}$ can be seen as an interpretation of **PF3** in terms of **LM6** or vice versa, because the result of this operation is $\mathbf{Pat} \circ \mathbf{Lee} + \mathbf{male} \circ \mathbf{female} + \mathbf{33} \circ \mathbf{66} + \text{noise}$ (since $\oslash = \circ$). This allows answering queries of the form "which filler in

[4] This solution relies on knowing that there is only one Lee & female record in the database, so it should not be considered as a sensible database operation but as an example demonstrating substitution transformations.

PF3 plays the same role as (something) in **LM6**?" by unbinding **PF3** $\oslash$ **LM6** with the required filler HV, resulting in the noisy answer HV.

Note that Solution 1 resembles the standard processing of records in databases. We first identify the record and then check the value of the role of interest. *Solutions* 2b and 3 are examples of a different type of computing sometimes called "holistic mapping" [176] or "transformation without decomposition" [323]. We call it "holistic transformation" to not confuse it with the transformation of input data into HVs or with mapping in analogical reasoning. This holistic transformation of HVs is commonly illustrated by an example well known under the name "Dollar of Mexico" [177, 180, 182].

In essence, the "Dollar of Mexico" show-case solves some simple "proportional analogies" of the form A:B :: C:D as (i.e., United States:Mexico :: Dollar:?). These analogies are also known to be solvable by addition and subtraction of the "neural" word embeddings of the corresponding concepts [274, 318]. Following a similar approach, the authors in [316] proposed to improve the results by training shallow neural networks using a dependency path of relations between terms in sentences.

It should be noted that there is no direct analog to this kind of processing by holistic transformation (using geometric properties of the representational space) in the conventional symbol manipulation [323]. The holistic transformation of HVs can be seen as a parallel alternative to the conventional sequential search.

Learning holistic transformations from examples. Learning systematic transformations from examples was investigated in [300, 301] for HRR. Previously, this capability was shown in [323] only for manually constructing the so-called transformation HV. In [300, 301], the transformation HV was obtained from several training pairs of HVs. One of the proposed approaches to obtaining the transformation HV was to use the gradient descent by iterating through all examples until the optimization converged. The experiments demonstrated that the learned transformation HVs were able to generalize to previously unseen compositional structures with novel elements. A high level of systematicity was indicated by the ability of transformation HVs to generalize to novel elements in structures of a complexity higher than the structures provided as training examples. The capability of BSC to learn holistic transformations was also presented in [178, 182]. However, the disadvantage of such holistic transformations is their bidirectionality, which is due to the fact that the unbinding operation in BSC is equivalent to the binding operation. This complication can be resolved by using either the permutation or an additional associative memory as the binding operation as proposed in [76].

The holistic transformation of the kind considered above was used to associate, e.g., sensory and motor/action data via binding HVs. For example, in [214, 216], BSC was applied to form a scene representation in the experiments with honey bees. It was shown to mimic learning in honey bees using the transformation as in [178, 182]. In [77], an HDC/VSA-based approach for learning behaviors, based on observing and associating sensory and motor/action data represented by HVs, was proposed. A potential shortcoming of the approaches to learning holistic transformations presented above is that the objects/relations are assumed to be dissimilar to each other. The learning might not work as expected given that there is some similarity structure between the objects/relations used as training examples. This direction deserves further investigation.

3.1.2 *Analogical Reasoning.*

We begin with a brief introduction to analogical reasoning by summarizing basic analogical processes as well as some of their properties. Two final sections discuss applications of HDC/VSA to modeling analogical retrieval and mapping.

Basics of analogical reasoning modeling. Modeling analogical reasoning in humans is an important frontier for Artificial Intelligence. Because it allows analogical problem solving as well as even more universal cognitive processes taking the problem structure into account and applying knowledge acquired in different domains. Analogical reasoning theories [114, 115, 117, 118, 227] usually consider and model the following basic processes: description, retrieval (also known as access or search), mapping, and inference.

Any model for analogical reasoning usually works with analogical episodes (or simply analogs). The description process concerns the representation of analogical episodes. The analogical episodes are usually modeled as systems of hierarchical relations (predicates), consisting of elements in the form of entities (objects) and relations of various hierarchical levels. Entities belong to some subject domain (e.g., the sun and planet) and are described by attributes (features, properties), which in essence are relations with a single argument (e.g., mass and temperature). Relations (e.g., attract, more, and cause) define relations between the elements of analogical episodes. Arguments of relations may be objects, attributes, and other relations. It is assumed that a collection of base (source) analogical episodes is stored in a memory.

The retrieval process searches the memory (in models, the base of analogical episodes) to find the closest analog(s) for the given target (input, query) analog. The similarity between episodes is used as a criterion for the search. Once the base analogical episode(s) closest to the target is identified, the mapping process finds the correspondences between the elements of two analogical episodes: the target and the base ones. The inference process concerns the transfer of the knowledge from the base analogical episode(s) to the target analogical episode. This new knowledge obtained about the target may, e.g., provide the solution to the problem specified by the target analogical episode. Because the analogical reasoning is not a deductive mechanism, the candidate inferences are only hypotheses and must be evaluated and checked (see, e.g., [115] and references therein).

Cognitive science has identified quite a few properties demonstrated by subjects when performing analogical reasoning. Two types of similarity that influence processing of analogical episodes are distinguished. Structural similarity (which should not be confused with the structural similarity in HDC/VSA) reflects how the elements of analogs are arranged with respect to each other, that is, in terms of the relations between the elements [74, 117, 118]. Analogs are also matched by the "surface" or "superficial" similarity [85, 114] based on common analogs' elements or a broader "semantic" similarity [74, 154, 401], based on, e.g., joint membership in a taxonomic category or on similarity of characteristic feature vectors. Experiments based on human assessment of similarities and analogies confirmed that both surface (semantic) and structural similarity are necessary for sound retrieval [85]. The structural similarity in the retrieval process is considered less important than that in the mapping process; however, the models of retrieval that take into account only the surface similarity are considered inadequate. These properties are expected to be demonstrated by the computational models of analogy [116]. In the following sections, we discuss applications of HDC/VSA for analogical reasoning.

Analogical retrieval. It is known that humans retrieve some types of analogs more readily than others. Psychologists identified different types of similarities and ordered them according to the ease of retrieval [85, 114, 363, 413]. The similarity types are summarized in Table 18, relative to the base analogical episode. Simple examples of animal stories (adapted by Plate from [401]) with those similarity types are also presented. All analogical episodes have the same first-order relations (in the example, `bite()` and `flee()`). There are also higher-order relations `cause(bite, flee)` and `cause(flee, bite)` and attribute relations (`dog, human, mouse, cat`).

In addition to common first-order relations, the **literal similarity (LS)** also assumes both the same higher-order relations (in the example, the single relation `cause(bite, flee)`) and

Table 18. Types of Analogical Similarity

Similarity type	Common 1st-order relations	Common high-order relations	Common object attributes	Examples using animal episodes

Base				`dog(Spot); human(Jane);`
				`cause(bite(Spot, Jane), flee(Jane, Spot))`

Literal Similarity	✓	✓	✓	`dog(Fido); human(John);`
				`cause(bite(Fido, John), flee(John, Fido))`

Surface Features	✓	✗	✓	`dog(Fido); human(John);`
				`cause(flee(John, Fido), bite(Fido, John))`

True Analogy	✓	✓	✗	`mouse(Mort); cat(Felix);`
				`cause(bite(Felix, Mort), flee(Mort, Felix))`

First-order Only Relations	✓	✗	✗	`mouse(Mort); cat(Felix);`
				`cause(flee(Mort, Felix), bite(Felix, Mort))`

object attributes. The **true analogy (AN)** has the same higher-order relations, but different object attributes. The **surface features (SF)** has the same object attributes but different higher-order relations. The **first-order only relations (FOR)** differ in both higher-order relations and attributes. For the analogical retrieval, it is believed that the retrievability order is expressed as LS $\geq$ SF $>$ AN $\geq$ FOR [85, 114, 363, 413].

Researchers studying analogical reasoning proposed a number of heuristics-based models of the analogical retrieval. The most influential of them are still "many are called but few are chosen," which operate with symbolic structures [85], and analog retrieval by constraint satisfaction using localist neural network structures [401]. The structure of analogical episodes should be taken into account in their similarity estimation. This requires alignment and finding correspondences between elements of the analogical episodes, as in the mapping (Section 3.1.2 below), which is computationally expensive. Moreover, unlike for mapping, where only two analogical episodes are considered, in the retrieval process alignment should be repeated for the target analogical episode and each of the base analogical episodes, making such an implementation of the retrieval prohibitive.

To reduce the computational costs, a two-stage **filter and refine (F&R)** approach is used in the traditional models of analogical retrieval. At the filtering step, the target analogical episode is compared to all the base analogical episodes using a low-cost similarity of their feature vectors (that only counts the frequency of symbolic names in the analogical episodes, without taking into account the structure). The most similar base analogical episodes are selected as prospective candidates. At the refining step, the candidates are then compared to the target analogical episode by the value of their structural similarity. Computationally expensive mapping algorithms (Section 3.1.2 below) are used for calculating the structural similarity. As the final result, the analogical episodes with the highest structural similarity are returned.

HDC/VSA have been applied to modeling of the analogical retrieval by Plate (see, e.g., [319, 320, 324, 326] and references therein). In HDC/VSA, both the set of structure elements and their arrangements influence their HV similarity, so that similar structures (in this case, analogical episodes) produce similar HVs. Because the HV similarity measures are not computationally expensive, the two-stage F&R approach of the traditional models is not needed. Using HRR, it was shown that the results obtained by a single-stage HV similarity estimation are consistent with both the empirical results in psychological experiments as well as the aforementioned leading traditional models of the analogical retrieval. Note that Plate experimented with analogical episodes different from those tested in the leading models, but they still belong to the proper similarity types, as shown in Table 18. Similar results were also reported in [330] for SBDR using Plate's episodes.

The study [338] applied SBDR to represent the analogical episodes in the manner of [330]. However, the performance was evaluated using the test bases of the most advanced models of analogical retrieval. The results demonstrated some increase in the recall and a noticeable increase in the precision compared to the leading traditional (two-stage) models. The authors also compared the computational complexity and found that in most of the cases the HDC/VSA approach had advantages over the traditional models.

Analogical mapping. The most influential models of analogical mapping include the **Structure Mapping Engine (SME)** [79] and its versions and further developments [84], as well as the **Analogical Constraint Mapping Engine (ACME)** [147]. SME is a symbolic model that uses a local-to-global alignment process to determine correspondences between the elements of analogical episodes. SME's drawback is a rather poor account of semantic similarity. Also, structure matching in SME is computationally expensive, so it is prohibitively expensive to use SME during retrieval by comparing input to each of (many) analogical episodes in the memory containing all analogical episodes for a structure-sensitive comparison.

ACME is a localist connectionist model that determines analogical mappings using a parallel constraint satisfaction network. Unlike SME, ACME relies not only on the structural information, but also takes into account semantic and pragmatic constraints. ACME is usually even more computationally expensive than SME.

Further models of mapping based on HRR and BSC were proposed that use techniques based on holistic transformations (Section 3.1.1). One of the limitations of these studies is that the approach was not demonstrated to be scalable to large analogical episodes. HRR was also used in another model for the analogical mapping [74] called DRAMA, where the similarity between HVs was used to initialize a localist network involved in the mapping.

In [330], similarity of HVs (formed with SBDR) of the analogical episodes was used for their mapping. However, this technique worked only for the most straightforward mapping cases. In [331], several alternative techniques for mapping with SBDR were proposed (including direct similarity mapping, re-representation by substitution of identical HVs, and parallel traversing of structures using higher-level roles) and some of them were demonstrated on complex analogies. However, the techniques are rather sophisticated and used sequential operations.

In [386], a kind of re-representation of an analog's element HVs was proposed to allow the analogical mapping of the resultant HVs on the basis of similarity. The re-representation approach included the superposition of two HVs. One of those HVs was obtained as the HV for the episode's element using the usual representational scheme of the episode, e.g., the HV used for the retrieval (compositional structure representation by HVs, see Section 2.2.4 in [222]). This HV took into account the semantic similarity. The other HV was the superposition of HVs of elements higher-level roles. This took into account the structural similarity. The proposed procedure was tested in several experiments on simple analogical episodes used in previous studies (e.g., in [326, 331], and on rather complex analogical episodes previously used only in the leading state-of-the-art models, e.g., "Water Flow-Heat Flow," "Solar System-Atom," and "Schools" [114, 147]. It produced the correct mapping results. The analogical inference was also considered. The computational complexity of the proposed approach was rather low and was largely affected by the dimensionality of HVs.

The problem with the current models of HDC/VSA for analogical mapping is that they lack interaction and competition of consistent alternative mappings. They could probably be improved by using an approach involving the associative memory akin to [109].

Finally, an important aspect of HDC/VSA usage for analogical mapping and reasoning is the compatibility with the existing well-established formats of knowledge representation. This will facilitate the unification of symbolic and subsymbolic approaches for cognitive modeling and

Artificial Intelligence. The work in [272] presented a proof of concept of the mapping between Resource Description Framework Schema ontology and HDC/VSA.

Graph isomorphism. The ability to identify some form of "structural isomorphism" is an important component of analogical mapping [79, 326]. The abstract formulation of isomorphism is graph isomorphism. In [109], an interesting scheme was proposed for finding the graph isomorphism with HDC/VSA and associative memory. The scheme used the mechanism proposed in [250]. The paper presented the HDC/VSA-based implementation of the algorithm proposed in [317], which used replicator equations and treated the problem as a maximal-clique-finding problem. In essence, the HDC/VSA-based implementation transformed the continuous optimization problem into a high-dimensional space where all aspects of the problem to be solved were represented as HVs. The simulation study (unfortunately performed on a simple graph) showed that the distributed version of the algorithm using the mechanism from [250] mimics the dynamics of the localist implementation from [317].

3.1.3 Cognitive Modeling. In this part, we briefly cover known examples of using HDC/VSA for modeling particular cognitive capabilities, such as sequence memorization or problem-solving, for cognitive tasks like the Wason task [71], n-back task [120], Wisconsin card sorting test [174], Raven's Progressive Matrices [354], or the Tower of Hanoi [393].

HDC/VSA as a component of cognitive models. As argued in [195], HDC/VSA is an important tool for cognitive modeling. In cognitive science, HDC/VSA has been commonly used as a part of computational models replicating experimental data obtained from humans. For example, in [22] HVs were used as the representational scheme of a computational model. The model was tested using categorization studies considering three competing accounts of concepts: "prototype theory," "exemplar theory," and "theory theory." The model was shown to be able to replicate experimental data from categorization studies for each of the accounts. It is also worth mentioning that there are numerous works using context HVs (Section 2.2.1) to form models replicating the results obtained in language-related cognitive studies, see, e.g., [41, 166–169, 172, 358, 378, 399].

Modeling human working memory with HDC/VSA. A topic that was studied by different research groups working on HDC/VSA is sequence memorization and recall. For example, it was demonstrated in [128] (see Section 3.3 in [222]) that a HDC/VSA-based representation of sequences performed better than localist representations when compared on the standard benchmark for behavioral effects. Some studies [20, 38, 70, 96, 121, 190, 292, 359] demonstrated how the recall of sequences represented in HVs (Section 3.3 in [222]), albeit with slightly different encoding, can reproduce the performance of human subjects on remembering sequences. This is profound as it demonstrates that simple HDC/VSA operations can reproduce basic experimental findings of human memory studies. An alternative model was proposed in [30]. Importantly, this work linked the neuroscience literature to the modeling of memorizing sequences with HDC/VSA.

Raven's Progressive Matrices. Raven's Progressive Matrices is a nonverbal test used to measure general human intelligence and abstract reasoning. Some simple geometrical figures are placed in a matrix of panels, and the task is to select the figure for an empty panel from a set of possibilities [253]. Note that the key work related to Raven's progressive matrices goes back to the 1930s (see [356]). It was widely used to test human fluid intelligence and abstract reasoning since the 1990s [31]. The task was first brought to the scope of HDC/VSA in [354, 355] using HRR and its subsequent implementation in spiking neurons. Later studies [75, 251] demonstrated that other HDC/VSA models (BSC and MAP, respectively) can also be used to create systems capable of solving a limited set of Raven's Progressive Matrices containing only the progression rule.

In all studies the key ingredient of the solution was the representation of a geometrical panel by its HV (e.g., giving access to the symbolic representation of the panel followed by using role-filler bindings of the shapes and their quantity present in the panel). Subsequently, the HV corresponding to the transformation between the HVs of the adjacent pairs of panels was obtained using the ideas from [301] (see Section 3.1.1). The transformation HV was then used to form a prediction HV for the blank panel in the matrix. The candidate answer with the HV most similar to the prediction HV was then chosen as the answer to the test. All of the previously described studies have two limitations: First, they assume the perception system provides the symbolic representations that support the reasoning for solving Raven's Progressive Matrices test, and, second, they only support the progression rule. [144] addressed these limitations by positioning VSA/HDC as a common language between a neural network (to solve the perception issue) and a symbolic logical reasoning engine (to support more rules). Specifically, it exploited the superposition of multiplicative bindings in a neural network to describe raw sensory visual objects in a panel and used **Fourier Holographic Reduced Representations (FHRR)** to efficiently emulate a symbolic logical reasoning with a rich set of rules [144].

The Tower of Hanoi task. The Tower of Hanoi task, which is a simple mathematical puzzle, is another example of a task used to assess problem solving capabilities. The task involves three pegs and a fixed number of disks of different sizes with holes in the middle such that they can be placed on the pegs. Given a starting position, the goal is to move the disks to a target position. Only one disk can be moved at a time and a larger disk cannot be placed on top of a smaller disk.

In [393], an HDC/VSA-based model capable of solving the Tower of Hanoi tasks was presented. The binding and superposition operations were used to form HVs of the current position and a set of goals identified by the model. The model implemented a known algorithm for solving the task given valid starting and target positions. The performance of the model was compared to that of humans regarding of time delays, which were found to be qualitatively similar.

Modeling the visual similarity of words. Modeling human perception of word similarity with HDC/VSA was based on the experimental data obtained for human subjects in [62]. The task was to model the human patterns of delays in priming tasks with the similarity values of sequence HVs obtained from various HDC/VSA models and for various schemes for sequence representation (Section 3.3 in [222]).

In the task of modeling restrictions on the perception of word similarity, four types of restrictions (a total of 20 similarity patterns) were summarized in [128]. In [128], the BSC model was employed to represent symbols in their positions; various string representations with substrings were also used. Symbols in positions with correlated HVs to represent nearby positions were studied in [49] for the BSC, FHRR, and HRR models. Their results demonstrated partial satisfaction of the restrictions. However, substring representations from [51] with HRR, and symbols-at-correlated positions representations from [333] with SBDR using permutations as well as the ones from [335] with FHRR met all the restrictions for certain choices of the similarity measure and values of a scheme's hyperparameters. In the task of finding correlations between human and model similarity data, [333, 335] demonstrated results were on a par with those of the string kernel similarity measures from [129].

General-purpose rule-based and logic-based inference with HVs. The study in [391] presented a spiking neurons model that was positioned as a general-purpose neural controller. The controller was playing a role analogous to a production system capable of applying inference rules. HDC/VSA and their operations played a key role in the model providing the basis for representing symbols and their relations. The model was demonstrated to perform several tasks: repeating the alphabet, repeating the alphabet starting from a particular letter, and answering simple questions similar to the ones in Section 3.1.1. Another realization of a production system with the Tensor Product Representations model was demonstrated in [65]. Several examples of hierarchical reasoning using the superposition operation on context HVs representing hyponyms to form representations of hypernyms were presented in [230].

In [396], it was demonstrated how HVs can be used to represent a knowledge base with clauses for further performing deductive inference on them. The work widely used negation for logical inference, which was also discussed in [234]. Reasoning with HRR using modus ponens and modus tollens rules was demonstrated in [244]. The works in [127, 376, 377] discussed the usage of VSA/HDC for the realization of context logic language and demonstrated the inference procedures.

3.1.4 Computer Vision and Scene Analysis. This section summarizes different aspects of using HDC/VSA for processing visual data. This is one of the newest and least explored application areas of HDC/VSA.

Visual analogies. In [426], a simple analogy-making scenario was demonstrated on 2D images of natural objects (e.g., bird, horse, and automobile). This work took an image representing a particular category, e.g., a bird. The HVs of images were obtained through using convolutional neural networks (Section 3.4.3 in [222]) and cellular automata computations (see [428] for the method description). Several (e.g., 50 in [426]) such binary HVs (e.g., for images of different birds) were superimposed together to form the HV of a category, e.g.,

$$\mathbf{land} = \mathbf{animal} \circ \mathbf{horse} + \mathbf{vehicle} \circ \mathbf{automobile}, \tag{4}$$

$$\mathbf{air} = \mathbf{animal} \circ \mathbf{bird} + \mathbf{vehicle} \circ \mathbf{airplane}. \tag{5}$$

The category HVs were used to form some statements using the HDC/VSA operations. Inspired by the well-known example of “Dollar of Mexico?” (as in the techniques of Section 3.1.1), it was shown that one could perform queries of a similar form as “What is the Automobile of Air?” (**AoA**) but using HVs formed from the 2D images:

$$\mathbf{AoA} = \mathbf{air} \oslash (\mathbf{land} \oslash \mathbf{automobile}). \tag{6}$$

The system demonstrated a high accuracy (98%) of correct analogy-making on previously unseen images of automobiles.

Reasoning on visual scenes and Visual Question Answering. Visual Question Answering is defined as a task where an agent should answer a question about a given visual scene. In [290], a trainable model was presented that used HDC/VSA for this task. The model in [290] differs from the state-of-the-art solutions that usually include a combination of a recurrent neural network (handles questions and provides answers) and a convolutional neural network (handles visual scenes). The model included two parts. The first part transformed a visual scene into an HV describing the scene using a neural network. This part used only one feed-forward neural network, which took a visual scene and returned its HV. The second part of the model defined the item memory of atomic HVs as well as HVs of questions along with their evaluation conditions in terms of cosine similarity thresholds. The neural network was trained to produce HVs associated with a dataset of simple visual scenes (two figures in various combinations of four possible shapes, colors, and positions). The gradient descent used errors from the question answering on the training data, which included five predefined questions. It was shown that the trained network successfully produced HVs that answered questions for new unseen visual scenes. The five considered questions were answered with 100% accuracy. On previously unseen questions the model demonstrated an accuracy in the range of 60–72%.

Similarly to [290], there were attempts in [144, 198, 263] to train neural networks to output HVs representing a structured description of scenes (see also Section 3.4.3 in [222]), which could then be used for computing visual analogies.

Another approach to Visual Question Answering with HDC/VSA was outlined in [231, 233], where a visual scene was first preprocessed to identify objects and construct a scene data structure called the causal matrix (it stored some object attributes including positions). This data structure describing a scene was transformed into an HV that could then be queried using HDC/VSA operations similar to those from [290]. In [202], it was applied to a dataset constructed to facilitate visual navigation in human-centered environments. This approach was further extended from Visual Question Answering to Visual Dialog in [232].

Another application of HDC/VSA for representation and reasoning on visual scenes, similar in its spirit to the Visual Question Answering, was presented in [412]. The approach represented visual scenes in the form of HVs using Fourier HRR. The paper transformed continuous positions in an image to complex-valued HVs such that the spatial relation between positions was preserved in the HVs (see the "fractional power encoding" in Sections 3.2.1 and 3.4.2 in [222]). During the evaluation, handwritten digits and their positions were identified using a neural network with an attention mechanism. Then the identified information was used to create a complex-valued compositional HV describing the scene. Such scene HVs were used to answer relational queries like "which digit is below 2 and to the left of 1?"

A further exploration of this task was presented in [254]. The approach in [412] also demonstrated solving a simple navigation problem in a maze. In [228], navigation tasks in 2D environments were further studied by using HVs as an input to a neural network producing outputs directing the movement. Neural networks trained with HVs demonstrated the best performance among methods considered. There was also a recent attempt to implement this continuous representation with spiking neurons [67].

3.2 Cognitive Architectures

HDC/VSA have been used as an important component of several bio-inspired cognitive architectures. Here, we briefly describe these proposals.

3.2.1 *Semantic Pointer Architecture Unified Network.*

The most well-known example of a cognitive architecture using HDC/VSA is called "Spaun" for Semantic Pointer Architecture Unified Network (see its overview in [73] and a detailed description in [72]). Spaun is a large spiking neural network (2.5 million neurons) that uses HRR for data representation and manipulation. In the architecture, HVs play the role of "semantic pointers" [22] aiming to integrate connectionist and symbolic approaches. It has an "arm" for drawing its outputs as well as "eyes" for sensing inputs in the form of 2D images (handwritten or typed characters). Spaun (without modifying the architecture) was demonstrated in eight different cognitive tasks that require different behaviors. The tasks used to demonstrate the capabilities of Spaun were as follows: copy drawing of handwritten digits, handwritten digit recognition, reinforcement learning on a three-armed bandit task, serial working memory, counting of digits, question answering given a list of handwritten digits, rapid variable creation, and fluid syntactic or semantic reasoning. The same principles were used in [52] to represent the WordNet knowledge base with HVs allowing enriching, e.g., Spaun, with a memory storing some prior knowledge.

3.2.2 *Associative-Projective Neural Networks.*

The cognitive architecture of **Associative-Projective Neural Networks (APNNs)** that use the SBDR model was proposed and presented in [234, 238, 241, 242, 337]. The goal was to show how to construct a complex hierarchical model of the world that presumably exists in humans' and higher animals' brains, as a step toward Artificial Intelligence.

Two hierarchy types were considered: the compositional (part-whole) as well as the categorization or generalization (class-instance or is-a) ones. An example of the compositional hierarchy is letters → words → sentences → paragraphs → text. Another example is the description of knowledge base episodes in terms of their elements of various complexity from attributes to objects to relations to higher-order relations (see Section 3.1.2). An example of the categorization hierarchy is dog → spaniel → spaniel Rover or apple → big red apple → this big red apple in hand.

The proposed world model relies on models of various modalities, including sensory ones (visual, acoustic, tactile, motoric, etc.) and more abstract modalities (linguistics, planning, reasoning, abstract thinking, etc.), that are organized in hierarchies. The models are required for objects of different nature, e.g., events, real objects, feelings, attributes, and so on. Models (their representations) of various modalities can be combined, resulting in multi-modal representations of objects and associating them with the behavioral schemes (reactions to objects or situations), see details in [234, 238, 241, 242, 337].

The APNN architecture is based on HDC/VSA (though it was proposed long before the terms HDC and VSA appeared); in particular, models are represented by SBDR (Section 2.3.8 in [222]). An approach to formation, storage, and modification of hierarchical models was proposed. This is facilitated by the capability to represent in HVs of fixed dimensionality (for items of various complexity and generality) various heterogeneous data types, e.g., numeric data, images, words, sequences, structures (Section 3 in [222]). As usual for HDC/VSA, the model HVs can be constructed on-the-fly (without learning). APNNs have a multi-module, multi-level, and multi-modal design. A module forms, stores, and processes many HVs representing models of objects of a certain modality and of a certain level of compositional hierarchy. A module's HVs are constructed from HVs obtained from other modules, such as lower-level modules of the same modality, or from modules of other modalities. The lowest level of the compositional hierarchy consists of modules providing a representation grounding (atomic HVs).

For SBDR, a HV is similar to the HVs of its elements of a lower compositional hierarchy level, as well as to the HVs of the higher level, of which the HV is an element. So, using similarity search (in the item memory of levels) it is possible to recover both the lower-level element HVs and compositional HVs of the higher level.

Each module has a long-term memory where it stores its HVs. A Hopfield-like distributed autoassociative memory [98, 99, 125] was suggested as a module memory. It performs the clean-up procedure for noisy or partial HVs by a similarity search. However, its unique property is the formation of the second main hierarchy type, i.e., of generalization (class-instance) hierarchies. It is formed when many similar (correlated) HVs are stored (memorized), based on the idea of Hebb's cell assemblies including cores (subsets of HV 1-components often occurring together, corresponding to, e.g., typical features of categories and object-prototypes) and fringes (features of specific objects), see [125, 337].

It is acknowledged that a world model comprising domain(s) specific knowledge as well as information about an agent itself is necessary for any intelligent behavior. Such a model allows comprehension of the world by an intelligent agent and assists it in its interactions with the environment, e.g., through predictions of action outcomes.

The main problem with the APNN architecture is that not all its aspects have been modeled. For example, there are the following questions, which do not have exact answers:

- How do we extract objects and their parts of various hierarchical levels?
- How do we determine the hierarchical level an object belongs to?

- How do we work with an object that may belong to different hierarchy levels and modules?
- How do we represent objects invariant of their transformations?

Also, modeling cores and fringes formation in distributed auto-associative memory is still fragmentary as of now. Finally, it worth noting that similar ideas are currently being developed in the context of deep neural networks [63, 119].

3.2.3 Hierarchical Temporal Memory. An interesting connection between HDC/VSA and a well-known architecture called **Hierarchical Temporal Memory (HTM)** [132] was presented in [308]. The work showed how HTM can be trained in its usual sequential manner to support basic HDC/VSA operations: binding and superposition of sparse HVs, which are natively used by HTM. Even though permutations were not discussed, it is likely that they also could be implemented, so that HTM could be seen as another HDC/VSA model, which additionally has a learning engine in its core.

3.2.4 Learning Intelligent Distribution Agent. A version of the well-known symbolic cognitive architecture Learning Intelligent Distribution Agent [97], working with HVs, was presented in [389]. In particular, the Modular Composite Representations model was used [388]. Moreover, memory mechanisms in the proposed architecture were also related to HDC/VSA: An extension of Sparse Distributed Memory [175], known as Integer Sparse Distributed Memory [390], was used. The usage of HVs allowed resolving some of the issues with the original model [97] such as representation capability, flexibility, and scalability.

3.2.5 Memories for Cognitive Architectures. Memory is one of the key components of any cognitive architecture and for modeling cognitive abilities of humans. There is a plethora of memory models. For example, MINERVA 2 [146] is an influential computational model of long-term memory. However, in its original formulation MINERVA 2 was not very suitable for an implementation in connectionist systems. Nevertheless, it was demonstrated in [193] that the Tensor Product Representations model (Section 2.3.2 in [222]) can be used to formalize MINERVA 2 as a fixed size tensor of order four. Moreover, it was demonstrated that the lateral inhibition mechanism for HDC/VSA [109] and HRR can be used to approximate MINERVA 2 with HVs. HVs allowed compressing the exact formulation of the model, which relies on tensors, into several HVs, thus making the model more computationally tractable at the cost of lossy representation in HVs.

Another example of using HVs (with HRR) for representing concepts is a Holographic Declarative Memory [3, 190, 192] related to BEAGLE [172] (see Section 2.2.1). It was proposed as a declarative and procedural memory in cognitive architectures. It was shown that the memory can account for many effects such as primacy, recency, probability estimation, interference between memories, and others.

In [166, 169], BEAGLE (Section 2.2.1) was extended to store (instead of one context HV per word) episodic memory of the observed data as HVs of all the contexts. This extension was called the instance theory of semantics. Each word was represented by an atomic random HV. A word's context (a sentence) HV is constructed as a superposition of its word HVs and is stored in the memory. The HV of some query word is constructed as follows. First, the sim_{cos} of the query word HV and each context HV is calculated and raised to a power, producing a vector of "trace activations." Then, context HVs are weighted by traces and summed to produce the retrieved ("semantic") HV of the query word.

The study in [53] introduced a "weighted expectancy subtraction" mechanism that formed actual context HV as follows. First, the context HV produced the retrieved HV as explained above. Then, the retrieved HV was weighted and subtracted from the initial context HV. During the retrieval, the weighted HV of the second retrieval iteration was subtracted from the HV of the first retrieval iteration. This allowed flexibly controlling the construction of general versus specific semantic knowledge. The work in [305] proposed CogNGen, a core of a cognitive architecture that combines predictive processing based on neural generative coding and HDC/VSA models of human memory. The CogNGen architecture learns across diverse tasks and models human performance at larger scales.

4 DISCUSSION

4.1 Application Areas

4.1.1 Context HVs. When it comes to context HVs, Random Indexing and Bound Encoding of the Aggregate Language Environment have appeared as improvements to Latent Semantic Analysis, e.g., they do not require Singular Value Decomposition and can naturally take order information into account. However, they were largely overshadowed after the introduction of "neural" word embeddings in Word2vec [274] or GloVe [318]. The latter are the result of an iterative process, which takes numerous passes via training data to converge. At the same time, an important fact is that distributional models such as Latent Semantic Analysis can in fact benefit from some techniques used in neural word embeddings [248]. Concerning, e.g., Bound Encoding of the Aggregate Language Environment, as recently demonstrated in [171], the method can benefit from negative information. Nevertheless, the current de facto situation in the natural language processing community is that Bound Encoding of the Aggregate Language Environment and Random Indexing methods are rarely the first candidates when it comes to choosing word embeddings. However, since Bound Encoding of the Aggregate Language Environment has been proposed within cognitive science community, it still plays an important role in modeling cognitive phenomena related to memory and semantics [53, 166]. Also, in contrast to the iterative methods, Random Indexing and Bound Encoding of the Aggregate Language Environment only require a single pass through the training data to form context HVs. In some situations, this could be an advantage, especially since the natural language processing community is becoming increasingly concerned about the computational costs of algorithms [395].

4.1.2 Classification. While right now the classification with HDC/VSA is flourishing, there are still important aspects that are often not taken into account in these studies.

Formation of HVs. An important aspect of the formation of HVs is the initial extraction of features from raw data such as 2D images or acoustic signals. Usually, directly transforming raw data into HVs does not result in a good performance, so an additional step of extracting meaningful features is required.

Another important aspect is that, when constructing HVs from feature data for classification, in most cases the transformation of data into HVs is somewhat ad hoc. While there likely will not be a straightforward answer to how transforming data into HVs, it is still important to mention several issues.

It is a well-known fact that the advantage of nonlinear transformation is that classes not linearly separable in the original representation, might become linearly separable after a proper nonlinear transformation to a high-dimensional space (often called lift). This allows using not only k-Nearest Neighbor classifiers but also well-developed linear classifiers to solve problems that are not linearly separable. So nonlinearity is an important aspect of transforming data into HVs. All transformations of data into HVs that we are aware of seem to be nonlinear. However, there are no studies that scrutinize and characterize the nonlinearity properties of HVs obtained from the compositional approach. Moreover, most of the studies choose a particular transformation of numeric vector and stick to it. One of the most common choices is randomized "float" coding [224, 339, 344, 417].

There is, however, a recent study [89, 90] that established a promising connection between kernel methods [6, 347] and the fractional power encoding for representing numeric vectors as well as an earlier algorithm for approximating a particular type of kernels (tree kernels) with the HRR model [430]. In our opinion, the transformation of data into HVs is a hyperparameter of the model and using, e.g., cross-validation to choose the most promising transformation will likely be the best strategy when considering a range of different datasets.

Choice of classifier. As we saw in Section 2.3, centroids are probably the most common approach to forming a classifier in HDC/VSA. This is understandable, since centroids have an important advantage in terms of computational costs—they are very easy to compute. However, as pointed out in [184], the result of superposition does not provide generalization in itself, it is just a representation of combinations of HVs of training samples. Practically, it means that the centroids are not the best performing approach when it comes to classification performance. One way to improve the performance is to assign weights when including new samples into centroids [136, 342]. It was also shown that the perceptron learning rule in [158] and loss-based objective in [139] might significantly improve centroids. Earlier work on HV-based classifiers, e.g., [236, 237, 332, 336] also used linear perceptron and Support Vector Machine classifiers with encouraging results. Note that a large-margin perceptron usually trains much faster than a Support Vector Machine for big data, while providing classification quality at the same level as the Support Vector Machine and usually much higher than that of the standard perceptron. Another recent result [64] is that centroids can be easily combined with a known conventional classifier: generalized learning vector quantization [370]. Using an HDC/VSA transformation of data to HV, the authors obtained state-of-the-art classification results on a benchmark [80]. In general, we believe that when inventing new mechanisms of classification with HDC/VSA, it is important to report the results on collections of datasets instead of only a handful of datasets. For example, for feature-based classification, the UCI Machine Learning Repository [66] and subsets thereof (e.g., [80]) are a common choice (examples of HDC/VSA using it are [64, 92, 210]). For univariate temporal signals, the UCR Time Series Archive [57] is a good option that was used, e.g., in [375]. If a reported mechanism targets a more specific application area, then it would be desirable to evaluate it on a relevant collection for that area.

Many other types of classifiers such as k-Nearest Neighbors [187] are also likely to work with HVs as their input. When HVs are generated by a nonlinear transformation (in [187], using an HDC/VSA-guided convolutional neural network feature extractor), k-Nearest Neighbors classifier forms an "explicit memory" in memory-augmented neural networks or Neural Turing Machines [123]. The contents of the explicit memory can be compressed using outer products with randomized labels [208]. As mentioned in the previous section, linear classifiers can be used with nonlinearly transformed HVs. For example, the ridge regression, which is commonly used for randomized neural networks [371], performed well with HVs [361]. However, not all conventional classifiers work well with HVs [1]. That is because some of the algorithms (e.g., decision tree or Naïve Bayes) assume that any component of a vector can be interpreted on its own. It is a reasonable assumption when components of vectors are meaningful features, but in HVs a component does not usually have a meaningful interpretation. In the case of HDC/VSA with sparse representations, special attention should be given to classifiers that benefit from sparsity. Examples of such classifiers are the sparse Support Vector Machine [78] and winnow algorithm [252].

Applications in machine learning beyond classification. There are also efforts to apply HDC/VSA within machine learning outside of classification. Examples of such efforts are using data transformed into HVs for clustering [7, 137], unsupervised learning [279, 306], multi-task learning

[32–34], distributed learning [149, 361], model compression [32, 143, 361, 362], and ensemble learning [25, 409].

It is expected that fractional power encoding [89, 229, 319] (Section 3.2.1 in [222]) is going to be a particularly fruitful method for enabling new applications beyond classification. This expectation is based on two facts. First, fractional power encoding is known to approximate kernels, which allows for an efficient implementation of kernel methods. There are already examples of its use to implement methods for probability density estimation [89, 90], kernel regression [89, 90], Gaussian processes-based mutual information exploration [101], representing probability statements [100], path integration [68], and reinforcement learning [9]. Second, fractional power encoding provides a simple but powerful way for representing numeric data in HVs, which allows numerous applications relying on such data. Some recent examples include simulation and prediction of dynamical systems [406], reasoning on 2D images [87, 254, 412], navigation in 2D environments [228, 412], representation of time series [375], and even modeling in neuroscience [67, 86].

We did not devote separate sections to these efforts as the studies are still scarce, but the interested readers are kindly referred to the above works for the initial investigations on these topics.

4.1.3 Real-world Use-cases and New Application Areas. Section 2 demonstrated that there have been numerous attempts to apply HDC/VSA in a diverse range of scenarios spanning from communications to analogical reasoning. As we can see from Section 2.3, the most recent uptick in the research activity was applying HDC/VSA to classification tasks. In the near future, we are likely to see them being applied to solving classification tasks in new domains. Examples of such new domains recently appeared in, e.g., [407], where HDC/VSA was applied to branch prediction in processor cores and [380], where HDC/VSA was applied to food profiling.

Concerning the applications of analogical reasoning (Section 3.1.2), the major bottleneck is still the transformation of textual, speech or pictorial descriptions of analogical episodes to directed ordered acyclic graphs that can then be transformed into HVs (Section 3.5.2 in [222]). Note that this problem concerns not only analogical reasoning based on HDC/VSA but all methods that use predicate-based descriptions as inputs.

Nevertheless, there is still a considerable way to go to demonstrate how HDC/VSA-based solutions scale up to real-world problems. We, however, strongly believe that, similarly to the modern reincarnation of connectionist models, eventually research will distill the niches where the advantages of HDC/VSA are self-evident. Currently, one promising niche seems to be the time series classification [375], particularly in-sensor classification of biomedical signals [288] and prosthetic grasping [303]. Furthermore, the exploration of HDC/VSA in novel application domains should be continued. For instance, there were recent applications in communication [140, 150, 151, 199] and in distributed systems [383] (see Section 2.1.2), which were not foreseen by the community. Another recent example is the attempt to apply HDC/VSA to robotics problems [265–267, 284, 296, 298, 404].

4.2 Interplay with Neural Networks

4.2.1 HVs as Input to Neural Networks. One of the most obvious ways to make an interplay between HDC/VSA and neural networks is by using HVs to represent input to neural networks. This is a rather natural combination of the two, because, in essence, neural networks often work with distributed representations. So, processing information distributed in HVs is not an issue for neural networks. However, since HVs are high dimensional, it is not always possible to use them as the input: The size of neural networks' input layer should be set to D (e.g., in [259], whereby a fully connected "readout" layer for a task of choice was trained on D-dimensional input HVs) or even to a tensor composed of HVs (e.g., to represent each position in the retina by its HV, without superposition of HVs). Moreover, the local structure of the input signal space may become different from that used, e.g., in convolutional neural networks. This could require very different neural network architectures compared to modern deep neural networks.

There are, nevertheless, scenarios where using HVs with neural networks appeared to be beneficial. First, HVs are useful in situations when the data to be fed to a neural network are high dimensional and sparse. Then HVs can be used to form more compact distributed representations of these data. A typical example of using such high-dimensional and sparse data is *n*-gram statistics. There are works that studied tradeoffs between the dimensionality of HVs representing *n*-gram statistics (see Section 3.3.4 in [222]) and the performance of neural networks using these HVs as their input [1, 219]. These works demonstrated that it is possible to achieve the same or very similar classification performance with networks of much smaller size. Moreover, the degradation of the classification performance is gradual with the decreasing size of HVs, so their dimensionality can be used to control the tradeoff between the size of the network and its performance. On top of creating more compact representations, an additional advantage of HVs might lie in making HVs binary as in, e.g., Binary Spatter Codes. This might be leveraged in situations where the whole model is binarized [381].

Also, HVs may be useful when the size of input is not fixed but instead could vary for different inputs. Since neural networks are not flexible in changing their architecture, HDC/VSA can be used to take care of forming fixed size HVs for input of variable size. This mode of a neural network interface has been demonstrated in an automotive context to represent either varying number of intersections being crossed by a vehicle [7] or the dynamically changing environment around a vehicle [276, 280]. Further promising avenues for this mode are graphs and natural language processing, since there is a lot of structure in both, which can potentially be represented in HVs [183, 259]. Some investigations in this direction using Tensor Product Representations were presented in [36].

We foresee that this interface mode might expand the applicability of neural networks, as it allows relieving the pressure of forming the task either with fixed size input or in the form of, e.g., a sequence suitable for recurrent neural networks. However, it may require a replacement of the widely used convolutional layers. Although, there are new results [403] suggesting that fully connected neural networks might be a good architecture even for vision tasks.

4.2.2 The Use of Neural Networks for Producing HVs. Transforming data into HVs (see Section 3 in [222]) might be a non-trivial task, especially when data are unstructured and of non-symbolic nature as, e.g., in the case of images (Sections 3.4.1 and 3.4.2 in [222]). Also, those transformations are usually not learned. This challenge stimulates the interface between neural networks and HDC/VSA in the other direction, i.e., to transform activations of neural network layer(s) into HVs. For example, as mentioned in Section 3.4.3 in [222], it is very common to use activations of convolutional neural networks to form HVs of images. This is commonly done using standard pre-trained neural networks [285, 296, 427]. Two challenges here are to increase the dimensionality and change the format of the neural network representations to conform with the HV format requirements. The former one is generally addressed by expanding the dimensionality, e.g., by random projection, possibly with a subsequent binarization by thresholding [138, 296]. Some neural networks already produce binary vectors (see [285]), and the transformation into HVs was done by randomly repeating these binary vectors to get the necessary dimensionality. To address the latter one, in [139, 144, 187, 208], the authors guided a convolutional neural network to produce HDC/VSA-conforming vectors with the aid of proper attention, sharpening, and loss functions. The sign of HV components can be used to transform them into bipolar HVs (of the same dimensionality). These approaches train neural networks from scratch (as with meta-learning in

[139, 187, 208] or additive loss in [144]) such that the activations of the network resemble quasi-orthogonal HVs for, e.g., images of unrelated classes. In [285, 296], the authors superimposed HVs obtained from several neural networks, which improved the results in applications. Yet another promising avenue is make the processes of classification and reconstruction (i.e., generation) of raw sensory data simultaneously. One particular realization of this idea, called "bridge networks," was recently presented in [304]. Finally, it is worth mentioning that a neural network does not necessarily need to produce HVs, but it can benefit from the HDC/VSA operations by improving its retrieval performance through superimposing multiple permuted versions of an output vector, as demonstrated in [56].

4.2.3 HDC/VSA for Simplifying Neural Networks. In [2], it was shown that it is possible to treat the functionality of binarized neural networks with the ideas from high-dimensional geometry. The paper has demonstrated that binarized networks work because of the properties of binary high-dimensional spaces, i.e., the properties used in Binary Spatter Codes [179]. While it is an interesting qualitative result, it did not provide a concrete way to make the two benefit from each other. This is not obvious, since in the standard neural networks all weights are trained via backpropagation, which is rather different from the HDC/VSA principles.

There is, however, a family of randomized neural networks [371] where a part of the network is initialized randomly and stays fixed. There are two versions of such networks: feed-forward (e.g., random vector functional link networks [155] or extreme learning machines [152]) and recurrent (e.g., echo state networks [164] or reservoir computing [255]). The way the randomness is used in these networks can be expressed in terms of HDC/VSA operations for the both feed-forward [210] and recurrent [205] versions. Conventionally, randomized neural networks were used with real-valued representations. However, since it was realized that these networks can be interpreted in terms of HDC/VSA, it appeared natural to use binary/integer variants (as in Binary Spatter Codes and Multiply-Add-Permute) to produce activations of hidden layers of the networks. This opened the avenue for efficient hardware implementations of such randomized neural networks. Yet another connection between HDC/VSA and feed-forward randomized neural networks was demonstrated in [325] where it was shown that HRR's binding operation can be approximated by such networks. Finally, in [23] it was shown that the address mechanism from the Sparse Distributed Memory [175] approximates the attention mechanism [405] used in modern neural networks.

4.2.4 HDC/VSA for Explaining Neural Networks. It was discussed in Section 2.4 in [222] that the capacity theory [91] applies to different HDC/VSA models. As mentioned in the previous section, randomized recurrent neural networks, known as reservoir computing/echo state networks, can be formulated using HDC/VSA. Therefore, capacity theory can also be used to explain memory characteristics of reservoir computing. Moreover, using the abstract idea of dissecting a network into mapping and classifier parts [314], it is possible to apply capacity theory for predicting the accuracy of other types of neural networks (such as deep convolutional neural networks) [225].

In [264], it was shown that Tensor Product Representations approximate representations of structures learned by recurrent neural networks.

4.2.5 The Use of HDC/VSA with Spiking Neural Networks. Another direction of the interplay between HDC/VSA and neural networks is their usage in the context of **spiking neural networks (SNN)**. It is especially important in the context of emerging neuromorphic platforms [58, 273]. The main advantage HDC/VSA can bring into the SNN domain is the ease of transformation to spiking activities, either with rate-based coding for HDC/VSA models with scalar components or phase-to-timing coding for HDC/VSA models with phasor components. The Spaun cognitive architecture overviewed in Section 3.2.1 is one of the first examples where the HRR model was used in the context of SNN. The latest developments [93, 94] use FHRR and HRR to implement associative memory and k-Nearest Neighbor classifiers on SNN. Further, these memories were proposed as building blocks for the realization of a holistic HDC/VSA-based unsupervised learning pipeline on SNN [306]. While in [16] the Sparse Block Codes model was mapped to an SNN circuit. In other related efforts, an event-based dynamic vision sensor [142, 284] or an SNN [291, 436] was used to perform the initial processing of the input signals that were then transformed to HVs to form the prediction model.

These works provide some initial evidence of the expressiveness of HDC/VSA, on the one hand, and compatibility with SNNs, on the other. We therefore foresee that using HDC/VSA as a programming/design abstraction for various cognitive functionalities will soon manifest itself in the emergence of novel SNN-based applications.

4.2.6 "Hybrid" Solutions. By "hybrid" in this context we refer to solutions that use both neural networks and some elements of HDC/VSA. Currently, a particularly common primitive used in such hybrid solutions is the representation of a set of role-filler bindings or superposition of multiplicative bindings. For example, in [37] the weights of several neural networks were stored jointly by using the superposition operation, which alleviated the problem of "catastrophic forgetting." In [420], activations of layers of a deep neural network were used as filler HVs. They were bound to the corresponding random role HVs and all role-filler bindings were aggregated in a single superposition HV that in turn was used to successfully detect out-of-distribution data. Similarly, in [296, 299, 398], activations of several neural networks were combined together via HDC/VSA operations. In [296, 299], this idea was used to form a single HV compactly representing the aggregated neural networks-based image descriptor while in [398] outputs of multiple neural networks were fused together to solve classification problems. In [107], the superposition of role-filler bindings was used to simultaneously represent the output of a deep neural network when solving multi-label classification tasks. In [144], the activations of the last layer generate a query HV that resembles the superposition of the visual objects available in a panel, whereby each object is uniquely represented by multiplicative binding of its attributes' HVs. In addition, a review of hybrid solutions combining Tensor Product Representations and neural networks such as Tensor Product Generation Networks [153] can be found in [387]. Finally, it is worth noting that all these solutions in some way relied on the idea of "computing in superposition" [204] suggesting that HVs can be used to simultaneously manipulate several pieces of information.

4.3 Open Issues

As introduced at the beginning of this survey, HDC/VSA originated from proposals of integrating the advantages of the symbolic approach to Artificial Intelligence, such as compositionality and systematicity, and those of the neural networks approach to Artificial Intelligence (connectionism), such as vector-based representations, learning, and grounding. There is also the "neural-symbolic computing" [59, 60], or "neurosymbolic AI" [61] community that suggests hybrid approaches to Artificial Intelligence. The key idea is to form an interface so that symbolic and connectionist approaches can work together. At present, HDC/VSA and neural-symbolic computing seem to be rather separate fields that can benefit from synergy. Moreover, the works developing cognitive architectures and Artificial Intelligence with HDC/VSA are rather limited [72, 73, 337].

So far, a major advantage of the HDC/VSA models has been their ability to use HVs in a single unified format to represent data of varied types and modalities. Moreover, the use of HDC/VSA operations allows introducing compositionality into representations. The prerequisite, however, is that the representation of the input data to be transformed into HVs should be able to specify the compositionality explicitly. Nevertheless, despite this advantage, HDC/VSA is usually overlooked in the context of neural-symbolic computing, which calls for establishing a closer interaction between these two communities.

In fact, most of the works use HDC/VSA to reimplement the symbolic Artificial Intelligence primitives or the machine learning functionality with HVs, in a manner suitable for emerging unconventional computing hardware. At the same time, when transforming data into HVs, machine learning is used rarely if at all. Learning is used mainly for training a classifier based on the already constructed HVs. In most recent studies, the narrative is often to demonstrate solutions to some simple classification or similarity search problems. In so, the quality of the results is comparable to the state-of-the-art solutions, but the energy/computational costs required by an HDC/VSA-based solution are only a fraction of those of the baseline approaches. These developments suggest that HDC/VSA might find one of their niches in application areas known as "tiny machine learning" and "edge machine learning." Nevertheless, there is an understanding that manually designing transformations of data into HVs for certain modalities (e.g., 2D images) is a challenging task. This stimulates attempts that use modern learning-based approaches, such as deep neural networks, for producing HVs (see Section 4.2.2). The current attempts, however, are rather limited, since they focus heavily on using HVs formed from the representations produced by neural networks to solve some downstream machine learning tasks (e.g., similarity search or classification).

Another approach would be to discover general principles for combining neural networks and HDC/VSA, but currently there are only few such efforts [36, 37, 143, 144, 304, 431] (see also Section 4.2.6). For example, the Tensor Product Representations operations in [36], and the HDC/VSA operations in [144], are introduced into the neural network machinery. These attempts are timely for the connectionist approach to Artificial Intelligence, since, despite recent uptick of deep neural networks [247], there is a growing awareness within the connectionist community that reaching Artificial General Intelligence is going to require a much higher level of generalization than that available in modern neural networks [122, 124, 145, 262]. A recent proposal in [387] could be considered more HDC/VSA oriented.

One of the milestones toward reaching Artificial General Intelligence is solving the problem of compositionality. For example, [124] stressed the importance of various aspects of binding for achieving compositionality and generalization. The framework of HDC/VSA has a dedicated on-the-fly operation for binding (Section 2.2.3 in [222]), which does not require any training. The neural implementation of binding in the context of SNNs is still an open issue. There are, however, two recent proposals [439, 440] aiming at addressing this issue.

Further, [122] suggested that various inductive biases are required for human-level generalization, including compositionality and discovery of causal dependencies. HDC/VSA have a potential to achieve this through analogical reasoning (Section 3.1.2). However, the progress is held back by the lack of mechanisms to build the analogical episodes, e.g., by observing the outer world or by just reading texts. The analogical episodes should include two major types of hierarchy, the compositional ("part-whole") one, and the generalization ("is-a") hierarchy. We believe that associative memories [125] may provide one way to form "is-a" hierarchies (see the discussion in [337]), but this topic has not yet been studied extensively in the context of HDC/VSA. In terms of forming part-whole hierarchies from 2D images, a recent conceptual proposal was given in [145]. The essence of the proposal is learning to parse 2D images by training a neural network and using the similarity of the learnt high-dimensional vector representations for analogical reasoning. An interesting direction for future work is to see how such representations can be paired with the analogical reasoning capabilities of HDC/VSA. However, all the above and other proposals from the connectionist community rely on learning all the functional lacking in neural networks. But there are also discussions on the innate structure of neural networks [262]. For the sake of fairness, it should be noted that current HDC/VSA also lack ready implementations for most of the above-mentioned functionality. There are a lot of open problems that should be addressed to build truly intelligent systems. Below we list some of them. Some problems related to the internals of HDC/VSA are the following:

- Recovery. Recovering element HVs from compositional HVs. Section 2.2.5 in [222] presented some ideas for recovering the content of HVs. However, for most of the HDC/VSA models, knowledge of all but one bound HV is required. This makes the recovery problem combinatorial (but see a proposal in [88, 197]).
- Similarity. In many of the HDC/VSA models, the results of the binding operation are not similar as soon as a single input HV is dissimilar. While it is often convenient, or even desired, that the result of the binding operation is dissimilar to its input HVs, the price to be paid is weak or no similarity in the resultant HV that includes the same HVs in slightly different combinations. This might hinder the similarity search that is at the heart of HDC/VSA and required in many operations such as the recovery or clean-up procedures.
- Memory. Quantity and allocation of item memories. How many item memories and which HVs of all available HVs should be placed in each of them? Types of the item memories to be used? List memories provide reliable retrieval but are problematic for generalization. Distributed auto-associative memories have problems with dense HVs (but see [350]).
- Generalization of HVs. How do we form generalized HVs containing typical features of objects and is-a hierarchies of objects but preserve HVs of specific objects as well? Distributed auto-associative memories have potential for generalization by unsupervised learning [125].
- Generativity. Is it possible to make a new meaningful compositional HV without constructing it from atomic HVs? Is it possible to produce meaningful input (e.g., fake image or sound, as in deep neural networks) from some generated generalized or specific compositional HV?
- Similarity-based associations. How do we select the particular association needed in the context from myriads of possible associations? For example, between HVs of a part and of a whole or between HVs of a class and of a class instance.
- Parsing. How do we parse the raw sensory inputs into a meaningful part-whole hierarchy of objects?
- Holistic representation. HVs are holistic representations. However, for the comparison of objects we may need to operate with their parts. Part HVs can be found given a particular holistic HV. Is it possible to form holistic HVs of very different inputs of the same class so that they are similar enough to be found in memory?
- Dynamics. Representation, storage, similarity-based search, and replaying of spatial-temporal data (e.g., a kind of video).
- Probabilistic representation. Representation of object(s) reflecting the probabilities assigned to them.
- Learning. How do we learn, for example, the most suitable transformation of input data into HVs for a given problem? Also, learning HV for behaviors, including reinforcement learning.

Let us also touch on problems specific not only to HDC/VSA:

- Representation invariance. To recognize the same object in various situations and contexts, we need some useful invariance of representation that makes similar diverse manifestations of the same object.
- Context-dependent representation. Representing data as objects and getting the proper representation of an object in a particular context.
- Context-dependent similarity. For example, depending on the context a pizza is similar to a cake but also to a frisbee. How can such a context-dependent similarity be implemented?

- Context-dependent processing. The type of processing to be applied to data should take into account the context, such as bottom-up and top-down expectations or system goals.
- Hierarchies. Forming “part-whole” and “is-a” hierarchies. Which level of “part-whole” hierarchy does an object belong to? An object can belong to various levels for different scenes of the same nature. This is also connected to the image scale. An object can belong to very many hierarchies for scenes of varied nature. Concerning “is-a” (class-instance) hierarchy, an object can belong to various hierarchies of classes, subclasses, and so on, in different contexts.
- Cause-effect. Cause-effect extraction in new situations could be done by analogy to familiar ones. Generalizations and specifications using is-a hierarchy are possible.
- Interface with symbolic representations. Designers of cognitive agents have to solve the dual problem of both building a world model and building it so that it can be expressed in symbols to interact with humans.
- The whole system control. Most of the solutions generally rely on conventional deterministic mechanisms for flow control. It is, however, likely that the control of the system should also be trainable, so that the system could adjust it for new tasks.

All the problems described above are rarely (if at all) considered in the scope of HDC/VSA. To the best of our knowledge, one study that discussed some of these problems is [337]. Moreover, it is not fully clear which of these problems are related to the general problems of building Artificial General Intelligence and which ones are due to the architectural peculiarities of neural networks and HDC/VSA. In other words, the separation provided above is not necessarily unequivocal. We believe, however, that building Artificial General Intelligence will require facing these problems anyway. Finally, we hope that insights from HDC/VSA, symbolic Artificial Intelligence, and neural networks will contribute to the solution.

5 CONCLUSION

In this two-part survey, we provided comprehensive coverage of the computing framework known under the names Hyperdimensional Computing and Vector Symbolic Architectures. Part I of the survey [222] covered existing models and transformations of input data of various types into distributed representations. In this Part II, we focused on known applications of Hyperdimensional Computing/Vector Symbolic Architectures including the use in cognitive modeling and cognitive architectures. We also discussed the open problems along with promising directions for the future work. We hope that for newcomers, this two-part survey will provide a useful guide of the field, as well as facilitate its exploration and the identification of fruitful directions for research and exploitation. For the practitioners, we hope that the survey will broaden the vision of the field beyond their specialization. Finally, we expect that it will accelerate the convergence of this interdisciplinary field to discipline with common terminology and solid theoretical foundations.

ACKNOWLEDGMENTS

We thank three reviewers, the editors, and Pentti Kanerva for their insightful feedback as well as Linda Rudin for the careful proofreading that contributed to the final shape of the survey.

REFERENCES

[1] P. Alonso, K. Shridhar, D. Kleyko, E. Osipov, and M. Liwicki. 2021. HyperEmbed: Tradeoffs between resources and performance in NLP tasks with hyperdimensional computing enabled embedding of n-gram statistics. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN’21)*. 1–9.

[2] A. G. Anderson and C. P. Berg. 2018. The high-dimensional geometry of binary neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR'18)*. 1–15.

[3] N. Arora, R. L. West, A. Brook, and M. A. Kelly. 2018. Why the Common Model of the mind needs holographic a-priori categories. *Proc. Comput. Sci.* 145 (2018), 680–690.

[4] F. Asgarinejad, A. Thomas, and T. Rosing. 2020. Detection of epileptic seizures from surface EEG using hyperdimensional computing. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC'20)*. 536–540.

[5] H. Aujla, M. J. C. Crump, M. T. Cook, and R. K. Jamieson. 2019. The Semantic Librarian: A search engine built from vector-space models of semantics. *Behav. Res. Methods* 51, 6 (2019), 2405–2418.

[6] B. Scholkopf and A. J. Smola. 2002. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* Adaptive Computation and Machine Learning Series.

[7] T. Bandaragoda, D. De Silva, D. Kleyko, E. Osipov, U. Wiklund, and D. Alahakoon. 2019. Trajectory clustering of road traffic in urban environments using incremental machine learning in combination with hyperdimensional computing. In *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC'19)*. 1664–1670.

[8] I. Barclay, C. Simpkin, G. Bent, T. La Porta, D. Millar, A. Preece, I. Taylor, and D. Verma. 2022. Trustable service discovery for highly dynamic decentralized workflows. *Fut. Gener. Comput. Syst.* 134 (2022), 236–246.

[9] M. Bartlett, T. C. Stewart, and J. Orchard. 2022. Biologically-based neural representations enable fast online shallow reinforcement learning. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'22)*. 2981–2987.

[10] T. Basaklar, Y. Tuncel, S. Y. Narayana, S. Gumussoy, and U. Y. Ogras. 2021. Hypervector design for efficient hyperdimensional computing on edge devices. In *Proceedings of the tinyML Research Symposium (tinyML'21)*. 1–9.

[11] P. beim Graben and S. Gerth. 2012. Geometric representations for minimalist grammars. *J. Logic Lang. Inf.* 21 (2012), 393–432.

[12] P. beim Graben, M. Huber, W. Meyer, Ronald Romer, C. Tschope, and M. Wolff. 2022. Vector symbolic architectures for context-free grammars. *Cogn. Comput.* 14 (2022), 733–748.

[13] P. beim Graben and R. Potthast. 2012. A Dynamic Field account of language-related brain potentials. In *Principles of Brain Dynamics: Global State Interactions*. 93–112.

[14] S. Benatti, E. Farella, E. Gruppioni, and L. Benini. 2014. Analysis of robust implementation of an EMG pattern recognition based control. In *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS'14)*. 45–54.

[15] S. Benatti, F. Montagna, V. Kartsch, A. Rahimi, D. Rossi, and L. Benini. 2019. Online learning and classification of EMG-based gestures on a parallel ultra-low power platform using hyperdimensional computing. *IEEE Trans. Biomed. Circ. Syst.* 13, 3 (2019), 516–528.

[16] G. Bent, C. Simpkin, Y. Li, and A. Preece. 2022. Hyperdimensional computing using time-to-spike neuromorphic circuits. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'22)*. 1–8.

[17] G. Bhat, N. Tran, H. Shill, and U. Y. Ogras. 2020. w-HAR: An activity recognition dataset and framework using low-power wearable devices. *Sensors* 20, 18 (2020), 1–26.

[18] R. Billmeyer and K. Parhi. 2021. Biological gender classification from fMRI via hyperdimensional computing. In *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*. 578–582.

[19] B. H. Bloom. 1970. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM* 13, 7 (1970), 422–426.

[20] P. Blouw and C. Eliasmith. 2013. A neurally plausible encoding of word order information into a semantic vector space. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'13)*. 1905–1910.

[21] P. Blouw and C. Eliasmith. 2015. Constraint-based parsing with distributed representations. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'15)*. 238–243.

[22] P. Blouw, E. Solodkin, P. Thagard, and C. Eliasmith. 2016. Concepts as semantic pointers: A framework and computational model. *Cogn. Sci.* 40, 5 (2016), 1128–1162.

[23] T. Bricken and C. Pehlevan. 2021. Attention approximates sparse distributed memory. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS'21)*. 1–15.

[24] H. A. Burkhardt, D. Subramanian, J. Mower, and T. Cohen. 2019. Predicting adverse drug-drug interactions with neural embedding of semantic predications. In *Proceedings of the AMIA Annual Symposium*. 992–1001.

[25] A. Burrello, S. Benatti, K. Schindler, L. Benini, and A. Rahimi. 2021. An ensemble of hyperdimensional classifiers: Hardware-friendly short-latency seizure detection with automatic iEEG electrode selection. *IEEE J. Biomed. Health Inf.* 25, 4 (2021), 935–946.

[26] A. Burrello, L. Cavigelli, K. Schindler, L. Benini, and A. Rahimi. 2019. Laelaps: An energy-efficient seizure detection algorithm from long-term human iEEG recordings without false alarms. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'19)*. 752–757.

[27] A. Burrello, K. Schindler, L. Benini, and A. Rahimi. 2018. One-shot learning for iEEG seizure detection using end-to-end binary operations: Local binary patterns with hyperdimensional computing. In *Proceedings of the IEEE Biomedical Circuits and Systems Conference (BioCAS'18)*. 1–4.

[28] A. Burrello, K. Schindler, L. Benini, and A. Rahimi. 2020. Hyperdimensional computing with local binary patterns: One-shot learning of seizure onset and identification of ictogenic brain regions using short-time iEEG recordings. *IEEE Trans. Biomed. Eng.* 67, 2 (2020), 601–613.

[29] W. R. Caid, T. S. Dumais, and S. I. Gallant. 1995. Learned vector-space models for document retrieval. *Inf. Process. Manage.* 31, 3 (1995), 419–429.

[30] R. Calmus, B. Wilson, Y. Kikuchi, and C. I. Petkov. 2019. Structured sequence processing and combinatorial binding: Neurobiologically and computationally informed hypotheses. *Philos. Trans. Roy. Soc. B* 375, 1791 (2019), 1–13.

[31] P. A. Carpenter, M. A. Just, and P. Shell. 1990. What one intelligence test measures: A theoretical account of the processing in the Raven progressive matrices test. *Psychol. Rev.* (1990).

[32] C.-Y. Chang, Y.-C. Chuang, E.-J. Chang, and A.-Y. A. Wu. 2021. MulTa-HDC: A multi-task learning framework for hyperdimensional computing. *IEEE Trans. Comput.* 70, 8 (2021), 1269–1284.

[33] C.-Y. Chang, Y.-C. Chuang, and A.-Y. A. Wu. 2020. IP-HDC: Information-preserved hyperdimensional computing for multi-task learning. In *Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS'20).* 1–6.

[34] C.-Y. Chang, Y.-C. Chuang, and A.-Y. A. Wu. 2020. Task-projected hyperdimensional computing for multi-task learning. In *Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI'20) (IFIP Advances in Information and Communication Technology, Vol. 583).* 241–251.

[35] E. Chang, A. Rahimi, L. Benini, and A. Wu. 2019. Hyperdimensional computing-based multimodality emotion recognition with physiological signals. In *Proceedings of the IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS'19).* 137–141.

[36] K. Chen, Q. Huang, H. Palangi, P. Smolensky, K. Forbus, and J. Gao. 2020. Mapping natural-language problems to formal-language solutions using structured neural representations. In *Proceedings of the International Conference on Machine Learning (ICML'20).* 1566–1575.

[37] B. Cheung, A. Terekhov, Y. Chen, P. Agrawal, and B. Olshausen. 2019. Superposition of many models into one. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS'19).* 10868–10877.

[38] X. Choo and C. Eliasmith. 2010. A spiking neuron model of serial-order recall. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'10).* 2188–2193.

[39] X. Choo and C. Eliasmith. 2013. General instruction following in a large-scale biologically plausible brain model. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'10).* 322–327.

[40] Y.-C. Chuang, C.-Y. Chang, and A.-Y. A. Wu. 2020. Dynamic hyperdimensional computing for improving accuracy-energy efficiency trade-offs. In *Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS'20).* 1–5.

[41] C. M. Chubala and R. K. Jamieson. 2013. Recoding and representation in artificial grammar learning. *Behav. Res. Methods* 45, 2 (2013), 470–479.

[42] T. Cohen and D. Widdows. 2009. Empirical distributional semantics: Methods and biomedical applications. *J. Biomed. Inf.* 42, 2 (2009), 390–405.

[43] T. Cohen and D. Widdows. 2017. Embedding of semantic predications. *J. Biomed. Inf.* 68 (2017), 150–166.

[44] T. Cohen and D. Widdows. 2018. Bringing order to neural word embeddings with embeddings augmented by random permutations (EARP). In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL'18).* 465–475.

[45] T. Cohen, D. Widdows, and T. Rindflesch. 2014. Expansion-by-analogy: A vector symbolic approach to semantic search. In *Proceedings of the International Symposium on Quantum Interaction (QI'14) (Lecture Notes in Computer Science, Vol. 8951).* 54–66.

[46] T. Cohen, D. Widdows, R. W. Schvaneveldt, P. Davies, and T. C. Rindflesch. 2012. Discovering discovery patterns with predication-based semantic indexing. *J. Biomed. Inf.* 45, 6 (2012), 1049–1065.

[47] T. Cohen, D. Widdows, C. Stephan, R. Zinner, J. Kim, T. Rindflesch, and P. Davies. 2014. Predicting high-throughput screening results with scalable literature-based discovery methods. *CPT: Pharmacom. Syst. Pharmacol.* 3, 10 (2014), 1–9.

[48] T. Cohen, D. Widdows, L. De Vine, R. W. Schvaneveldt, and T. C. Rindflesch. 2012. Many paths lead to discovery: Analogical retrieval of cancer therapies. In *Proceedings of the International Symposium on Quantum Interaction (QI'12) (Lecture Notes in Computer Science, Vol. 7620).* 90–101.

[49] T. Cohen, D. Widdows, M. Wahle, and R. W. Schvaneveldt. 2013. Orthogonality and orthography: Introducing measured distance into semantic space. In *Proceedings of the International Symposium on Quantum Interaction (QI'13) (Lecture Notes in Computer Science, Vol. 8369).* 34–46.

[50] G. Cormode and S. Muthukrishnan. 2005. An improved data stream summary: The count-min sketch and its applications. *J. Algor.* 55, 1 (2005), 58–75.

[51] G. E. Cox, G. Kachergis, G. Recchia, and M. N. Jones. 2011. Toward a scalable holographic word-form representation. *Behav. Res. Methods* 43, 3 (2011), 602–615.

[52] E. Crawford, M. Gingerich, and C. Eliasmith. 2016. Biologically plausible, human-scale knowledge representation. *Cogn. Sci.* 40, 4 (2016), 782–821.

[53] M. J. C. Crump, R. K. Jamieson, B. T. Johns, and M. N. Jones. 2020. Controlling the retrieval of general vs specific semantic knowledge in the instance theory of semantic memory. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'20).* 1–7.

[54] F. Cumbo, E. Cappelli, and E. Weitschek. 2020. A brain-inspired hyperdimensional computing approach for classifying massive DNA methylation data of cancer. *Algorithms* 13, 9 (2020), 1–13.

[55] A. Curtidor, T. N. Baidyk, and E. M. Kussul. 2021. Analysis of random local descriptors in face recognition. *Electronics* 10, 11 (2021), 1–19.

[56] I. Danihelka, G. Wayne, B. Uria, N. Kalchbrenner, and A. Graves. 2016. Associative long short-term memory. In *Proceedings of the International Conference on Machine Learning (ICML'16).* 1986–1994.

[57] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh anbd Michael, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh. 2019. The UCR time series archive. *IEEE/CAA J. Autom. Sin.* 6, 6 (2019), 1293–1305.

[58] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang, and H. Wang. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 1 (2018), 82–99.

[59] A. d'Avila Garcez et al. 2019. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *J. Appl. Logics* 6, 4 (2019), 611–632.

[60] A. d'Avila Garcez, K. B. Broda, and D. M. Gabbay. 2002. *Neural-Symbolic Learning System: Foundations and Applications.* Springer-Verlag, Berlin.

[61] A. d'Avila Garcez and L. C. Lamb. 2020. Neurosymbolic AI: The 3rd wave. arXiv:2012.05876. Retrieved from https://arxiv.org/abs/2012.05876.

[62] C. J. Davis. 2010. The spatial coding model of visual word identification. *Psychol. Rev.* 117, 3 (2010), 713–758.

[63] Z. Deng, Z. Fryer, B. Juba, R. Panigrahy, and X. Wang. 2021. Provable hierarchical lifelong learning with a sketch-based modular architecture. arXiv:2112.10919. Retrieved from https://arxiv.org/abs/2112.10909.

[64] C. Diao, D. Kleyko, J. M. Rabaey, and B. A. Olshausen. 2021. Generalized learning vector quantization for classification in randomized neural networks and hyperdimensional computing. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'21).* 1–9.

[65] C. P. Dolan and P. Smolensky. 1989. Tensor product production system: A modular architecture and representation. *Connect. Sci.* 1, 1 (1989), 53–68.

[66] D. Dua and C. Graff. 2019. UCI Machine Learning Repository. Retrieved from http://archive.ics.uci.edu/ml.

[67] N. S.-Y. Dumont and C. Eliasmith. 2020. Accurate representation for spatial cognition using grid cells. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'20).* 2367–2373.

[68] N. S.-Y. Dumont, J. Orchard, and C. Eliasmith. 2022. A model of path integration that connects neural and symbolic representation. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'22).* 3662–3668.

[69] M. Eggimann, A. Rahimi, and L. Benini. 2021. A 5 μW standard cell memory-based configurable hyperdimensional computing accelerator for always-on smart sensing. *IEEE Trans. Circ. Syst. I: Regul. Pap.* 68, 10 (2021), 4116–4128.

[70] J. M. Eich. 1982. A composite holographic associative recall model. *Psychol. Rev.* 89, 6 (1982), 627–661.

[71] C. Eliasmith. 2005. Cognition with neurons: A large-scale, biologically realistic model of the Wason task. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'05)*, Vol. 27. 1–6.

[72] C. Eliasmith. 2013. *How to Build a Brain: A Neural Architecture for Biological Cognition.* Oxford University Press.

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

[74] C. Eliasmith and P. Thagard. 2001. Integrating structure and meaning: A distributed model of analogical mapping. *Cogn. Sci.* 25, 2 (2001), 245–286.

[75] B. Emruli, R. W. Gayler, and F. Sandin. 2013. Analogical mapping and inference with binary spatter codes and sparse distributed memory. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'13).* 1–8.

[76] B. Emruli and F. Sandin. 2014. Analogical mapping with sparse distributed memory: A simple model that learns to generalize from examples. *Cogn. Comput.* 6, 1 (2014), 74–88.

[77] B. Emruli, F. Sandin, and J. Delsing. 2015. Vector space architecture for emergent interoperability of systems by learning from demonstration. *Biologic. Insp. Cogn. Arch.* 11 (2015), 53–64.

[78] K. Eshghi and M. Kafai. 2016. *Support Vector Machines with Sparse Binary High-dimensional Feature Vectors.* Technical Report. Hewlett Packard Labs.

[79] B. Falkenhainer, K. D. Forbus, and D. Gentner. 1989. The structure-mapping engine: Algorithm and examples. *Artif. Intell.* 41, 1 (1989), 1–63.

[80] M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim. 2014. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* 15 (2014), 3133–3181.

[81] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. 2002. Placing search in context: The concept revisited. *ACM Trans. Inf. Syst.* 20, 1 (2002), 116–131.

[82] J. M. Fishbein and C. Eliasmith. 2008. Integrating structure and meaning: A new method for encoding structure for text classification. In *Proceedings of the European Conference on Information Retrieval (ECIR'08) (Lecture Notes in Computer Science, Vol. 4956)*. 514–521.

[83] V. Fock. 1932. Konfigurationsraum und zweite quantelung. *Physik* 75 (1932), 622–647.

[84] K. D. Forbus, R. W. Ferguson, A. Lovett, and D. Gentner. 2017. Extending SME to handle large-scale cognitive modeling. *Cogn. Sci.* 41, 5 (2017), 1152–1201.

[85] K. D. Forbus, D. Gentner, and K. Law. 1995. MAC/FAC: A model of similarity-based retrieval. *Cogn. Sci.* 19, 2 (1995), 141–205.

[86] E. P. Frady, P. Kanerva, and F. T. Sommer. 2018. A framework for linking computations and rhythm-based timing patterns in neural firing, such as phase precession in hippocampal place cells. In *Proceedings of the Annual Conference on Cognitive Computational Neuroscience (CCN'18)*. 1–5.

[87] E. P. Frady, S. J. Kent, P. Kanerva, B. A. Olshausen, and F. T. Sommer. 2018. Cognitive neural systems for disentangling compositions. In *Proceedings of the Cognitive Computing*. 1–3.

[88] E. P. Frady, S. J. Kent, B. A. Olshausen, and F. T. Sommer. 2020. Resonator networks, 1: An efficient solution for factoring high-dimensional, distributed representations of data structures. *Neural Comput.* 32, 12 (2020), 2311–2331.

[89] E. P. Frady, D. Kleyko, C. J. Kymn, B. A. Olshausen, and F. T. Sommer. 2021. Computing on functions using randomized vector representations. arXiv:2109.03429. Retrieved from https://arxiv.org/abs/2109.03429.

[90] E. P. Frady, D. Kleyko, C. J. Kymn, B. A. Olshausen, and F. T. Sommer. 2022. Computing on functions using randomized vector representations (in brief). In *Proceedings of the Neuro-Inspired Computational Elements Conference (NICE'22)*. 115–122.

[91] E. P. Frady, D. Kleyko, and F. T. Sommer. 2018. A theory of sequence indexing and working memory in recurrent neural networks. *Neural Comput.* 30 (2018), 1449–1513.

[92] E. P. Frady, D. Kleyko, and F. T. Sommer. 2021. Variable binding for sparse distributed representations: Theory and applications. *IEEE Trans. Neural Netw. Learn. Syst.* 99 (2021), 1–14.

[93] E. P. Frady, G. Orchard, D. Florey, N. Imam, R. Liu, J. Mishra, J. Tse, A. Wild, F. T. Sommer, and M. Davies. 2020. Neuromorphic nearest-neighbor search using Intel's Pohoiki Springs. In *Proceedings of the Neuro-Inspired Computational Elements Workshop (NICE'20)*. 1–10.

[94] E. P. Frady and F. T. Sommer. 2019. Robust computation with rhythmic spike patterns. *Proc. Natl. Acad. Sci. U.S.A.* 116, 36 (2019), 18050–18059.

[95] N. A. Garcia Fragoso, T. N. Baidyk, and E. M. Kussul. 2020. Speaker recognition using LIRA neural networks. *Int. J. Electr. Comput. Eng.* 14, 1 (2020), 14–22.

[96] D. R. J. Franklin and D. J. K. Mewhort. 2015. Memory as a hologram: An analysis of learning and recall. *Can. J. Exp. Psychol.* 69, 1 (2015), 115–135.

[97] S. Franklin, T. Madl, D. D'mello, and J. Snaider. 2013. LIDA: A systems-level architecture for cognition, emotion, and learning. *IEEE Trans. Auton. Mental Dev.* 6, 1 (2013), 19–41.

[98] A. A. Frolov, D. Husek, and D. A. Rachkovskij. 2006. Time of searching for similar binary vectors in associative memory. *Cybernet. Syst. Anal.* 42, 5 (2006), 615–623.

[99] A. A. Frolov, D. A. Rachkovskij, and D. Husek. 2002. On informational characteristics of Willshaw-like autoassociative memory. *Neural Netw. World* 12, 2 (2002), 141–157.

[100] P. M. Furlong and C. Eliasmith. 2022. Fractional binding in vector symbolic architectures as quasi-probability statements. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'22)*. 259–266.

[101] P. M. Furlong, T. C. Stewart, and C. Eliasmith. 2022. Fractional binding in vector symbolic representations for efficient mutual information exploration. In *Proceedings of the ICRA Workshop: Towards Curious Robots: Modern Approaches for Intrinsically-Motivated Intelligent Behavior*. 1–5.

[102] S. I. Gallant. 1991. A practical approach for representing context and performing word sense disambiguation using neural networks. *Neural Comput.* 3, 3 (1991), 293–309.

[103] S. I. Gallant. 2000. Context vectors: A step toward a grand unified representation. In *Proceedings of the International Workshop on Hybrid Neural Systems (Lecture Notes in Computer Science, Vol. 1778)*. 204–210.

[104] S. I. Gallant. 2022. Orthogonal matrices for MBAT vector symbolic architectures, and a "Soft" VSA representation for JSON. arXiv:2202.04771. Retrieved from https://arxiv.org/abs/2202.04771.

[105] S. I. Gallant and W. R. Caid. 1993. MatchPlus: A context vector system for document retrieval. In *Proceedings of the Workshop on Human Language Technology*. 396.

[106] S. I. Gallant and P. Culliton. 2016. Positional binding with distributed representations. In *Proceedings of the International Conference on Image, Vision and Computing (ICIVC'16)*. 108–113.

[107] A. Ganesan, H. Gao, S. Gandhi, E. Raff, T. Oates, J. Holt, and M. McLean. 2021. Learning with holographic reduced representations. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS'21)*. 1–15.

[108] R. W. Gayler. 2003. Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. In *Proceedings of the Joint International Conference on Cognitive Science (ICCS/ASCS'03)*. 133–138.

[109] R. W. Gayler and S. D. Levy. 2009. A distributed basis for analogical mapping: New frontiers in analogy research. In *Proceedings of the New Frontiers in Analogy Research, Second International Conference on the Analogy (ANALOGY'09)*. 165–174.

[110] L. Ge and K. K. Parhi. 2020. Classification using hyperdimensional computing: A review. *IEEE Circ. Syst. Mag.* 20, 2 (2020), 30–47.

[111] L. Ge and K. K. Parhi. 2021. Seizure detection using power spectral density via hyperdimensional computing. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'21)*. 7858–7862.

[112] L. Ge and K. K. Parhi. 2022. Applicability of hyperdimensional computing to seizure detection. *IEEE Open J. Circ. Syst.* 3 (2022), 59–71.

[113] P. Genssler and H. Amrouch. 2021. Brain-inspired computing for wafer map defect pattern classification. In *Proceedings of the IEEE International Test Conference (ITC'21)*. 123–132.

[114] D. Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cogn. Sci.* 7, 2 (1983), 155–170.

[115] D. Gentner and J. Colhoun. 2010. Analogical processes in human thinking and learning. In *Towards a Theory of Thinking*. Springer, 35–48.

[116] D. Gentner and K. D. Forbus. 2011. Computational models of analogy. *Cogn. Sci.* 2, 3 (2011), 266–276.

[117] D. Gentner and F. Maravilla. 2017. Analogical reasoning. *Int. Handbk. Think. Reas.* 2 (2017), 186–203.

[118] D. Gentner and L. Smith. 2012. Analogical reasoning. *Encycl. Hum. Behav.* 2 (2012), 130–136.

[119] B. Ghazi, R. Panigrahy, and J. Wang. 2019. Recursive sketches for modular deep learning. In *Proceedings of the International Conference on Machine Learning (ICML'19)*. 2211–2220.

[120] J. Gosmann and C. Eliasmith. 2015. A spiking neural model of the n-Back task. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'15)*. 812–817.

[121] J. Gosmann and C. Eliasmith. 2021. CUE: A unified spiking neuron model of short-term and long-term memory. *Psychol. Rev.* 128, 1 (2021), 104–124.

[122] A. Goyal and Y. Bengio. 2020. Inductive biases for deep learning of higher-level cognition. arXiv:2011.15091. Retrieved from https://arxiv.org/abs/2011.15091.

[123] A. Graves, G. Wayne, and I. Danihelka. 2014. Neural Turing machines. arXiv:1410.5401. Retrieved from https://arxiv.org/abs/14105401.

[124] K. Greff, S. van Steenkiste, and J. Schmidhuber. 2020. On the binding problem in artificial neural networks. arXiv:2012.05208. Retrieved from https://arxiv.org/abs/2012.05208.

[125] V. I. Gritsenko, D. A. Rachkovskij, A. A. Frolov, R. W. Gayler, D. Kleyko, and E. Osipov. 2017. Neural distributed autoassociative memories: A survey. *Cybernet. Comput. Eng.* 2, 188 (2017), 5–35.

[126] R. Guirado, A. Rahimi, G. Karunaratne, E. Alarcón, A. Sebastian, and S. Abadal. 2022. Wireless on-chip communications for scalable in-memory hyperdimensional computing. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'22)*. 1–8.

[127] H. R. Schmidtke. 2021. Multi-modal actuation with the activation bit vector machine. *Cogn. Syst. Res.* 66 (2021), 162–175.

[128] T. Hannagan, E. Dupoux, and A. Christophe. 2011. Holographic string encoding. *Cogn. Sci.* 35, 1 (2011), 79–118.

[129] T. Hannagan and J. Grainger. 2012. Protein analysis meets visual word recognition: A case for string kernels in the brain. *Cogn. Sci.* 36, 4 (2012), 575–606.

[130] Z. S. Harris. 1968. *Mathematical Structures of Language*. New York, Interscience Publishers.

[131] E. Hassan, Y. Halawani, B. Mohammad, and H. Saleh. 2021. Hyper-dimensional computing challenges and opportunities for AI applications. *IEEE Access* (2021), 1–15.

[132] J. Hawkins, D. Dubinsky, and S. Ahmad. 2011. *Hierarchical Temporal Memory*. Technical Report. Numenta, Inc.

[133] X. He, X. Yang, S. Zhang, J. Zhao, Y. Zhang, E. Xing, and P. Xie. 2020. Sample-efficient deep learning for COVID-19 diagnosis based on CT scans. medRxiv. 1–10. Retrieved from https://doi.org/10.1101/2020.04.13.20063941

[134] R. Hecht-Nielsen. 1994. Context vectors: General purpose approximate meaning representations self-organized from raw data. *Comput. Intell.: Imitat. Life* 3, 11 (1994), 43–56.

[135] M. Heddes, I. Nunes, T. Givargis, A. Nicolau, and A. Veidenbaum. 2022. Hyperdimensional hashing: A robust and efficient dynamic hash table. arXiv:2205.07850. Retrieved from https://arxiv.org/abs/2205.07850.

[136] A. Hernandez-Cane, N. Matsumoto, E. Ping, and M. Imani. 2021. OnlineHD: Robust, efficient, and single-pass online learning using hyperdimensional system. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'21)*. 56–61.

[137] A. Hernández-Cano, Y. Kim, and M. Imani. 2021. A framework for efficient and binary clustering in high-dimensional space. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'21)*. 1859–1864.

[138] M. Hersche, L. Benini, and A. Rahimi. 2020. Binarization methods for motor-imagery brain–computer interface classification. *IEEE J. Emerg. Select. Top. Circ. Syst.* 10, 4 (2020), 567–577.

[139] M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi. 2022. Constrained few-shot class-incremental learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'22)*. 1–19.

[140] M. Hersche, S. Lippuner, M. Korb, L. Benini, and A. Rahimi. 2021. Near-channel classifier: Symbiotic communication and classification in high-dimensional space. *Brain Inf.* 8 (2021), 1–15.

[141] M. Hersche, J. R. Millan, L. Benini, and A. Rahimi. 2018. Exploring embedding methods in binary hyperdimensional computing: A case study for motor-imagery based brain-computer interfaces. arXiv:1812.05705. Retrieved from https://arxiv.org/abs/1812.05705.

[142] M. Hersche, E. Mello Rella, A. Di Mauro, L. Benini, and A. Rahimi. 2020. Integrating event-based dynamic vision sensors with sparse hyperdimensional computing: A low-power accelerator with online learning capability. In *Proceedings of the IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED'20)*. 169–174.

[143] M. Hersche, P. Rupp, L. Benini, and A. Rahimi. 2020. Compressing subject-specific brain-computer interface models into one model by superposition in hyperdimensional space. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'20)*. 246–251.

[144] M. Hersche, M. Zeqiri, L. Benini, A. Sebastian, and A. Rahimi. 2022. A neuro-vector-symbolic architecture for solving Raven's progressive matrices. arXiv:2203.04571. Retrieved from https://arxiv.org/abs/2203.04571.

[145] G. Hinton. 2021. How to represent part-whole hierarchies in a neural network. arXiv:2102.12627. Retrieved from https://arxiv.org/abs/2102.12627.

[146] D. L. Hintzman. 1984. MINERVA 2: A simulation model of human memory. *Behav. Res. Methods Instrum. Comput.* 16, 2 (1984), 96–101.

[147] K. J. Holyoak and P. Thagard. 1989. Analogical mapping by constraint satisfaction. *Cogn. Sci.* 13, 3 (1989), 295–355.

[148] Y.-R. Hsiao, Y.-C. Chuang, C.-Y. Chang, and A.-Y. A. Wu. 2021. Hyperdimensional computing with learnable projection for user adaptation framework. In *Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI'21)*. 436–447.

[149] C.-Y. Hsieh, Y.-C. Chuang, and A.-Y. A. Wu. 2021. FL-HDC: Hyperdimensional computing design for the application of federated learning. In *Proceedings of the IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS'21)*. 1–5.

[150] C.-W. Hsu and H.-S. Kim. 2019. Collision-tolerant narrowband communication using non-orthogonal modulation and multiple access. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'19)*. 1–6.

[151] C.-W. Hsu and H.-S. Kim. 2020. Non-orthogonal modulation for short packets in massive machine type communications. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'20)*. 1–6.

[152] G. Huang, Q. Zhu, and C. Siew. 2006. Extreme learning machine: Theory and applications. *Neurocomputing* 70, 1–3 (2006), 489–501.

[153] Q. Huang, P. Smolensky, X. He, L. Deng, and D. Wu. 2018. Tensor product generation networks for deep NLP modeling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT'18)*. 1263–1273.

[154] J. E. Hummel and K. J. Holyoak. 1997. Distributed representations of structure: A theory of analogical access and mapping. *Psychol. Rev.* 104, 3 (1997), 427–466.

[155] B. Igelnik and Y. H. Pao. 1995. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Trans. Neural Netw.* 6 (1995), 1320–1329.

[156] M. Imani, C. Huang, D. Kong, and T. Rosing. 2018. Hierarchical hyperdimensional computing for energy efficient classification. In *Proceedings of the ACM/ESDA/IEEE Design Automation Conference (DAC'18)*. 1–6.

[157] M. Imani, J. Hwang, T. Rosing, A. Rahimi, and J. M. Rabaey. 2017. Low-power sparse hyperdimensional encoder for language recognition. *IEEE Des. Test* 34, 6 (2017), 94–101.

[158] M. Imani, D. Kong, A. Rahimi, and T. Rosing. 2017. VoiceHD: Hyperdimensional computing for efficient speech recognition. In *Proceedings of the IEEE International Conference on Rebooting Computing (ICRC'17)*. 1–8.

[159] M. Imani, J. Messerly, F. Wu, W. Pi, and T. Rosing. 2019. A binary learning framework for hyperdimensional computing. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'19)*. 126–131.

[160] M. Imani, J. Morris, S. Bosch, H. Shu, G. De Micheli, and T. Rosing. 2019. AdaptHD: Adaptive efficient training for brain-inspired hyperdimensional computing. In *Proceedings of the IEEE Biomedical Circuits and Systems Conference (BioCAS'19)*. 1–4.

[161] M. Imani, T. Nassar, A. Rahimi, and T. Rosing. 2018. HDNA: Energy-efficient DNA sequencing using hyperdimensional computing. In *Proceedings of the IEEE International Conference on Biomedical and Health Informatics (BHI'18)*. 271–274.

[162] M. Imani, S. Salamat, B. Khaleghi, M. Samragh, F. Koushanfar, and T. Rosing. 2019. SparseHD: Algorithm-hardware co-optimization for efficient high-dimensional computing. In *Proceedings of the IEEE Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM'19).* 190–198.

[163] M. Imani, X. Yin, J. Messerly, S. Gupta, M. Niemier, X. S. Hu, and T. Rosing. 2020. SearcHD: A memory-centric hyperdimensional computing with stochastic training. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 39, 10 (2020), 2422–2433.

[164] H. Jaeger. 2002. *Tutorial on Training Recurrent Neural Networks, Covering BPTT, RTRL, EKF and the Echo State Network Approach.* Technical Report GMD Report 159, German National Research Center for Information Technology.

[165] P. Jakimovski, H. R. Schmidtke, S. Sigg, L. W. F. Chaves, and M. Beigl. 2012. Collective communication for dense sensing environments. *J. Amb. Intell. Smart Environ.* 4, 2 (2012), 123–134.

[166] R. K. Jamieson, J. E. Avery, B. T. Johns, and M. N. Jones. 2018. An instance theory of semantic memory. *Comput. Brain Behav.* 1, 2 (2018), 119–136.

[167] B. T. Johns. 2019. Mining a crowdsourced dictionary to understand consistency and preference in word meanings. *Front. Psychol.* 10, 268 (2019), 1–14.

[168] B. T. Johns and R. K. Jamieson. 2019. The influence of place and time on lexical behavior: A distributional analysis. *Behav. Res. Methods* 51, 6 (2019), 2438–2453.

[169] B. T. Johns and M. N. Jones. 2015. Generating structure from experience: A retrieval-based model of language processing. *Can. J. Exp. Psychol.* 69, 3 (2015), 233–251.

[170] B. T. Johns, M. N. Jones, and D. J. K. Mewhort. 2019. Using experiential optimization to build lexical representations. *Psychonom. Bull. Rev.* 26 (2019), 103–126.

[171] B. T. Johns, D. J. K. Mewhort, and M. N. Jones. 2019. The role of negative information in distributional semantic learning. *Cogn. Sci.* 43, 5 (2019), 1–30.

[172] M. N. Jones and D. J. K. Mewhort. 2007. Representing word meaning and order information in a composite holographic lexicon. *Psychol. Rev.* 114, 1 (2007), 1–37.

[173] A. Joshi, J. T. Halseth, and P. Kanerva. 2016. Language geometry using random indexing. In *Proceedings of the International Symposium on Quantum Interaction (QI'16).* 265–274.

[174] I. Kajic and T. C. Stewart. 2021. Biologically constrained large-scale model of the Wisconsin card sorting test. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'21).* 2295–2301.

[175] P. Kanerva. 1988. *Sparse Distributed Memory.* The MIT Press.

[176] P. Kanerva. 1997. Fully distributed representation. In *Proceedings of the Real World Computing Symposium (RWC'97).* 358–365.

[177] P. Kanerva. 1998. Dual role of analogy in the design of a cognitive computer. In *Advances in Analogy Research: Integration of Theory and Data from the Cognitive, Computational, and Neural Sciences.* 164–170.

[178] P. Kanerva. 2000. Large patterns make great symbols: An example of learning from example. In *Proceedings of the International Workshop on Hybrid Neural Systems (Lecture Notes in Computer Science, Vol. 1778).* 194–203.

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

[180] P. Kanerva. 2010. What we mean when we say "What's the dollar of Mexico?": Prototypes and mapping in concept space. In *Proceedings of the AAAI Fall Symposium. Quantum Informatics for Cognitive, Social, and Semantic Processes.* 2–6.

[181] P. Kanerva, J. Kristoferson, and A. Holst. 2000. Random indexing of text samples for Latent Semantic Analysis. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'00).* 1036.

[182] P. Kanerva, G. Sjödin, J. Kristoferson, R. Karlsson, B. Levin, A. Holst, J. Karlgren, and M. Sahlgren. 2001. Computing with large random patterns. In *The Foundations of Real-World Intelligence.* 251–272.

[183] J. Karlgren and P. Kanerva. 2019. High-dimensional distributed semantic spaces for utterances. *Nat. Lang. Eng.* 25, 4 (2019), 503–517.

[184] J. Karlgren and P. Kanerva. 2021. Semantics in high-dimensional space. *Front. Artif. Intell.* 4 (2021), 1–6.

[185] J. Karlgren and M. Sahlgren. 2001. From words to understanding. In *The Foundations of Real-World Intelligence*, 294–308.

[186] G. Karunaratne, M. Le Gallo, M. Hersche, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi. 2021. Energy efficient in-memory hyperdimensional encoding for spatio-temporal signal processing. *IEEE Trans. Circ. Syst. II: Express Briefs* 68, 5 (2021), 1725–1729.

[187] G. Karunaratne, M. Schmuck, M. Le Gallo, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi. 2021. Robust high-dimensional memory-augmented neural networks. *Nat. Commun.* 12, 1 (2021), 1–12.

[188] N. Karvonen, J. Nilsson, D. Kleyko, and L. Lorna Jimenez. 2019. Low-power classification using FPGA–an approach based on cellular automata, neural networks, and hyperdimensional computing. In *Proceedings of the IEEE International Conference On Machine Learning And Applications (ICMLA'19).* 370–375.

[189] A. Kazemi, M. M. Sharifi, Z. Zou, M. Niemier, X. S. Hu, and M. Imani. 2021. MIMHD: Accurate and efficient hyperdimensional inference using multi-bit in-memory computing. In *Proceedings of the IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED'21).* 1–6.

[190] M. A. Kelly, N. Arora, R. L. West, and D. Reitter. 2020. Holographic declarative memory: Distributional semantics as the architecture of memory. *Cogn. Sci.* 44, 11 (2020), 1–34.

[191] M. A. Kelly, M. Ghafurian, R. L. West, and D. Reitter. 2020. Indirect associations in learning semantic and syntactic lexical relationships. *J. Mem. Lang.* 115 (2020), 1–14.

[192] M. A. Kelly, K. Kwok, and R. L. West. 2015. Holographic declarative memory and the fan effect: A test case for a new memory module for ACT-R. In *Proceedings of the International Conference on Cognitive Modeling (ICCM'15).* 148–153.

[193] M. A. Kelly, D. J. K. Mewhort, and R. L. West. 2017. The memory tesseract: Mathematical equivalence between composite and separate storage memory models. *J. Math. Psychol.* 77 (2017), 142–155.

[194] M. A. Kelly, D. Reitter, and R. L. West. 2017. Degrees of separation in semantic and syntactic relationships. In *Proceedings of the International Conference on Cognitive Modeling (ICCM'17).* 199–204.

[195] M. A. Kelly and R. L. West. 2012. From vectors to symbols to cognition: The symbolic and sub-symbolic aspects of vector-symbolic cognitive models. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'12).* 1768–1773.

[196] M. A. Kelly, Y. Xu, J. Calvillo, and D. Reitter. 2020. Which sentence embeddings and which layers encode syntactic structure? In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'20).* 2375–2381.

[197] S. J. Kent, E. P. Frady, F. T. Sommer, and B. A. Olshausen. 2020. Resonator networks, 2: Factorization performance and capacity compared to optimization-based methods. *Neur. Comput.* 32, 12 (2020), 2332–2388.

[198] S. J. Kent and B. A. Olshausen. 2017. A vector symbolic approach to scene transformation. In *Proceedings of the Annual Conference on Cognitive Computational Neuroscience (CCN'17).* 1–2.

[199] H.-S. Kim. 2018. HDM: Hyper-dimensional modulation for robust low-power communications. In *Proceedings of the IEEE International Conference on Communications (ICC'18).* 1–6.

[200] Y. Kim, M. Imani, N. Moshiri, and T. Rosing. 2020. GenieHD: Efficient DNA pattern matching accelerator using hyperdimensional computing. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'20).* 115–120.

[201] Y. Kim, M. Imani, and T. Simunic. 2018. Efficient human activity recognition using hyperdimensional computing. In *Proceedings of the International Conference on the Internet of Things (IOT'18).* 1–6.

[202] D. E. Kirilenko, A. K. Kovalev, E. Osipov, and A. I. Panov. 2021. Question answering for visual navigation in human-centered environments. In *Proceedings of the Mexican International Conference on Artificial Intelligence (MICAI'21).* 31–45.

[203] D. Kleyko, C. Bybee, C. J. Kymn, B. A. Olshausen, A. Khosrowshahi, D. E. Nikonov, F. T. Sommer, and E. P. Frady. 2022. Integer factorization with compositional distributed representations. In *Proceedings of the Neuro-Inspired Computational Elements Conference (NICE'22).* 73–80.

[204] D. Kleyko, M. Davies, E. P. Frady, P. Kanerva, S. J. Kent, B. A. Olshausen, E. Osipov, J. M. Rabaey, D. A. Rachkovskij, A. Rahimi, and F. T. Sommer. 2021. Vector symbolic architectures as a computing framework for emerging hardware. *Proc. IEEE* 110, 10 (2022), 1538–1571.

[205] D. Kleyko, E. P. Frady, M. Kheffache, and E. Osipov. 2022. Integer echo state networks: Efficient reservoir computing for digital hardware. *IEEE Trans. Neural Netw. Learn. Syst.* 33, 4 (2022), 1688–1701.

[206] D. Kleyko, R. W. Gayler, and E. Osipov. 2020. Commentaries on "Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception" [Science Robotics Vol. 4 Issue 30 (2019) 1-10]. arXiv:2003.11458. Retrieved from https://arxiv.org/abs/2003.11458.

[207] D. Kleyko, R. Hostettler, W. Birk, and E. Osipov. 2015. Comparison of machine learning techniques for vehicle classification using road side sensors. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC'15).* 572–577.

[208] D. Kleyko, G. Karunaratne, J. M. Rabaey, A. Sebastian, and A. Rahimi. 2022. Generalized key-value memory to flexibly adjust redundancy in memory-augmented networks. *IEEE Trans. Neural Netw. Learn. Syst.* 99 (2022), 1–6.

[209] D. Kleyko, S. Khan, E. Osipov, and S. P. Yong. 2017. Modality classification of medical images with distributed representations based on cellular automata reservoir computing. In *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI'17).* 1053–1056.

[210] D. Kleyko, M. Kheffache, E. P. Frady, U. Wiklund, and E. Osipov. 2021. Density encoding enables resource-efficient randomly connected neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 8 (2021), 3777–3783.

[211] D. Kleyko, N. Lyamin, E. Osipov, and L. Riliskis. 2012. Dependable MAC layer architecture based on holographic data representation using hyper-dimensional binary spatter codes. In *Proceedings of the Multiple Access Communications (MACOM'12) (Lecture Notes in Computer Science, Vol. 7642).* 134–145.

[212] D. Kleyko and E. Osipov. 2014. Brain-like classifier of temporal patterns. In *Proceedings of the International Conference on Computer and Information Sciences (ICCOINS'14)*. 104–113.

[213] D. Kleyko and E. Osipov. 2014. On bidirectional transitions between localist and distributed representations: The case of common substrings search using vector symbolic architecture. *Proc. Comput. Sci.* 41 (2014), 104–113.

[214] D. Kleyko, E. Osipov, M. Bjork, H. Toresson, and A. Oberg. 2015. Fly-the-bee: A game imitating concept learning in bees. *Proc. Comput. Sci.* 71 (2015), 25–30.

[215] D. Kleyko, E. Osipov, and R. W. Gayler. 2016. Recognizing permuted words with vector symbolic architectures: A cambridge test for machines. *Proced. Comput. Sci.* 88 (2016), 169–175.

[216] D. Kleyko, E. Osipov, R. W. Gayler, A. I. Khan, and A. G. Dyer. 2015. Imitation of honey bees' concept learning processes using vector symbolic architectures. *Biologic. Insp. Cogn. Arch.* 14 (2015), 57–72.

[217] D. Kleyko, E. Osipov, N. Papakonstantinou, and V. Vyatkin. 2018. Hyperdimensional computing in industrial systems: The use-case of distributed fault isolation in a power plant. *IEEE Access* 6 (2018), 30766–30777.

[218] D. Kleyko, E. Osipov, N. Papakonstantinou, V. Vyatkin, and A. Mousavi. 2015. Fault detection in the hyperspace: Towards intelligent automation systems. In *Proceedings of the IEEE International Conference on Industrial Informatics (INDIN'15)*. 1219–1224.

[219] D. Kleyko, E. Osipov, D. De Silva, U. Wiklund, V. Vyatkin, and D. Alahakoon. 2019. Distributed representation of n-gram statistics for boosting self-organizing maps with hyperdimensional computing. In *Proceedings of the International Andrei Ershov Memorial Conference on Perspectives of System Informatics (PSI) (Lecture Notes in Computer Science, Vol. 11964)*. 64–79.

[220] D. Kleyko, E. Osipov, and U. Wiklund. 2018. Vector-based analysis of the similarity between breathing and heart rate during paced deep breathing. In *Proceedings of the Computing in Cardiology Conference (CinC'18)*. 1–4.

[221] D. Kleyko, E. Osipov, and U. Wiklund. 2019. A hyperdimensional computing framework for analysis of cardiorespiratory synchronization during paced deep breathing. *IEEE Access* 7 (2019), 34403–34415.

[222] D. Kleyko, D. A. Rachkovskij, E. Osipov, and A. Rahimi. 2022. A survey on hyperdimensional computing aka vector symbolic architectures, Part I: Models and data transformations. *ACM Comput. Surv.* (2022).

[223] D. Kleyko, A. Rahimi, R. W. Gayler, and E. Osipov. 2020. Autoscaling Bloom filter: Controlling trade-off between true and false positives. *Neural Comput. Appl.* 32 (2020), 3675–3684.

[224] D. Kleyko, A. Rahimi, D. A. Rachkovskij, E. Osipov, and J. M. Rabaey. 2018. Classification and recall with binary hyperdimensional computing: Tradeoffs in choice of density and mapping characteristic. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 12 (2018), 5880–5898.

[225] D. Kleyko, A. Rosato, E. P. Frady, M. Panella, and F. T. Sommer. 2020. Perceptron theory for predicting the accuracy of neural networks. arXiv:2012.07881. Retrieved from https://arxiv.org/abs/2012.17881.

[226] Y. Knight, M. Spranger, and L. Steels. 2015. A vector representation of fluid construction grammar using holographic reduced representations. In *Proceedings of the EuroAsianPacific Joint Conference on Cognitive Science (EAPCogSci'15)*. 560–565.

[227] B. Kokinov and R. M. French. 2003. Computational models of analogy-making. *Encycl. Cogn. Sci.* 1 (2003), 113–118.

[228] B. Komer and C. Eliasmith. 2020. Efficient navigation using a scalable, biologically inspired spatial representation. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'20)*. 1532–1538.

[229] B. Komer, T. C. Stewart, A. R. Voelker, and C. Eliasmith. 2019. A neural representation of continuous space using fractional binding. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'19)*. 2038–2043.

[230] C. Kommers, V. Ustun, A. Demski, and P. S. Rosenbloom. 2015. Hierarchical reasoning with distributed vector representations. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'15)*. 1171–1176.

[231] A. K. Kovalev, A. I. Panov, and E. Osipov. 2020. Hyperdimensional representations in semiotic approach to AGI. In *Proceedings of the Artificial General Intelligence (AGI'20) (Lecture Notes in Computer Science, Vol. 12177)*. 231–241.

[232] A. K. Kovalev, M. Shaban, A. A. Chuganskaya, and A. I. Panov. 2021. Applying vector symbolic architecture and semiotic approach to visual dialog. In *Proceedings of the International Conference on Hybrid Artificial Intelligence Systems (HAIS'21)*. 243–255.

[233] A. K. Kovalev, M. Shaban, E. Osipov, and A. I. Panov. 2021. Vector semiotic model for visual question answering. *Cogn. Syst. Res.* 71 (2021), 52–63.

[234] E. M. Kussul. 1992. *Associative Neuron-like Structures.* Naukova Dumka. [in Russian]

[235] E. M. Kussul and T. N. Baidyk. 2004. Improved method of handwritten digit recognition tested on MNIST database. *Image Vis. Comput.* 22, 12 (2004), 971–981.

[236] E. M. Kussul, T. N. Baidyk, V. V. Lukovich, and D. A. Rachkovskij. 1993. Adaptive neural network classifier with multifloat input coding. In *Proceedings of the International Conference on Neural Networks and Their Applications (NEURO'93)*. 209–216.

[237] E. M. Kussul, T. N. Baidyk, V. V. Lukovich, and D. A. Rachkovskij. 1994. Adaptive high performance classifier based on random threshold neurons. In *Proceedings of the European Meeting on Cybernetics and Systems (EMCSR'94)*. 1687–1694.

[238] E. M. Kussul, T. N. Baidyk, and D. C. Wunsch. 2010. *Neural Networks and Micromechanics*. Springer.

[239] E. M. Kussul, T. N. Baidyk, D. C. Wunsch, O. Makeyev, and A. Martin. 2006. Permutation coding technique for image recognition system. *IEEE Trans. Neural Netw.* 17, 6 (2006), 1566–1579.

[240] E. M. Kussul, L. M. Kasatkina, D. A. Rachkovskij, and D. C. Wunsch. 1998. Application of random threshold neural networks for diagnostics of micro machine tool condition. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'98)*, Vol. 1. 241–244.

[241] E. M. Kussul and D. A. Rachkovskij. 1991. Multilevel assembly neural architecture and processing of sequences. In *Neurocomputers and Attention: Connectionism and Neurocomputers*, Vol. 2. 577–590.

[242] E. M. Kussul, D. A. Rachkovskij, and T. N. Baidyk. 1991. Associative-projective neural networks: Architecture, implementation, applications. In *Proceedings of the International Conference on Neural Networks and Their Applications (NEURO'91)*. 463–476.

[243] E. M. Kussul, D. A. Rachkovskij, and T. N. Baidyk. 1991. On image texture recognition by associative-projective neurocomputer. In *Proceedings of the Intelligent Engineering Systems through Artificial Neural Networks (ANNIE'91)*. 453–458.

[244] V. Kvasnička and J. Pospíchal. 2006. Deductive rules in holographic reduced representation. *Neurocomputing* 69, 16–18 (2006), 2127–2139.

[245] M. Laiho, J. H. Poikonen, P. Kanerva, and E. Lehtonen. 2015. High-dimensional computing with sparse vectors. In *Proceedings of the IEEE Biomedical Circuits and Systems Conference (BioCAS'15)*. 1–4.

[246] T. K. Landauer and S. T. Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction, and representation of knowledge. *Psychol. Rev.* 104, 2 (1997), 211–240.

[247] Y. LeCun, Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature* 521 (2015), 436–444.

[248] O. Levy, Y. Goldberg, and I. Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Trans. Assoc. Comput. Ling.* 3 (2015), 211–225.

[249] S. Levy, S. Bajracharya, and R. W. Gayler. 2013. Learning behavior hierarchies via high-dimensional sensor projection. In *Proceedings of the AAAI Conference on Learning Rich Representations from Low-Level Sensors*. 25–27.

[250] S. D. Levy and R. W. Gayler. 2009. 'Lateral inhibition' in a fully distributed connectionist architecture. In *Proceedings of the International Conference on Cognitive Modeling (ICCM'09)*. 1–6.

[251] S. D. Levy, C. Lowney, W. Meroney, and R. W. Gayler. 2014. Bracketing the beetle: How Wittgenstein's understanding of language can guide our practice in AGI and cognitive science. In *Proceedings of the Artificial General Intelligence (AGI'14) (Lecture Notes in Computer Science, Vol. 8598)*. 73–84.

[252] N. Littlestone. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Mach. Learn.* 2, 4 (1988), 285–318.

[253] A. Lovett, K. D. Forbus, and J. Usher. 2010. A structure-mapping model of Raven's progressive matrices. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'10)*, Vol. 32. 2761–2766.

[254] T. Lu, A. R. Voelker, B. Komer, and C. Eliasmith. 2019. Representing spatial relations with fractional binding. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'19)*. 2214–2220.

[255] M. Lukosevicius and H. Jaeger. 2009. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* 3, 3 (2009), 127–149.

[256] K. Lund and C. Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behav. Res. Methods Instrum. Comput.* 28, 2 (1996), 203–208.

[257] J. Luo, Y. Cao, and Y. Wu. 2018. Towards decomposed linguistic representation with holographic reduced representation. OpenReview Preprint.

[258] D. Ma, R. Thapa, and X. Jiao. 2021. MoleHD: Ultra-low-cost drug discovery using hyperdimensional computing. arXiv:2106.02894. Retrieved from https://arxiv.org/abs/2106.02894.

[259] Y. Ma, M. Hildebrandt, V. Tresp, and S. Baier. 2018. Holistic representations for memorization and inference. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI'18)*. 1–11.

[260] Y. Ma and B. Ma. 2022. Multimodal sentiment analysis on unaligned sequences via holographic embedding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'22)*. 8547–8551.

[261] A. X. Manabat, C. R. Marcelo, A. L. Quinquito, and A. Alvarez. 2019. Performance analysis of hyperdimensional computing for character recognition. In *Proceedings of the International Symposium on Multimedia and Communication Technology (ISMAC'19)*. 1–5.

[262] G. Marcus. 2020. The next decade in AI: Four steps towards robust artificial intelligence. arXiv:2002.06177. Retrieved from https://arxiv.org/abs/2002.01677.

[263] N. Maudgalya, B. A. Olshausen, and S. J. Kent. 2020. Vector symbolic visual analogies. In *Proceedings of the AAAI Symposium on Conceptual Abstraction and Analogy in Natural and Artificial Intelligence*.

[264] R. T. McCoy, T. Linzen, E. Dunbar, and P. Smolensky. 2019. RNNs implicitly implement tensor-product representations. In *Proceedings of the International Conference on Learning Representations (ICLR'19)*. 1–22.

[265] N. McDonald, R. Davis, L. Loomis, and J. Kopra. 2021. Aspects of hyperdimensional computing for robotics: Transfer learning, cloning, extraneous sensors, and network topology. In *Disruptive Technologies in Information Sciences.* 1–14.

[266] M. Mendes, A. P. Coimbra, and M. M. Crisostomo. 2012. Robot navigation based on view sequences stored in a sparse distributed memory. *Robotica* 30, 4 (2012), 571–581.

[267] M. Mendes, M. M. Crisostomo, and A. P. Coimbra. 2008. Robot navigation using a sparse distributed memory. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'08).* 53–58.

[268] A. Menon, A. Natarajan, R. Agashe, D. Sun, M. Aristio, H. Liew, Y. S. Shao, and J. M. Rabaey. 2022. Efficient emotion recognition using hyperdimensional computing with combinatorial channel encoding and cellular automata. *Brain Inf.* 9 (2022), 1–13.

[269] A. Menon, A. Natarajan, L. I. Galindez Olascoaga, Y. Kim, B. Benedict, and J. M. Rabaey. 2022. On the role of hyperdimensional computing for behavioral prioritization in reactive robot navigation tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'22).* 1–7.

[270] A. Menon, D. Sun, M. Aristio, H. Liew, K. Lee, and J. M. Rabaey. 2021. A highly energy-efficient hyperdimensional computing processor for wearable multi-modal classification. In *Proceedings of the IEEE Biomedical Circuits and Systems Conference (BioCAS'21).* 1–4.

[271] J. I. Q. Mercado, R. B. Fernandez, and M. A. R. Salinas. 2020. Semantic similarity estimation using vector symbolic architectures. *IEEE Access* 8 (2020), 109120–109132.

[272] C. Mercier, H. Chateau-Laurent, F. Alexandre, and T. Viéville. 2021. Ontology as neuronal-space manifold: Towards symbolic and numerical artificial embedding. In *Proceedings of the KRHCAI Workshop on Knowledge Representation for Hybrid & Compositional AI.* 1–11.

[273] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha. 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 6197 (2014), 668–673.

[274] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS'13).* 1–9.

[275] G. A. Miller and W. G. Charles. 1991. Contextual correlates of semantic similarity. *Lang. Cogn. Process.* 6, 1 (1991), 1–28.

[276] F. Mirus, P. Blouw, T. C. Stewart, and J. Conradt. 2019. An investigation of vehicle behavior prediction using a vector power representation to encode spatial positions of multiple objects and neural networks. *Front. Neurorobot.* 13 (2019), 1–17.

[277] F. Mirus, P. Blouw, T. C. Stewart, and J. Conradt. 2019. Predicting vehicle behaviour using LSTMs and a vector power representation for spatial positions. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN'19).* 113–118.

[278] F. Mirus, T. C. Stewart, and J. Conradt. 2018. Towards cognitive automotive environment modelling: Reasoning based on vector representations. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN'18).* 55–60.

[279] F. Mirus, T. C. Stewart, and J. Conradt. 2020. Detection of abnormal driving situations using distributed representations and unsupervised learning. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN'20).* 363–368.

[280] F. Mirus, T. C. Stewart, and J. Conradt. 2020. The importance of balanced data sets: Analyzing a vehicle trajectory prediction model based on neural networks and distributed representations. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'20).* 1–8.

[281] I. S. Misuno, D. A. Rachkovskij, and S. V. Slipchenko. 2005. Vector and distributed representations reflecting semantic relatedness of words. *Math. Mach. Syst.* 3 (2005), 50–66. [in Russian]

[282] I. S. Misuno, D. A. Rachkovskij, S. V. Slipchenko, and A. M. Sokolov. 2005. Searching for text information with the help of vector representations. *Probl. Program.* 4 (2005), 50–59. [in Russian]

[283] J. Mitchell and M. Lapata. 2010. Composition in distributional models of semantics. *Cogn. Sci.* 34, 8 (2010), 1388–1429.

[284] A. Mitrokhin, P. Sutor, C. Fermuller, and Y. Aloimonos. 2019. Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception. *Sci. Robot.* 4, 30 (2019), 1–10.

[285] A. Mitrokhin, P. Sutor, D. Summers-Stay, C. Fermuller, and Y. Aloimonos. 2020. Symbolic representation and learning with hyperdimensional computing. *Front. Robot. AI* (2020), 1–11.

[286] A. Moin, A. Zhou, S. Benatti, A. Rahimi, L. Benini, and J. M. Rabaey. 2019. Analysis of contraction effort level in EMG-based gesture recognition using hyperdimensional computing. In *Proceedings of the IEEE Biomedical Circuits and Systems Conference (BioCAS'19).* 1–4.

[287] A. Moin, A. Zhou, A. Rahimi, S. Benatti, A. Menon, Tamakloe S, J. Ting, N. Yamamoto, Y. Khan, F. Burghardt, L. Benini, A. C. Arias, and J. M. Rabaey. 2018. An EMG gesture recognition system with flexible high-density sensors and brain-inspired high-dimensional classifier. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'18).* 1–5.

[288] Ali Moin, Andy Zhou, Abbas Rahimi, Alisha Menon, Simone Benatti, George Alexandrov, Senam Tamakloe, Jonathan Ting, Natasha Yamamoto, Yasser Khan, F. Burghardt, L. Benini, A. C. Arias, and J. M. Rabaey. 2021. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. *Nat. Electr.* 4, 1 (2021), 54–63.

[289] F. Montagna, A. Rahimi, S. Benatti, D. Rossi, and L. Benini. 2018. PULP-HD: Accelerating brain-inspired high-dimensional computing on a parallel ultra-low power platform. In *Proceedings of the ACM/ESDA/IEEE Design Automation Conference (DAC'18)*. 1–6.

[290] G. Montone, J. K. O'Regan, and A. V. Terekhov. 2017. Hyper-dimensional computing for a visual question-answering system that is trainable end-to-end. arXiv:1711.10185. Retrieved from https://arxiv.org/abs/1711.10185.

[291] J. Morris, H. W. Lui, K. Stewart, B. Khaleghi, A. Thomas, T. Marback, B. Aksanli, E. Neftci, and T. Rosing. 2022. HyperSpike: HyperDimensional computing for more efficient and robust spiking neural networks. In *Proceedings of the Design, Automation and Test in Europe Conference (DATE'22)*. 664–669.

[292] B. B. Murdock. 1982. A theory for the storage and retrieval of item and associative information. *Psychol. Rev.* 89, 6 (1982), 609–626.

[293] F. R. Najafabadi, A. Rahimi, P. Kanerva, and J. M. Rabaey. 2016. Hyperdimensional computing for text classification. In *Proceedings of the Design, Automation and Test in Europe Conference (DATE'16)*.

[294] M. Nazemi, A. Esmaili, A. Fayyazi, and M. Pedram. 2020. SynergicLearning: Neural network-based feature extraction for highly-accurate hyperdimensional learning. In *Proceedings of the IEEE/ACM International Conference On Computer Aided Design (ICCAD'20)*. 1–9.

[295] P. Neubert and P. Protzel. 2018. Towards hypervector representations for learning and planning with schemas. In *Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz'18) (Lecture Notes in Computer Science, Vol. 11117)*. 182–189.

[296] P. Neubert and S. Schubert. 2021. Hyperdimensional computing as a framework for systematic aggregation of image descriptors. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 16938–16947.

[297] P. Neubert, S. Schubert, and P. Protzel. 2016. Learning vector symbolic architectures for reactive robot behaviours. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'16)*. 1–3.

[298] P. Neubert, S. Schubert, and P. Protzel. 2019. An introduction to hyperdimensional computing for robotics. *Künstl. Intell.* 33, 4 (2019), 319–330.

[299] P. Neubert, S. Schubert, K. Schlegel, and P. Protzel. 2021. Vector semantic representations as descriptors for visual place recognition. In *Proceedings of the Robotics: Science and Systems (RSS'21)*. 1–11.

[300] J. Neumann. 2000. Learning holistic transformation of HRR from examples. In *Proceedings of the International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies (KES'00)*. 557–560.

[301] J. Neumann. 2002. Learning the systematic transformation of holographic reduced representations. *Cogn. Syst. Res.* 3, 2 (2002), 227–235.

[302] I. Nunes, M. Heddes, T. Givargis, A. Nicolau, and A. Veidenbaum. 2022. GraphHD: Efficient graph classification using hyperdimensional computing. In *Proceedings of the Design, Automation and Test in Europe Conference (DATE'22)*. 1485–1490.

[303] L. I. Galindez Olascoaga, A. Menon, M. Ibrahim, and J. M. Rabaey. 2022. A brain-inspired hierarchical reasoning framework for cognition-augmented prosthetic grasping. In *Proceedings of the AAAI Workshop on Combining Learning and Reasoning*. 1–9.

[304] W. Olin-Ammentorp and M. Bazhenov. 2021. Bridge networks: Relating inputs through vector-symbolic manipulations. In *Proceedings of the International Conference on Neuromorphic Systems (ICONS'21)*. 1–6.

[305] A. Ororbia and M. A. Kelly. 2022. CogNGen: Constructing the kernel of a hyperdimensional predictive processing cognitive architecture. arXiv:2204.00619. Retrieved from https://arxiv.org/abs/2204.00619.

[306] E. Osipov, S. Kahawala, D. Haputhanthri, T. Kempitiya, D. De Silva, D. Alahakoon, and D. Kleyko. 2021. HyperSeed: Unsupervised learning with vector symbolic architectures. To appear in IEEE Trans. Neural Netw. 2022.

[307] E. Osipov, D. Kleyko, and A. Legalov. 2017. Associative synthesis of finite state automata model of a controlled object with hyperdimensional computing. In *Proceedings of the Annual Conference of the IEEE Industrial Electronics Society (IECON'17)*. 3276–3281.

[308] D. E. Padilla and M. D. McDonnell. 2014. A neurobiologically plausible vector symbolic architecture. In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC'14)*. 242–245.

[309] U. Pale, T. Teijeiro, and D. Atienza. 2021. Systematic assessment of hyperdimensional computing for epileptic seizure detection. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC'21)*. 6361–6367.

[310] U. Pale, T. Teijeiro, and D. Atienza. 2022. Hyperdimensional computing encoding for feature selection on the use case of epileptic seizure detection. arXiv:2205.07654. Retrieved from https://arxiv.org/abs/2205.07654.

[311] U. Pale, T. Teijeiro, and D. Atienza. 2022. Multi-centroid hyperdimensional computing approach for epileptic seizure detection. *Front. Neurol.* 13 (2022), 1–13.

[312] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. 2000. Latent Semantic Indexing: A probabilistic analysis. *J. Comput. Syst. Sci.* 61, 2 (2000), 217–235.

[313] N. Papakonstantinou, S. Proper, B. O'Halloran, and I. Y. Tumer. 2014. Simulation based machine learning for fault detection in complex systems using the functional failure identification and propagation framework. In *Proceedings of the ASME Computers and Information in Engineering Conference (CIE'14)*, Vol. 1B. 1–10.

[314] V. Papyan, X. Y. Han, and D. L. Donoho. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proc. Natl. Acad. Sci.* 117, 40 (2020), 24652–24663.

[315] D. V. Pashchenko, D. A. Trokoz, A. I. Martyshkin, M. P. Sinev, and B. L. Svistunov. 2020. Search for a substring of characters using the theory of non-deterministic finite automata and vector-character architecture. *Bull. Electr. Eng. Inf.* 9, 3 (2020), 1238–1250.

[316] A. Paullada, B. Percha, and T. Cohen. 2020. Improving biomedical analogical retrieval with embedding of structural dependencies. In *Proceedings of the SIGBioMed Workshop on Biomedical Language Processing (BioNLP'20)*. 38–48.

[317] M. Pelillo. 1999. Replicator equations, maximal cliques, and graph isomorphism. *Neural Comput.* 11, 8 (1999), 1933–1955.

[318] J. Pennington, R. Socher, and C. D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1532–1543.

[319] T. A. Plate. 1994. *Distributed Representations and Nested Compositional Structure*. Ph.D. Thesis. University of Toronto.

[320] T. A. Plate. 1994. Estimating analogical similarity by dot-products of holographic reduced representations. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS'94)*. 1109–1116.

[321] T. A. Plate. 1995. Holographic reduced representations. *IEEE Trans. Neural Netw.* 6, 3 (1995), 623–641.

[322] T. A. Plate. 1997. A common framework for distributed representation schemes for compositional structure. In *Proceedings of the Connectionist Systems for Knowledge Representation and Deduction*. 15–34.

[323] T. A. Plate. 1997. Structure matching and transformation with distributed representations. In *Connectionist-Symbolic Integration*. 1–19.

[324] T. A. Plate. 2000. Analogy retrieval and processing with distributed vector representations. *Int. J. Knowl. Eng. Neural Netw.* 17, 1 (2000), 29–40.

[325] T. A. Plate. 2000. Randomly connected sigma-pi neurons can form associative memories. *Comput. Neural Syst.* 11, 4 (2000), 321–332.

[326] T. A. Plate. 2003. *Holographic Reduced Representations: Distributed Representation for Cognitive Structures*. Center for the Study of Language and Information (CSLI), Stanford, CA.

[327] P. Poduval, Z. Zou, H. Najafi, H. Homayoun, and M. Imani. 2021. StocHD: Stochastic hyperdimensional system for efficient and robust learning from raw data. In *Proceedings of the ACM/ESDA/IEEE Design Automation Conference (DAC'21)*. 1–6.

[328] M. O. Rabin and D. Scott. 1959. Finite automata and their decision problems. *IBM J. Res. Dev.* 3, 2 (1959), 114–125.

[329] D. A. Rachkovskij. 1996. Application of stochastic assembly neural networks in the problem of interesting text selection. *Neural Netw. Syst. Inf. Proc.* (1996), 52–64. [in Russian]

[330] D. A. Rachkovskij. 2001. Representation and processing of structures with binary sparse distributed codes. *IEEE Trans. Knowl. Data Eng.* 3, 2 (2001), 261–276.

[331] D. A. Rachkovskij. 2004. Some approaches to analogical mapping with structure sensitive distributed representations. *J. Exp. Theor. Artif. Intell.* 16, 3 (2004), 125–145.

[332] D. A. Rachkovskij. 2007. Linear classifiers based on binary distributed representations. *Inf. Theor. Appl.* 14, 3 (2007), 270–274.

[333] D. A. Rachkovskij. 2021. Shift-equivariant similarity-preserving hypervector representations of sequences. arXiv:2112.15475. Retrieved from https://arxiv.org/abs/2112.15475.

[334] D. A. Rachkovskij and T. V. Fedoseyeva. 1990. On audio signals recognition by multilevel neural network. In *Proceedings of the International Symposium on Neural Networks and Neural Computing (NEURONET'90)*. 281–283.

[335] D. A. Rachkovskij and D. Kleyko. 2022. Recursive binding for similarity-preserving hypervector representations of sequences. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'22)*. 1–8.

[336] D. A. Rachkovskij and E. M. Kussul. 1998. DataGen: A generator of datasets for evaluation of classification algorithms. *Pattern Recogn. Lett.* 19, 7 (1998), 537–544.

[337] D. A. Rachkovskij, E. M. Kussul, and T. N. Baidyk. 2013. Building a world model with structure-sensitive sparse binary distributed representations. *Biol. Insp. Cogn. Arch.* 3 (2013), 64–86.

[338] D. A. Rachkovskij and S. V. Slipchenko. 2012. Similarity-based retrieval with structure-sensitive sparse binary distributed representations. *Comput. Intell.* 28, 1 (2012), 106–129.

[339] D. A. Rachkovskij, S. V. Slipchenko, E. M. Kussul, and T. N. Baidyk. 2005. Sparse binary distributed encoding of scalars. *J. Autom. Inf. Sci.* 37, 6 (2005), 12–23.

[340] D. A. Rachkovskij, S. V. Slipchenko, and I. S. Misuno. 2010. Intelligent processing of proteomics data to predict glioma sensitivity to chemotherapy. *Cybernet. Comput.* 161 (2010), 90–105. [in Russian]

[341] D. A. Rachkovskij, S. V. Slipchenko, I. S. Misuno, E. M. Kussul, and T. N. Baidyk. 2005. Sparse binary distributed encoding of numeric vectors. *J. Autom. Inf. Sci.* 37, 11 (2005), 47–61.

[342] A. Rahimi, S. Benatti, P. Kanerva, L. Benini, and J. M. Rabaey. 2016. Hyperdimensional biosignal processing: A case study for EMG-based hand gesture recognition. In *Proceedings of the IEEE International Conference on Rebooting Computing (ICRC'16).* 1–8.

[343] A. Rahimi, S. Datta, D. Kleyko, E. P. Frady, B. Olshausen, P. Kanerva, and J. M. Rabaey. 2017. High-dimensional computing as a nanoscalable paradigm. *IEEE Trans. Circ. Syst. I: Regul. Pap.* 64, 9 (2017), 2508–2521.

[344] A. Rahimi, P. Kanerva, L. Benini, and J. M. Rabaey. 2019. Efficient biosignal processing using hyperdimensional computing: Network templates for combined learning and classification of ExG signals. *Proc. IEEE* 107, 1 (2019), 123–143.

[345] A. Rahimi, P. Kanerva, J. del R. Millan, and J. M. Rabaey. 2017. Hyperdimensional computing for noninvasive brain-computer interfaces: Blind and one-shot classification of EEG error-related potentials. In *Proceedings of the EAI International Conference on Bio-inspired Information and Communications Technologies (BICT'17).* 19–26.

[346] A. Rahimi, P. Kanerva, and J. M. Rabaey. 2016. A robust and energy efficient classifier using brain-inspired hyperdimensional computing. In *Proceedings of the IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED'16).* 64–69.

[347] A. Rahimi and B. Recht. 2007. Random features for large-scale kernel machines. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS'07)*, Vol. 20. 1–8.

[348] A. Rahimi, A. Tchouprina, P. Kanerva, J. Millán, and J. M. Rabaey. 2017. Hyperdimensional computing for blind and one-shot classification of EEG error-related potentials. *Mobile Netw. Appl.* (2017), 1–12.

[349] M. Rahimzadeh, A. Attar, and S. M. Sakhaei. 2021. A fully automated deep learning-based network for detecting COVID-19 from a new and large lung CT scan dataset. *Biomed. Sign. Process. Contr.* 68 (2021), 1–14.

[350] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. 2021. Hopfield networks is all you need. In *Proceedings of the International Conference on Machine Learning (ICML'21).* 1–95.

[351] O. Räsänen. 2015. Generating hyperdimensional distributed representations from continuous valued multivariate sensory input. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'15).* 1943–1948.

[352] O. Räsänen and S. Kakouros. 2014. Modeling dependencies in multiple parallel data streams with hyperdimensional computing. *IEEE Sign. Process. Lett.* 21, 7 (2014), 899–903.

[353] O. Räsänen and J. Saarinen. 2016. Sequence prediction with sparse distributed hyperdimensional coding applied to the analysis of mobile phone use patterns. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 9 (2016), 1878–1889.

[354] D. Rasmussen and C. Eliasmith. 2011. A neural model of rule generation in inductive reasoning. *Top. Cogn. Sci.* 3, 1 (2011), 140–153.

[355] D. Rasmussen and C. Eliasmith. 2014. A spiking neural model applied to the study of human performance and cognitive decline on raven's advanced progressive matrices. *Intelligence* 42 (2014), 53–82.

[356] J. Raven, J. C. Raven, and J. H. Court. 2000. *Manual for Raven's Progressive Matrices and Vocabulary Scales.* Oxford Psychologists Press.

[357] G. Recchia, M. N. Jones, M. Sahlgren, and P. Kanerva. 2010. Encoding sequential information in vector space models of semantics: Comparing holographic reduced representation and random permutation. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'10).* 865–870.

[358] G. Recchia, M. Sahlgren, P. Kanerva, and M. N. Jones. 2015. Encoding sequential information in semantic space models: Comparing holographic reduced representation and random permutation. *Comput. Intell. Neurosci.* (2015), 1–18.

[359] S. Reimann. 2021. The algebra of cognitive states: Towards modelling the serial position curve. In *Proceedings of the International Conference on Cognitive Modeling (ICCM'21).* 1–7.

[360] P. Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'95).* 448–453.

[361] A. Rosato, M. Panella, and D. Kleyko. 2021. Hyperdimensional computing for efficient distributed classification with randomized neural networks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'21).* 1–10.

[362] A. Rosato, M. Panella, E. Osipov, and D. Kleyko. 2021. On effects of compression with hyperdimensional computing in distributed randomized neural networks. In *Proceedings of the International Work-Conference on Artificial Neural Networks (IWANN'21) (Lecture Notes in Computer Science, Vol. 12862).* 155–167.

[363] B. H. Ross. 1989. Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. *J. Exp. Psychol.: Learn. Mem. Cogn.* 15, 3 (1989), 456–468.

[364] H. Rubenstein and J. B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM* 8, 10 (1965), 627–633.

[365] M. Sahlgren. 2001. Vector-based semantic analysis: Representing word meanings based on random labels. In *Proceedings of the ESSLI Workshop on Semantic Knowledge Acquisition and Categorization.* 1–21.

[366] M. Sahlgren. 2005. An introduction to random indexing. In *Proceedings of the International Conference on Terminology and Knowledge Engineering (TKE'05).* 1–9.

[367] M. Sahlgren, A. Holst, and P. Kanerva. 2008. Permutations as a means to encode order in word space. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'08).* 1300–1305.

[368] F. Sandin, B. Emruli, and M. Sahlgren. 2017. Random indexing of multidimensional data. *Knowl. Inf. Syst.* 52 (2017), 267–290.

[369] S. Sangali, M. Hersche, L. Benini, and A. Rahimi. 2020. Evolvable hyperdimensional computing: Unsupervised regeneration of associative memory to recover faulty components. In *Proceedings of the IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS'20).* 281–285.

[370] A. Sato and K. Yamada. 1996. Generalized learning vector quantization. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS'96).* 423–429.

[371] S. Scardapane and D. Wang. 2017. Randomness in neural networks: An overview. *Data Min. Knowl. Discov.* 7 (2017), 1–18.

[372] K. Schindler and A. Rahimi. 2021. A primer on hyperdimensional computing for iEEG seizure detection. *Front. Neurol.* 12 (2021), 1–12.

[373] K. Schlegel, F. Mirus, P. Neubert, and P. Protzel. 2021. Multivariate time series analysis for driving style classification using neural networks and hyperdimensional computing. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV'21).* 602–609.

[374] K. Schlegel, P. Neubert, and P. Protzel. 2022. A comparison of vector symbolic architectures. *Artif. Intell. Rev.* 55 (2022), 4523–4555.

[375] K. Schlegel, P. Neubert, and P. Protzel. 2022. HDC-MiniROCKET: Explicit time encoding in time series classification with hyperdimensional computing. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'22).* 1–8.

[376] H. R. Schmidtke. 2021. Reasoning and learning with context logic. *J. Reliab. Intell. Environ.* 7, 2 (2021), 171–185.

[377] H. R. Schmidtke and S. Coelho. 2022. Scales and hedges in a logic with analogous semantics. In *Proceedings of the Annual Conference on Advances in Cognitive Systems (ACS'22).* 1–20.

[378] T. M. Schubert, T. Cohen, and S. Fischer-Baum. 2020. Reading the written language environment: Learning orthographic structure from statistical regularities. *J. Mem. Lang.* 114 (2020), 1–12.

[379] R. W. Schvaneveldt, D. E. Meyer, and C. A. Becker. 1976. Lexical ambiguity, semantic context, and visual word recognition. *J. Exp. Psychol.: Hum. Percept. Perf.* 2, 2 (1976), 243–256.

[380] T. Shahroodi, M. Zahedi, C. Firtina, M. Alser, S. Wong, O. Mutlu, and S. Hamdioui. 2022. Demeter: A fast and energy-efficient food profiler using hyperdimensional computing in memory. arXiv:2206.01932. Retrieved from https://arxiv.org/abs/2206.01932.

[381] K. Shridhar, H. Jain, A. Agarwal, and D. Kleyko. 2020. End to end binarized neural networks for text classification. In *Proceedings of the Workshop on Simple and Efficient Natural Language Processing (SustaiNLP'20).* 29–34.

[382] C. Simpkin, I. Taylor, G. Bent, G. de Mel, and R. K. Ganti. 2018. A scalable vector symbolic architecture approach for decentralized workflows. In *Proceedings of the International Conference on Advanced Collaborative Networks, Systems and Applications (COLLA'18).* 1–7.

[383] C. Simpkin, I. Taylor, G. A. Bent, G. de Mel, S. Rallapalli, L. Ma, and M. Srivatsa. 2019. Constructing distributed time-critical applications using cognitive enabled services. *Fut. Gener. Comput. Syst.* 100 (2019), 70–85.

[384] C. Simpkin, I. Taylor, D. Harborne, G. Bent, A. Preece, and R. K. Ganti. 2020. Efficient orchestration of node-RED IoT workflows using a vector symbolic architecture. *Fut. Gener. Comput. Syst.* 111 (2020), 117–131.

[385] S. V. Slipchenko. 2005. Distributed representations for the processing of hierarchically structured numerical and symbolic information. *Syst. Technol.* 6 (2005), 134–141. [in Russian]

[386] S. V. Slipchenko and D. A. Rachkovskij. 2009. Analogical mapping using similarity of binary distributed representations. *Inf. Theor. Appl.* 16, 3 (2009), 269–290.

[387] P. Smolensky, R. T. McCoy, R. Fernandez, M. Goldrick, and J. Gao. 2022. Neurocompositional computing: From the central paradox of cognition to a new generation of AI systems. AI Magazine. 1–15. https://doi.org/10.1002/aaai.12065

[388] J. Snaider and S. Franklin. 2014. Modular composite representation. *Cogn. Comput.* 6 (2014), 510–527.

[389] J. Snaider and S. Franklin. 2014. Vector LIDA. *Proc. Comput. Sci.* 41 (2014), 188–203.

[390] J. Snaider, S. Franklin, S. Strain, and E. O. George. 2013. Integer sparse distributed memory: Analysis and results. *Neural Netw.* 46 (2013), 144–153.

[391] T. C. Stewart, X. Choo, and C. Eliasmith. 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 (CogSci'10).* 1100–1105.

[392] T. C. Stewart, X. Choo, and C. Eliasmith. 2014. Sentence processing in spiking neurons: A biologically plausible left-corner parser. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'14)*. 1533–1538.

[393] T. C. Stewart and C. Eliasmith. 2011. Neural cognitive modelling: A biologically constrained spiking neuron model of the tower of Hanoi task. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'11)*. 656–661.

[394] T. C. Stewart and C. Eliasmith. 2013. Parsing sequentially presented commands in a large-scale biologically realistic brain model. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'13)*. 3460–3467.

[395] E. Strubell, A. Ganesh, and A. McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'19)*. 3645–3650.

[396] D. Summers-Stay. 2019. Propositional deductive inference by semantic vectors. In *Proceedings of the Intelligent Systems and Applications (IntelliSys'19) (Advances in Intelligent Systems and Computing, Vol. 1037)*. 810–820.

[397] P. Sutor, D. Summers-Stay, and Y. Aloimonos. 2018. A computational theory for life-long learning of semantics. In *Proceedings of the International Conference on Artificial General Intelligence (AGI'18)*. 217–226.

[398] P. Sutor, D. Yuan, D. Summers-Stay, C. Fermuller, and Y. Aloimonos. 2022. Gluing neural networks symbolically through hyperdimensional computing. arXiv:2205.15534. Retrieved from https://arxiv.org/abs/2205.15534.

[399] V. Taler, B. T. Johns, and M. N. Jones. 2020. A large scale semantic analysis of verbal fluency across the aging spectrum: Data from the Canadian longitudinal study on aging. *J. Gerontol.: Psychol. Sci.* 75, 9 (2020), 221–230.

[400] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz. 2012. Theory and practice of Bloom filters for distributed systems. *IEEE Commun. Surv. Tutor.* 14, 1 (2012), 131–155.

[401] P. Thagard, K. J. Holyoak, G. Nelson, and D. Gochfeld. 1990. Analog retrieval by constraint satisfaction. *Artif. Intell.* 46, 3 (1990), 259–310.

[402] R. Thapa, B. Lamichhane, D. Ma, and X. Jiao. 2021. SpamHD: Memory-efficient text spam detection using brain-inspired hyperdimensional computing. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'21)*. 84–89.

[403] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy. 2021. MLP-Mixer: An all-MLP architecture for vision. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS'21)*. 1–12.

[404] J. B. Travnik and P. M. Pilarski. 2017. Representing high-dimensional data to intelligent prostheses and other wearable assistive robots: A first comparison of tile coding and selective Kanerva coding. In *Proceedings of the International Conference on Rehabilitation Robotics (ICORR'17)*. 1443–1450.

[405] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS'17)*. 1–11.

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

[407] I. Vougioukas, A. Sandberg, and N. Nikoleris. 2021. Branch predicting with sparse distributed memories. arXiv:2110.09166. Retrieved from https://arxiv.org/abs/2110.09166.

[408] R. Wang, F. Kong, H. Sudler, and X. Jiao. 2021. HDAD: Hyperdimensional computing-based anomaly detection for automotive sensor attacks. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'21)*. 461–464.

[409] R. Wang, D. Ma, and X. Jiao. 2022. EnHDC: Ensemble learning for brain-inspired hyperdimensional computing. arXiv:2203.13542. Retrieved from https://arxiv.org/abs/2203.13542.

[410] N. D. Watkinson, T. Givargis, V. Joe, A. Nicolau, and A. Veidenbaum. 2021. Class-modeling of septic shock with hyperdimensional computing. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC'21)*. 1653–1659.

[411] N. D. Watkinson, V. Joe, T. Givargis, A. Nicolau, and A. Veidenbaum. 2021. Detecting COVID-19 related pneumonia on CT scans using hyperdimensional computing. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC'21)*. 3970–3973.

[412] E. Weiss, B. Cheung, and B. A. Olshausen. 2016. A neural architecture for representing and reasoning about spatial relationships. OpenReview Preprint.

[413] C. M. Wharton, K. J. Holyoak, P. E. Downing, T. E. Lange, T. D. Wickens, and E. R. Melz. 1994. Below the surface: Analogical similarity and retrieval competition in reminding. *Cogn. Psychol.* 26, 1 (1994), 64–101.

[414] D. Widdows. 2008. Semantic vector products: Some initial investigations. In *Proceedings of the AAAI Symposium on Quantum Interaction (AAAI'08)*. 1–8.

[415] D. Widdows and T. Cohen. 2010. The semantic vectors package: New algorithms and public tools for distributional semantics. In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC'10)*. 9–15.

[416] D. Widdows and T. Cohen. 2015. Graded semantic vectors: An approach to representing graded quantities in generalized quantum models. In *Proceedings of the International Symposium on Quantum Interaction (QI'15) (Lecture Notes in Computer Science, Vol. 9535)*. 231–244.

[417] D. Widdows and T. Cohen. 2015. Reasoning with vectors: A continuous model for fast robust inference. *Logic J. IGPL* 23, 2 (2015), 141–173.

[418] D. Widdows and K. Ferraro. 2008. Semantic vectors: A scalable open source package and online technology management application. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'08)*. 1183–1190.

[419] D. Widdows, K. Howell, and T. Cohen. 2021. Should semantic vector composition be explicit? Can it be linear? In *Proceedings of the Workshop on Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science (SemSpace'21)*. 1–12.

[420] S. Wilson, N. Sünderhauf, and F. Dayoub. 2021. Hyperdimensional feature fusion for out-of-distribution detection. arXiv:2110.00214. Retrieved from https://arxiv.org/abs/2110.00214.

[421] M. Wolff, G. Wirsching, M. Huber, P. beim Graben, R. Römer, and I. Schmitt. 2018. A fock space toolbox and some applications in computational cognition. In *Proceedings of the International Conference on Speech and Computer (SPECOM'18)*. 757–767.

[422] J. Wong. 2018. *Negative Capacitance and Hyperdimensional Computing for Unconventional Low-power Computing*. Ph.D. Thesis, University of California, Berkeley.

[423] S. K. M. Wong, W. Ziarko, V. V. Raghavan, and P. C. N. Wong. 1987. On modeling of information retrieval concepts in vector spaces. *ACM Trans. Datab. Syst.* 12, 2 (1987), 299–321.

[424] Y. Yao, W. Liu, G. Zhang, and W. Hu. 2022. Radar-based human activity recognition using hyperdimensional computing. *IEEE Trans. Microw. Theory Techn.* 70, 3 (2022), 1605–1619.

[425] T. Yerxa, A. G. Anderson, and E. Weiss. 2018. The hyperdimensional stack machine. In *Proceedings of the Cognitive Computing*. 1–2.

[426] O. Yilmaz. 2015. Analogy making and logical inference on images using cellular automata based hyperdimensional computing. In *Proceedings of the International Conference on Cognitive Computation: Integrating Neural and Symbolic Approaches (COCO'15)*, Vol. 1583. 19–27.

[427] O. Yilmaz. 2015. Machine learning using cellular automata based feature expansion and reservoir computing. *J. Cell. Automata* 10, 5-6 (2015), 435–472.

[428] O. Yilmaz. 2015. Symbolic computation using cellular automata-based hyperdimensional computing. *Neural Comput.* 27, 12 (2015), 2661–2692.

[429] T. Yu, Y. Zhang, Z. Zhang, and C. De Sa. 2022. Understanding hyperdimensional computing for parallel single-pass learning. arXiv:2202.04805. Retrieved from https://arxiv.org/abs/2202.04805.

[430] F. M. Zanzotto and L. Dell'Arciprete. 2012. Distributed tree kernels. In *Proceedings of the International Conference on Machine Learning (ICML'12)*. 1–8.

[431] M. Zeman, E. Osipov, and Z. Bosnic. 2021. Compressed superposition of neural networks for deep learning in edge computing. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'21)*. 1–8.

[432] S. Zhang, R. Wang, J. J. Zhang, A. Rahimi, and X. Jiao. 2021. Assessing robustness of hyperdimensional computing against errors in associative memory. In *Proceedings of the International Conference on Application-specific Systems, Architectures and Processors (ASAP'21)*. 211–217.

[433] A. Zhou, R. Muller, and J. M. Rabaey. 2021. Incremental learning in multiple limb positions for electromyography-based gesture recognition using hyperdimensional computing. TechRxiv. 1–10. Retrieved from http://dx.doi.org/10.36227/techrxiv.16643257.v1

[434] A. Zhou, R. Muller, and J. M. Rabaey. 2021. Memory-efficient, limb position-aware hand gesture recognition using hyperdimensional computing. In *Proceedings of the tinyML Research Symposium (tinyML)*. 1–8.

[435] M. Zitnik, M. Agrawal, and J. Leskovec. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34, 13 (2018), 457–466.

[436] Z. Zou, H. Alimohamadi, A. Zakeri, F. Imani, Y. Kim, M. H. Najafi, and M. Imani. 2022. Memory-inspired spiking hyperdimensional network for robust online learning. *Sci. Rep.* 12 (2022), 1–13.

[437] Z. Zou, Y. Kim, H. Najafi, and M. Imani. 2021. ManiHD: Efficient hyper-dimensional learning using manifold trainable encoder. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'21)*. 850–855.

[438] D. A. Rachkovskij. 2022. Representation of spatial objects by shift-equivariant similarity-preserving hypervectors. *Neural Computing and Applications* (2022), 1–17.

[439] A. Renner, Y. Sandamirskaya, F. T. Sommer, and E. P. Frady. 2022. Sparse vector binding on spiking neuromorphic hardware using synaptic delays. In *International Conference on Neuromorphic Systems (ICONS'22)*. 1–5.

[440] G. Bent, C. Simpkin, Y. Li, and A. Preece. 2022. Hyperdimensional Computing using Time-to-spike Neuromorphic Circuits. In *International Joint Conference on Neural Networks (IJCNN'22)*. 1–8.

Received 8 November 2021; revised 31 May 2022; accepted 3 June 2022