The transformative potential of vector symbolic architecture for cognitive processing at the network edge

Graham Bent[a], Cai Davies[b], Marc Roig Vilamala[b], Yuhua Li[b], Alun Preece[b], Alex Vicente Sola[c], Gaetano Di Caterina[c], Paul Kirkland[c], Benomy Tutcher[d], and Gavin Pearson[e]

[a]Neurosynapse Ltd, UK
[b]Security, Crime and Intelligence Institute, Cardiff University, UK
[c]Neuromorphic Sensor Signal Processing Lab, EEE Dept, University of Strathclyde, UK
[d]Frazer-Nash Consultancy, UK
[d]Defence Science and Technology Laboratory, UK

ABSTRACT

Vector Symbolic Architecture (VSA), a.k.a. Hyperdimensional Computing (HDC) has transformative potential for advancing cognitive processing capabilities at the network edge. This paper examines how this paradigm offers robust solutions for AI and Autonomy within a future command, control, communications, computers, cyber, intelligence, surveillance and reconnaissance (C5ISR) enterprise by effectively modelling the cognitive processes required to perform Observe, Orient, Decide and Act (OODA) loop processing. The paper summarises the theoretical underpinnings, operational efficiencies, and synergy between VSA and current AI methodologies, such as neural-symbolic integration and learning. It also addresses major research challenges and opportunities for future exploration, underscoring the potential for VSA to facilitate intelligent decision-making processes and maintain information superiority in complex environments. The paper intends to serve as a cornerstone for researchers and practitioners to harness the power of VSA in creating next-generation AI applications, especially in scenarios that demand rapid, adaptive, and autonomous responses.

Keywords: Vector symbolic architecture, hyperdimensional computing, edge intelligence, C5ISR, OODA

1. INTRODUCTION

Artificial intelligence (AI) and machine learning (ML) are rapidly advancing UK Defence's abilities to understand situations and act at a higher tempo than our adversaries. An essential component in all defence operations is the command, control, communications, computers, cyber, intelligence, surveillance and reconnaissance (C5ISR) function that involves collecting, processing, exploiting and disseminating accurate information in a timely manner to support decision making. C5ISR activities are complex and multifaceted; they currently require significant input from human operators demanding a combination of cognitive capabilities including attention, perception, memory, decision-making, problem-solving, and communication. Whilst human cognitive capabilities are difficult to replicate in automated systems, this is precisely the challenge for generation-after-next AI and Autonomy for Intelligence, Surveillance and Reconnaissance (A^2ISR), which seeks to leverage recent advances in AI and ML to release capacity and maximise understanding in the C5ISR enterprise to maintain information advantage.

To fully achieve these goals, future C5ISR systems will require the capability to automatically perform a necessary and sufficient set of 'brain-like' cognitive functions leveraging the latest developments in AI technology. In the context of C5ISR this requires the capability for future AI to autonomously perform the range of cognitive functions needed for the observe, orient, decide and act (OODA) loop processing cycle. To perform these functions faster than our adversaries, specifically in dynamic and contested multi-domain operations, it has been argued that OODA loop processing needs to be increasingly compressed and that future C5ISR will need to be performed closer to the network edge.[1] However, the mainstream approaches in current AI, based on very large artificial neural networks (ANNs), needs massive computational and data resources, meaning these technologies

Send correspondence to Alun Preece: PreeceAD@cardiff.ac.uk

are tied to data centres and heavily reliant on cloud hosting with stable network connectivity. To realise the full benefits of C5ISR at the network edge, there needs to be a shift towards new AI and ML paradigms that: 1) Can integrate a range of AI approaches (symbolic, sub-symbolic, and hybrid) and across a set of distributed nodes (i.e., in a resilient and distributed manner); 2) Are resource efficient in bandwidth and computation; 3) Are effective, benchmarked against state of art AI in relevant domains including learning, reasoning, and planning; 4) Are data-frugal and noise robust; and 5) Are future proof by being compatible with the range of emerging technologies (e.g., neuromorphic hardware, in-memory processing) that are designed specifically for lower size, weight, power and cost (SWaP-C).

In this paper, we describe how the brain-inspired *vector symbolic architecture* (VSA) a.k.a. *hyperdimensional computing* (HDC) paradigm[2,3] can achieve these requirements, and how VSA processing has transformative potential for advancing cognitive processing capabilities at the network edge. Moreover, in its uniform processing of symbolic and sub-symbolic information, combined with its ability to meet necessary and sufficient cognitive processing functions, VSA is uniquely positioned to meet the requirements within a dynamic and contested operational context, across the OODA loop.

Section 2 explains the basis of VSA computing and the cognitive processing functions required for C5ISR in the context of the OODA loop. Section 3, as the main technical core of the paper, describes how the VSA representation can be used as the 'glue' to integrate existing AI/ML components in distributed edge-of-network settings and, importantly, how VSA can potentially be used to perform C5ISR AI/ML functions in a single unified framework, removing silos between currently-disparate capabilities. The paper concludes in Sections 4 and 5 by considering opportunities and challenges for future exploitation.

2. VSA AND HUMAN COGNITIVE PROCESS

VSA presents a significant paradigm shift in AI and cognitive computing, characterised by its unique handling of complex data structures and its efficient computational models. In VSA, concepts are symbolized using a common representational format known as a *hypervector*: a very high dimensional vector. The fundamental idea that underpins VSA is that practically anything can be represented as a symbolic hypervector. Because of this common representation, hypervectors can in turn be combined to create new hypervectors that represent more complex concepts. This is the basis by which VSA integrates sub-symbolic and symbolic processing: by treating sub-symbolic vector representations as symbols that can be manipulated. While large, VSA hypervectors are often binary, making them highly compact.

2.1 Real-World Application of VSA: Enhanced C5ISR Systems

A practical application of VSA lies in enhancing C5ISR systems. VSA's proficiency in efficiently managing high-dimensional data and executing complex pattern recognition is ideal for the diverse information processed in C5ISR systems.

Consider surveillance drones as an example: VSA can amalgamate and analyse data from various sensors—visual, infrared, and radar—to form a comprehensive environmental understanding. The data from each can be represented as hypervectors and the various hypervector operations can provide the capability to integrate symbolic (structured data like coordinates) and sub-symbolic (unstructured data such as imagery) information aids in real-time decision-making and threat assessment. Furthermore, the scalability and flexibility of VSA ensure adaptability to evolving C5ISR systems, accommodating new sensors and data types. This adaptability is crucial in the rapidly advancing defence technology sector.

VSA thus offers a comprehensive toolkit for addressing complex, multi-faceted problems in AI, rendering it a potentially indispensable asset in contemporary engineering and computational research and development.

2.2 Human Cognition Process

Human cognition encompasses the wide array of mental processes involved in acquiring knowledge and understanding through experience, the senses and thought. It includes aspects such as perception, attention, memory, learning, reasoning and planning, and language and communication. There has been a considerable amount of research into ways of characterizing these different human cognitive functions as programmable representations. In this paper we consider an approach that builds on the fundamental concept within human cognition, that allows individuals to acquire, code, store, recall, and decode information about phenomena in their everyday spatial environment: this is known as a *cognitive map*. Cognitive maps enable humans to understand the relationships between things in their world, to make sense of these relationships leading to what is termed general intelligence. Originated by psychologist Edward C. Tolman,[4] the term originally referred to our mental representations of the layout of environments, such as towns or buildings. Cognitive maps also allow humans to do much more including decision-making, problem-solving, reasoning and planning a course of action.

The ability to form cognitive maps is evidence of the elaborate functionality within our brains. When we visit a new city, for instance, we create a mental picture of how streets interconnect, where landmarks are located, and how to move from one point to another. Over time, this mental map becomes more detailed and allows for more efficient navigation within the environment.

Cognitive maps are inherently adaptive; they can be updated with new information as an environment changes or as we learn more about a previously un-mapped domain. This adaptability signifies the higher-order cognitive capabilities for learning and memory consolidation. Current machine learning and deep learning suffer from the stability-plasticity dilemma, i.e., catastrophic forgetting of previously learned knowledge when attempting to absorb new information and intransigence of significantly resisting to incorporate new information while retaining old knowledge. VSA framework breaks the stability-plasticity dilemma due to its innate property of superposition to retain all learned information and adapt by absorbing new information, making it free from forgetting and flexible to adapt.

2.3 Cognitive Processing Required for C5ISR

Current C5ISR processing is essentially a human processing task that combines many different types of cognitive processes. Given that the aim of exploiting AI in C5ISR is to automate many of these cognitive human capabilities, we need to first define where these cognitive processes are required in the context of C5ISR. To address this question, we consider the typical C5ISR sequence of operations that are performed using the OODA loop model.[5] The OODA loop serves as a critical framework for decision-making and information processing, by encapsulating a sequence of steps that enable military forces to process information and respond effectively to threats.

Observe: This step involves the collection of real-time data from a diverse array of sensors and sources. The data can come from satellite imagery, human intelligence (HUMINT), signals intelligence (SIGINT), and other surveillance and reconnaissance systems. The goal at this stage is to gather a comprehensive picture of the operational environment, including enemy positions, movement, and potential vulnerabilities or threats. In terms of the cognitive processing, the observe step requires the capability to perform the functions of **perception** and **attention**. The latter has a key role to play in making effective use of limited computational resources.

Orient: Orientation is the analytical step where the collected data is synthesized and evaluated in the context of the operational environment, including cultural, geographical, and situational factors. During this step, military analysts are essentially interpreting the data, comparing it against existing knowledge, and integrating it with their understanding of the tactical and strategic picture. We argue that, from a cognition perspective, orientation is essentially the process of comparing the current observed world state with memorized cognitive maps. It is in this step that C5ISR personnel must consider the implications of new information and update their cognitive maps of the battlefield accordingly (i.e., learning). An important element of the orientation process is therefore the capability to compare the current cognitive maps with previous experience, so that some efficient method of storing (memory) and comparing cognitive maps is required. In terms of the cognitive processing, the orientation step therefore requires the functions of **memory** and **learning**.

Decide: The decision step involves formulating a course of action based on the analysis conducted during the orientation step. Decision-makers weigh the intelligence reports, consider the available options, evaluate potential risks and benefits, and determine the best strategy to pursue. The decision step is critical as it sets the stage for how forces will act and react to emerging situations. The reasoning and planning required in this step are complex cognitive function that usually require interpretation of the current cognitive map(s), to plan what are the required actions to achieve the desired mission goal(s), i.e. the course of action (COA). Ideally the COA should be a continuous decision-making process based on the current perceived state of the world, such that changes can be rapidly accounted for and new COA created in response. In terms of the cognitive processing, the decide step therefore requires the functions of **memory**, **learning**, **reasoning** and **planning**.

Act: Action is the final step of the OODA loop, i.e. it is the implementation of the chosen COA. In military C5ISR processing, this could manifest in various ways, such as repositioning forces, launching a strike, executing a defensive manoeuvre, or adjusting surveillance assets for better coverage. The goal in this step is to carry out decisions effectively and efficiently, to achieve the desired tactical or strategic objective. The result of performing the required actions needs to be continually assessed; any changes to the current cognitive map, and hence actions performed, will change the state of the perceived environment and used in the OODA loop to continuously refine the actions. In terms of the cognitive processing, the act step therefore needs to access all of the cognitive capabilities of the earlier steps. It is also the step that needs to communicate and co-ordinate decisions and actions, and will often make significant use the cognitive functions of **language** and **communication**, potentially in the context of limited communications bandwidth.

The OODA loop is therefore an essential concept within military C5ISR processing that facilitates rapid and effective decision-making in the face of uncertainty. By efficiently observing the battlefield, orienting through analysis, deciding on the best course of action, and acting to achieve mission objectives, military units can maintain situational awareness and operational superiority. The continuous and iterative nature of the OODA loop ensures that forces can adapt to evolving situations and maintain an edge in an ever-changing conflict environment. The rationale for deploying AI in C5ISR is to speed up the cycle using real-time data feeds, advanced analytics, artificial intelligence and machine learning. This is to enhance each step with an emphasis on performing these functions at the tactical edge of network environments with limited processing and communications capacity.

2.4 VSA Offers an Integrated Approach to Cognitive Processing

Achieving 'brain-like' cognitive processing potentially requires a computing framework that operates in a similar manner to the human brain. Traditional AI approaches include symbolic and connectionist methods.[6] The symbolic approach represents information via symbols and their relations, and solves problems or infers new knowledge through the processing of these symbols. In the alternative connectionist approach, information is processed in a network of simple interconnected computational units often called artificial neural networks.

Neuro-symbolic AI combines symbolic and connectionist AI to leverage the strengths of both. Neural networks excel at processing complex data like images and audio but struggle with explainability and manual adjustments. Symbolic AI, on the other hand, is easier to understand and fine-tune but can't handle complex data due to the overwhelming number of rules required. By combining these approaches, neuro-symbolic AI aims to harness complex data processing with explainability.

VSA, inspired by Eliasmith's work in *How to build a Brain*,[7] offers a potential solution by representing semantic concepts as high-dimensional vectors, enabling brain-like cognitive processing. This approach uses the Semantic Pointer Architecture (SPA), where vectors symbolically represent concepts. Current AI models, like large language models (LLMs), build symbolic vector spaces to perform tasks resembling human reasoning. However, LLMs lack VSA's consistent ability to process both symbolic and sub-symbolic information and are less resource-efficient.

The basic operations performed using VSA processing are shown schematically in Figure 1. They begin with hypervector generation followed by a process that binds and bundles hypervectors to create new hypervectors, which can subsequently unbound to recover the original vectors from a clean-up memory. Importantly, the new vectors semantically represent the vectors of which they are composed (i.e., vector of vectors, or a symbol of symbols) and that it is possible to efficiently deconstruct these vectors to obtain the component vectors.

In the context of military applications, VSA has been demonstrated as an efficient technique for discovering and re-tasking assets in bandwidth-constrained edge of network scenarios.[8–10] VSA has also been shown to be applicable to a wide range of functions that are directly applicable to C5ISR, such as signal processing,[11] robot control,[12] and sensor fusion.[13] Recent developments have demonstrated that VSA can also be used in the context of AI/ML, particularly in neuro-symbolic processing.[14]

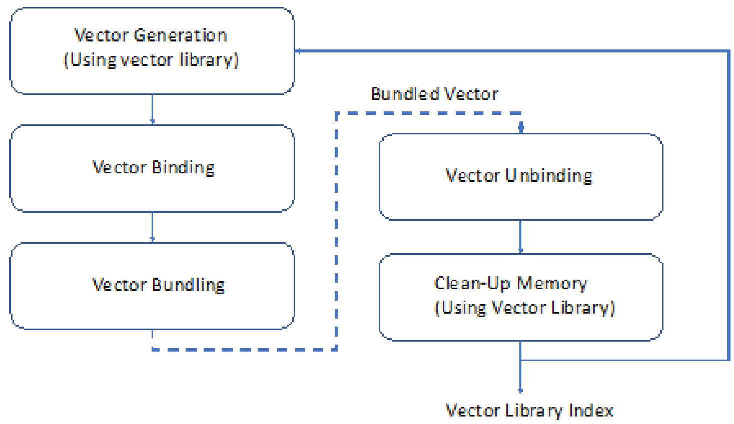

Figure 1. Schematic of the basic sequence steps in the VSA processing cycle which comprises: hypervector generation, binding, bundling, unbinding and clean-up memory.

3. LEVERAGING VSA FOR ENHANCED COGNITIVE CAPABILITIES IN C5ISR

The intent of the following four sub-sections is to demonstrate in technical detail how VSA can be effective in implementing the necessary and sufficient cognitive functions across the OODA loop, including perception and attention (observe and orient), memory (orient and decide), decision-making, problem-solving and communication (decide and act).

3.1 Observe: Hypervector Generation

In the observe step of the OODA loop the challenge is how to represent the wide range of sensors and sources of information in a form that can be readily processed by the subsequent steps. As noted earlier, the basic idea that underpins VSA is that practically anything can be represented as a symbolic vector in a high dimensional vector space. An essential property of such vectors is that when the vector dimension D is large, any two randomly generated hypervectors are orthogonal (i.e., vector dot product or hamming similarity is zero) with a very high probability. Similarly, when lower dimensional vectors are randomly projected into a higher dimensional vector space, they become quasi-orthogonal.[15] The orthogonality property ensures that, up to some limit determined by D, vectors can be bundled together (i.e., addition with some form of normalization). The resulting compound vector (or vector of vectors) has the same dimension D as the vectors of which it is composed, with the *holographic* property that the component vectors are distributed across the bundled vector and can be recovered with high probability using the clean-up memory. In the context of C5ISR this facilitates the construction of hypervectors that represent descriptions of things as diverse as imagery, time-varying signals, sensor characteristics and intelligence reports. Some of these are described in the following sections.

3.1.1 Encoding Using Random Hypervectors

A basic mechanism for generating VSA hypervectors is to begin with a set of random hypervectors that represent what we term atomic vectors (i.e., symbols). These vectors are stored in memory (i.e., the clean-up memory). To represent more complex things, the atomic vectors can be bound and bundled together using the VSA algebraic operators and these compound vectors of vectors can in turn be used to describe more complex things in a hierarchical representation. In the context of C5ISR, previous work has demonstrated that it is possible to represent complex document structures (e.g., an intelligence report), sensors and devices (e.g., sensor images, combat radios) and higher-level configurations of these assets (e.g., communications plans) as vectors of vectors.[16]

Storing these compound vectors in a distributed memory (usually locally with the entity that is being described by the vector) it has been demonstrated that it is possible to perform asset discovery and to reconfigure the assets into new configurations in edge of Tactical Communications and Information Systems (TacCIS) network settings by exchanging and locally interpreting the compound VSA vectors.[14,16]

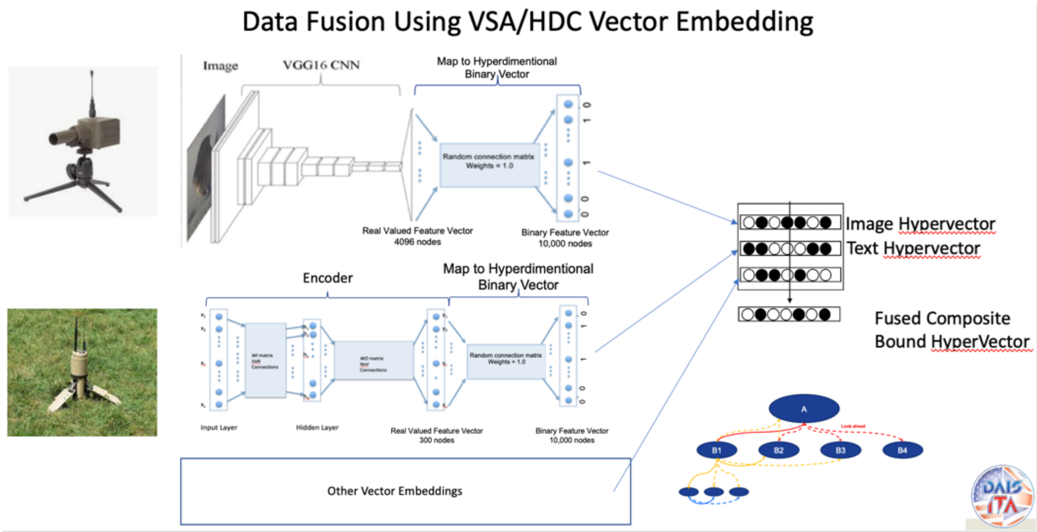

Figure 2. A schematic illustrating the data fusion process using VSA vector embedding, depicting the workflow for combining sensors and sources that are mapped into hypervectors using different types of ANN encoding.

3.1.2 Encoding Using Non-Random Hypervectors

An alternative to using random vectors that results in hypervectors with the same common encoded representation as the random hypervectors, is to build up a representation of an entity by using vectors that in some way semantically represent the structure of the entity. Artificial neural networks are essentially techniques for learning such real valued vector representations which can then be mapped into hypervectors. This is achieved firstly through a process called 'standardisation' where the vectors are transformed into pseudo-orthogonal vectors[17] and secondly using a 'random projection' process that creates vectors with the required dimension and preserves the holographic distributed representation properties of the resulting hypervectors.

One of the main advantages of this mapping is that real valued vectors of different dimension can be mapped into a common VSA representation which then facilitates the binding and bundling of vectors from different sources into a single semantic vector representation as a form of data fusion. An example is illustrated in Figure 2 showing the combining of data from a roadside camera, an unattended ground sensor and other sources, each processed using different ANNs. Similar fusion of information from open source information (OSINF) sources has recently been demonstrated, where the real valued vectors that represent pieces of OSINF are created using natural language transformer models and these are bundled with vectors that represent associated meta-data to facilitate search.[18] A further advantage of the VSA representation is that the resultant hypervector retains the holographic distributed representation of the component hypervectors. It was therefore possible to simply truncate the vectors for transmission over low bandwidth networks and still retain a semantic representation of the complex entity being represented.

3.1.3 Image Encoding

There are several approaches for encoding images as VSA hypervectors. The main approach has been to utilise the capabilities of convolutional neural networks (CNN) to produce vector representations of the input images. Mapping the vectors from the layers of CNN into a common hypervector representation enables new classifiers to be rapidly developed using one-shot or few shot learning i.e., an efficient form of transfer learning.[19] One additional benefit of using VSA to perform this type of transfer learning task is that it becomes possible to bundle vectors from all layers into a single vector and it has been shown that this significantly improves the classification performance of new classifiers and facilitates the capability to identify that a new image is an outlier that has not been previously seen during training of the CNN.[20]

Without a trained CNN it is also possible to use VSA to directly encode an image. An encoding scheme was presented in[12] that simply represented an image by encoding each pixel in the image by its X and Y co-ordinates and colour intensity as random hypervectors, and binding and bundling these vectors to create a composite image hypervector. This hypervector was in turn bound to a random hypervector that represented the class of the image to create an image classifier that did not require a large amount of training data. However, this approach did not perform well when the objects in the image were spatially translated or rotated. An alternative encoding approach has been demonstrated in papers by Neubert et al.[17] This approach uses a different encoding of position using random vectors and this does provide for some spatial translation and combines this technique that encodes parts of the image into a single hypervector. Importantly, the resulting VSA approach performs significantly better than other state of the art techniques for image matching and retrieval for visual place recognition and are particularly relevant to C5ISR processing.

The current state of the art using VSA uses a vector semantic representations (VSR) to construct hierarchical vector description of an image by encoding the image based on the spatial layout of its semantic entities.[21] This approach uses image segmentation to identify entities in the scene and then represents the spatial relationships between objects using vector binding. The resulting vectors are hierarchically bundled into a single VSR vector. The evaluation demonstrated that the place recognition performance is on par or better than the compared existing approaches. The approach does rely on the quality of the image segmentation but that the general concept is expected to be applicable to other tasks and to also benefit from future developments on extracting semantic (and other) information from images.

3.1.4 Spatio-temporal Signal Encoding

The processing of time varying signals and spatio-temporal signals using VSA has been largely dominated by the desire to classify these signals. The classical approach is described in[22] where the encoding of the time varying signal from biosignals using the standard VSA operations of binding and bundling but making specific use of the VSA permutation operator. The results demonstrated that electromyography (EMG) signals sampled at 500Hz from multiple sensors on the subjects arm could be used to successfully classify hand gesture recognition, electroencephalography (EEG) based signals could be used in the context of brain–computer interfaces, and that electrocorticography (ECoG) based signals could be analysed for seizure detection. In all cases the VSA implementation outperformed a standard support vector machine based classifier and training time was significantly faster.

A similar approach[23] has been used for voice recognition achieving comparable performance to alternative deep neural network approaches but with significantly faster training times (around $\times 5$). Further benefits of using VSA hypervectors has recently been demonstrated by IBM using their in-memory processing technology[11] to obtain a peak classification accuracy of 98.9% (within 0.04 percent of the baseline), while achieving 1.80–8.83$\times$ higher energy efficiency over a dedicated digital CMOS encoder and a 284$\times$ gain energy efficiency over an encoder running on a low-power general purpose computing platform.

VSA has also been used to classify driving style from a number of times series signals generated by vehicle sensors.[24] The paper reports that the VSA approach not only achieved similar or slightly better classification results than a range of other state of the art techniques, but also significantly reduced training time and increased data efficiency. In addition, the authors note that the proposed VSA model allows implementation in neuromorphic hardware based on spiking neural networks (SNNs).

One of the best existing methods for time series classification, in terms of accuracy and computation time, is MiniROCKET.[25] In a recent paper Schlegel et al.[26] have shown that the internal high-dimensional representation of MiniROCKET is well suited to be complemented by the algebra of VSA. The paper describes a VSA implementation called HDC-MiniROCKET, where the original algorithm is only a special case. HDC-MiniROCKET equalled or outperformed MiniROCKET on all of the 128 URC standard time series datasets.[27] Using VSA has therefore been demonstrated to offer significant benefits compared to existing state of the art for spatio-temporal signal encoding.

3.1.5 Encoding of other information sources

C5ISR requires not just the processing of information from sensors but also from intelligence reports and other information sources. This information is often reported as text. In previous work it has been demonstrated that text can be encoded from a basic set of atomic random hypervectors that can be as simple as representing the

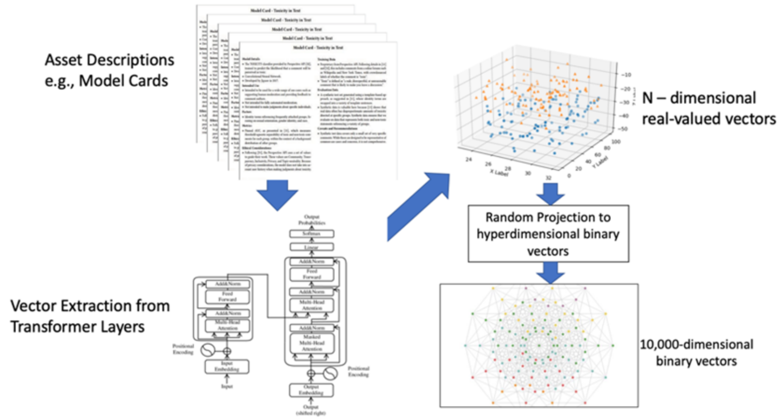

Figure 3. An example of semantic vector generation using transformer networks, such as large language models, showing the transition from asset descriptions to hyperdimensional binary vectors.

letters of an alphabet plus some random vectors to represent punctuation. Using a hierarchical VSA binding and bundling scheme it is possible to build vectors to represent words, sentences, paragraphs, etc, up to complete documents.[28]

An alternative to using random vectors to represent complex entities is the use of semantic vector representations that are learned using what are termed vector space embedding techniques. In a vector space embedding, real-world entities—words, images, sounds, and other data objects—can be mapped into numerical vectors in a way that reflects their semantic meaning or characteristics. The result is that entities with similar meanings or properties are represented by vectors that are closer together in the vector embedded space. In the case of text, the vector space embeddings are the cornerstone of the latest developments in AI, being most familiar in large language model applications such as ChatGPT for text and DALL-E for images. The vector representations within these applications are learned using hierarchical transformer neural networks where the different layers of the transformer represent different aspects of the input sequence of words or other symbols. Analogous to the way in which vectors from a CNN can be extracted and combined using VSA the same approach can be used for these transformer models, and this facilitates combining these vectors with other information (e.g., vectors representing meta data, images) without the need to retrain the LLM. This has recently been demonstrated as illustrated in Figure 3 as a highly efficient approach to text search.

In summary, for the observe step of the OODA loop VSA provides a way to represent C5ISR data from a wide range of sensors and sources in a common representational format (i.e., hypervectors). This in turn facilitates the capability to rapidly integrate information into further hypervector representations and for these to be compactly stored in memory. In the following sections we describe how these hypervector representations can then be used to perform the required cognitive operations in the other steps of the OODA loop.

3.2 Orient: Knowledge Representation and Cognitive Maps

The orientation step of the OODA loop is where the vector representations generated in the observe step need to be rationalised with existing knowledge to determine if the observed state of the world is aligned with previously-acquired knowledge from memory.

In terms of cognitive functions, this is the process of comparing what is being observed with memories of the world state. To perform this cognitive function, it is necessary to be able to represent these world states in a form that can be efficiently stored in memory and that facilitates rapid comparison and matching.

A simple example of orientation can be understood from the semantic image processing example above. The semantic hypervector representation of a complex scene is a compact representation that can be efficiently stored in memory. Neubert et al.[29] has demonstrated this approach to be an extremely efficient state-of-the-art method to locate the vehicle or robot in the physical world (i.e., orientate to the position). It has also been demonstrated that this is possible even when there are significant changes in the image due to light conditions, weather, etc, and is therefore a robust way of orientating to location accounting for uncertainty in the observations. Importantly 'location' in this context can also be represented as a hypervector that not only represents the physical location but using VSA binding can also represent the 'location' in different semantic representations (e.g., this is my home or place of work) which can also be represented by hypervectors. VSA facilitates these different representations to be hierarchically linked. Orientation is therefore a process of mapping what is being observed to a 'memory' of existing knowledge. In this section we argue that hypervector representations of cognitive maps are a way to achieve these objectives and show through a simple example how cognitive maps can be learned directly from observation.

Another major advantage of using VSA to encode images compared to state-of-the-art techniques is that since the vector representation is a single highly compact hypervector the corresponding matching, based on vector similarity, is significantly faster to compute than existing methods.

The more traditional AI approach to representing these relationships and hierarchies of relationships is to use knowledge graphs (KGs). Liang et al.[30] presents a survey of current state of the art. Since it is possible to construct hypervector representations of any graph structure, it is possible to represent KGs as hypervectors. Poduval et al.[31] describes an approach to hypervector graph representation which is termed GraphHD. In this representation the graph nodes are themselves hypervectors which semantically represent the node concepts and the links between the nodes are represented using the VSA binding and bundling operations. Hypervector representations of graphs facilitates the capability to rapidly discover similar graph structures using simple vector matching operations, and importantly sub-graph structures within larger graphs.[32] It is also possible to efficiently navigate graph structures using VSA unbinding operations. VSA representations can therefore be used to perform KG representation and multi-hop reasoning using very efficient vector operations.

Whilst it is possible to represent KGs in this way, recent developments in AI have clearly demonstrated, just as Eliasmith and others conjectured, that vector space embedding and performing operations on vectors in the vector space (e.g., moving through the vector space and transforming or mapping between vector spaces) surprisingly produces the capability to perform human-like cognitive functions. One of the main benefits of vector space embedding is that the complexities of the real world can be mapped in such a way that simple vector operations in the vector space can mirror complex reasoning in the real world. As an example, the embedding application Word2Vec which uses triplets of co-occurring words to construct the vector space can perform complex reasoning using simple vector operations (e.g., subtracting the vector for 'male' from the vector for 'king' and adding the vector for 'female' results in the vector for 'queen'). In the case of LLMs (e.g., GPT4) this is expanded using a simple attention mechanism to consider long word sequences that are mapped (encoded) into a vector embedding space that represents the structure of language. Mapping any sequence of words (i.e., symbols) into this embedding space produces an abstract vector that can be transformed (decoded) into a different representation (e.g., another language) or to predict the next word (symbol) or a sequence that represents the answer to a question.

This raises the possiblity to represent KGs in a semantic vector space and importantly to facilitate both reasoning over KGs and graph completion (i.e., inferring relationships that are not in the observed graph) using vector operations. This partially explains the motivation for representing KGs as graph neural networks (GNNs)[30] and leads to the question of the possibility to do something similar using hypervectors. A recent paper[33] has described how it is possible to do this using an approach termed *cognitive map learner* (CML). In the paper the authors demonstrate that it is possible to learn the structure of any graph as a semantic vector representation. Intriguingly this is done by randomly exploring the graph and the authors show that the full graph structure can be inferred without the need to explore all possible paths (i.e., it automatically performs graph completion).

The approach is also applicable to constructing other types of cognitive map and the paper also describes how this concept can be applied to control a simple robot to perform tasks such as following and fleeing from 'prey' or 'predator'. To illustrate how this applies in the orient step of the OODA loop we use a simple illustration of an AI agent that must learn a cognitive map of its observed environment. The observed environment comprises a grid that contains barriers through which the agent cannot pass. The agent performs short random excursions from random locations based on a set of possible actions in the environment that avoids the barriers. Initially

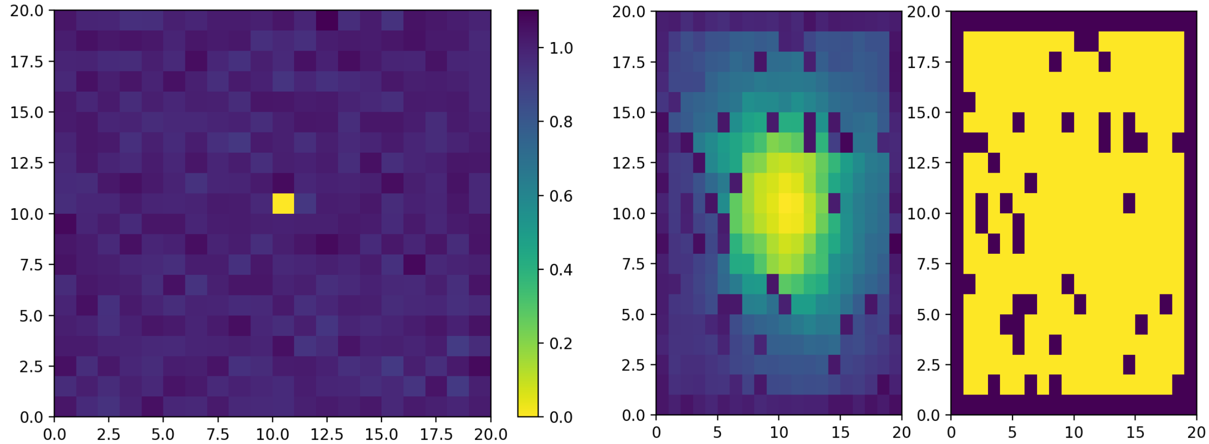

Figure 4. (a) Initial CML composed of random vectors. The colour coding is a measure of the cosine distance from the reference point in the grid. (b) Trained CML where the colour coded cosine distance corresponds to the average path distances from the reference location to all other locations that avoids the barriers. (c) The corresponding 'real world' location grid and the position of the barriers.

the cognitive map (CM) vector space comprises random vectors with one random vector representing each cell of the grid. These actions are also initially represented by random vectors with one vector for each possible action.

The initial CM is illustrated in Figure 4 (a) where the cosine distance between the central cell vector and each of the other cell vectors is colour coded. The agent has zero cosine distance to its own location and the cosine distance to any other part of the cognitive map is ≈ 1.0 (i.e., the vectors are quasi-orthogonal). So, the only safe move is to stay where it is. The CM vectors and the corresponding action vectors are updated following each random excursion using a simple learning rule. The corresponding situation following training is shown in Figure 4 (b) when viewed from the same reference location.

The colour coding reflects the distance in the vector space, and it can be shown that this corresponds to the average path distances to that location that avoids the barriers. Since the agent cannot pass through a barrier the vector distance to any of these locations is still 1.0. In addition to learning the vector space the algorithm also learns the vector representation of the 'real world' actions in that vector space. The result is that that complex navigation in the 'real world' environment is analogous to simple vector transformations in the CM vector space. These properties will be further exploited in the 'Decide' step of the OODA loop. Figure 4 (c) shows the 'real world' location grid and the barrier positions. Whilst this is a simple example illustrating the concept of creating a cognitive map it is important to recognise that this can be extended to much more complex observation environments with much larger action spaces. Since the CM is expressed as a number of vectors using VSA the vectors can be bundled to create a single vector that describes the entire CM. When observing a new situation the corresponding CM in memory can be rapidly identified.

In summary, VSA can be used to represent knowledge from knowledge graphs and as cognitive maps and also facilitates the capability to construct and learn new knowledge representations as hypervectors.

3.3 Decide: Reasoning and Planning

The decide step in the OODA loop is the step in which, having oriented to a cognitive map based on what has and is being observed, a course of action (COA) has to be determined. This requires the cognitive functions of reasoning and planning to achieve a desired goal.

So, can planning be achieved using a VSA representation? The CML approach described in the orient step, demonstrates a way of constructing an abstract vector space representation of the observed 'world state' (the cognitive map) and so the planning challenge becomes one of navigating not in the 'real world' but in the vector space using simple vector operations.[33] gives details of how this can be achieved since when learning the mapping between the 'real world' and the abstract vector space that in parallel the mapping between actions in the real

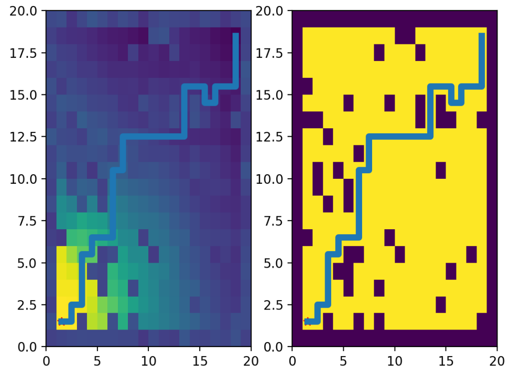

Figure 5. (a) shows the path followed in the CM vector space. (b) shows the path followed in the 'real world' grid avoiding the barriers.

world and corresponding vectors in the vector space are also learned. This makes it possible to move between a current state to a goal state using vector additions.

The algorithm chooses the best action vector at each step (analogous to the A* algorithm) and adds the best action vector to the current state vector to determine the next state. Mapping back into the real world gives results in performing the best action at each step with the result that the associated agent follows a trajectory that ends at the required goal. The approach has some similarities to reinforcement learning and whilst it is possible to perform reinforcement learning tasks using VSA[34] the agent here has not been trained to search for a goal or given any prior knowledge of paths that lead to goals and so, unlike reinforcement learning, the CML has not learned a policy, rather the planning step is simple a product of moving through a vector space using vectors that represent 'real world' actions.

This suggests a novel approach to **handling uncertainty** in these planning situations. Uncertainty can be defined as being in a state of limited knowledge, where it is impossible to exactly describe the existing state, a future outcome, or more than one possible outcome. Since VSA represents both state and action as vector operations in the vector space, this potentially provides a novel mechanism for handling uncertainty by discovering plan solutions that are still constrained by the possible action vectors and also to infer areas of the vector space that can be reached using possible action vectors but have not been explored. This is likely to be an area of fruitful future VSA research.

Figure 5 gives an example of a typical trajectory followed to reach a specified goal whilst avoiding intervening barriers. It should be noted that the algorithm can be implemented iteratively, or alternatively the trajectory can be determined in memory with the sequence of action vectors being stored before the trajectory is followed.

In this simple example, the mapping between the vector space and the 'real world' locations is easy to understand but in other environments complex actions in the 'real world' are still mapped into individual vectors and so the important message is that planning done in the abstract vector space (i.e., cognitive map) is a much simpler problem than planning in real world space.

So, can this be extended to hierarchical planning and what are the benefits? In[35] an approach for assembling CMLs into hierarchical configuration using the binding and bundling properties of VSA is described. The paper addresses integrating and orchestrating multiple CMLs together as a finite state machine which is a notoriously difficult task with traditional ANNs.[36] Using VSA facilitates the hierarchical linking of independently developed cognitive maps such that actions performed in the higher-level map can be mirrored in the lower-level maps. A simple example is illustrated in Figure 6. A KG that describes how to achieve a goal as a set of tasks can be mapped into a higher level CM that is linked to a lower level CM where an agent achieved the goal whilst avoiding the barriers. The agent has to plan a COA to achieve the goals of the higher level tasks. This is precisely what hierarchical planning systems are currently trying to achieve and so VSA approaches to this challenge offer a potential solution.

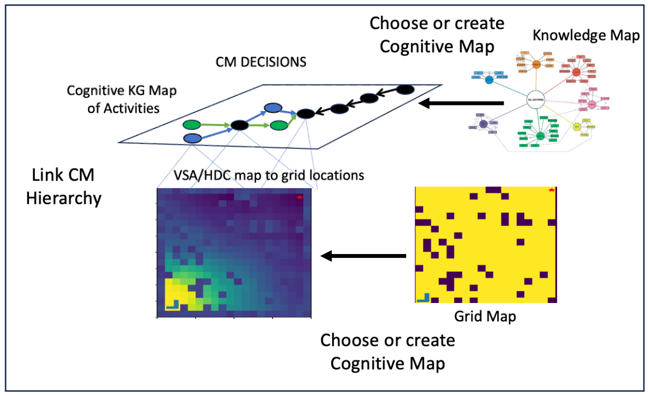

Figure 6. Illustration of hierarchical CMs comprising a higher level CM mapped from a knwledge graph and the lower level CM mapped from the positional grid map.

3.4 Act: Binding Decisions to Actions

As stated earlier, this final step of the OODA loop is to carry out the planned COA effectively and efficiently to achieve the desired tactical or strategic objective. Whilst this is the final step in OODA, it drives the next iteration of the loop since it is the step that actually progresses the goals—it controls and monitors the actions that are being taken to achieve the goals.

The act step implements the real world actions that are suggested by the decide step and by monitoring the effects of the actions to ensure that they are having the desired effect. In the context of the example described above, the cognitive action required from the CM may need to be further interpreted into a sequence of sub-actions that need to be taken to achieve this (e.g., turn left might require a sequence of actions to control a vehicle or robot to perform the required manoeuvre). This could be achieved by having yet lower levels of CM that describe these sub-actions but VSA is also ideally suited to this type of requirement since the required sequence of actions can be represented as vector of bundled sub-action vectors and this can be bound to the action vector and stored in memory. When an action vector is received it is a simple unbinding operation to obtain the required sub action sequence.

The act step has to ensure that the required actions can and have been taken. This is typically achieved through the subsequent observe step (direct changes that are taking place being observed) or inferred from observation (subsequent orientation step). How this is achieved in practice will be specific to the particular application.

4. OPPORTUNITIES AND CHALLENGES FOR VSA EXPLOITATION

In the preceding sections we have attempted to show how VSA is a powerful uniform framework for performing the types of cognitive processing tasks that will be required for future C5ISR. In each step of the OODA loop, we have explained that it is possible to represent a wide variety of things from sensors and sources through to knowledge graphs and cognitive maps as semantic hypervectors. We have shown, through several referenced examples, that these representations facilitate the capability to perform several of the cognitive functions required for C5ISR, from object classification through to goal directed planning. Because of the common vector representation, the potential exists to perform all the OODA loop processing in a unified framework, removing silos between C5ISR functions. In this paper we have referenced a number of works that have already demonstrated that VSA representations have many benefits, when compared to other state of the art solutions for performing equivalent cognitive processing functions. The properties of the VSA hypervector representation also exhibit a number of other properties that might also be exploited for future C5ISR applications, and these are discussed below.

4.1 Opportunities to Exploit VSA

In the general context of AI and ML, VSA-based learning tackles the challenges that are inherently hard for conventional machine learning by exploiting the unique properties of the VSA framework that are especially appealing for edge computing and hardware implementation. There are therefore many opportunities to exploit the properties of VSA for future C5ISR. These properties include:

Noise robust: VSA represents the information (data) in holographic, distributed hypervectors. Unlike the computing framework used by traditional machine learning—where an error of a single data value would lead to erroneous data and hence erroneous prediction—a data value is distributed and represented in a hypervector in high dimensional space, which minimises information loss even if some bits of the hypervector are flipped or corrupted. This feature makes VSA-based learning robust to noise in data and tolerable to errors in communication.[37]

Data-frugal: Mainstream machine learning generally requires large volumes of training data. In contrast, the flexible data encoding schemes of VSA bypass many of the data and learning problems associated with traditional machine learning. As VSA encodes data in hypervectors, it naturally facilitates a mechanism to seamlessly and easily unify multimodal data, via a simple association and superposition of hypervectors of different data models. Similarly, VSA superposition property offers an unrivalled systematic tool for dealing with data issues such as missing values and tackling learning problems such as one/few-shot learning and continual learning.

Computationally-efficient: VSA facilitates the encoding, processing, and storing of data in a common vector space, enabling in-memory computing which eliminates the latency of data movement between CPU and memory.[38] VSA-based learning utilises simple but very efficient algebraic operations of hypervectors, primarily leveraging its binding property to combine data features and its superposition property to capture complex relationships and nuanced patterns in data. VSA learning is typically done via one-pass of the data but can also be done iteratively to minimise model errors like traditional learning. Unlike floating point operations used in traditional machine learning, distributed VSA bitwise operations use simple binding (bitwise multiplication, permutation/cyclic shift), bundling (bitwise addition), and comparison (bitwise XOR) to make the computing process parallel and hence fast and scalable.

Energy-efficient and hardware-friendly: Due to bitwise manipulations of hypervectors, VSA operations are friendly for implementation and consume minimal energy.[14,16,39] This energy efficiency makes VSA learning superior over traditional machine learning on edge devices where computing power and energy capacity are often severely constrained.[40] The feature of all bitwise operations makes VSA hardware-friendly when implementing learning algorithms on circuits.

Next generation processor-compatibility: VSA hypervector representations are ideally suited to being implemented in next generation neuromorphic processing devices such as SNNs and 'in-memory' processing technologies[11,14,16] which not only provide the potential for ultra energy efficient processing[41] but also, in the case of SNNs, offer potentially significant advantages for military applications when coupled to emerging devices such as event-based cameras[42,43] and ultra high speed processing using spiking photonic devices.[44]

All these properties clearly have resonance with the challenges of A^2ISR and specifically where these functions have to be performed in challenging edge of network tactical environments where size, weight, power and cost (SWaP-C), and limited communications bandwidth are major considerations, and where speed of response to the increasing tempo of the modern battlespace is increasingly important.

4.2 Challenges

Section 3 has shown that there are potentially many solutions to meet the challenge of the cognitive processing functions required for each step of the OODA loop for future C5ISR Solutions. For each cognitive function—perception, attention, memory, decision-making, problem-solving and communication—there is evidence that VSA is effective, in most cases benchmarked against the state of the art in AI. While these works can be regarded as offering 'point solutions', the most compelling argument for the applicability of VSA is that it provides the mathematical tools for *combining* these diverse representations into alternative vector representations for performing different tasks.

Whilst there is a common representation of 'everything' in the form of hypervectors, different works have made different modelling choices, notably in the handing of data dimensions including time, space, and ontology. Each specific modelling choice has strengths and limitations. The key challenge, however, is *how to best link the different processing stages into a continuously operating cycle and particularly at the network edge.* This is also one of the main challenges for AI in general, which is striving for a mechanism to engineer heterogeneous AI solutions as a type of 'plug & play' approach where models can be added together and run in parallel without the need for retraining. The mechanism suggested in[35] for combining CMLs using VSA offers a potential solution to this challenge. The challenge is to determine the most efficient ways in which this can be achieved.

A related challenge is how best to use VSA as an integration technology to harness the strengths of other AI approaches, while mitigating their weaknesses. Work combining VSA approaches with LLMs and other transformer models is at a relatively early stage and likely to prove a fruitful area for research and development in the near term.

5. CONCLUSIONS

In this paper we have shown how VSA is a powerful framework for performing the types of cognitive processing tasks that will be required for future C5ISR. In each step of the OODA loop, we have explained that it is possible to represent a wide variety of assets, from sensors and sources through to knowledge graphs and cognitive maps, as semantic hypervectors and explained how state of the art efficient analytics (e.g., adaptive, distributed, federated neuro-symbolic and probabilistic AI) can be performed across OODA using VSA. We have also identified that the representation of both state and action using VSA hypervectors offers a potential novel approach to handling uncertainty in the OODA planning and decision phase and is likely to be a fruitful direction for future research. We have shown through several referenced examples—that integrate symbolic and connectionist approaches to AI—that VSA hypervector representations facilitate the capability to perform many of the cognitive functions required for C5ISR, from object classification through to goal-directed planning, using VSA techniques that perform all the processing in a uniform representational framework.

The paper has provided an extensive set of references that describe the state of the art of VSA. These demonstrate that VSA solutions perform as well or better than existing state of the art solutions, usually with lower computational complexity. VSA hypervector representation is complementary to many of the current developments in AI/ML. It indeed provides a potential mechanism to meet one of the major outstanding research challenges, by providing a mechanism to engineer heterogeneous AI solutions as a type of 'plug & play' solution, where models can be added together and run in parallel without the need for retraining.

ACKNOWLEDGMENTS

This work was funded by Dstl under project number U129, contract number DSTL0000022661. The contents include material subject to ©Crown copyright (2024), Dstl. This information is licensed under the Open Government Licence v3.0. To view this licence, visit https://www.nationalarchives.gov.uk/doc/open-government-licence/ Paper number: DSTL/CP161732.

REFERENCES

[1] Dstl, "The defence implications of edge computing," *DSTL/CR132147, DSTL/TR132145* (2021).

[2] Kleyko, D., Rachkovskij, D., Osipov, E., and Rahimi, A., "A survey on hyperdimensional computing a.k.a. vector symbolic architectures, part i: Models and data transformations," *ACM Computing Surveys (CSUR)* **55**, 1–52 (2022).

[3] Kleyko, D., Rachkovskij, D., Osipov, E., and Rahimi, A., "A survey on hyperdimensional computing a.k.a. vector symbolic architectures, part ii: Applications, cognitive models, and challenges," *ACM Computing Surveys (CSUR)* **55**, 1–40 (2023).

[4] Tolman, E. C., "Cognitive maps in rats and men.," *Psychological Review* **55**(4), 189 (1948).

[5] Richards, C., "Boyd's OODA loop," *Necesse* (2020).

[6] Youheng, Z., "A historical review and philosophical examination of the two paradigms in artificial intelligence research," *European Journal of Artificial Intelligence and Machine Learning* **2**(2), 24–32 (2023).

[7] Eliasmith, C., [*How to Build a Brain: A Neural Architecture for Biological Cognition*], Oxford (2013).

[8] Simpkin, C., Taylor, I., Bent, G. A., De Mel, G., and Rallapalli, S., "Decentralized microservice workflows for coalition environments," in [*2017 IEEE SmartWorld*], IEEE (2017).

[9] Simpkin, C., Taylor, I., Bent, G. A., de Mel, G., Rallapalli, S., Ma, L., and Srivatsa, M., "Constructing distributed time-critical applications using cognitive enabled services," *Future Generation Computer Systems* **100**, 70–85 (2019).

[10] Simpkin, C., Taylor, I., Harborne, D., Bent, G., Preece, A., and Ganti, R. K., "Efficient orchestration of Node-RED IoT workflows using a vector symbolic architecture," *Future Generation Computer Systems* **111**, 117–131 (2020).

[11] Karunaratne, G., Gallo, M. L., Hersche, M., Cherubini, G., Benini, L., Sebastian, A., and Rahimi, A., "Energy efficient in-memory hyperdimensional encoding for spatio-temporal signal processing," *IEEE Transactions on Circuits and Systems II: Express Briefs* **68**, 1725–1729 (2021).

[12] Mitrokhin, A., Sutor, P., Fermüller, C., and Aloimonos, Y., "Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception," *Science Robotics* **4** (2019).

[13] Menon, A., Natarajan, A., Agashe, R., Sun, D., Aristio, M., Liew, H., Shao, Y. S., and Rabaey, J. M., "Efficient emotion recognition using hyperdimensional computing with combinatorial channel encoding and cellular automata," *Brain Informatics* **9** (2021).

[14] Bent, G., Simpkin, C., Li, Y., and Preece, A., "Energy efficient spiking neural network neuromorphic processing to enable decentralised service workflow composition in support of multi-domain operations," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*], **12113**, 479–499, SPIE (2022).

[15] Rachkovskij, D. A., "Estimation of vectors similarity by their randomized binary projections," *Cybernetics and Systems Analysis* **51**(5), 808–818 (2015).

[16] Bent, G., Simpkin, C., Taylor, I., Rahimi, A., Karunaratne, G., Sebastian, A., Millar, D., Martens, A., and Roy, K., "Energy efficient 'in memory' computing to enable decentralised service workflow composition in support of multi-domain operations," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*], Pham, T. and Solomon, L., eds., **11746**, 414–435, SPIE (2021).

[17] Neubert, P. and Schubert, S., "Hyperdimensional computing as a framework for systematic aggregation of image descriptors," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* , 16933–16942 (2021).

[18] Davies, C., Meek, S., Hawkins, P., Tutcher, B., Preece, A., and Bent, G., "Vector symbolic architecture for semantic data discovery in coalitions," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*], **in press**, SPIE (2024).

[19] Hersche, M., Karunaratne, G., Cherubini, G., Benini, L., Sebastian, A., and Rahimi, A., "Constrained few-shot class-incremental learning," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* , 9047–9057 (2022).

[20] Wilson, S., Sunderhauf, N., and Dayoub, F., "Hyperdimensional feature fusion for out-of-distribution detection," *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* , 2643–2653 (2023).

[21] Neubert, P., Schubert, S., Schlegel, K., and Protzel, P., "Vector semantic representations as descriptors for visual place recognition.," in [*Robotics: Science and Systems*], 1–11 (2021).

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

[23] Imani, M., Kong, D., Rahimi, A., and Simunic, T., "VoiceHD: Hyperdimensional computing for efficient speech recognition," *2017 IEEE International Conference on Rebooting Computing (ICRC)* , 1–8 (2017).

[24] Schlegel, K., Mirus, F., Neubert, P., and Protzel, P., "Multivariate time series analysis for driving style classification using neural networks and hyperdimensional computing," in [*2021 IEEE Intelligent Vehicles Symposium (IV)*], 602–609, IEEE (2021).

[25] Dempster, A., Schmidt, D. F., and Webb, G. I., "MiniRocket: A very fast (almost) deterministic transform for time series classification," *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (2020).

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

[27] Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., and Keogh, E. J., "The UCR time series archive," *IEEE/CAA Journal of Automatica Sinica* **6**, 1293–1305 (2018).

[28] Simpkin, C., Taylor, I., Preece, A., Bent, G. A., Tomsett, R., Ganti, R. K., and Worthington, O., "Coalition C3 information exchange using binary symbolic vectors," in [*MILCOM 2019–2019 IEEE Military Communications Conference (MILCOM)*], 784–789 (2019).

[29] Neubert, P., Schubert, S., and Protzel, P., "An introduction to hyperdimensional computing for robotics," *KI–Künstliche Intelligenz* **33**, 319–330 (2019).

[30] Liang, K. Y., Meng, L., Liu, M., Liu, Y., Tu, W., Wang, S., Zhou, S., Liu, X., and Sun, F., "A survey of knowledge graph reasoning on graph types: Static, dynamic, and multimodal," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , doi: 10.1109/TPAMI.2024.3417451 (2024).

[31] Poduval, P., Alimohamadi, H., Zakeri, A., Imani, F., Najafi, M. H., Givargis, T., and Imani, M., "GrapHD: Graph-based hyperdimensional memorization for brain-like cognitive learning," *Frontiers in Neuroscience* **16**, 757125 (2022).

[32] Gayler, R. and Levey, S., "A distributed basis for analogical mapping," in [*New Frontiers in Analogy Research, Proceedings of the Second International Conference on Analogy, ANALOGY-2009*], (2009).

[33] Stöckl, C., Yang, Y., and Maass, W., "Local prediction-learning in high-dimensional spaces enables neural networks to plan," *Nature Communications* **15:2344** (2024).

[34] Ni, Y., Abraham, D., Issa, M., Kim, Y., Mercati, P., and Imani, M., "Efficient off-policy reinforcement learning via brain-inspired computing," in [*Proceedings of the Great Lakes Symposium on VLSI 2023*], 449–453 (2023).

[35] McDonald, N., "Modularizing and assembling cognitive map learners via hyperdimensional computing," *arXiv preprint arXiv:2304.04734* (2023).

[36] Duan, S., Yu, S., and Príncipe, J. C., "Modularizing deep learning via pairwise learning with kernels," *IEEE Transactions on Neural Networks and Learning Systems* **33**(4), 1441–1451 (2021).

[37] Zhang, S., Wang, R., Zhang, J. J., Rahimi, A., and Jiao, X., "Assessing robustness of hyperdimensional computing against errors in associative memory," in [*2021 IEEE 32nd International Conference on Application-specific Systems, Architectures and Processors (ASAP)*], 211–217, IEEE (2021).

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

[39] Rahimi, A., Kanerva, P., and Rabaey, J. M., "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in [*Proceedings of the 2016 international symposium on low power electronics and design*], 64–69 (2016).

[40] Yu, X., Zhou, M., Asgarinejad, F., Gungor, O., Aksanli, B., and Rosing, T., "Lightning talk: Private and secure learning at the edge with hyperdimensional computing," in [*2023 60th ACM/IEEE Design Automation Conference (DAC)*], (2023).

[41] Bihl, T., Farr, P., Di Caterina, G., Vicente-Sola, A., Manna, D., Kirkland, P., Liu, J., and Combs, K., "Exploring spiking neural networks (SNN) for low size, weight, and power (SWaP) benefits," in [*Proceedings of the 57th Hawaii International Conference on System Sciences*], (2024).

[42] Vicente-Sola, A., Manna, D. L., Kirkland, P., Di Caterina, G., and Bihl, T. J., "Spiking neural networks for event-based action recognition: A new task to understand their advantage," *Available at SSRN 4559724* (2023).

[43] Kirkland, P., Di Caterina, G., Soraghan, J., and Matich, G., "Neuromorphic technologies for defence and security," in [*Emerging Imaging and Sensing Technologies for Security and Defence V; and Advanced Manufacturing Technologies for Micro-and Nanosystems in Security and Defence III*], **11540**, 113–130, SPIE (2020).

[44] Robertson, J., Kirkland, P., Alanis, J. A., Hejda, M., Bueno, J., Di Caterina, G., and Hurtado, A., "Ultrafast neuromorphic photonic image processing with a VCSEL neuron," *Scientific Reports* **12**(1), 4874 (2022).