Explainability as an Emergent Property of Brain-Inspired Human Cyber Physical Networks

Charles J. Gish[1,3(✉)], Javier Villalba-Diez[2,3], and Joaquin Ordieres-Mere[1]

[1] Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid, c/ José Gutiérrez Abascal 2, 28006 Madrid, Spain
charles.gish@hs-heilbronn.de

[2] Department of Mechanical Engineering, Universidad de La Rioja, Edificio Departamental, c/ San José de Calasanz 31, 26004 Logroño, Spain

[3] Heilbronn University of Applied Sciences, Bildungscampus Nord, 74076 Heilbronn, Germany

Abstract. As AI becomes more widespread and deep learning models are integrated into various aspects of industry, scientists and engineers face the challenge of integrating human operators and end users into the design and operation of cognitive cyber physical systems. This challenge is exacerbated by the fact that many deep learning models employ black box approaches that lack transparent, human-interpretable algorithms. This paper addresses the challenge faced by proposing a human cyber physical network model, inspired by the human brain. In contrast to contemporary deep learning models, ours leverages a vector symbolic architecture to interactively learn human behavior and to develop cognition within the networks. To test this exploratory model, a two-part simulation is conducted. Using a modified industrial human-machine interaction dataset, we create structural networks of human, cyber, and physical object representations. These objects and their situational contexts are then encoded as hyperdimensional vectors. With context-dependent thinning, our model builds analogical episodes featuring distributed, parallel associative memories. The proposed model is shown to have analogical reasoning capabilities, with object nodes learning structural characteristics such as their hierarchical, or part-whole relationships within a network. Functional characteristics, such as human motion patterns and is-a relationships are learned as well. The model's accuracy and performance can be transparently audited using established algorithms from network science. The results in this paper indicate that by designing systems as brain-inspired networks of human, cyber, and physical objects, vector symbolic architectures can be used to learn their structure and function by human-interpretable methods. Thus, the accountability inherent in the proposed model increases AI explainability in human cyber physical systems.

Keywords: Explainable AI · Human Cyber Physical System · Vector Symbolic Architecture

M. Kurosu and A. Hashizume (Eds.): HCII 2025, LNCS 15767, pp. 344–363, 2025.
https://doi.org/10.1007/978-3-031-93838-2_22

1 Introduction

The transition from Industry 4.0 to Industry 5.0 presents a significant challenge in designing production systems that are human-centric, resilient, and sustainable [38]. To address this challenge, researchers are focusing on the design and implementation of cognitive human-cyber-physical systems (HCPS). These systems comprise complex networked agents operating at various levels of manufacturing hierarchies, interacting across human, cyber, and physical domains [11,35].

Graph neural networks (GNNs) have been heavily researched as a way to learn representations in complex networked systems [14,41]. However, learning over local manifolds in hierarchical, complex cyber physical systems limits how much compositionality can be represented [33].

The authors in [5] used a graph encoder-decoder model to learn action patterns in graphs of human and physical object nodes. A later paper [19], proposed a dataset called the Collaborative Action Dataset (CoAx), based on the dataset in [5]. CoAx was used to learn human motion patterns in a number of tasks, including human-robot collaboration in an industrial setting. However, the artificial neural networks (ANNs) typically employed in GNNs and in convolutional neural networks suffer from the variable binding problem [12], or the lack of ability to connect abstract symbols to real-world features - an aspect of human cognition.

The most interesting problem with current machine learning approaches in regard to this paper, however, is that the lack of interpretability and explainability in the models seems to scale with their complexity. As is pointed out in [21], there is a trade-off between prediction capabilities in deep learning models, and their ability to to explain how said predictions were made. Neuro-symbolic AI has been focused on as a means of increasing machine learning explainability in many fields, according to a 2024 review [39]. One approach discussed in the review involved the use of a vector symbolic architecture (VSA) together with a neural network [7], but fell short of improving explainability due to the model's computation in the neural network.

VSAs have been employed to model human-like cognitive functionality and analogical reasoning within hierarchical data structures [8,28]. Using VSAs without the augmentation of neural networks would likely improve explainability in machine learning models for HCPS. Nevertheless, the advancement in this domain is impeded by the lack of methods to construct analogical episodes through world observation [18].

An exemplary system embodying the ideal HCPS qualities of cognition, robustness, and efficiency is the human complex brain network [3]. Brain-inspired complex networks have been utilized in the human and physical domains of sociotechnical systems [23,32,34]. However, there is a scarcity of literature on employing brain-inspired networks across human, cyber, and physical domains. Additionally, there are limited concrete industrial use cases featuring operational implementations of VSAs [17,18]. Thus, there is an unexplored research area in the combined use of brain-inspired complex networks

and VSAs to tackle Industry 5.0 challenges, such as cognitive HCPS design, as well as an opportunity to evaluate the explainability of such systems from a new perspective.

To create value in an Industry 5.0 context, manufacturing organizations must integrate humans and machines within complex environments where heterogeneous data sources present interoperability challenges [25]. Beyond integration, humans must play a central role in production, necessitating machine intelligence capable of collaborating with humans at various hierarchical levels [40]. The objective of developing systems with human-like artificial intelligence involves overcoming the problem of compositionality, as well as achieving generalization and causal discovery [18]. Several VSA models have attempted to address these goals, but whether they provide a holistic solution applicable to HCPS design remains an open question.

Another open question - the one addressed in this paper - is whether explainability would emerge as a property of such a model. The authors in [4] propose a taxonomy for the evaluation approaches for interpretability in machine learning, or rather, the ability of a system to *explain* its reasoning. Their functionally-grounded, application-grounded, and human-grounded approaches would provide a measure of rigor to the evaluation of explainability in a VSA-based model.

VSA models exist that learn hierarchical compositions from both sensor and actuator interaction patterns, as well as from noisy human-machine interactions [6,20]. While these models can address Industry 5.0 interoperability issues and human-machine integration, they do not place humans in a central role. Furthermore, they often utilize centralized architectures, requiring the transmission or broadcasting of sensor states onto a network before actuator updates, which can diminish efficiency in wireless sensor networks [37].

In functionally-grounded approaches to the evaluation of interpretability [4], formal definitions of interpretable models are used as proxies to evaluate explainability in a model. In the composition of a hierarchical network such as an HCPS, defined graph theoretical models from network science could be used as proxies for thier efficiency optimization.

Certain VSAs, such as associative-projective neural networks (APNNs), can generalize from novel input patterns to known output patterns, typically employing autoassociative memories [29] or implementing both auto- and hetero-associative memories related through spectral graph theory [13]. However, the vectors are randomly generated to enforce the near orthogonality necessary for Hebbian learning and hyperdimensional computing in general [24]. Any operation on, or permutation of, random binary vectors produces a new vector that must be bound as a role or filler to be meaningful. Consequently, VSAs typically require supervised learning with labels and explicitly defined structures [16].

Domain expertise is required for application-grounded approaches [4], in which a human with knowledge of a particular task evaluates the quality of an

explanation in the system. Such knowledge could apply to parameters needed to optimize classification scores for generalizability to other similar HCPS.

Despite VSAs' capability for causal discovery through analogical reasoning, there is a notable lack of practical examples [18], especially for HCPS. This is likely due to the absence of mechanisms for building analogical episodes.

In this paper, we therefore propose a brain-inspired model for cognitive human cyber physical networks (HCPNs), employing methods from neuroscience and network science within an APNN to address the limitations of existing VSAs. The model addresses the variable binding problem inherent in contemporary deep learning and GNN models, while providing a mechanism to build real-world representations of HCPS that can be implemented in cloud applications or on constrained IoT devices. Taken as a whole, the proposed model enables the construction of HCPS episodes with analogical reasoning capability. Through a simulation with the model, the aim of this paper is to answer the following research question:

RQ: *As a consequence of its interpretable methods, is explainability an emergent property of a brain-inspired HCPN model?*

The remainder of this paper is structured as follows. In Sect. 2, fundamental theoretical concepts from the intersection of neuroscience, network science, and VSAs are provided. These concepts are then used in Sect. 3, as a foundation upon which an HCPN model and APNN implementation are based. In Sect. 4, the model is audited in a simulation that includes: the use of a modified dataset from the literature to build HCPNs and their associated analogs; the retrieval of learned symbolic representations for part-whole inference; and the mapping of related representation components for is-a inference. For the latter two parts of the simulation, the explainability is evaluated using functionally-grounded and application-grounded approaches [4]. Finally, conclusions are drawn from the simulation, future work is discussed, and the question of whether explainability exists in the proposed model is addressed.

2 Theoretical Concepts

2.1 Brain-Inspired Structural, Functional, and Effective Networks

Human-cyber-physical systems (HCPS) are complex networks of interconnected agents [11,35]. Such systems can be described by their structure, function, and behavior [10]. Two important technologies that support the design and operation of robust and efficient HCPS are the industrial internet of things (IIoT) and digital twins (DTs) [38]. It is conceivable that the nodes of these complex networks of of human, cyber, and physical agents could be modeled as DTs, with their links defined as IIoT network connections. The structure, function and behavior of the HCPS could then be modeled as well.

The human brain, as a robust and efficient complex system, exemplifies structural, functional, and effective networks, making it an ideal model for

HCPS design [3,9]. Structural networks within brain models represent anatomical connections shaped by temporally correlated functional activity in different regions of the brain. Functional networks are defined by time-dependent interactions between brain regions, which in turn, influence structural network changes. Finally, effective networks capture causal associations between neural regions [3]. An example of an HCPS analog of this concept could be publish-subscribe connections between DTs of sensor nodes as structural network links, with instantaneous messages of sensor states as functional network links, and event trigger values as effective links [31]. Neuroimaging techniques like fMRI identify structural networks in brains by creating correlation matrices that can be thresholded into adjacency matrices [3]. In the HCPS analogy, structural links between statistically relevant agents could be identified from temporal correlations in their data, as illustrated in Fig. 1.

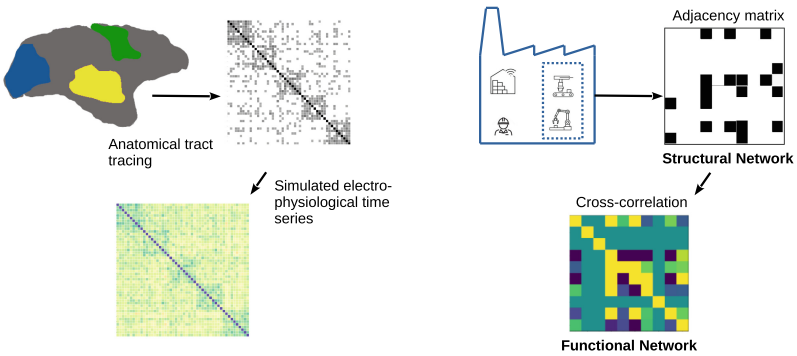


Fig. 1. Illustration of cross-correlation and threshold adapted from [3] (left) and its concept adapted to a factory (right)

Brain networks exhibit properties of scale-free networks, reflecting both robustness and efficiency [1,3]. These properties are observed in brain-inspired networks applied to human and physical domains [23,32]. Extending this concept to the cyber domain requires symbolic representations of structural, functional, and effective relationships for human and physical assets. Such representations are essential for applying brain-inspired models to explainable machine learning for HCPS.

2.2 Vector Symbolic Architectures and Context-Dependent Thinning

The eigenvalues and eigenvectors of a graph's Laplacian matrix are related to many defining aspects of the underlying network [22]. Both [32] and [23] highlight the importance of the second smallest eigenvalue - the Fiedler value - in characterizing network attributes such as connectivity and robustness in human and physical domains. From these examples in the literature, it follows

that if human and physical assets were nodes in a graph, the Laplacian eigenvalues and eigenvectors could serve as meaningful representations of their contexts in the cyber domain.

Example: an adjacency matrix that results from a threshold matrix of correlations between human, cyber, and physical objects has column vectors $\mathbf{a}_1, \mathbf{a}_2$, and $\mathbf{a}_3$, each corresponding to the H, C, and P nodes in a directed graph (see Fig. 2). If the Laplacian matrix for the graph is symmetric, then its eigenvectors will be orthogonal with respect to one another [36], which is also a property required of symbolic representations in VSAs [18]. The eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, ordered by increasing magnitude of their eigenvalues $\lambda_1, \lambda_2, \lambda_3$ may then symbolically represent the graph nodes and their directed adjacency matrix column vectors.

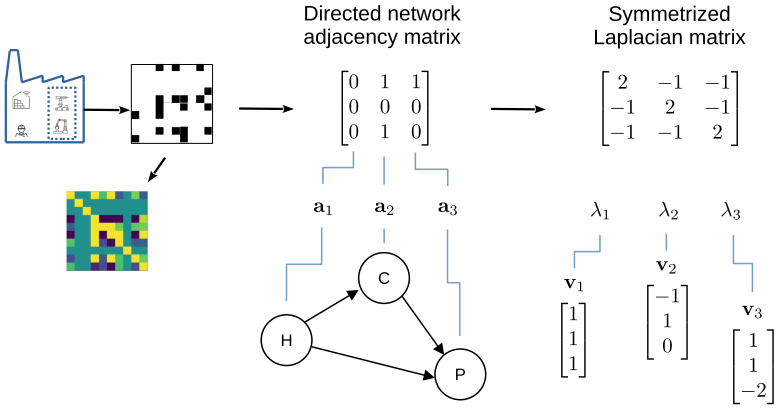


Fig. 2. .

The scenario in the example works, provided the column- and eigenvectors are encoded in a meaningful manner. However, as [29] point out, the challenge is to represent such numerical vectors by hyperdimensional vectors (HDVs) that preserve similarity to the originals.

In contemporary VSAs [15,26], binding and bundling operations such as circular convoltution and XOR are capable of encoding graphs. While these representations preserve unstructured similarity and control the density of their HDVs, they often produce what is known as 'superposition catastrophe' [30]. This results in erroneous similarities between vectors having different structures, where an HDV composed of elements ab, cd, and ef would be similar to an HDV having a composite element ad or bf.

As an example, suppose we create an HDV representation of eigenvector $\mathbf{v}_1 = (1, 0, 1)$ by binding its element position roles (**p_1**, **p_2**, **p_3**) to their corresponding value fillers (**one**, **zero**, **one**), then bundling each element into a composite HDV to give **eigvec_1** $= [(\mathbf{p\_1} \otimes \mathbf{one}) \oplus (\mathbf{p\_2} \otimes \mathbf{zero}) \oplus (\mathbf{p\_3} \otimes \mathbf{one})]$.[1]

[1] The operator $\otimes$ denotes binding, and $\oplus$ is bundling.

Because only unstructured similarity is preserved, it could be impossible to distinguish between **eigvec_1** and **eigvec_2**. This is because the latter would represent $\mathbf{v}_2 = (0, 1, 1)$ and be composed of elements such as $(\mathbf{p\_2} \otimes \mathbf{one})$.

On the other hand, the context-dependent thinning (CDT) of sparse binary distributed representations (SBDRs) proposed in [28,29] preserve both structured and unstructured similarity, and an SBDR composed of elements *ab*, *cd*, and *ef* would be *less* similar to an SBDR having a composite element *ad* or *bf* when using permutations of the components to represent their positions or order. Such a permutation is denoted $\mathbf{X}_{\sim\mathbf{y}}$, with $\mathbf{X}$ the SBDR and $\mathbf{y}$ the position number or order.

Additive CDT [28] is achieved by taking a composite SBDR that is a disjunctive superposition of two or more SBDRs, for example $\mathbf{Z} = \mathbf{A} \vee \mathbf{B} \vee \mathbf{C}$, and then carrying out a thinning operation, which is an iterative conjunction of $\mathbf{Z}$ with different permutations of itself. The thinning is denoted with angle brackets, with $\langle \mathbf{Z} \rangle = \langle \mathbf{A} \vee \mathbf{B} \vee \mathbf{C} \rangle$ having a subset of the active bits from each of its components $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. The total number of active bits in $\langle \mathbf{Z} \rangle$ is proportional to the number of active bits in each component SBDR.

From the above example, the SBDR representations for eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ using CDT notation would be $\mathbf{eigvec\_1} = \langle \langle \mathbf{p\_1} \vee \mathbf{one}_{\sim 1} \rangle \vee \langle \mathbf{p\_2} \vee \mathbf{zero}_{\sim 2} \rangle \vee \langle \mathbf{p\_3} \vee \mathbf{one}_{\sim 3} \rangle \rangle$, and $\mathbf{eigvec\_2} = \langle \langle \mathbf{p\_1} \vee \mathbf{zero}_{\sim 1} \rangle \vee \langle \mathbf{p\_2} \vee \mathbf{one}_{\sim 2} \rangle \vee \langle \mathbf{p\_3} \vee \mathbf{one}_{\sim 3} \rangle \rangle$. Structured similarity is preserved, and a different subset of active bits from $\mathbf{p\_2}$ are present in $\langle \mathbf{p\_2} \vee \mathbf{zero}_{\sim 2} \rangle$ than are present in $\langle \mathbf{p\_2} \vee \mathbf{one}_{\sim 2} \rangle$. Thus, **eigvec_1** and **eigvec_2** can be easily distinguished from each other.

2.3 Associative-Projective Neural Networks

APNNs [18,28,29] are made up of modules at different hierarchical levels, having different modalities. Modules are in turn made up of neural fields, which are sets of binary neurons having the same dimension as the SBDRs upon which they operate. Each module has a neural field designated as an associative memory that stores a set of SBDRs for later recall, and one or more neural fields used as buffers to store SBDRs temporarily, similar to RAM. The neural fields are connected by projective connections, which can either transmit unchanged SBDRs between fields, or activate certain bits in order to permute them. A simple example of a parallel APNN module architecture adapted from [29] is shown in Fig. 3.

The lowest level in an APNN contains elementary component SBDRs, which through CDT operations, are composed to form higher-level representations. In such an architecture, a noisy or incomplete version of an SBDR can be used as a probe to find the most similar SBDR in memory and return a complete or cleaned up version.

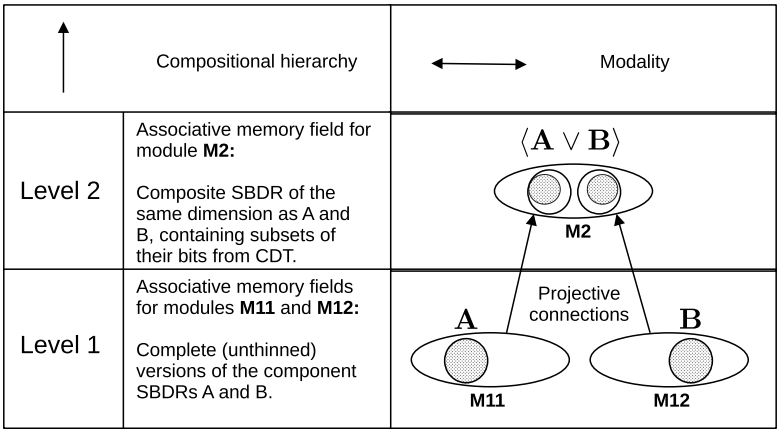


Fig. 3. APNN architecture diagram adapted from [29]

Operations using this procedure include finding a whole from its part by probing for example, **M2** with **A** to get $\langle \mathbf{A} \vee \mathbf{B} \rangle$, or returning a filler for a role [28] by using $\langle \mathbf{A} \vee \mathbf{B} \rangle \wedge \neg \mathbf{A}$ to get noisy filler $\mathbf{B}'$, which is then used in probing **M12** to get the cleaned up filler **B**. These operations are reversible, and analogous operations can be used to retrieve a part given a whole, and a role given a filler. These operations could be used to infer, for example, an entire network of cyber physical assets from a single SBDR representing a human node. This implies the capability for brain-like analogical reasoning: description of an HCPS using symbolic representations, retrieval of the correct modules, mapping corresponding components, and inferring knowledge based on their similarities. The model proposed in the next section aims to enable this capability.

3 A Brain-Inspired Model for Human Cyber Physical Networks

3.1 Human Cyber Physical Networks

In order to build analogical episodes, such as analogs of HCPS, real-world data must be converted to symbolic representations of systems. We therefore propose a brain-inspired model for human cyber physical networks (HCPNs). As described in the example from Sect. 2.1, time-series data of changing parameters are used to establish structural communication links between human, cyber, or physical assets. Given a set of time-stamped states or discrete values, velocities are calculated for each time t at each node in a pre-determined node pool. The Pearson correlation coefficient [2] is then calculated using the timestamped velocity vectors $\mathbf{x}$ and $\mathbf{y}$, from each pair of nodes. The coefficient is normalized, and the lag between the time series determines the sign of the resulting correlation score CS as shown by

$$CS = |\mathrm{r}|\mathrm{sgn}(\mathrm{lag}(\mathbf{x}, \mathbf{y})), \tag{1}$$

where

$$\mathrm{r} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}, \tag{2}$$

with $s_\mathbf{x}$ and $s_\mathbf{y}$ representing the standard deviation of $\mathbf{x}$ and $\mathbf{y}$, respectively. The correlation scores make up the elements of the correlation matrix. If a given element in this matrix is at or above the correlation threshold (CT), then a link is established between the two nodes. Since the structural HCPN is a directed network as shown in Fig. 2, the direction of the edge in the resulting graph is determined by $\mathrm{sgn}(\mathrm{lag}(\mathbf{x}, \mathbf{y}))$.

After thresholding, a structural HCPN is generated. The average number of links, or average degree $\langle d \rangle$ determines the average amount of information in an SBDR. This is because the more links a node has, the more dense the column vector of the adjacency matrix is for that node. The way information is distributed in the HCPN is determined by the degree distribution p_k, or the probability that a given node in the HCPN will have k neighbors [1].

Since human-centrism is fundamental to HCPS design, the nodes in the HCPN should behave such that they are influenced the most by the human nodes. Thus, we bias the human entries in the correlation matrix with a human weight (HW), which results in the human nodes having a high number of links to other nodes, or a high out-degree d_{out} in the HCPN. This bias can be seen as similar to the preferential attachment for nodes in scale-free networks [1]. We hypothesize that the analogical reasoning capability of our model, as in the case of human brains, is related to the scale-free property. It would also seem reasonable to conclude that by using the interpretable methods from network science for auditing and improvement, the proposed HCPN model would exhibit explainability.

3.2 Sparse Binary Distributed Representations of Laplacian Matrix Eigenvectors

In order to represent the information in the structural HCPNs, a method of encoding roles and filler values for individual vector elements is needed. The roles can be randomly generated SBDRs [29]. However, it is also practical to have a method to encode arbitrary numerical values for vector element fillers. This enables unsupervised learning of novel adjacency matrix column vectors, as well as Laplacian matrix eigenvectors.

For this purpose, we adapt a method from [27] to encode numeric values by setting a number of consecutive SBDR bits active, at an index value proportional to the numeric value. All other bits in the SBDR are zeros.

These SBDR fillers are then bound to randomly generated roles, using CDT as in the example from Subsect. 2.2. This is illustrated in Fig. 4, for an HCPN with $N = 3$ nodes. With this encoding scheme for SBDR representations of Laplacian matrix eigenvectors, we propose an APNN implementation for machine learning in HCPNs.

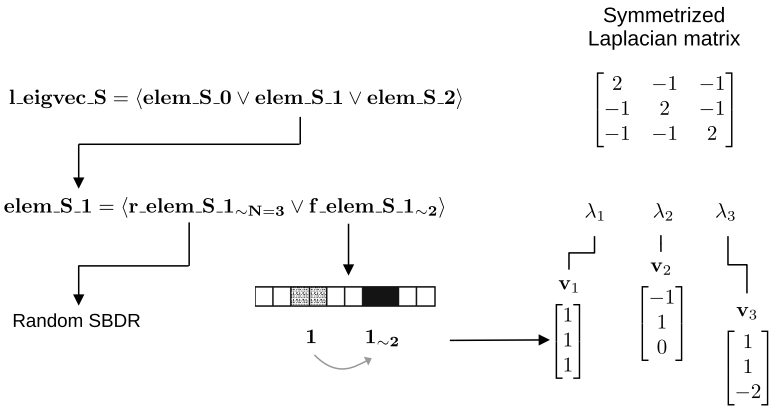


Fig. 4. SBDR encoding scheme adapted from [27], used in a vector example

3.3 The Laplacian Associative-Projective Neural Network

As discussed in Subsect. 2.3, APNNs [29] can be used to enable analogical reasoning capabilities in systems. We propose the Laplacian associative-projective neural network (LAPNN) to unlock this capability for the HCPN model. The LAPNN associative memory[2] layout is shown in Fig. 5.

At the base level **L0**, the role and filler SBDRs **r_actn_xx** and **f_actn_xx** are components of the composite vector **actn_xx** in **L1**, that represents actions defined in an action-object dataset [19]. The velocity measurements in the timestamped data for each HCPN node are represented by the composite SBDR **vel_xx** at level **L1**. In general, the composition hierarchy is as described in Subsect. 2.3. All nodes in the HCPN contain an LAPNN node, forming a distributed associative memory. One notable composite SBDR is **AGG** in level **L5**. This top level representation is an aggregation of all LAPNN modules for nodes in the next lower network hierarchy. In other words, **AGG** can be probed to return an entire HCPN, or aggregation of **Node** SBDRs.

[2] Although the effective component of the HCPN **E** is shown here, it was not used or discussed in this paper.

L5	$\mathbf{Node} = \langle \mathbf{S} \vee \mathbf{F} \vee \mathbf{E} \vee \mathbf{AGG} \rangle$
L4	$\mathbf{F} = \langle \text{vel\_actn} \vee \text{ctxt\_class} \vee \text{ctxt\_inst} \rangle$ $\quad \mathbf{E} = \text{l\_eigvec\_E} = \langle \text{elem\_E\_0} \vee \ldots \vee \text{elem\_E\_N} \rangle$ $\mathbf{S} = \langle \text{l\_eigvec\_S} \vee \text{adjvec\_S} \vee \text{ctxt\_hrchy} \vee \text{ctxt\_grnd} \rangle$
L3	$\text{adjvec\_S} = \langle \text{elem\_S\_0} \vee \ldots \vee \text{elem\_S\_N} \rangle$ $\text{l\_eigvec\_S} = \langle \text{elem\_S\_0} \vee \ldots \vee \text{elem\_S\_N} \rangle$ $\quad \text{ctxt\_hrchy} = \langle \text{r\_ctxt\_hrchy} \vee \text{f\_ctxt\_hrchy} \rangle$ $\text{ctxt\_grnd} = \langle \text{r\_ctxt\_grnd} \vee \text{f\_ctxt\_grnd} \rangle$ $\quad \text{vel\_actn} = \langle \text{vel\_actn\_00} \vee \ldots \vee \text{vel\_actn\_N}_c \rangle$ $\text{ctxt\_class} = \langle \text{r\_ctxt\_class} \vee \text{f\_ctxt\_class} \rangle$ $\quad \text{ctxt\_inst} = \langle \text{r\_ctxt\_inst} \vee \text{f\_ctxt\_inst} \rangle$ $\text{elem\_E\_xx} = \langle \text{r\_elem\_E\_xx}_{\sim N} \vee \text{f\_elem\_E\_xx}_{\sim xx} \rangle$
L2	$\text{elem\_S\_xx} = \langle \text{r\_elem\_S\_xx}_{\sim N} \vee \text{f\_elem\_S\_xx}_{\sim xx} \rangle$ $\text{r\_ctxt\_hrchy}_{\sim N}$ $\quad \text{f\_ctxt\_hrchy}$ $\quad \text{r\_ctxt\_grnd}_{\sim N}$ $\quad \text{f\_ctxt\_grnd}$ $\text{vel\_actn\_xx} = \langle \text{vel\_xx} \vee \text{actn\_xx} \rangle$ $\quad \text{r\_ctxt\_class}_{\sim N}$ $\quad \text{f\_ctxt\_class}$ $\text{r\_ctxt\_inst}_{\sim N}$ $\quad \text{f\_ctxt\_inst}$ $\quad \text{r\_elem\_E\_xx}_{\sim N}$ $\quad \text{f\_elem\_E\_xx}_{\sim xx}$
L1	$\text{r\_elem\_S\_xx}_{\sim N}$ $\quad \text{f\_elem\_S\_xx}_{\sim xx}$ $\quad \text{r\_elem\_S\_xx}_{\sim N}$ $\quad \text{f\_elem\_S\_xx}_{\sim xx}$ $\text{vel\_xx} = \langle \text{vel\_xx\_00} \vee \ldots \vee \text{vel\_xx\_N}_d \rangle$ $\quad \text{actn\_xx} = \langle \text{r\_actn\_xx} \vee \text{f\_actn\_xx} \rangle$
L0	$\text{vel\_xx\_xx} = \text{r\_vel\_xx\_xx}_{\sim xx}$ $\quad \text{r\_actn\_xx}$ $\quad \text{f\_actn\_xx}$

Fig. 5. LAPNN hierarchical memory layout

4 Simulation and Evaluation of Explainability

4.1 Data Preparation and Procedure

The data preparation code, HCPN and LAPNN models, and simulation code were all written in Python 3.10.12. The simulation was run on a PC with Ubuntu 22.04 installed.

For the simulation, the Collaborative Action Dataset for human motion forecasting [19] was chosen. The dataset features six different human subjects, each performing the following three tasks: 1) valve terminal plug n play, 2) valve assembly, and 3) collaborative soldering. Each task was carried out in a total of ten takes, and computer vision image frames were recorded for each take. The general directory tree structure for the dataset is shown in Fig. 6 (left). To prepare the data for the simulation, the dataset was modified to include a velocity entry for each timestamped frame. This was calculated using the relative x, y, z position data between consecutive frames as shown in Fig. 6 (center). A pool of DT HCPN nodes was created from Web of Things [31] Thing Description files, and each node object was initialized with structural and effective edges, and is illustrated for example nodes A and B in Fig. 6 (right). The pseudocode for the latter two procedures is given as Algorithms 1 and 2 in the Appendix. Finally, the simulation was carried out in two steps: a part-whole inference, and an is-a inference.

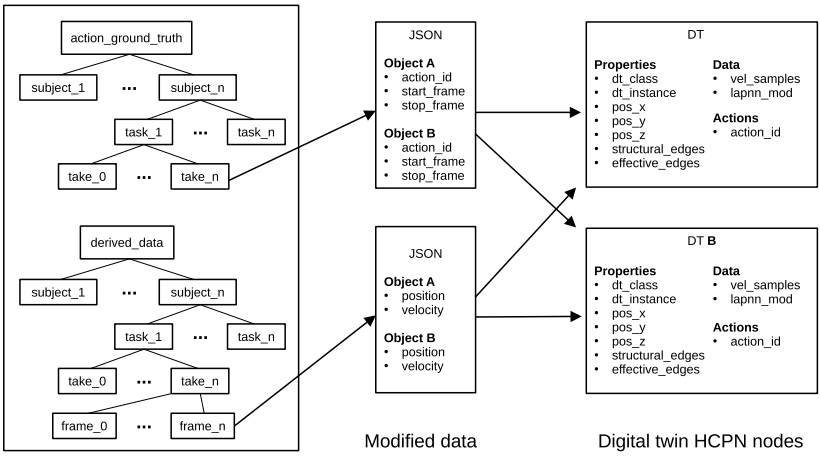


Fig. 6. Creation of example HCPN nodes A and B from CoAx dataset

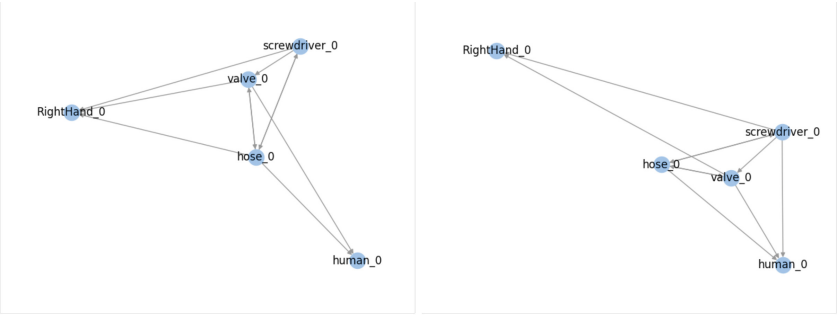
In the part-whole inference step, three unique HCPNs corresponding to the three tasks in the dataset, as well as their analogs (LAPNNs) were built for a single human subject in one take, for training. Another HCPN and its analog were built for another human subject, in a different take for testing. The LAPNN representation of the HCPN human node from the testing side was used as a probe to infer which HCPN it belongs to (to which *whole* HCPN the human *part* belongs) on the training side. This was a functionally-grounded evaluation [1] of explainability in the model, in that the program used a known parameter, adjusting the correlation threshold when building structural HCPN links.

In the is-a inference step, three versions of the same HCPN corresponding to one task, as well as their analogs, were built for training in a single take. Each HCPN had a different human subject. An HCPN corresponding to the same task, as well as its analog, was built for testing in another take. The HCPN was for one of the same three human subjects. The LAPNN functional component of the human node in the test HCPN was mapped and compared to the LAPNN functional components in each of the human nodes in the learned HCPNs (*is* the velocity vector similar to that of *a* known pattern?) seen for subject x. This was an application-grounded eveluation [1], because a domain expert (programmer) adjusted the human weight parameter of the correlation algorithm for a constant correlation threshold.

4.2 Part-Whole Inference

The tasks 1–3 were selected from the dataset for training. Pseudorandom numbers 5 and 6 were generated for the subject and the take, respectively. The training context was initialized, and the three structural HCPNs were created. The HCPN for task 1 is shown in Fig. 7a. Each Node's LAPNN module was gener-

ated, and all of the module nodes were aggregated into one composite LAPNN module for each HCPN.



(a) Subject 5, task 1 (CT = 0.5) (b) Subject 6, task 1 (CT = 0.5)

Fig. 7. Training and Testing HCPNs

Pseudorandom numbers generated for subject 6, take 2, and task 1. A test context was initialized and the HCPN (Fig. 7b) corresponding to task 1 was created for take 2. Its LAPNN modules and aggregate module node were generated as well.

The human node of the test HCPN was then used as a probe on the aggregated LAPNN module nodes for the three training HCPNs, for different values of the correlation threshold (CT), resulting in similarity scores x_1, x_2, and x_3. Probing the aggregated nodes was equivalent to the operations

$$\begin{aligned} &\textbf{human\_0}_{task\_1} \wedge \textbf{task\_1}, \\ &\textbf{human\_0}_{task\_1} \wedge \textbf{task\_2}, \text{ and} \\ &\textbf{human\_0}_{task\_1} \wedge \textbf{task\_3}. \end{aligned}$$

The correct class condition was true when the probe from the test context resulted in the similarity score corresponding to task 1 in the training context (x_1) being the highest. The results are shown in Table 1 with the deviation from the mean $x_1 - \bar{x}$ and standard deviation s.

The model correctly classified the test HCPN human each time. However, the CT value alone does not appear to have an effect on part-whole classification performance.

4.3 Is-A Inference

For the is-a training, the chosen human subjects were 4, 5, and 6. Pseudorandom numbers 2 and 9 were generated for the training task and take. For the test HCPN, task 2 was taken to be identical to that of the training case. Pseudorandom number 8 was generated for the test take, and 6 was generated for the test subject.

Table 1. Data for part-whole inference with correlation threshold CT

CT	Correct class	$x_1 - \bar{x}$	s
0.9	True	0.04636	0.04015
0.8	True	0.04654	0.04031
0.7	True	0.04658	0.04035
0.6	True	0.04611	0.03994
0.5	True	0.04584	0.03970
0.4	True	0.04592	0.03977
0.3	True	0.04592	0.03977
0.2	True	0.04584	0.03970
0.1	True	0.04640	0.04019
0.0	True	0.04585	0.03971

The CT value was set in code to be 0.9 for the remainder of the simulation. The training and test contexts were initialized, and the HCPNs and LAPNNs were generated in a manner similar to that in the part-whole step.

The functional LAPNN component of the human node in the test HCPN was then mapped to the corresponding components of the three human nodes in the training HCPN. It was used as a probe for different values of the human weight parameter (HW), to get a similarity score for each. The equivalent operations are

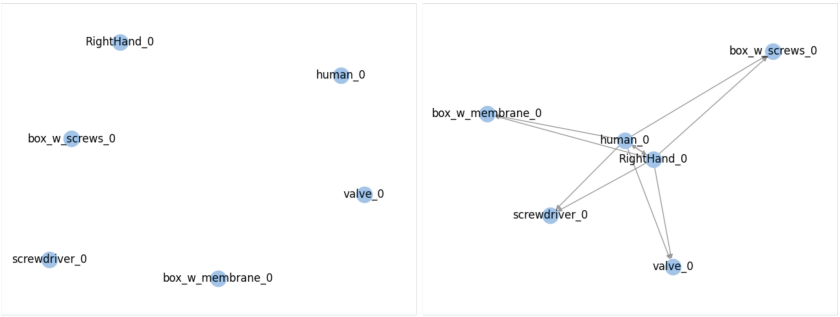
$$\mathbf{human\_0\_F}_{subject\_6} \wedge \mathbf{human\_0\_F}_{subject\_4},$$
$$\mathbf{human\_0\_F}_{subject\_6} \wedge \mathbf{human\_0\_F}_{subject\_5}, \text{ and}$$
$$\mathbf{human\_0\_F}_{subject\_6} \wedge \mathbf{human\_0\_F}_{subject\_6}.$$

As in the part-whole classification, the correct class condition was true for the similarity score correspong to subject 6, or x_3, to be the highest. The resulting data are shown in Table 2.

Table 2. Data for is-a inference with human weight parameter HW

HW	Correct class	$x_3 - \bar{x}$	s
0.00	True	0.00023	0.00040
0.25	True	0.00049	0.00051
0.50	False	−0.00019	0.00033
0.75	False	0.00000	0.00000
1.00	False	0.00000	0.00000
1.25	False	0.00000	0.00000
1.50	True	$4.24331e^{-5}$	$3.67482e^{-5}$
1.75	True	0.00015	0.00026
2.00	True	0.00049	0.00084

Interestingly, the test subject was classified correctly in training HCPNs where CT was high and HW was low, and where both CT and HW were high. In the first case, none of the nodes in the HCPN were connected (Fig. 8a). In the latter, the human-related nodes in the HCPN had a relatively higher number of links than the rest of the nodes (Fig. 8b). This suggests that when links exist in the network, classification performance is better when more are attached to a few nodes in the HCPN.



(a) Subject 6, task 2 (HW = 0.0) (b) Subject 6, task 2 (HW = 2.0)

Fig. 8. Training HCPNs

5 Conclusions and Future Work

In this paper, we proposed a brain-inspired model for human cyber physical networks (HCPNs), as well as a Laplacian associative-projective neural network (LAPNN).

We hypothesized that through symbolic representation of the HCPN Laplacian eigenvectors, analogical reasoning capability could be achieved. In a two-part experiment, it was found that analogical episodes, or analogs of HCPNs, can be built or *described* using correlated time series data for pairs of human, cyber, and physical object nodes. The LAPNN modules for each node were found to be capable of *retrieving* their whole network when used as probes in a distributed associative memory. Furthermore, the functional time-dependent changes in data associated with human-object interaction can be *mapped* to corresponding patterns between nodes. Thereafter, they can correctly identify (i.e. *infer*) a known human subject under certain conditions.

The test HCPN was easily identified among three learned HCPNs. However, this seems to be independent of the thresholding used in building structural links, as well as the average degree of the structural HCPNs. The functional activity component of the HCPNs can be used under certain conditions to identify human subjects, from their behavior patterns and, specifically, their

velocities within 3D space. This appears to be related to the degree distribution of the HCPNs.

One indication of weakness in the model is that the SBDRs for nodes in a given HCPN are quite similar to one another, compared to the strongly dissimilar vectors seen in other VSAs [15,26]. This is seen in the relatively low standard deviations in Tables 1 and 2. A possible cause is that the velocity vector representations (in level **L3** of the LAPNN memory) are highly thinned, and do not have roles associated with their filler values. This is left for future investigation.

The simulation in this paper brought to light possibilities for future development of the HCPN and LAPNN models. For example, functional HCPNs can very likely be created in order to learn and generalize dynamics or closed-loop control algorithms. It seems plausible that effective HCPNs can be used for human-grounded evaluations of the model's explainability with non-expert operators as well. For example, the modified dataset used in this paper could be generated through hand-guiding a collaborative robot, and the model's accuracy of the end-effector pose predictions could then be evaluated.

The evaluations carried out in this paper indicate that the analogical inference capability of the LAPNN in the simulation can be improved by interpretable methods from neuroscience and network science. We therefore answer the **RQ** from Sect. 1 in the affirmative, and conclude that explainability is indeed an emergent property of the proposed HCPN model.

Appendix

Algorithm 1

```
procedure READ_HCPNODES(directory_path)
    Initialize an empty list dt_hcpnode_pool
    for each file in directory_path do
        if file extension is '.json' then
            Load file as DT_HCPNode object
            Initialize node with data from file
            Append node to dt_hcpnode_pool
        end if
    end for
    Create a ranking dictionary to order nodes
    for each node in dt_hcpnode_pool do
        Compute rank
        Store node and rank in the ranking dictionary
    end for
    Sort dt_hcpnode_pool based on rank
    Return sorted dt_hcpnode_pool
end procedure
```

Algorithm 2

procedure DATASET_TO_CONTEXT(subject, task, take, ground_truth_dir, hcpnode_pool, weight, threshold)
    Define overlap frame range for action transitions
    Construct file paths for ground truth and derived data
    Load ground truth data from JSON file
    for each action sequence in ground truth **do**
        Identify start and stop frames
        for each node in hcpnode_pool **do**
            Match node class with ground truth class
            Assign effective edges based on matched actions and objects
        end for
        Assign velocity data from derived files to the corresponding nodes:
         Thin out data arrays exceeding 60 elements
         Attach velocity data to node's temporary storage
        Identify structural edges using velocity correlation:
         Use multithreading to process each node's connections
         Apply correlation thresholds to determine neighbors
    end for
    Return updated hcpnode_pool with structural and effective edges
end procedure

References

1. Barabási, A.L.: Network Science. Cambridge University Press, Cambridge (2016). Illustrated edition
2. Barsalou, M.A., Smith, J.: Applied Statistics Manual: A Guide to Improving and Sustaining Quality with Minitab. ASQ Quality Press, Milwaukee (2018)
3. Bullmore, E., Sporns, O.: Complex brain networks: graph theoretical analysis of structural and functional systems. Nat. Rev. Neurosci. **10**(3), 186–198 (2009). https://doi.org/10.1038/nrn2575, https://www.nature.com/articles/nrn2575
4. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning (2017). https://doi.org/10.48550/arXiv.1702.08608, arXiv:1702.08608
5. Dreher, C.R.G., Wächter, M., Asfour, T.: Learning object-action relations from bimanual human demonstration using graph networks. http://arxiv.org/abs/1908.08391, arXiv:1908.08391 [cs] (2019)
6. Emruli, B., Sandin, F., Delsing, J.: Vector space architecture for emergent interoperability of systems by learning from demonstration. Biol. Inspir. Cogn. Archit. **11**, 53–64 (2015). https://doi.org/10.1016/j.bica.2014.11.015, https://www.sciencedirect.com/science/article/pii/S2212683X14000784
7. Furlong, P.M., Eliasmith, C.: Modelling neural probabilistic computation using vector symbolic architectures. Cogn. Neurodyn. (2023). https://doi.org/10.1007/s11571-023-10031-7
8. Gayler, R.W.: Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. arXiv:cs/0412059 (2004)
9. Gentner, D., Smith, L.: Analogical reasoning. In: Encyclopedia of Human Behavior, pp. 130–136. Elsevier (2012). https://doi.org/10.1016/B978-0-12-375000-6.00022-7, https://linkinghub.elsevier.com/retrieve/pii/B9780123750006000227

10. Goel, A.K., Rugaber, S., Vattam, S.: Structure, behavior, and function of complex systems: the structure, behavior, and function modeling language. Artif. Intell. Eng. Design Anal. Manuf. **23**(1), 23–35 (2009). https://doi.org/10.1017/S0890060409000080, https://www.cambridge.org/core/product/identifier/S0890060409000080/type/journalarticle
11. Golovianko, M., Terziyan, V., Branytskyi, V., Malyk, D.: Industry 4.0 vs. industry 5.0: co-existence, transition, or a hybrid. Procedia Comput. Sci. **217**, 102–113 (2023). https://doi.org/10.1016/j.procs.2022.12.206, https://www.sciencedirect.com/science/article/pii/S1877050922022840
12. Greff, K., van Steenkiste, S., Schmidhuber, J.: On the binding problem in artificial neural networks. arXiv:2012.05208 [cs] (2020)
13. Haga, T., Fukai, T.: Multiscale representations of community structures in attractor neural networks. PLOS Comput. Biol. **17**, e1009296 (2021). https://doi.org/10.1371/journal.pcbi.1009296
14. Hamilton, W.L.: Graph representation learning. Synthesis Lect. Artif. Intell. Mach. Learn. **14**(3), 1–159 (2020)
15. Kanerva, P.: Binary spatter-coding of ordered *K*-tuples. In: von der Malsburg, C., von Seelen, W., Vorbrüggen, J.C., Sendhoff, B. (eds.) ICANN 1996. LNCS, vol. 1112, pp. 869–873. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61510-5_146
16. Kempitiya, T., Alahakoon, D., Osipov, E., Kahawala, S., De Silva, D.: A two-layer self-organizing map with vector symbolic architecture for spatiotemporal sequence learning and prediction. Biomimetics **9**(3), 175 (2024). https://doi.org/10.3390/biomimetics9030175, https://www.mdpi.com/2313-7673/9/3/175
17. Kleyko, D., et al.: Vector symbolic architectures as a computing framework for emerging hardware. Proc. IEEE Inst. Electr. Electron. Eng. **110**(10), 1538–1571 (2022). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10588678/
18. Kleyko, D., Rachkovskij, D., Osipov, E., Rahimi, A.: A survey on hyperdimensional computing aka vector symbolic architectures, part II: applications, cognitive models, and challenges. ACM Comput. Surv. **55**(9), 175:1–175:52 (2023). https://doi.org/10.1145/3558000, https://dl.acm.org/doi/10.1145/3558000
19. Lagamtzis, D., Schmidt, F., Seyler, J., Dang, T.: CoAx: collaborative action dataset for human motion forecasting in an industrial workspace. In: Proceedings of the 14th International Conference on Agents and Artificial Intelligence, pp. 98–105. SCITEPRESS - Science and Technology Publications, Online Streaming (2022). https://doi.org/10.5220/0010775600003116, https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0010775600003116
20. Levy, S., Bajracharya, S., Gayler, R.: Learning behavior hierarchies via high-dimensional sensor projection. In: AAAI Workshop - Technical Report, pp. 25–27 (2013)
21. Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable AI: a review of machine learning interpretability methods. Entropy **23**(1), 18 (2020). https://doi.org/10.3390/e23010018, https://pmc.ncbi.nlm.nih.gov/articles/PMC7824368/
22. Mohar, B.: Laplace eigenvalues of graphs-a survey. Discrete Math. **109**(1), 171–183 (1992). https://doi.org/10.1016/0012-365X(92)90288-Q, https://www.sciencedirect.com/science/article/pii/0012365X9290288Q
23. Murakami, M., Kominami, D., Leibnitz, K., Murata, M.: Drawing inspiration from human brain networks: construction of interconnected virtual networks. Sensors **18**(4), 1133 (2018). https://doi.org/10.3390/s18041133, https://www.mdpi.com/1424-8220/18/4/1133

24. Neubert, P., Schubert, S., Protzel, P.: An introduction to hyperdimensional computing for robotics. KI - Künstliche Intell. **33**(4), 319–330 (2019). https://doi.org/10.1007/s13218-019-00623-z
25. Ordieres-Meré, J., Gutierrez, M., Villalba-Díez, J.: Toward the industry 5.0 paradigm: increasing value creation through the robust integration of humans and machines. Computers in Industry **150**, 103947 (2023). https://doi.org/10.1016/j.compind.2023.103947, https://www.sciencedirect.com/science/article/pii/S0166361523000970
26. Plate, T.A.: Distributed representations and nested compositional structure. Ph.D., University of Toronto, Canada (1994). aAINN97247 ISBN-10: 0315972475
27. Purdy, S.: Encoding data for HTM systems (2016). https://doi.org/10.48550/arXiv.1602.05925, arXiv:1602.05925 [cs, q-bio]
28. Rachkovskij, D.: Representation and processing of structures with binary sparse distributed codes. IEEE Trans. Knowl. Data Eng. **13**, 261–276 (2001). https://doi.org/10.1109/69.917565
29. Rachkovskij, D., Kussul, E., Baidyk, T.: Building a world model with structure-sensitive sparse binary distributed representations. Biol. Inspir. Cogn. Archit. **3**, 64–86 (2013). https://doi.org/10.1016/j.bica.2012.09.004
30. Rachkovskij, D.A., Kussul, E.M.: Binding and normalization of binary sparse distributed representations by context-dependent thinning. Neural Comput. **13**(2), 411–452 (2001). https://doi.org/10.1162/089976601300014592, https://direct.mit.edu/neco/article/13/2/411-452/6479
31. Tzavaras, A., Mainas, N., Petrakis, E.G.: Thing ontologies for the semantic web of things. In: 2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA), pp. 1–8. IEEE (2022)
32. Villalba-Diez, P.: The Lean Brain Theory: Complex Networked Lean Strategic Organizational Design. Taylor & Francis Ltd., Boca Raton (2017)
33. Villalba-Díez, J., Molina, M., Ordieres-Meré, J., Sun, S., Schmidt, D., Wellbrock, W.: Geometric deep lean learning: deep learning in industry 4.0 cyber-physical complex networks. Sensors **20**(3), 763 (2020). https://doi.org/10.3390/s20030763, https://www.mdpi.com/1424-8220/20/3/763
34. Villalba-Díez, J., Ordieres-Meré, J., Rubio-Valdehita, S.: The lean brain theory. brain-like lean manufacturing systems. Procedia CIRP **57**, 140–145 (2016). https://doi.org/10.1016/j.procir.2016.11.025, https://www.sciencedirect.com/science/article/pii/S2212827116311787
35. Wang, B., Zheng, P., Yin, Y., Shih, A., Wang, L.: Toward human-centric smart manufacturing: a human-cyber-physical systems (HCPS) perspective. J. Manuf. Syst. **63**, 471–490 (2022). https://doi.org/10.1016/j.jmsy.2022.05.005, https://linkinghub.elsevier.com/retrieve/pii/S0278612522000759
36. Wang, X., Mieghem, P.V.: Orthogonal eigenvector matrix of the laplacian. In: 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), pp. 358–365. IEEE, Bangkok (2015). https://doi.org/10.1109/SITIS.2015.35, http://ieeexplore.ieee.org/document/7400588/
37. Wieselthier, J., Gam, D., Ephremides, A.: On the construction of energy-efficient broadcast and multicast trees in wireless networks, vol. 2, pp. 585–594 (2000). https://doi.org/10.1109/INFCOM.2000.832232
38. Zhang, C., et al.: Towards new-generation human-centric smart manufacturing in industry 5.0: a systematic review. Adv. Eng. Inform. **57**, 102121 (2023). https://doi.org/10.1016/j.aei.2023.102121, https://www.sciencedirect.com/science/article/pii/S1474034623002495

39. Zhang, X., Sheng, V.S.: Neuro-symbolic AI: explainability, challenges, and future trends (2024). https://doi.org/10.48550/arXiv.2411.04383, arXiv:2411.04383 [cs]
40. Zhou, J., Zhou, Y., Wang, B., Zang, J.: Human–cyber–physical systems (HCPSs) in the context of new-generation intelligent manufacturing. Engineering **5**(4), 624–636 (2019). https://doi.org/10.1016/j.eng.2019.07.015, https://linkinghub.elsevier.com/retrieve/pii/S2095809919306514
41. Zhou, J.: Graph neural networks: a review of methods and applications (2020)