# A Brain-Inspired Hierarchical Reasoning Framework for Cognition-Augmented Prosthetic Grasping


## Abstract

This paper proposes a hierarchical framework that enables reasoning across multiple levels of abstraction, from perception to high level control. The framework relies primarily in Hyperdimensional (HD) computing, a versatile brain inspired computing paradigm that can produce composite distributed representations for efficient classification, and can also facilitate symbolic reasoning. We present this framework through a use case corresponding to cognition-augmented prosthetic grasping, where intent is probabilistically predicted in order to aid the user in successfully selecting and executing the most appropriate gripping mode. Experiments show that HD computing can constitute a mathematical formalism capable of integrating heterogeneous aspects of cognition under a common framework.


## Introduction

Data-driven artificial intelligence (AI) and machine learning methods have demonstrated great success at perception-based benchmarks, sparking great interest in their adoption within various disciplines, including biomedical engineering, robotics, and edge computing. Yet, the deployment of AI in practical scenarios may present heterogeneous requirements that range from the inclusion of uncertainty-capturing models of the world to the incorporation of expert knowledge and symbolic reasoning over it. This calls for mathematical formalisms and frameworks that can integrate heterogeneous algorithms and representations.

Furthermore, human cognition is hierarchical by nature. At the low level, cognition is established by the integration of massive sensory data from disparate sources. High level cognition involves abstract reasoning supported by preconceived notions of the world's properties. This hierarchical scheme ultimately drives human *action*, which, in turn, restarts the sense-reason-act closed-loops human brains are uniquely endowed with.

This paper posits that Hyperdimensional (HD) computing (Kanerva 2009), known also as Vector Symbolic Architectures (VSA) (Gayler 2004), constitutes a mathematical formalism capable of integrating the heterogenous aspects of cognition described above. HD computing builds upon a carefully designed algebra of vectors, that enables symbolic computations as well as the composite encoding of heterogenous information. Moreover, due to its high-capacity and distributed nature, HD computing is robust against noise (Rahimi et al. 2018) and can be memory efficient (Zhou, Muller, and Rabaey 2020).

The posture above is motivated by the multiple successful implementations of HD computing available in literature; earlier on in its development focusing primarily on symbolic reasoning (Gayler 1998; Kanerva 2009), and more recently demonstrating its usefulness for various perception tasks in the fields of robotics (Neubert, Schubert, and Protzel 2019) and biosignal processing and classification (Rahimi et al. 2018). For perception taks, in particular, HD computing enables the compact encoding of composite spatio-temporal and frequency distribution structures by exploiting the properties of high-dimensional binary and bipolar vector spaces (Kleyko et al. 2021). Classification often takes place by means of a simple nearest-centroid search, with the class-centroids also represented as high dimensional vectors and stored in an *associative memory*.

In this paper, we explore the capabilities of HD computing as a formalism to integrate classification, probabilistic inference and symbolic reasoning within a single hierarchical framework. We envision that this framework can enable reasoning and learning across different levels of abstraction by fully describing it in the HD space. The contributions of this paper can be summarized as follows:

**Proposal and illustration of a hierarchical reasoning framework**. Through a synthetic prosthetic grasping example, we illustrate how the proposed framework can hierarchically represent the different levels of abstraction of the application: at the lowest (perception) level it relies on biosignal classification; at an intermediate layer it relies on a Dynamic Bayesian Network (DBN) to infer user intent over time; and at the highest layer it relies on symbolic reasoning for control and feedback.

**Mapping of a probabilistic graphical model to HD space.** We demonstrate how to map the DBN mentioned above to the HD space. We show that this mapping allows us to store the conditional probability tables (CPTs) of the model in superposition, with the potential of reducing memory footprint, and enabling an end-to-end HD representation for the proposed framework.


Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

**Discussion of the implications and extensions of the proposed framework** We discuss the implications of the current framework and detail future work directions on 1) the relation between the symbolic reasoning layer of the framework and it's potential to enable continual learning, 2) the potential inclusion of state-of-the-art tractable probabilistic models within the framework and 3) our view of how this framework will perform in other applications and benchmarks.

## Hyperdimensional computing

Inspired by the understanding that the brain's computations rely on massive circuits of neurons and synapses, HD computing operates on pseudo-random hyperdimensional vectors through a well defined computational algebra of vectors, different from linear algebra (Kanerva 2009). Pairing the aforementioned representation and algebra with an associative memory yields a powerful and versatile architecture that enables efficient learning and reasoning.

### Main components

HD computing can materialize through different models, each with their own set of properties. For an extensive list see, for example (Rahimi et al. 2017). In general, all HD computing instances rely on the following components (Neubert, Schubert, and Protzel 2019):

**Hyperdimensional vector spaces** The computational primitives are vectors with dimensions higher than $10^2$, denoted here as $\mathbb{V}$. Depending on the model, these vectors can be real-numbered tensors (Smolensky 1990), real and complex valued vectors (Plate 1994), binary vectors (Kanerva et al. 1997) or bipolar vectors (Gayler 1998). In this paper we use the Multiply Add Permute model (Gayler 1998), which relies on the latter bipolar representation. HD computing exploits the properties of hyperdimensional spaces, such as high-capacity, the near orthogonality of random vectors and robustness to noise.

**HD computing operations** HD computing relies on three main operations:

- Bundling $+$: Combines two input vectors into a *similar* output vector.
- Binding $\otimes$: Combines two input vectors into a *dissimilar* (nearly orthogonal to the inputs) output vector. Binding is associative and commutative and distributes over bundling. Finally, binding an HD vector with itself results in the unit vector.
- Permutation (also called protect) $\rho$: Permuting a vector produces a dissimilar vector to the original one. This operation is often used to "protect" ordering that could be lost due to the associative and distributive rules of binding. Note that permutations distribute over the previous two operations.

**Item Memory (I.M.)** Stores the randomly generated *seed* HD vectors that will be used for the rest of the computation. As such, the item memory functions as a codebook, where each HD vector has an assigned meaning and therefore constitutes a symbol.

### Data structures

Through the components described above, HD computing offers the possibility of encoding data structures into compound HD vectors. Additional computations on these data structures can be performed through the operations described above (Kleyko et al. 2021). Examples of common data structures are:

*1) Key,value pairs*: Binding can be used to represent associations between elements in a key-value pair / variable (categorical) value fashion. Such a relation could, for example be a prosthetic user's activities of daily living and their location: $\{\mathrm{ACT}\otimes\mathrm{cook}, \mathrm{LOC}\otimes\mathrm{kitchen}\}$, where all the components are random HD vectors saved in item memory, the key is denoted with upper case and the value or filler with lower case.

*2) Sets and histograms*: Sets can be represented by binding the HD vectors corresponding to the elements in the set. For example, the set $S = \{d, c, b, a\}$ can be represented in HD space through bundling as $\mathrm{S} = \mathrm{d} + \mathrm{c} + \mathrm{b} + \mathrm{a}$, where the elements of S are seed HD vectors stored in item memory. For sets with repeated elements, such as $H = \{a, a, b, c, c, c\}$, a frequency distribution can also be represented through bundling as $\mathrm{H} = \mathrm{a} + \mathrm{a} + \mathrm{b} + \mathrm{c} + \mathrm{c} = 2\mathrm{a} + \mathrm{b} + 3\mathrm{c}$. The frequency of a specific HD vector, for example, $\mathrm{a}$ can be approximately recovered from H through their dot product: $dot(\mathrm{H}, \mathrm{a})/D \sim 2$, where $D$ denotes vector dimension.

*3) Sequences*: The ordering of the components in a sequence can be preserved through the permutation operation $\rho$. As such, the sequence $Z = (a, b, c)$ can be represented in HD space as $\mathrm{Z} = \rho^2(\mathrm{a}) + \rho^1(\mathrm{b}) + \rho^0(\mathrm{c})$, where $\rho^n(\mathrm{v})$ denotes that v is permuted $n$ times.[1]

### Parsing representations and queries

The data structures above can be combined to create compound representations (Kleyko et al. 2021). For example, suppose User1 commonly cooks at 4 pm, then their data record is specified as $\mathrm{D}_{1,4\mathrm{pm}} = \mathrm{ACT} \otimes \mathrm{cook} + \mathrm{LOC} \otimes \mathrm{kitchen}$. In contrast, User2 tends to work out at 4 pm such that their record is $\mathrm{D}_{2,4\mathrm{pm}} = \mathrm{ACT} \otimes \mathrm{run} + \mathrm{LOC} \otimes \mathrm{gym}$. To probe which activity User1 performs at 4 pm, we can *unbind*[2] the ACT key from the $\mathrm{D}_{1,4\mathrm{pm}}$ record, since:

$$\begin{aligned}\mathrm{D}_{1,4\mathrm{pm}} \otimes \mathrm{ACT} &= \mathrm{ACT} \otimes (\mathrm{ACT} \otimes \mathrm{cook} + \mathrm{LOC} \otimes \mathrm{kitchen}) \\ &= \mathrm{cook} + \mathrm{ACT} \otimes \mathrm{LOC} \otimes \mathrm{kitchen} \\ &= \mathrm{cook} + noise \approx \mathrm{cook},\end{aligned}$$

since the term $\mathrm{ACT} \otimes \mathrm{LOC} \otimes \mathrm{kitchen}$ is dissimilar to the seed vectors in I.M. and can therefore be regarded as noise. The primitive $\mathrm{cook}$ can then be recovered by choosing the vector in I.M. with the highest similarity to $\mathrm{cook} + noise$. Thus, this probing operation "cleans up" the noise of the compound representation.

## Proposed hierarchical framework

The proposed hierarchical framework is illustrated in Figure 1 and corresponds to a *cognition-augmented* prosthetic grasping use case.

[1] In practice, permutation commonly consists of cyclic shifts.

[2] in the Multiply-Add-Permute model user in this paper, unbinding uses the same operation as binding.

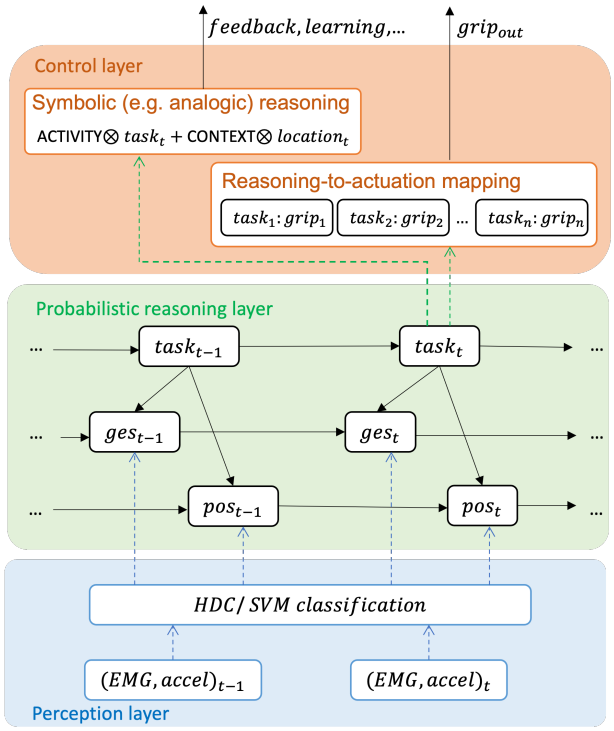


Figure 1: Proposed hierarchical framework.

## Use case description

This synthetic use case is intended to augment the EMG-based gesture recognition application in (Moin et al. 2021) and (Zhou, Muller, and Rabaey 2020). The goal of the cited works is to classify EMG (and accelerometer in the latter) data into a set of gestures, which can be used by prosthesis users to select the appropriate *grip* mode (Figure 3.b shows an example of gesture to grip mappings). Our proposed used case is motivated by the observation that many prosthesis users struggle to remember and select the gesture-to-grip mapping while trying to engage in their daily activities . Therefore, we propose to augment this application beyond gesture classification by incorporating user-intent prediction, which can ultimately be used in aiding the prosthesis user throughout the grip selection and execution process. Moreover, as described later on in the paper, the higher level specification of user intent can be used for analogical reasoning towards identification of and adaptation to new environments.

## Hierarchical reasoning framework

The proposed framework comprises three layers:

**1) Perception layer** The role of the perception layer, highlighted in blue, is to map heterogeneous sensory inputs to higher level descriptive concepts, such as hand-gesture and arm position. Therefore, for this layer, we reuse the representations and models proposed in (Zhou, Muller, and Rabaey 2020). In this work, EMG signals are windowed over 50 milisecond sections, and are mapped to spatiotemporal HD vector representations. The input "query" vectors are then mapped to specific gestures by means of a nearest centroid classifier. The centroids in the model are also specified in HD space by means of class-prototype HD vectors, and classification takes place by selecting the class that maximizes the similarity (measured by Hamming distance) between its prototype and the query vector. Accelerometer data is used to classify the user's arm position through a Support Vector Machine (SVM). In (Zhou, Muller, and Rabaey 2020), this position information is used to improve gesture classification performance in a dual-stage classification fashion. For our framework, we feed the classified arm position as well as the classified gesture to the following layer.

**2) Probabilistic reasoning layer** The second layer of abstraction, highlighted in green, builds a high-level probabilistic world model of the application. In our use case, we realize this model with a DBN that encodes the probabilistic relations among the tasks performed by the user and the observed hand-gestures and arm-positions. This DBN makes a first order Markov assumption for the temporal relations among tasks, gestures and positions.

The probabilistic reasoning layer in our example framework performs a filtering on this DBN. That is, it computes the posterior distribution over the task at time $t$, given all evidence over $ges$ and $pos$ to date. That is, we comput $\Pr(task_t|\mathbf{e}_{0:t})$, with $\mathbf{e} = \{ges, pos\}$. In practice, the inference algorithm maintains a belief over the task at time $t$ ($\Pr(task_t|\mathbf{e}_{0:t})$) and updates it at time $t+1$ given the evidence over $t+1$ and $t$ in a recursive fashion (Russell and Norvig 2002):

$$\begin{aligned}\Pr(task_{t+1}|ges_{0:t+1}, pos_{0:t+1}) = \alpha \Pr(ges_{t+1}|task_{t+1}, ges_t)\cdot \\ \Pr(pos_{t+1}|task_{t+1}, pos_t) \sum_{task_t} \Pr(task_{t+1}|task_t)\cdot \\ \Pr(task_t|ges_{0:t}, pos_{0:t}),\end{aligned} \tag{1}$$

where $Pr(task_t|ges_{0:t}, pos_{0:t})$ is calculated in the previous iteration with evidence at time $t$ and $t-1$, $task_t$ is marginalized, and $\alpha$ is a normalization constant that ensures that the posteriors sum to 1.

We map the DBN to HD space[3] by using the data structures discussed in the HD computing Section. First, we generate random HD vector seeds for each value of the $task$, $ges$ and $pos$ variables, and store them in the Value-I.M. Then, as showed in Figure 2, we randomly project the probability of each value to HD space. To achieve this, we perform the scalar multiplication of each probability value (scaled by e.g. $10^3$ such that we only store integers) times a randomly generated HD vector. Then we bind this real-valued vector with the HD vectors corresponding to the values of the conditions (e.g. conditioned on $task_1$ and $ges_1$). Note that, in this example, we apply the permute operation to $ges_1$ in order to signal that this variable took place at time $t-1$. Finally, we superimpose (through bundling) all the conditional probability vectors, such that we only store a single HD vector per CPT in the model.

[3]The method of mapping a model to the HD space is inspired by the model superposition method proposed in (Cheung et al. 2019).

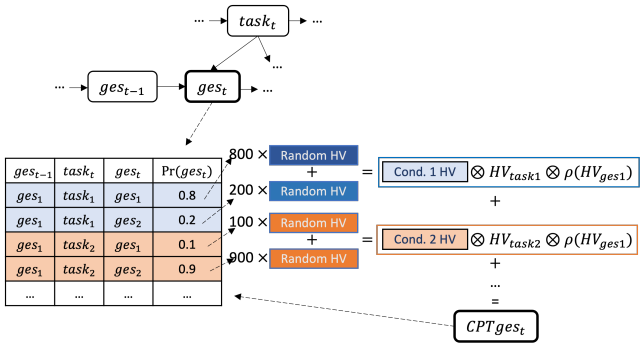


Figure 2: Mapping of a CPT to HD assuming binary variables in the DBN.

At inference time, the probabilities are recovered from the superimposed CPTs by unbinding them from their condition (which will normally be the observed or marginalized variables), taking the dot product between the HD vector corresponding to the value of interest, and dividing by the dimension of the HD space (as discussed for histograms in the data structure section). For example, starting from the CPT illustrated in Figure 2, the value of $\Pr(ges_t = ges_1 | task_t = task_2, ges_{t-1} = ges_1)$ can be recovered through $dot(CPT_{ges_t} \otimes HV_{task_2} \otimes \rho(HV_{ges_1}), HV_{ges_1})/(1000 \cdot D)$, where 1000 aounts for the scaling factor used during the mapping (whose purpose is to facilitate integer storage). Inference can be again performed with equation 1.

Despite a potential information loss, the motivation behind mapping this probabilistic model to the HD space is two fold: 1) The potential for memory efficiency for large models, since space complexity of each CPT remains constant. We leave the analysis of HD vector storage capacity as well as a precise quantification of the memory efficiency for future work, but prior work has found that storing in superposition can lead to resource savings (Zhou, Muller, and Rabaey 2020; Cheung et al. 2019). 2) The second motivation for this mapping is that encoding the entire framework with the same primitives can be advantageous, as it can potentially enable reasoning and, eventually, learning across all levels of abstraction including the Control Layer described next.

**3) Control layer** The last layer serves as an *interface* between the proposed framework and other external components of the system. There are several ways in which this layer can be realized, depending on the needs of the system and the application. This versatility is further facilitated by the HD representation:

- Symbolic reasoning: for example, to transfer knowledge to unknown contexts and environment through analogical reasoning. Subsection *Context Knowledge Transfer Using Symbolic Reasoning* within the experimental section discusses this idea in detail.
- Reasoning to actuation (input-output) mapping: the predicted user intent (i.e. belief in the set of tasks at time $t$ given evidence) can, for example, be used to fine tune the output grip selection. One way of doing that in HD space is through recall of reactive behaviour (Neubert, Schubert, and Protzel 2017). In this technique, a representation encodes sensor-action pairs, such that behavior can be resembled at run-time by extracting the action corresponding to the current sensor value. The posterior distribution can further be used to weight this recall representation, an interesting direction that we leave for future work.

# Experiments

## Experimental setup and dataset

We validated our approach with the dataset in (Zhou, Muller, and Rabaey 2020), which includes measurements from a 64-channel EMG electrode array with an onboard tri-axial accelerometer, shown in Figure 3.a. We consider 3 hand-gestures and 4 arm positions, as illustrated in 3.b and 3.c. In addition, we define 3 possible tasks based on common activities of daily living: brushing hair, tying shoelace and open jar. Sequential data over the different tasks is not available in (**?**), therefore, we generate a 10000 sample sequential dataset by sampling an ideal DBN modeled with expert knowledge and we use a random 70%, 30% train-test split over 10 trials to evaluate the performance of the first two layers of the proposed framework.

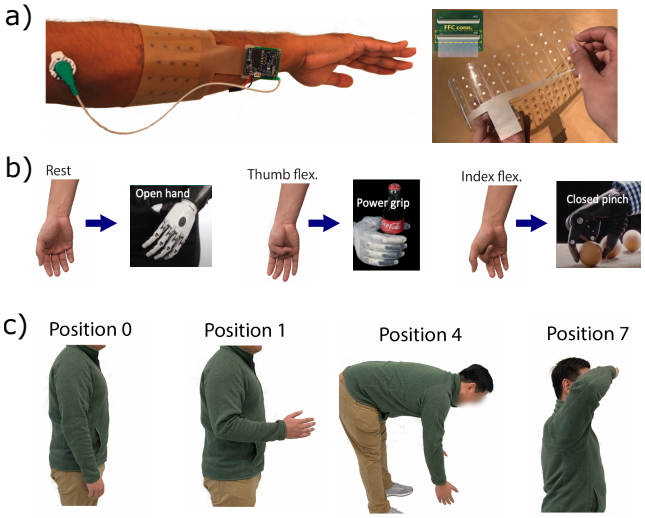


Figure 3: Experimental setup, based on (Zhou, Muller, and Rabaey 2020; Moin et al. 2021) .

## Task description

The task we evaluated included predicting the most likely task at time $t$, given evidence over $ges$ and $pos$ over time. That is, we evaluate the accuracy of predicting the most likely $task_t$ after evaluating the filtering algorithm described above. Note that the evidence variables in the DBN, $ges$ and $pos$, are classified at the perception layer using HD computing nearest centroids and an SVM classifier, respectively, as described before.

Moreover, we evaluate the performance of two DBNs with different structures: the *complex* model has the same structure as the modelled depicted in Figure 1, and the *simple* model ignores the time relationships among gestures and positions, that is, $ges_t$ and $pos_t$ only depend on $task_t$.

## Results

Figure 4 shows an example of the posterior distributions evaluated with each of the aforementioned DBNs (simple in blue and complex in red) when using a standard tabular CPT representation (cross markers) and when mapping CPTs to HD space (circle markers). The real distribution (from the DBN we sampled to generate the datasets) is shown with a black dotted line. It is clear that the posterior calculated by the complex model is closer to the real distribution. However, note that the HD space mapping of the complex model degrades more than in the case of the simple model.

This is expected due to the limited superposition capacity of the HD representation, which is strained more by the complex than by the simple model.

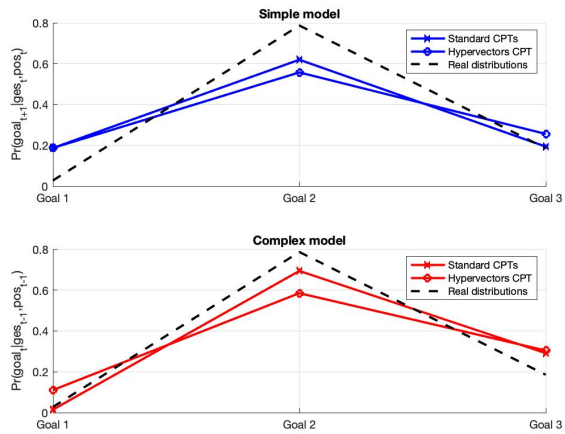


Figure 4: Examples of posterior distributions

Figure 5 compares the task prediction accuracy of all the models. Overall, filtering accuracy is preserved throughout the 10 time slices shown, and the complex model shows superior performance than the simple one. Yet, once again, the HD representation loses nearly 3% accuracy for the complex model, whereas the simple model handles the HD representation without any accuracy degradation.

Figure 5

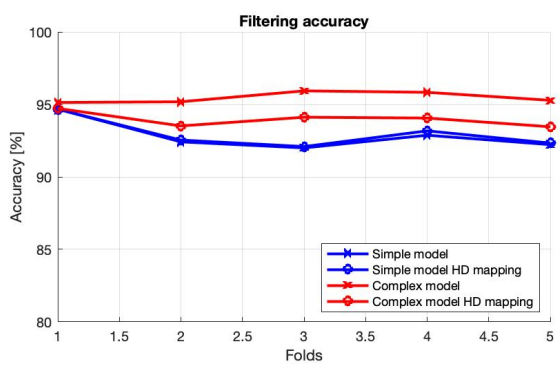


Figure 5: Filtering accuracy over 10 time slices for all the models.

These experiments show there is a clear trade-off between model complexity and HD model encoding capabilities. An interesting future work direction would be to systematically quantify the superposition capacity of HD vectors when used to represent graphical probabilistic models such as the DBN studied herewith.

## Context Knowledge Transfer Using Symbolic Reasoning

The Control Layer of the proposed framework can allow us to deduce generalizable outcomes that can be carried from one context to another. This is especially important because learning is typically performed based on sporadic samples that are attributed to a limited number of contexts.

This section presents some examples of how this capabilities can be exploited for the considered use case.

In most practical contexts, learning is performed based on limited samples generated from a fixed set of sensors. Most commonly, these samples are available under a limited number of contexts. For example, in this paper's use case, data might only be available for the interactions of a prosthetic arm with objects in a kitchen.

However, partial groups of training samples generated from different contexts can be mutually exclusive. In other words, some samples may exhibit common features with respect to learnt tasks among different contexts, whereas other samples lead to different task definitions depending on the context. The question is: how can we optimize our learning method such that it can systematically transfer task definitions between contexts so that our system can adapt to unknown settings in some contexts?

For example, suppose the DBN in Figure 1 was trained solely with samples from a kitchen setting (denoted in HD space as $\mathrm{kit}$). Suppose that the DBN's current prediction of $task_t =$ is opening jar (denoted as $\mathrm{oj}$). Imagine we did not have access to training data from an "front door" location ($\mathrm{fd}$), such that the DBN may not be able to generalize to this context. However, we know that the twisting motion sensed by the EMG in the open jar case may be similar to the twisting motion sensed in the open door case ($\mathrm{od}$) for the front door setting. Therefore, instead of attempting to predict the most likely activity with the "front-door unaware DBN", we can rely on our high-level analogical knowledge. This example follows a similar analogical reasoning example corresponding to answering the query "What is the Dollar of Mexico"? (Kanerva 2010).

We can represent the two scenarios through the following record HD vectors:

$$\mathrm{H}_{\mathrm{context1}} = \mathrm{ACT} \otimes \mathrm{oj} + \mathrm{LOC} \otimes \mathrm{kit}$$
$$\mathrm{H}_{\mathrm{context2}} = \mathrm{ACT} \otimes \mathrm{od} + \mathrm{LOC} \otimes \mathrm{fd}$$

Then we can bind the two records such that the values occupying the same keys are bound:

$$\mathrm{H}_{\mathrm{context1} \otimes \mathrm{context2}} = \mathrm{oj} \otimes \mathrm{od} + \mathrm{kit} \otimes \mathrm{fd}.$$

Then we can ask the following query "what is the equivalent task to open jar in a front door environment?" by probing the above structure with open jar HD vector:

$$\begin{aligned} \mathrm{H}_{\mathrm{context1} \otimes \mathrm{context2}} \otimes \mathrm{oj} &= (oj \otimes od + kit \otimes fd) \otimes oj \\ &= oj \otimes od \otimes oj + noise \approx od. \end{aligned} \quad (2)$$

We can therefore conclude that the analogous to opening jar in a new "front door" environment is opening door.

These symbolic reasoning mechanisms could further be used to provide feedback to the probabilistic model for continual learning and adaptation. Moreover, the analogical reasoning example explained here could be exploited when training data collection is too costly and expert knowledge is available.

## Discussion and future work

### Implications of the proposed framework

- The proposed probabilistic model within our framework is rather simple, and, as such, we were able to store it in superposition without significant performance degradation. For real life applications, it will be necessary to perform more complete analyses on the capacity of the models.

- Overall, the considered use case is simple, but it opens the door for further investigation into more complete hierarchical frameworks.

### Future work

- We'd like to study our framework with other datasets and benchmarks corresponding to real life scenarios.
- Compiling the probabilistic model to a more tractable representation (e.g. probabilistic circuits), is an interesting direction to follow next. We would also like to evaluate whether this type of model can be mapped to HD space like in the DBN case.
- Beyond analogical reasoning, this framework can benefit from other forms of symbolic reasoning.

## References


Cheung, B.; Terekhov, A.; Chen, Y.; Agrawal, P.; and Olshausen, B. 2019. Superposition of many models into one. *arXiv preprint arXiv:1902.05522*.

Gayler, R. W. 1998. Multiplicative binding, representation operators & analogy (workshop poster).

Gayler, R. W. 2004. Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. *arXiv preprint cs/0412059*.

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

Kanerva, P. 2010. What we mean when we say" What's the dollar of Mexico?": Prototypes and mapping in concept space. In *2010 AAAI fall symposium series*.

Kanerva, P.; et al. 1997. Fully distributed representation. *PAT*, 1(5): 10000.

Kleyko, D.; Davies, M.; Frady, E. P.; Kanerva, P.; Kent, S. J.; Olshausen, B. A.; Osipov, E.; Rabaey, J. M.; Rachkovskij, D. A.; Rahimi, A.; et al. 2021. Vector Symbolic Architectures as a Computing Framework for Nanoscale Hardware. *arXiv preprint arXiv:2106.05268*.

Moin, A.; Zhou, A.; Rahimi, A.; Menon, A.; Benatti, S.; Alexandrov, G.; Tamakloe, S.; Ting, J.; Yamamoto, N.; Khan, Y.; et al. 2021. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. *Nature Electronics*, 4(1): 54–63.

Neubert, P.; Schubert, S.; and Protzel, P. 2017. *Learning vector symbolic architectures for reactive robot behaviours*. Universitätsbibliothek Chemnitz.

Neubert, P.; Schubert, S.; and Protzel, P. 2019. An introduction to hyperdimensional computing for robotics. *KI-Künstliche Intelligenz*, 33(4): 319–330.

Plate, T. 1994. Distributed representations and nested compositional structure (PhD thesis). *University of Toronto, Toronto, ON, Canada*.

Rahimi, A.; Datta, S.; Kleyko, D.; Frady, E. P.; Olshausen, B.; Kanerva, P.; and Rabaey, J. M. 2017. High-dimensional computing as a nanoscalable paradigm. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(9): 2508–2521.

Rahimi, A.; Kanerva, P.; Benini, L.; and Rabaey, J. M. 2018. Efficient biosignal processing using hyperdimensional computing: Network templates for combined learning and classification of exg signals. *Proceedings of the IEEE*, 107(1): 123–143.

Russell, S.; and Norvig, P. 2002. Artificial intelligence: a modern approach.

Smolensky, P. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, 46(1-2): 159–216.

Zhou, A.; Muller, R.; and Rabaey, J. 2020. Memory-Efficient, Limb Position-Aware Hand Gesture Recognition using Hyperdimensional Computing. In *Research Symposium on Tiny Machine Learning*.