Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
Journal Track

# A Theoretical Perspective on Hyperdimensional Computing (Extended Abstract)*

**Anthony Thomas, Sanjoy Dasgupta, Tajana Rosing**
Department of Computer Science and Engineering, UC San Diego, La Jolla CA
{ahthomas,dasgupta,tajana}@eng.ucsd.edu

## Abstract

Hyperdimensional (HD) computing is a set of neurally inspired methods for computing on high-dimensional, low-precision, distributed representations of data. These representations can be combined with simple, neurally plausible algorithms to effect a variety of information processing tasks. HD computing has recently garnered significant interest from the computer hardware community as an energy-efficient, low-latency, and noise-robust tool for solving learning problems. We present a novel mathematical framework that unifies analysis of HD computing architectures, and provides general, non-asymptotic, sufficient conditions under which HD information processing techniques will succeed.

## 1 Introduction

Hyperdimensional (HD) computing is an emerging area at the intersection of computer architecture and theoretical neuroscience [Kanerva, 2009]. It is based on the observation that brains are able to perform complex tasks using circuitry that: (1) uses low power, (2) requires low precision, and (3) is highly robust to data corruption. HD computing aims to carry over similar design principles to a new generation of digital devices that are highly energy-efficient, fault tolerant, and well-suited to natural information processing [Rahimi *et al.*, 2018].

The wealth of recent work on neural networks also draws its inspiration from the brain, but modern instantiations of these methods have diverged from the desiderata above. The success of these networks has rested upon choices that are not neurally plausible, most notably significant depth and training via backpropagation. Moreover, from a practical perspective, training these models often requires high precision and substantial amounts of energy. While a large body of literature has sought to ameliorate these issues with neural networks, these efforts have largely been designed to address specific performance limitations. By contrast, the properties above emerge naturally from the basic architecture of HD computing.

Hyperdimensional computing focuses on the very simplest neural architectures. Typically, there is a single, static, mapping from inputs $x$ to much higher-dimensional "neural" representations $\phi(x)$ living in some space $\mathcal{H}$. All computational tasks are performed in $\mathcal{H}$-space, using simple, operations like element-wise additions and dot products. The mapping $\phi$ is often taken to be random, and the embeddings have coordinates that have low precision; for instance, they might take values $-1$ and $+1$. The entire setup is elementary and lends itself to fast, low-power hardware realizations.

Indeed, a cottage industry has emerged around developing optimized implementations of HD computing based algorithms on hardware accelerators [Imani *et al.*, 2017; Rahimi *et al.*, 2018; Gupta *et al.*, 2018; Schmuck *et al.*, 2019; Salamat *et al.*, 2019; Imani *et al.*, 2019]. Broadly speaking, this line of work touts HD computing as an energy efficient, low-latency, and noise-resilient alternative to conventional realizations of general purpose ML algorithms like support vector machines, multilayer perceptrons, and nearest-neighbor classifiers. While this work has reported impressive performance benefits, there has been relatively little formal treatment of HD computing as a tool for general purpose learning.

Our work has two broad aims. The first, more modest, goal is to introduce the area of hyperdimensional computing to a machine learning audience. The second is to develop a particular mathematical framework for understanding and analyzing these models. The recent literature has suggested a variety of different HD architectures that conform to the overall blueprint given above, but differ in many important details. We present a unified treatment of many such architectures that enables their properties to be compared. The most basic types of questions we wish to answer are:

1. How can individual items, sets of items, and sequences of items, be represented and stored in $\mathcal{H}$-space, in a manner that permits reliable decoding?
2. What kinds of noise can be tolerated in $\mathcal{H}$-space?
3. What kinds of structure in the input $x$-space are preserved by the mapping to $\mathcal{H}$-space?
4. What is the power of linear separators on the $\phi$-representation?

Some of these questions have been introduced in the HD

*Published in the Journal of Artificial Intelligence Research (JAIR)

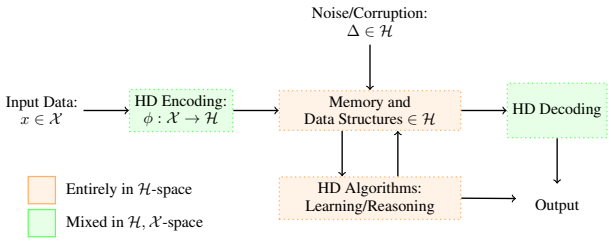


Figure 1: The flow of data in HD computing. Data is mapped from the input space to HD-space under an encoding function $\phi : \mathcal{X} \to \mathcal{H}$. HD representations of data are stored in data structures and may be corrupted by noise or hardware failures. HD representations can be used as input for learning algorithms or other information processing tasks and may be decoded to recover the input data.

computing literature and studied in isolation [Plate, 2003; Gallant and Okaywe, 2013; Kleyko *et al.*, 2018; Frady *et al.*, 2018]. In this work we address these questions formally and in greater generality.

## 2 Introduction to HD Computing

We being by providing an introduction to the fundamentals of HD computing and some brief discussion of its motivations in the neuroscience literature.

Neuroscience has proven to be a rich source of inspiration for the machine learning community: from the perceptron [Rosenblatt, 1958], which introduced a simple and general-purpose learning algorithm for linear classifiers, to neural networks [Rumelhart *et al.*, 1986], to convolutional architectures inspired by visual cortex [Fukushima, 1980], to sparse coding [Olshausen and Field, 1996] and independent component analysis [Bell and Sejnowski, 1995]. One of the most consequential discoveries from the neuroscience community, underlying much research at the intersection of neuroscience and machine learning, has been the notion of *high-dimensional distributed representations* as the fundamental data structure for diverse types of information. In the neuroscience context, these representations are also typically *sparse*.

The notion of high-dimensional, distributed, data representations has engendered a number of computational models that have collectively come to be known as *vector symbolic architectures* (VSA) [Levy and Gayler, 2008]. In general, VSAs provide a systematic way to generate and manipulate high-dimensional representations of symbols so as to implement cognitive operations like association between related concepts. Notable examples of VSAs include "holographic reduced representations" [Plate, 1995; Plate, 2003], "binary spatter codes" [Kanerva, 1994; Kanerva, 1995], and "matrix binding of additive terms" [Gallant and Okaywe, 2013]. HD computing can be seen as a successor to these early VSA models, with a strong additional slant towards hardware and computational efficiency. While our treatment focuses primarily on recent work on HD computing, many of our results apply to these earlier VSA models as well.

An overview of data-flow in HD computing is given in Figure 1. The first step in HD computing is encoding, which maps a piece of input data to its high-dimensional representation under some function $\phi : \mathcal{X} \to \mathcal{H}$. The nature of $\phi$ depends on the type of input and the choice of $\mathcal{H}$. The space $\mathcal{H}$ is some $d$-dimensional inner-product space defined over the real numbers or a subset thereof. For computational reasons, it is common to restrict $\mathcal{H}$ to be defined over integers in a limited range $[-b, b]$.

The HD representations of data can be combined, using simple element-wise operators, to affect various information processing tasks. Two particularly important operators are "bundling" and "binding." The bundling operator is used to superimpose multiple HD representations into a single point. For instance, given a set of items $\mathcal{S} \subset \mathcal{X}$, one represents $\mathcal{S}$ by bundling together the encodings of constituent items: $\phi(\mathcal{S}) = \oplus_{a \in \mathcal{S}} \phi(a)$. The binding operator is used to construct representations of structured data in which points consists of feature-value pairs. For a feature $f \in \mathcal{F}$ taking on a value $a \in \mathcal{X}$, its embedding is constructed as $\psi(f) \otimes \phi(a)$. These representations can also typically be "decoded" to recover the atomic inputs. In addition to storing and recalling patterns, HDC can be used for learning problems like classification and regression. HDC learning techniques typically take the form of simple, linear (on the HD version of data), algorithms like the perceptron or Winnow.

In the remainder, we consider a number of popular techniques for encoding and decoding both discrete data, and data defined on a Euclidean space. We develop an analytic framework that unifies analysis of many of these techniques and readily leads to simple sufficient conditions under which various HD information processing algorithms will succeed.

## 3 Encoding and Decoding Discrete Data

A central object in HD computing is the mapping from inputs to their high-dimensional representations. The design of this mapping, typically referred to as "encoding" in the literature on HD computing, has been the subject of considerable attention, and there is a wide range of possible encoding methods targeting different types of data. Some of these have been introduced in the HD computing literature and studied in isolation [Plate, 2003; Gallant and Okaywe, 2013; Kleyko *et al.*, 2018]. We here present a novel unifying framework in which to study these mappings and to characterize their key properties in a non-asymptotic setting. We first discuss the encoding and decoding of *sets* in some detail. Many HD encoding procedures for more complex data types such as sequences essentially amount to transforming the data into a set and then applying the standard set-encoding method.

### 3.1 Finite Sets

Let $\mathcal{A} = \{a_i\}_{i=1}^m$ be some finite alphabet of $m$ symbols. Symbols $a \in \mathcal{A}$ are mapped to $\mathcal{H}$ under an encoding function $\phi : \mathcal{A} \to \mathcal{H}$. Our goal in this section is to consider the encoding of sets $\mathcal{S}$ whose elements are drawn from $\mathcal{A}$. The HD representation of $\mathcal{S}$ is constructed by superimposing the embeddings of the constituent elements using the bundling operator $\oplus : \mathcal{H} \times \mathcal{H} \to \mathcal{H}$. The encoding of $\mathcal{S}$ is defined to be $\phi(\mathcal{S}) = \oplus_{a \in \mathcal{S}} \phi(a)$. We first focus on the intuitive setting in which $\oplus$ is the element-wise sum and then address other forms of bundling.

To determine if some $a \in \mathcal{A}$ is contained in $\mathcal{S}$, we check if the dot product $\langle \phi(a), \phi(\mathcal{S}) \rangle$ exceeds some fixed threshold. If the codewords $\{\phi(a) : a \in \mathcal{A}\}$ are orthogonal and have a constant length $L$, then we have $\langle \phi(a), \phi(\mathcal{S}) \rangle = L^2 \mathbb{1}(a \in \mathcal{S})$, where $\mathbb{1}$ is the indicator function which evaluates to one if its argument is true and zero otherwise. However, when the codewords are not perfectly orthogonal, we have $\langle \phi(a), \phi(\mathcal{S}) \rangle = L\mathbb{1}(a \in \mathcal{S}) + \Delta$, where $\Delta$ is the "cross-talk" caused by interference between the codewords. In order to decode reliably, we must ensure the contribution of the cross-talk is small and bounded. We formalize this using the notion of incoherence popularized in the sparse coding literature. We define incoherence formally as [Donoho *et al.*, 2005]:

**Definition 1.** ***Incoherence****. For $\mu \geq 0$, we say $\phi : \mathcal{A} \to \mathcal{H}$ is $\mu$-incoherent if for all distinct $a, a' \in \mathcal{A}$, we have*

$$|\langle \phi(a), \phi(a') \rangle| \leq \mu L^2$$

*where $L = \min_{a \in \mathcal{A}} \|\phi(a)\|$.*

When $d \geq m$, it is possible to have codewords that are mutually orthogonal, whereupon $\mu = 0$, however we are generally interested in results that do not require $d \geq m$.

The cross-talk noise can be bounded in terms of the incoherence of $\phi$, which in turn leads to a simple threshold rule for decoding:

**Theorem 1.** *Let $L = \min_{a \in \mathcal{A}} \|\phi(a)\|$ and let the bundling operator be the element wise sum. To decode whether an element $a$ lies in set $S$, we use the rule*

$$\langle \phi(a), \phi(S) \rangle \geq \frac{1}{2} L^2.$$

*This gives perfect decoding for sets of size $\leq s$ if $\phi$ is $1/(2s)$-incoherent.*

We now turn to practical methods for generating incoherent codebooks.

### 3.2 Random Codebooks

In practice, each $\phi(a)$ is usually generated by sampling from some distribution over $\mathcal{H}$ or a subset of $\mathcal{H}$ [Kanerva, 2009; Kleyko *et al.*, 2018; Rahimi *et al.*, 2018]. We typically require that this distribution is factorized so that coordinates of $\phi(a)$ are i.i.d.. Intuitively, the incoherence condition stipulated in Theorem 1 will hold if dot products between two different codewords are concentrated around zero. Furthermore, we would like it to be the case that this concentration occurs quickly as the encoding dimension is increased. It turns out that a fairly broad family of simple distributions, known as the sub-Gaussians, satisfies these properties.

We can show that codebooks sampled from any sub-Gaussian distribution are incoherent with high-probability, provided one chooses $d$ to be large enough:

**Theorem 2.** *Let $\phi$ be $\sigma$-sub-Gaussian. Then, for $\mu > 0$,*

$$\mathbb{P}(\exists\, a \neq a' \in \mathcal{A} \text{ s.t. } |\langle \phi(a), \phi(a') \rangle| \geq \mu L^2) \leq$$
$$m^2 \exp\left(-\frac{\mu^2 \kappa L^2}{2\sigma^2}\right)$$

*where $\kappa = (\min_a \|\phi(a)\|^2)/(\max_a \|\phi(a)\|^2)$.*

Throughout our work, we consider three running examples of codeword distributions (although our approach is generic to any sub-Gaussian distribution). In the first, we consider codewords sampled from the uniform distribution over the $d$-dimensional unit-cube. In the second, codewords are sampled from the $d$-dimensional standard-Gaussian distribution. In either case, one can show that, with high-probability $\mu = O(\sqrt{(\ln m)/d})$, meaning that (by invoking Theorem 1) the encoding dimension scales quadratically with the number of elements in the set, but only logarithmically in the alphabet size and probability of error. However, we show that the dependence on $s$ can be reduced to linear in exchange for a slightly weaker bound that just guarantees that any arbitrary set will be correctly decoded with high-probability. Finally, we also consider sparse binary codewords, which have an interesting connection to the Bloom filter [Bloom, 1970], which is a popular approximate data structure for representing sets.

### 3.3 Encoding Structures

We are often interested in representing more complex data types, such as objects with multiple attributes or "features." In general, we suppose that we observe a set of features $\mathcal{F}$ whose values are assumed to lie in some set $\mathcal{A}$. Let $\psi : \mathcal{F} \to \mathcal{H}$ be an embedding of features, and $\phi : \mathcal{A} \to \mathcal{H}$ be an embedding of values. We associate a feature with its value through use of the *binding* operator $\otimes : \mathcal{H} \times \mathcal{H} \to \mathcal{H}$ that creates an embedding for a (feature,value) pair. For a feature $f \in \mathcal{F}$ taking on a value $a \in \mathcal{A}$, its embedding is constructed as $\psi(f) \otimes \phi(a)$. A data point $\mathbf{x} = \{(f_i \in \mathcal{F}, x_i \in \mathcal{A})\}_{i=1}^n$ consists of $n$ such pairs. For simplicity, we assume each $x$ possesses all attributes, although our analysis also applies to the case that $x$ possesses only some subset of attributes. The entire embedding for $\mathbf{x}$ is constructed as [Plate, 2003]:

$$\phi(\mathbf{x}) = \bigoplus_{i=1}^{n} \psi(f_i) \otimes \phi(x_i) \tag{1}$$

As with sets we would typically like $\phi(\mathbf{x})$ to be *decodable* in the sense that we can recover the value associated with a particular feature, and *comparable* in the sense that $\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ is reflective of a reasonable notion of similarity between $\mathbf{x}$ and $\mathbf{x}'$. It turns out that these desiderata are again related to a notion of incoherence in the embeddings. We omit a technical discussion here, since the gist is similar to the discussion of sets above, but full detail is available in [Thomas *et al.*, 2021].

### 3.4 Robustness to Noise

We conclude our treatment of encoding discrete data by exploring the noise robustness properties of the encoding methods discussed above using the formalism of incoherence. We consider some unspecified noise process which corrupts the encoding of a set $\mathcal{S} \subset \mathcal{A}$ of size at most $s$ according to $\tilde{\phi}(\mathcal{S}) = \phi(\mathcal{S}) + \Delta_{\mathcal{S}}$. We say $\Delta_{\mathcal{S}}$ is $\rho$-bounded if:

$$\max_{a \in \mathcal{A}} |\langle \phi(a), \Delta_{\mathcal{S}} \rangle| \leq \rho.$$

We are interested in understanding the conditions under which we can still decode reliably.

**Theorem 3.** *Suppose $\mathcal{S}$ has size $\leq s$ and $\Delta_{\mathcal{S}}$ is $\rho$-bounded. We can correctly decode $\mathcal{S}$ using the thresholding rule from Theorem 1 if:*

$$\frac{\rho}{L^2} + s\mu < \frac{1}{2}$$

*where $L = \min_{a \in \mathcal{A}} \|\phi(a)\|_2$.*

We specialize this analysis to several different practical types of HD architecture and address both passive noise, and adversarial noise in which $\Delta_{\mathcal{S}}$ might be chosen maliciously, with knowledge of the codebook, so as to deliberately cause a decoding error.

## 4 Encoding Euclidean Data and Learning

We additionally address encoding of data defined over a Euclidean space $\mathcal{X} \subset \mathbb{R}^m$. In general, one typically here desires that the HD encoding preserve a notion of distance on the original data, which we formalize as follows:

**Definition 2.** ***Distance-Preserving Embedding****: Let $\delta_{\mathcal{X}}$ be a distance function on $\mathcal{X} \subset \mathbb{R}^n$ and $\delta_H$ be a distance function on $\mathcal{H}$. We say $\phi$ preserves $\delta_{\mathcal{X}}$ under $\delta_H$ if, there exist functions $\alpha, \beta : \mathbb{Z}^+ \to \mathbb{R}$ such that $\beta(d)/\alpha(d) \to 0$ as $d \to \infty$, and:*

$$\delta_{\mathcal{H}}(\phi(\mathbf{x}), \phi(\mathbf{x}')) \in \alpha(d)\delta_{\mathcal{X}}(\mathbf{x}, \mathbf{x}') \pm \beta(d), \qquad (2)$$

*for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$.*

We typically wish the distance function $\delta_{\mathcal{H}}$ on $\mathcal{H}$ to be simple to compute. In practice, it is often taken to be the Euclidean, Hamming, or angular distance. We show that this definition is satisfied by several popular forms of encoding in HDC and provide closed form expressions for $\alpha(d), \beta(d)$, which allows practitioners to determine an encoding dimension ($d$) sufficient to preserve a distance with desired fidelity.

Distance preservation also has important implications for learning on HD data representations. We show that distance preservation is a sufficient condition to establish preservation of neighborhood/cluster structure, robustness to various forms of noise, and in some cases, preservation of linear separability. The latter point is important because the learning algorithms used in HDC are predominantly linear, and thus, at a minimum, encoding should preserve linear separability to the extent it exists in the data. We additionally show that, under a particular type of encoding method, linear separators on the HD version of the data are sparse. That is, they depend only on a small number coordinates in the representation. Our analysis links properties of the data–for instance a notion of margin or separability between classes/clusters–with the choice of encoding dimension, which is an important practical consideration. Moreover, we examine how certain types of encoding can actually increase the power of linear separators on the HD version of the data. That is: linear decision boundaries learned on the HD data can sometimes capture nonlinear decision boundaries on the original data–much like kernel methods from machine learning and statistics. Indeed, many of the encoding techniques used in HD computing can be seen as approximate feature maps for different types of kernel function on the original data. We discuss these connections in greater detail in our main paper.

## 5 Conclusion and Discussion

A key question addressed here and by several pieces of prior work is to bound the magnitude of crosstalk noise in terms of the encoding dimension ($d$), and properties of the data, like the number of items to encode ($s$), and the alphabet size ($m$). Prior analysis [Plate, 2003; Gallant and Okaywe, 2013; Kleyko *et al.*, 2018; Frady *et al.*, 2018] recovers the same asymptotic relationship as we do, but either relies on specific assumptions about the method used to generate the codewords, or relies on asymptotic approximations. Our approach based on sub-Gaussianity and incoherence formalizes this analysis in the non-asymptotic setting.

In summary, our formalism using the notion of incoherence allows us to decouple the analysis of encoding, decoding and noise-robustness from any particular method for generating codewords and readily yields rigorous bounds in the non-asymptotic setting. Our approach unifies a large swath of HD computing under a single analytic framework, and enables us to offer general conditions under which thresholding based decoding schemes and simple learning methods will succeed.

## Acknowledgements

This work was supported in part by CRISP, one of six centers in JUMP, an SRC program sponsored by DARPA, in part by an SRC-Global Research Collaboration grant, GRC TASK 3021.001, GRC TASK 2942.001, DARPA-PA-19-03-03 Agreement HR00112090036, and also NSF grants 1527034, 1730158, 1826967, 2100237, 2112167, 2052809, 2003279, 1830399, 1911095, 2028040, and 1911095.

## References

[Bell and Sejnowski, 1995] Anthony Bell and Terence Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[Bloom, 1970] Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[Donoho *et al.*, 2005] David L Donoho, Michael Elad, and Vladimir N Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, 2005.

[Frady *et al.*, 2018] E Paxon Frady, Denis Kleyko, and Friedrich T Sommer. A theory of sequence indexing and working memory in recurrent neural networks. *Neural Computation*, 30(6):1449–1513, 2018.

[Fukushima, 1980] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.

[Gallant and Okaywe, 2013] Stephen I Gallant and T Wendy Okaywe. Representing objects, relations, and sequences. *Neural Computation*, 25(8):2038–2078, 2013.

[Gupta *et al.*, 2018] Saransh Gupta, Mohsen Imani, and Tajana Rosing. Felix: Fast and energy-efficient logic in

memory. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–7. IEEE, 2018.

[Imani *et al.*, 2017] Mohsen Imani, Abbas Rahimi, Deqian Kong, Tajana Rosing, and Jan M Rabaey. Exploring hyperdimensional associative memory. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 445–456. IEEE, 2017.

[Imani *et al.*, 2019] Mohsen Imani, Sahand Salamat, Saransh Gupta, Jiani Huang, and Tajana Rosing. Fach: Fpga-based acceleration of hyperdimensional computing by reducing computational complexity. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pages 493–498. ACM, 2019.

[Kanerva, 1994] Pentti Kanerva. The spatter code for encoding concepts at many levels. In *International Conference on Artificial Neural Networks*, pages 226–229. Springer, 1994.

[Kanerva, 1995] Pentti Kanerva. A family of binary spatter codes. In *International Conference on Artificial Neural Networks*, volume 1, pages 517–522, 1995.

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

[Kleyko *et al.*, 2018] Denis Kleyko, Abbas Rahimi, Dmitri A Rachkovskij, Evgeny Osipov, and Jan M Rabaey. Classification and recall with binary hyperdimensional computing: Tradeoffs in choice of density and mapping characteristics. *IEEE Transactions on Neural Networks and Learning Systems*, 29(12):5880–5898, 2018.

[Levy and Gayler, 2008] Simon D Levy and Ross Gayler. Vector symbolic architectures: A new building material for artificial general intelligence. In *Conference on Artificial General Intelligence*, pages 414–418. IOS Press, 2008.

[Olshausen and Field, 1996] Bruno Olshausen and David Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[Plate, 1995] Tony A Plate. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641, 1995.

[Plate, 2003] Tony Plate. *Holographic Reduced Representation: Distributed Representation for Cognitive Structures*. CSLI Lecture Notes (CSLI- CHUP) Series. CSLI Publications, 2003.

[Rahimi *et al.*, 2018] Abbas Rahimi, Pentti Kanerva, Luca Benini, and Jan M Rabaey. Efficient biosignal processing using hyperdimensional computing: Network templates for combined learning and classification of ExG signals. *Proceedings of the IEEE*, 107(1):123–143, 2018.

[Rosenblatt, 1958] Frank Rosenblatt. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[Rumelhart *et al.*, 1986] David Rumelhart, James McClelland, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. MIT Press, 1986.

[Salamat *et al.*, 2019] Sahand Salamat, Mohsen Imani, Behnam Khaleghi, and Tajana Rosing. F5-hd: Fast flexible fpga-based framework for refreshing hyperdimensional computing. In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 53–62, 2019.

[Schmuck *et al.*, 2019] Manuel Schmuck, Luca Benini, and Abbas Rahimi. Hardware optimizations of dense binary hyperdimensional computing: Rematerialization of hypervectors, binarized bundling, and combinational associative memory. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 15(4):1–25, 2019.

[Thomas *et al.*, 2021] Anthony Thomas, Sanjoy Dasgupta, and Tajana Rosing. Theoretical foundations of hyperdimensional computing. *Journal of Artificial Intelligence Research*, 72:215–249, 2021.