

Torchhd: An Open Source Python Library to Support Research on Hyperdimensional Computing and Vector Symbolic Architectures

Mike Heddes¹

MHEDDES@UCI.EDU

Igor Nunes¹

IGORD@UCI.EDU

Pere Vergés¹

PVERGESB@UCI.EDU

Denis Kleyko²

DENIS.KLEYKO@RI.SE

Danny Abraham¹

DANNYA1@UCI.EDU

Tony Givargis¹

GIVARGIS@UCI.EDU

Alexandru Nicolau¹

NICOLAU@ICS.UCI.EDU

Alexander Veidenbaum¹

ALEXV@ICS.UCI.EDU

¹ *Department of Computer Science, University of California, Irvine, CA 92617, USA*

² *Intelligent Systems Lab, Research Institutes of Sweden, 164 40 Kista, Sweden*

Editor: Sebastian Schelter

Abstract

Hyperdimensional computing (HD), also known as vector symbolic architectures (VSA), is a framework for computing with distributed representations by exploiting properties of random high-dimensional vector spaces. The commitment of the scientific community to aggregate and disseminate research in this particularly multidisciplinary area has been fundamental for its advancement. Joining these efforts, we present Torchhd, a high-performance open source Python library for HD/VSA. Torchhd seeks to make HD/VSA more accessible and serves as an efficient foundation for further research and application development. The easy-to-use library builds on top of PyTorch and features state-of-the-art HD/VSA functionality, clear documentation, and implementation examples from well-known publications. Comparing publicly available code with their corresponding Torchhd implementation shows that experiments can run up to 100× faster. Torchhd is available at: <https://github.com/hyperdimensional-computing/torchhd>.

Keywords: hyperdimensional computing, vector symbolic architectures, distributed representations, machine learning, symbolic AI, Python library

1. Introduction

Hyperdimensional computing (HD) (Kanerva, 2009), also known as vector symbolic architectures (VSA) (Gayler, 2003), is a computing framework capable of forming compositional distributed representations. It originated within artificial intelligence and cognitive science (Smolensky, 1990; Kussul et al., 1991; Plate, 1991) from attempts to address limitations of earlier distributed representations (von der Malsburg, 1986; Fodor and Pylyshyn, 1988). HD/VSA forms a “concept space” by exploiting the geometry and algebra of high-

dimensional spaces. The central idea is to represent information with randomly generated vectors, called *hypervectors*. Together with a set of operations on these hypervectors, HD/VSA can represent compositional structures, which, in turn, enables features such as reasoning by analogy (Plate, 1994; Kanerva, 2000; Gayler and Levy, 2009; Rachkovskij and Slipchenko, 2012; Kleyko et al., 2015) and cognitive computing (Emruli et al., 2013; Rasmussen and Eliasmith, 2014; Hersche et al., 2023).

While the foundational ideas of HD/VSA have been around for some time, it was only recently that it started gaining momentum both theoretically (Frady et al., 2018; Thomas et al., 2021; Clarkson et al., 2023) and practically, attracting attention from the wider machine learning community (Graves et al., 2014; Rahimi et al., 2019; Neubert et al., 2019; Ganesan et al., 2021; Hersche et al., 2022). This interest is partially driven by the observation that the strengths of HD/VSA complement some of the limitations of current artificial neural networks (Greff et al., 2020; Smolensky et al., 2022), leading to hybrid approaches that could be viewed as neuro-symbolic (Hersche et al., 2023).

In face of the growing interest, there have been substantial efforts to consolidate and disseminate HD/VSA by its research community. The challenge for these initiatives has been the fact that HD/VSA involves many disciplines, including machine learning, neuroscience, electrical engineering, artificial intelligence, mathematics, and cognitive science. Keeping up with the latest advances is not trivial as publications are spread across different venues and disciplines. It is, therefore, important to pursue consolidation and integration efforts, which will facilitate further research endeavors. Despite detailed surveys of various aspects of the area (Ge and Parhi, 2020; Schlegel et al., 2022; Kleyko et al., 2022c, 2023), much less attention, however, has been paid towards streamlining the translation of concepts, methodologies, and algorithms into a coherent open source software.

To address this issue, we developed Torchhd—an open source library for HD/VSA. Up to date with the latest advances in the area, Torchhd aims to lower the barrier of entry to HD/VSA for novices and provides a high-performance execution platform for experienced researchers. To support a wide variety of HD/VSA primitives, applications, and research directions; the provided abstractions are designed to be modular, allowing users to adopt the aspects that support their workflow. Our design philosophy focuses first on ease-of-use, aiming to be accessible without limiting expressiveness. The ease of use allows for application-driven research to focus on the conceptual and methodological aspects of the project while, at the same time, Torchhd enables the study of individual elements of HD/VSA for theoretical research. The second goal of the library is to provide high-performance execution, enabling much faster evaluation of novel algorithms without hurting the Python idiomatic development experience. We believe that Torchhd will benefit the HD/VSA and the broader machine learning communities based on its following features: 1) state-of-the-art HD/VSA methods for transforming data into hypervector; 2) support for automatic differentiation for research on hybrid neuro-symbolic models; 3) Python idiomatic interface design for a flexible, developer-first experience; 4) high-performance execution, which means that applications can run orders of magnitude faster; 5) interfaces for data sets commonly used in the literature to evaluate and benchmark HD/VSA methods (129 data sets currently).

2. Torchhd

Torchhd builds on PyTorch (Paszke et al., 2019), a library for high-performance tensor computation. PyTorch and its ecosystem provide many machine learning utilities, support for various hardware accelerators, and built-in automatic differentiation (autodiff) (Paszke et al., 2017). Building on PyTorch enables the exchange of knowledge between HD/VSA and the wider machine learning community, and facilitates the cross-fertilization of ideas.

Torchhd supports six common HD/VSA models, namely binary spatter codes (BSC) (Kanerva, 1997), multiply-add-permute (MAP) (Gayler, 1998), holographic reduced representations (HRR), Fourier HRR (FHRR) (Plate, 1995), sparse block codes (SBC) (Laiho et al., 2015; Frady et al., 2023), and vector-derived transformation binding (VTB) (Gosmann and Eliasmith, 2019). These models differ in their specification of the high-dimensional space and their implementations of the fundamental operations: superposition and binding of hypervectors. Support for (F)HRR enabled the implementation of vector function architectures (Frady et al., 2021, 2022) which extends an approach for implementing large-scale kernel machines via random Fourier features (Rahimi and Recht, 2007). Our implementations are verified with extensive automatic unit testing. We also place emphasis on clear documentation, which is available at: <https://torchhd.readthedocs.io>

The library has six modules which provide the following functionality:

- **functional**: functions for generating hypervectors (Rachkovskij et al., 2005; Nunes et al., 2023); the operations on hypervectors (Schlegel et al., 2022); and a resonator network for factorizing hypervectors (Frady et al., 2020; Renner et al., 2022; Kleyko et al., 2022a; Langenegger et al., 2023).
- **embeddings**: classes for transforming scalars or feature vectors into hypervectors, ranging from simple hypervector lookups till similarity-preserving transformations of feature vectors (Kleyko et al., 2021; Thomas et al., 2021), some of which approximate well-known kernels (Rahimi and Recht, 2007; Frady et al., 2022).
- **models**: common classification models such as the centroid model with class prototypes and its various training algorithms (Rahimi et al., 2016a; Imani et al., 2017; Hernandez-Cane et al., 2021; Nunes et al., 2022), learning vector quantization (Diao et al., 2021), and regularized least squares (Kleyko et al., 2021).
- **memory**: methods for long-term storage and retrieval of hypervectors, such as sparse distributed memory (Kanerva, 1988), inspired by the cerebellum (Teeters et al., 2022), the (modern) Hopfield network (Hopfield, 1982; Krotov and Hopfield, 2016), and the attention mechanism (Vaswani et al., 2017; Ramsauer et al., 2020).
- **structures**: classes for HD/VSA data structures (Kleyko et al., 2022b) such as hash tables, graphs, binary trees, and finite state automata (Osipov et al., 2017; Yerxa et al., 2018; Heddes et al., 2022). This module facilitates the development of classical algorithms using HD/VSA.
- **datasets**: convenient access to 126 classification and 3 regression data sets commonly used in the literature. This includes a classification benchmark of UCI machine learning repository data sets (Dua and Graff, 2017) created by Fernández-Delgado et al. (2014). All data sets are interoperable with the PyTorch ecosystem.

Library	BSC	MAP	HRR	FHRR	Memory models	Data structures	Resonator Network	Auto-diff	Data sets
OpenHD	✗	✓	✗	✗	✗	✗	✗	✗	✗
HDTorch	✓	✓	✗	✗	✗	✗	✗	±	✗
VSA Toolbox	✓	✓	✓	✓	✗	✗	✗	✗	✗
Torchhd (ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: A qualitative assessment of the functionality of HD/VSA software.

We note that several other software for HD/VSA exist. Table 1 contrasts the features of Torchhd with those of OpenHD (Kang et al., 2022), HDTorch (Simon et al., 2022), and VSA Toolbox (Schlegel et al., 2022). As part of their review on HD/VSA models, Schlegel et al. (2022) implement a broader range of HD/VSA models, however, their VSA Toolbox misses many of the features that make Torchhd particularly useful for research and development of HD/VSA, see, for example, Table 1. Importantly, Torchhd is the first general software library for HD/VSA since the other software were developed for more specific purposes.

To demonstrate the performance of Torchhd, we implemented three classification tasks and compared the execution time to the original publicly available code. All the experiments ran on a machine with 20 Intel Xeon Silver 4114 CPUs, 93 GB of RAM, and 4 Nvidia TITAN Xp GPUs, only a single CPU or GPU was used. The results in Table 2 show that Torchhd is faster for all examples with an average of $24\times$ and $54\times$ faster for CPU and GPU, respectively. The provided abstractions ensure performant code by avoiding overly sequential execution, reinforced by our support for batch processing.

Paper	Original	CPU	GPU
EU languages (Rahimi et al., 2016b)	13111.73	542.83 ($24\times$)	125.89 ($104\times$)
EMG gestures (Rahimi et al., 2016a)	1152.32	28.44 ($41\times$)	28.95 ($40\times$)
VoiceHD (Imani et al., 2017)	277.97	37.74 ($7\times$)	16.17 ($17\times$)

Table 2: Execution time in seconds (and speedup)

3. Conclusions and Future Work

Torchhd is a Python library for hyperdimensional computing (HD) a.k.a. vector symbolic architectures (VSA) that aims to be simple, versatile, and highly performant. The library builds on PyTorch, thus, enabling the exchange of knowledge between the HD/VSA and the broader machine learning communities as well as making PyTorch’s vast ecosystem of deep learning tools available for HD/VSA. Torchhd is designed to include any HD/VSA model and already supports binary spatter codes, multiply-add-permute, (Fourier) holographic reduced representations, sparse block codes, and vector-derived transformation binding. We plan to keep the library up-to-date with novel developments in the area by including more HD/VSA models and supporting a broader spectrum of learning algorithms, such as differentiable learning, regression, density estimation, and clustering. Our work joins the community’s vital effort to consolidate and disseminate HD/VSA research.

Acknowledgments

We would like to thank Ross W. Gayler for helpful discussions on the design choices and desired functionality of the library; Rishikanth Chandrasekaran, Dheyay Desai, Jenny Lee, and Xiaofan Yu for their contributions to the library; and finally, Anthony Thomas for his support while resolving an issue with the random projection implementation.

DK received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 839179.

References

- Kenneth L Clarkson, Shashanka Ubaru, and Elizabeth Yang. Capacity analysis of vector symbolic architectures. *arXiv:2301.10352*, pages 1–49, 2023.
- Cameron Diao, Denis Kleyko, Jan M Rabaey, and Bruno A Olshausen. Generalized learning vector quantization for classification in randomized neural networks and hyperdimensional computing. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2021.
- Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>.
- Blerim Emruli, Ross W Gayler, and Fredrik Sandin. Analogical mapping and inference with binary spatter codes and sparse distributed memory. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2013.
- Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15(1):3133–3181, 2014.
- Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988.
- 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.
- E Paxon Frady, Spencer J Kent, Bruno A Olshausen, and Friedrich T Sommer. Resonator networks, 1: An efficient solution for factoring high-dimensional, distributed representations of data structures. *Neural Computation*, 32(12):2311–2331, 2020.
- E Paxon Frady, Denis Kleyko, Christopher J Kymn, Bruno A Olshausen, and Friedrich T Sommer. Computing on functions using randomized vector representations. *arXiv:2109.03429*, pages 1–33, 2021.
- E Paxon Frady, Denis Kleyko, Christopher J Kymn, Bruno A Olshausen, and Friedrich T Sommer. Computing on functions using randomized vector representations (in brief). In *Neuro-Inspired Computational Elements Conference (NICE)*, pages 115–122, 2022.

- E Paxon Frady, Denis Kleyko, and Friedrich T. Sommer. Variable binding for sparse distributed representations: Theory and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 34(5):2191–2204, 2023.
- Ashwinkumar Ganesan, Hang Gao, Sunil Gandhi, Edward Raff, Tim Oates, James Holt, and Mark McLean. Learning with holographic reduced representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–15, 2021.
- Ross W Gayler. Multiplicative binding, representation operators & analogy. In *Advances in Analogy Research: Integration of Theory and Data from the Cognitive, Computational, and Neural Sciences*, pages 1–4, 1998.
- Ross W Gayler. Vector symbolic architectures answer Jackendoff’s challenges for cognitive neuroscience. In *Joint International Conference on Cognitive Science (ICCS/ASCS)*, pages 133–138, 2003.
- Ross W Gayler and Simon D Levy. A distributed basis for analogical mapping: New frontiers in analogy research. In *New frontiers in Analogy Research, Second International Conference on the Analogy (ANALOGY)*, pages 165–174, 2009.
- Lulu Ge and Keshab K Parhi. Classification using hyperdimensional computing: A review. *IEEE Circuits and Systems Magazine*, 20(2):30–47, 2020.
- Jan Gosmann and Chris Eliasmith. Vector-derived transformation binding: An improved binding operation for deep symbol-like processing in neural networks. *Neural computation*, 31(5):849–869, 2019.
- Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing machines. *arXiv:1410.5401*, pages 1–25, 2014.
- Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv:2012.05208*, 2020.
- Mike Heddes, Igor Nunes, Tony Givargis, Alexandru Nicolau, and Alex Veidenbaum. Hyperdimensional hashing: A robust and efficient dynamic hash table. In *ACM/IEEE Design Automation Conference (DAC)*, pages 907–912, 2022.
- Alejandro Hernandez-Cane, Namiko Matsumoto, Eric Ping, and Mohsen Imani. OnlineHD: Robust, efficient, and single-pass online learning using hyperdimensional system. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 56–61, 2021.
- Michael Hersche, Geethan Karunaratne, Giovanni Cherubini, Luca Benini, Abu Sebastian, and Abbas Rahimi. Constrained few-shot class-incremental learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9057–9067, 2022.
- Michael Hersche, Mustafa Zeqiri, Luca Benini, Abu Sebastian, and Abbas Rahimi. A neuro-vector-symbolic architecture for solving Raven’s progressive matrices. *Nature Machine Intelligence*, 5(4):363–375, 2023.

- John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.
- Mohsen Imani, Deqian Kong, Abbas Rahimi, and Tajana Rosing. VoiceHD: Hyperdimensional computing for efficient speech recognition. In *International Conference on Rebooting Computing (ICRC)*, pages 1–8, 2017.
- Pentti Kanerva. *Sparse distributed memory*. MIT press, 1988.
- Pentti Kanerva. Fully distributed representation. In *Real World Computing Symposium (RWC)*, pages 358–365, 1997.
- Pentti Kanerva. Large patterns make great symbols: An example of learning from example. In *International Workshop on Hybrid Neural Systems*, volume 1778 of *Lecture Notes in Computer Science*, pages 194–203, 2000.
- Pentti Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, 1(2):139–159, 2009.
- Jaeyoung Kang, Behnam Khaleghi, Tajana Rosing, and Yeseong Kim. OpenHD: A GPU-powered framework for hyperdimensional computing. *IEEE Transactions on Computers*, 71(11):2753–2765, 2022.
- Denis Kleyko, Evgeny Osipov, Ross W Gayler, Asad I Khan, and Adrian G Dyer. Imitation of honey bees’ concept learning processes using vector symbolic architectures. *Biologically Inspired Cognitive Architectures*, 14:57–72, 2015.
- Denis Kleyko, Mansour Kheffache, E Paxon Frady, Urban Wiklund, and Evgeny Osipov. Density encoding enables resource-efficient randomly connected neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3777–3783, 2021.
- Denis Kleyko, Connor Bybee, Christopher J Kymn, Bruno A Olshausen, Amir Khosrowshahi, Dmitri E Nikonov, Friedrich T Sommer, and E Paxon Frady. Integer factorization with compositional distributed representations. In *Neuro-Inspired Computational Elements Conference (NICE)*, pages 73–80, 2022a.
- Denis Kleyko, Mike Davies, Edward Paxon Frady, Pentti Kanerva, Spencer J. Kent, Bruno A. Olshausen, Evgeny Osipov, Jan M. Rabaey, Dmitri A. Rachkovskij, Abbas Rahimi, and Friedrich T. Sommer. Vector symbolic architectures as a computing framework for emerging hardware. *Proceedings of the IEEE*, 110(10):1538–1571, 2022b.
- Denis Kleyko, Dmitri A Rachkovskij, Evgeny Osipov, and Abbas Rahimi. A survey on hyperdimensional computing aka vector symbolic architectures, part I: Models and data transformations. *ACM Computing Surveys*, 55(6):1–40, 2022c.
- Denis Kleyko, Dmitri A Rachkovskij, Evgeny Osipov, and Abbas Rahimi. A survey on hyperdimensional computing aka vector symbolic architectures, part II: Applications, cognitive models, and challenges. *ACM Computing Surveys*, 55(9):1–52, 2023.

- Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition. *Advances in neural information processing systems*, 29, 2016.
- Ernst M Kussul, Dmitri A Rachkovskij, and Tatyana N Baidyk. Associative-projective neural networks: Architecture, implementation, applications. In *International Conference on Neural Networks and Their Applications (NEURO)*, pages 463–476, 1991.
- Mika Laiho, Jussi H Poikonen, Pentti Kanerva, and Eero Lehtonen. High-dimensional computing with sparse vectors. In *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4, 2015.
- Jovin Langenegger, Geethan Karunaratne, Michael Hersche, Luca Benini, Abu Sebastian, and Abbas Rahimi. In-memory factorization of holographic perceptual representations. *Nature Nanotechnology*, 18(5):479–485, 2023.
- Peer Neubert, Stefan Schubert, and Peter Protzel. An introduction to hyperdimensional computing for robotics. *KI - Künstliche Intelligenz*, 33(4):319–330, 2019.
- Igor Nunes, Mike Heddes, Tony Givargis, Alexandru Nicolau, and Alex Veidenbaum. GraphHD: Efficient graph classification using hyperdimensional computing. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1485–1490, 2022.
- Igor Nunes, Mike Heddes, Tony Givargis, and Alexandru Nicolau. An extension to basis-hypervectors for learning from circular data in hyperdimensional computing. In *ACM/IEEE Design Automation Conference (DAC)*, 2023.
- Evgeny Osipov, Denis Kleyko, and Alexander Legalov. Associative synthesis of finite state automata model of a controlled object with hyperdimensional computing. In *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 3276–3281, 2017.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–4, 2017.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- Tony A Plate. Holographic reduced representations: Convolution algebra for compositional distributed representations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 30–35, 1991.
- Tony A Plate. Estimating analogical similarity by dot-products of holographic reduced representations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1109–1116, 1994.
- Tony A Plate. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641, 1995.

- Dmitri A Rachkovskij and Serge V Slipchenko. Similarity-based retrieval with structure-sensitive sparse binary distributed representations. *Computational Intelligence*, 28(1):106–129, 2012.
- Dmitri A Rachkovskij, Sergey V Slipchenko, Ernst M Kussul, and Tatyana N Baidyk. Sparse binary distributed encoding of scalars. *Journal of Automation and Information Sciences*, 37(6):12–23, 2005.
- Abbas Rahimi, Simone Benatti, Pentti Kanerva, Luca Benini, and Jan M Rabaey. Hyperdimensional biosignal processing: A case study for EMG-based hand gesture recognition. In *International Conference on Rebooting Computing (ICRC)*, pages 1–8, 2016a.
- Abbas Rahimi, Pentti Kanerva, and Jan M Rabaey. A robust and energy-efficient classifier using brain-inspired hyperdimensional computing. In *International Symposium on Low Power Electronics and Design (ISLPED)*, pages 64–69, 2016b.
- 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, 2019.
- Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, pages 1–8, 2007.
- Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.
- Daniel Rasmussen and Chris Eliasmith. A spiking neural model applied to the study of human performance and cognitive decline on Raven’s advanced progressive matrices. *Intelligence*, 42:53–82, 2014.
- Alpha Renner, Lazar Supic, Andreea Danielescu, Giacomo Indiveri, Bruno A Olshausen, Yulia Sandamirskaya, Friedrich T Sommer, and E Paxon Frady. Neuromorphic visual scene understanding with resonator networks. *arXiv:2208.12880*, pages 1–15, 2022.
- Kenny Schlegel, Peer Neubert, and Peter Protzel. A comparison of vector symbolic architectures. *Artificial Intelligence Review*, 55(6):4523–4555, 2022.
- William Andrew Simon, Una Pale, Tomas Teijeiro, and David Atienza. HDTorch: Accelerating hyperdimensional computing with GP-GPUs for design space exploration. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8, 2022.
- Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46:159–216, 1990.
- Paul Smolensky, Richard McCoy, Roland Fernandez, Matthew Goldrick, and Jianfeng Gao. Neurocompositional computing: From the central paradox of cognition to a new generation of AI systems. *AI Magazine*, 43(3):308–322, 2022.

Jeffrey L Teeters, Denis Kleyko, Pentti Kanerva, and Bruno A Olshausen. On separating long-and short-term memories in hyperdimensional computing. *Frontiers in Neuroscience*, 16:1–19, 2022.

Anthony Thomas, Sanjoy Dasgupta, and Tajana Rosing. Theoretical foundations of hyperdimensional computing. *Journal of Artificial Intelligence Research (JAIR)*, 72:215–249, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Christoph von der Malsburg. Am I thinking assemblies? In *Brain Theory*, pages 161–176, 1986.

Thomas Yerxa, Alexander Anderson, and Eric Weiss. The hyperdimensional stack machine. *Cognitive Computing*, pages 1–2, 2018.