ChatGPT

Key References and Publication Venues for VSAX Research

Foundational VSA Papers

- **Plate (1995)** – Introduced Holographic Reduced Representations (HRR), using circular convolution to bind symbols in high-dimensional vectors [1]. Citing Plate's seminal work grounds VSAX in classic VSA theory, as VSAX's HRR implementation directly builds on this foundational binding technique.

- **Kanerva (1988)** – Proposed *Sparse Distributed Memory*, the original inspiration for hyperdimensional representations [2]. Use this to highlight the roots of VSA in brain-inspired memory models, underscoring that VSAX's binary VSA mode traces back to Kanerva's early work.

- **Kanerva (1997)** – Introduced the *Binary Spatter Code* (BSC) for fully distributed representation [3]. This work is key for VSAX's binary VSA model; citing it emphasizes that VSAX's binary hypervectors follow Kanerva's approach to symbolic encoding with random high-dimensional bit vectors.

- **Kanerva (2009)** – A tutorial-style introduction to *Hyperdimensional Computing*. This article situates VSAs in a broader context; referencing it can help orient readers and validate the relevance of VSAX by pointing to recognized explanations of why high-dimensional random vectors are useful for cognitive computing.

- **Gayler (1998)** – Developed the *Multiply-Add-Permute (MAP)* architecture for vector binding. Cite this to credit the MAP model integrated in VSAX, showing that VSAX leverages established VSA variants. It reinforces that VSAX's design (supporting MAP alongside HRR and FHRR) is grounded in prior art.

(Optionally, foundational references like Smolensky (1990) on tensor product binding or Gayler (2003) on VSA's cognitive neuroscience relevance can further strengthen the theoretical background, linking VSAX to earlier efforts to unify connectionist and symbolic representations.)

Contemporary VSA Libraries and Surveys

- **Heddes et al. (2023)** – Presents *TorchHD*, a PyTorch-based library supporting multiple VSA models (BSC, MAP, HRR, FHRR, etc.) with GPU acceleration [4]. This is a direct analog to VSAX; citing TorchHD lets you position VSAX among modern tools, highlighting that VSAX similarly provides a comprehensive VSA toolkit (but in JAX) and learning from TorchHD's emphasis on performance and usability.

- **Kang et al. (2022)** – Introduces *OpenHD*, a GPU-accelerated framework for hyperdimensional computing [5]. Use this to compare VSAX's design decisions (JAX and XLA for GPU support) to other high-performance HD/VSA systems, underlining that efficient GPU utilization is a current trend that VSAX follows.

- **Simon et al. (2022)** – Describes *HDTorch*, which uses GPUs to speed up hyperdimensional computing for design-space exploration. This reference can be used to demonstrate the growing interest in optimized VSA libraries; it reinforces the argument that VSAX's GPU acceleration is a needed contribution in line with recent developments.

- **Schlegel et al. (2022)** – A comparative review of VSA models and tools. Citing this paper provides readers with a survey of the VSA landscape; it can be used in the manuscript to justify why integrating multiple VSA models (as VSAX does) is valuable, since different VSA formalisms have complementary strengths. It also signals that VSAX contributes to an active area with several emerging libraries.

- **Kleyko et al. (2022)** – A comprehensive review *"Vector Symbolic Architectures as a Computing Framework for Emerging Hardware"*. This *Proceedings of the IEEE* article (with VSA pioneers as co-authors) covers modern advancements and applications of VSA. It can be cited to show that VSAX aligns with the cutting edge of the field and addresses noted challenges (e.g. hardware efficiency, scalability), lending authority to the manuscript's motivation.

(Additionally, the two-part survey by Kleyko et al. (2022c, 2023) in ACM Computing Surveys offers an exhaustive overview of VSA models and applications. These can be referenced for readers seeking background, and to validate that VSAX's supported models and use-cases cover the major themes identified in the literature.)

Associative Memory and Resonator Networks

- **Hopfield (1982)** – The classic paper on auto-associative memory in neural networks [6]. Referencing Hopfield's work connects VSAX's associative memory (clean-up or item retrieval using resonator networks) to the broader concept of content-addressable memory. It underscores that VSAX's "resonator" associative recall is rooted in well-known principles of pattern completion and memory retrieval.

- **Krotov & Hopfield (2016)** – Introduced the modern *dense associative memory* (a generalized Hopfield network with higher capacity) [7]. Cite this to discuss high-capacity memory retrieval: it provides theoretical context suggesting how many patterns can be stored/recalled. In the manuscript, this can bolster any claim that VSAX's resonator network or associative memory module is designed to maximize storage capacity or draw inspiration from modern Hopfield-like networks.

- **Frady et al. (2020)** – Proposed *Resonator Networks* for factorizing distributed representations. This is directly relevant to VSAX's resonator-based associative memory: it's the technique VSAX uses to decompose composite codes (e.g., unbinding symbols from a bound pair). Citing Frady et al. validates the resonator approach as an established method for retrieving structured components from hypervectors, and supports positioning VSAX as building on state-of-the-art VSA research.

- **Langenegger et al. (2023)** – Demonstrated *in-memory factorization of holographic representations* using nanoscale devices. This advanced work shows high-speed, parallel VSA operations for associative recall. Include this reference to hint at future directions and the relevance of VSAX: even though VSAX is software, it aligns with approaches that aim to exploit hardware for fast VSA operations. It emphasizes that the resonator and factorization capabilities in VSAX are not only theoretical but of practical interest (e.g., for neuromorphic or memory-centric hardware).

- **Clarkson et al. (2023)** – Provides a theoretical *capacity analysis of VSAs*. Using this citation in discussions of associative memory will strengthen the manuscript's rigor. It quantifies limits on how many items can be stored and recalled with high-dimensional vectors. By citing this, you show that VSAX is developed with awareness of these limits, and possibly that VSAX's design (choice of dimension, operations, etc.) is informed by such analyses to ensure reliable high-capacity memory storage.

Structured Operators: Clifford and Unitary Approaches

- **Aerts et al. (2006)** – Reformulated Kanerva's Binary Spatter Codes in terms of *Geometric Algebra*, showing that the XOR binding operation corresponds to the geometric product [8]. This reference is useful when describing VSAX's structured compositional operators: it provides a mathematical foundation for treating VSA operations as algebraic products (akin to Clifford algebra). Citing this work can justify VSAX's "Clifford/phase-inspired" operator design by indicating prior research that connects VSA bindings to well-understood algebraic systems.

- **Gosmann & Eliasmith (2019)** – Introduced *Vector-Derived Transformation Binding (VTB)* as an improved binding operation using unitary transformations. This is directly relevant to VSAX if the library includes unitary or rotational binding (as in FHRR or other phase-algebra operators). Cite this source to position VSAX's operators as part of an effort to enhance binding operations – for example, noting that VSAX supports invertible or structured bindings similar in spirit to VTB, enabling more complex or deeper symbol structures than traditional convolution or XOR binding.

- **Plate (1995)** – *(Also foundational, but relevant here)* described circular convolution as a binding operation, which can be viewed as multiplication in the frequency domain (a unitary operation) [1]. Emphasize this in the manuscript's methodology section to explain that VSAX's operators (like FHRR's complex phase binding) preserve norms and allow exact unbinding, echoing Plate's original formulation. It reinforces that VSAX's "phase algebra" for binding has precedent in the literature and is grounded in sound mathematics.

(If needed, also mention recent advances like Schlegel et al. (2022) who compare various binding mechanisms, to show that VSAX's chosen operators were informed by comparative evaluations of VSA operations.)

Benchmarks and Frameworks for Symbolic Reasoning

- **Nickel et al. (2016)** – Introduced *Holographic Embeddings (HolE)* for knowledge graphs, which applies circular correlation (an HRR operation) to embed relational data. This paper is a prime example of evaluating VSA-style methods on structured reasoning tasks (link prediction in knowledge bases). Use it to demonstrate VSAX's applicability: citing HolE shows that VSAX's underlying methods have proven effective in knowledge graph reasoning tasks, one of the domains highlighted in your manuscript.

- **Hersche et al. (2023)** – Developed a neuro-vector-symbolic architecture to solve *Raven's Progressive Matrices*, a classic visual reasoning benchmark, using VSA principles. This reference can be used to validate the efficacy of VSA in complex reasoning tasks. In positioning VSAX, you can argue that a library like VSAX eases experimentation with such neurosymbolic solutions; for instance, citing Hersche et al. illustrates the kind of high-level reasoning problems that VSAX could help tackle.

- **Graves et al. (2014)** – Proposed the *Neural Turing Machine*, a neural network with an addressable external memory. While not a VSA, this work is relevant as a baseline framework for differentiable symbolic reasoning. By citing NTM, you can contrast how VSAX (with explicitly structured VSA operations) differs from or complements neural memory architectures. It sets context that memory-based reasoning is a recognized challenge in AI, one that VSAX approaches with a VSA methodology.

- **Greff et al. (2020)** – Analyzed the *binding problem in neural networks*, highlighting difficulties that purely neural approaches face in representing variable bindings. This reference supports the motivation for VSAX: you can use it to argue that VSAX's explicit binding operators (from VSA) address a known limitation in standard neural networks. Citing this work aligns your manuscript with broader neurosymbolic discourse, reinforcing why a tool like VSAX is timely.

- **Gayler & Levy (2009)** – Demonstrated analogy-making via VSA, using distributed representations to perform analogical mapping. This is evidence of evaluating structured reasoning (analogies) within a VSA framework. Including this reference shows that tasks requiring relational reasoning can be solved with VSA operations. It guides readers that VSAX can be positioned not just as a low-level library but as an enabler for high-level cognitive tasks (like analogy or metaphor processing) that have been of long-standing interest in AI.

Recommended Publication Venues

- **Journal of Machine Learning Research (JMLR)** – JMLR (especially its *Machine Learning Open Source Software* track) is well-suited for a research software paper like VSAX. It's a top ML journal that has published similar toolkits (e.g., TorchHD). Submitting to JMLR would frame VSAX as a contribution to machine learning methodology, with broad visibility and rigorous peer review.

- **Transactions on Machine Learning Research (TMLR)** – An open-access ML journal that rapidly reviews novel contributions. TMLR would be appropriate for VSAX given its focus on machine learning systems and the novelty of integrating neurosymbolic computing into a library. The journal's scope on both theoretical and implementation aspects of ML systems matches VSAX's blend of algorithmic and software innovation.

- **Neural Computation** – A respected journal for neural and cognitive modeling research. VSAX's fusion of neural-like high-dimensional vectors with symbolic operations aligns well with *Neural Computation*'s readership. A submission here would emphasize the neuroscientific inspiration and computational properties of VSAX (e.g., how VSAX can model cognitive processes or support neural-symbolic integration), in line with the journal's themes.

- **NeuroSymbolic AI Special Issues or Thematic Conferences** – Although not a journal, keep an eye on calls from venues like *Frontiers in Artificial Intelligence (Neurosymbolic AI section)* or special issues in journals like *IEEE Transactions on Neural Networks and Learning Systems* focusing on neurosymbolic computing. These can be good targets if you want to highlight the AI systems aspect of VSAX. For instance, a special issue on neurosymbolic reasoning would value VSAX as a tool that advances the state-of-the-art in combining symbolic reasoning with neural-like representations.

- **Journal of Open Source Software (JOSS)** – As a supplemental option, JOSS allows a short publication focusing on the software implementation and its utility. While less about AI theory, it ensures VSAX's availability and contributions are formally recognized in the scholarly record. This can complement a traditional journal paper by providing a citable reference for the VSAX library itself.

In summary, positioning VSAX in a top-tier AI/ML journal (JMLR, TMLR) or a specialized neurosymbolic computing venue will lend credibility and reach. Coupling the manuscript's citations of the above works with a carefully chosen publication venue will underscore both the scholarly rigor and the practical importance of the VSAX library.

---

1 2 3 4 5 6 7 jmlr.org
https://www.jmlr.org/papers/volume24/23-0300/23-0300.pdf

8 [cs/0610075] On Geometric Algebra representation of Binary Spatter Codes
https://arxiv.org/abs/cs/0610075