A Walsh Hadamard Derived Linear Vector Symbolic Architecture

Mohammad Mahmudul Alam[1], **Alexander Oberle**[1], **Edward Raff**[1,2], **Stella Biderman**[2], **Tim Oates**[1], **James Holt**[3]
[1]University of Maryland, Baltimore County,[2]Booz Allen Hamilton,
[3] Laboratory for Physical Sciences
m256@umbc.edu, aoberle1@umbc.edu, Raff_Edward@bah.com,
biderman_stella@bah.com, oates@cs.umbc.edu, holt@lps.umd.edu

Abstract

Vector Symbolic Architectures (VSAs) are one approach to developing Neuro-symbolic AI, where two vectors in $\mathbb{R}^d$ are 'bound' together to produce a new vector in the same space. VSAs support the commutativity and associativity of this binding operation, along with an inverse operation, allowing one to construct symbolic-style manipulations over real-valued vectors. Most VSAs were developed before deep learning and automatic differentiation became popular and instead focused on efficacy in hand-designed systems. In this work, we introduce the Hadamard-derived linear Binding (HLB), which is designed to have favorable computational efficiency, and efficacy in classic VSA tasks, and perform well in differentiable systems. Code is available at https://github.com/FutureComputing4AI/Hadamard-derived-Linear-Binding.

1 Introduction

Vector Symbolic Architectures (VSAs) are a unique approach to performing symbolic style AI. Such methods use a binding operation $\mathcal{B} : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}^d$, where $\mathcal{B}(x, y) = z$ denotes that two concepts/vectors x and y are connected to each other. In VSA, any arbitrary concept is assigned to vectors in $\mathbb{R}^d$ (usually randomly). For example, the sentence "the fat cat and happy dog" would be represented as $\mathcal{B}(fat, cat) + \mathcal{B}(happy, dog) = \boldsymbol{S}$. One can then ask, "what was happy" by *unbinding* the vector for happy, which will return a noisy version of the vector bound to happy. The unbinding operation is denoted $\mathcal{B}^*(x, y)$, and so applying $\mathcal{B}^*(S, happy) \approx dog$.

Because VSAs are applied over vectors, they offer an attractive platform for neuro-symbolic methods by having natural symbolic AI-style manipulations via differentiable operations. However, current VSA methods have largely been derived for classical AI tasks or cognitive science-inspired work. Many such VSAs have shown issues in numerical stability, computational complexity, or otherwise lower-than-desired performance in the context of a differentiable system.

As noted in [39], most VSAs can be viewed as a linear operation where $\mathcal{B}(a, b) = a^\top Gb$ and $\mathcal{B}^*(a, b) = a^\top Fb$, where G and F are $d \times d$ matrices. Hypothetically, these matrices could be learned via gradient descent, but would not necessarily maintain the neuro-symbolic properties of VSAs without additional constraints. Still, the framework is useful as all popular VSAs we are aware fit within this framework. By choosing G and F with specified structure, we can change the computational complexity from $\mathcal{O}(d^2)$, down to $\mathcal{O}(d)$ for a diagonal matrix.

In this work, we derive a new VSA that has multiple desirable properties for both classical VSA tasks, and in deep-learning applications. Our method will have only $\mathcal{O}(d)$ complexity for the binding step, is numerically stable, and equals or improves upon previous VSAs on multiple recent deep

1^{st} Workshop on NeuroAI @ 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

learning applications. Our new VSA is derived from the Walsh Hadamard transform, and so we term our method the Hadamard-derived linear Binding (HLB) as it will avoid the $\mathcal{O}(d \log d)$ normally associated with the Hadamard transform, and has better performance than more expensive VSA alternatives.

Related work to our own will be reviewed in Appendix A, including our baseline VSAs and their definitions. Our new HLB will be derived in section 2, showing it theoretically desirable properties. section 3 will empirically evaluate HLB in classical VSA benchmark tasks, and in two recent deep learning tasks, showing improved performance in each scenario. We then conclude in section 4.

2 Methodology

In this section we review the Hadamard transform, its properties, and how we derive our new HLB via these properties. It follows a similar intuition to the original HRR [32], followed by a projection step as suggested by [10]. However, by choosing the Hadamard transform, we obtain a closed-form solution that requires only linear operations, making HLB fast to apply in practice. Hadamard H_d is a square matrix of size $d \times d$ of orthogonal rows consisting of only $+1$s and -1s given in Equation 1 where $d = 2^n \; \forall \; n \in \mathbb{N} : n \geqslant 0$.

$$H_1 = [1] \qquad H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \cdots \qquad H_{2^n} = \begin{bmatrix} H_{2^{n-1}} & H_{2^{n-1}} \\ H_{2^{n-1}} & -H_{2^{n-1}} \end{bmatrix} \tag{1}$$

Vector symbolic architectures (VSA), such as Holographic Reduced Representations (HRR) employs circular convolution to represent compositional structure which is computed using the Fast Fourier Transform (FFT) [32]. However, it can be numerically unstable due to irrational multiplications of complex numbers. Prior work [10] devised a projection step to mitigate the numerical instability of the FFT and it's inverse, but we instead ask if re-deriving the binding/unbinding operations may yield better results if we use the favorable properties of the Hadamard transform as given in Lemma 2.1.

Lemma 2.1 (Hadamard Properties)**.** *Let H be the Hadamard matrix of size $d \times d$ that holds the following properties for $x, y \in \mathbb{R}^d$. First, $H(Hx) = dx$, and second $H(x + y) = Hx + Hy$.*

The bound composition of two vectors into a single vector space is referred to as BINDING ($\mathcal{B}$). The knowledge retrieval from a bound representation is known as UNBINDING ($\mathcal{B}^*$). We define the binding function by replacing the Fourier transform in circular convolution with the Hadamard transform given in Definition 2.1.

Definition 2.1 (Binding and Unbinding)**.** The binding of vectors $x, y \in \mathbb{R}^d$ in the Hadamard domain is defined in Equation 2 where $\odot$ is elementwise multiplication. The unbinding function is defined in a similar fashion, i.e., $\mathcal{B} = \mathcal{B}^*$.

$$\mathcal{B}(x, y) = \frac{1}{d} \cdot H(Hx \odot Hy) \tag{2}$$

Composite representation in vector symbolic architectures is defined by the summation of the bound vectors. We define a parameter $\rho \in \mathbb{N} : \rho \geqslant 1$ that denotes the number of vector pairs bundled in a composite representation. Given vectors $x_i, y_i \in \mathbb{R}^d$ and $\forall \; i \in \mathbb{N} : 1 \leqslant i \leqslant \rho$, we can define the composite representation χ as

$$\underset{\rho=1}{\chi} = \mathcal{B}(x_1, y_1) \qquad \underset{\rho=2}{\chi} = \mathcal{B}(x_1, y_1) + \mathcal{B}(x_2, y_2) \qquad \cdots \qquad \chi_\rho = \sum_{i=1}^{\rho} \mathcal{B}(x_i, y_i) \tag{3}$$

Next, we require the unbinding operation, which is defined via an inverse function in the following theorem. This will give a symbolic form of our unbinding step that retrieves the original component $\boldsymbol{x}$ being searched for, as well as a necessary noise component $\boldsymbol{\eta}^\circ$, which must exist whenever $\rho \geqslant 2$ items are bound together without expanding the dimension d.

Theorem 2.1 (Inverse Theorem)**.** *Given the identity function $Hx \cdot Hx^\dagger = \mathbb{1}$ where $x^\dagger$ is the inverse of x in the Hadamard domain, then $\mathcal{B}^*(\mathcal{B}(x_1, y_1) + \cdots + \mathcal{B}(x_\rho, y_\rho), y_i^\dagger) = \begin{cases} x_i & \text{if} \quad \rho = 1 \\ x_i + \eta_i^\circ & \text{else } \rho > 1 \end{cases}$*

where $x_i, y_i \in \mathbb{R}^d$ and η_i° is the noise component.

Proof of Theorem 2.1. We start from the identity function $Hx \cdot Hx^{\dagger} = \mathbb{1}$ and thus $Hx^{\dagger} = \frac{\mathbb{1}}{Hx}$. Now using Equation 2 we get,

$$\mathcal{B}^*(\mathcal{B}(x_1, y_1) + \cdots + \mathcal{B}(x_\rho, y_\rho), y_i^{\dagger}) = \frac{1}{d} \cdot H((Hx_1 \odot Hy_1 + \cdots + Hx_\rho \odot Hy_\rho) \odot \frac{1}{Hy_i})$$

$$= \frac{1}{d} \cdot H(Hx_i + \frac{1}{Hy_i} \odot \sum_{\substack{j=1 \\ j \neq i}}^{\rho} (Hx_j \odot Hy_j)) = x_i + \frac{1}{d} \cdot H(\frac{1}{Hy_i} \odot \sum_{\substack{j=1 \\ j \neq i}}^{\rho} (Hx_j \odot Hy_j)) \quad \textit{Lemma } 2.1$$

$$= \begin{cases} x_i & \text{if} \quad \rho = 1 \\ x_i + \eta_i^{\circ} & \text{else } \rho > 1 \end{cases}$$

□

To reduce the noise component and improve retrieval accuracy, [10, 32] proposes a projection step to the input vectors by normalizing them by the absolute value in the Fourier domain. While such identical normalization is not useful in the Hadamard domain since it will only transform the elements to $+1$s and -1s, we will define a projection step with only the Hadamard transformation without normalization given in Definition 2.2.

Definition 2.2 (Projection)**.** The projection function of x is defined by $\pi(x) = \frac{1}{d} \cdot Hx$.

If we apply the Definition 2.2 to the inputs in Theorem 2.1 then we get

$$\mathcal{B}^*(\mathcal{B}(\pi(x_1), \pi(y_1)) + \cdots + \mathcal{B}(\pi(x_\rho), \pi(y_\rho)), \pi(y_i)^{\dagger}) = \mathcal{B}^*(\frac{1}{d} \cdot H(x_1 \odot y_1 + \cdots x_\rho \odot y_\rho), \frac{1}{y_i})$$
$$= \frac{1}{d} \cdot H(\frac{1}{y_i} \odot (x_1 \odot y_1 + \cdots x_\rho \odot y_\rho)) \tag{4}$$

The retrieved value is projected onto the Hadamard domain, and to recover the original data we apply the reverse projection. Since the Hadamard matrix is its own inverse, in the reverse projection step we just apply the Hadamard transformation again which derives the output to

$$H(\frac{1}{d} \cdot H(\frac{1}{y_i} \odot (x_1 \odot y_1 + \cdots x_\rho \odot y_\rho))) = \frac{1}{y_i} \odot (x_1 \odot y_1 + \cdots + x_\rho \odot y_\rho)$$
$$= \begin{cases} x_i & \text{if} \quad \rho = 1 \\ x_i + \sum\limits_{j=1,\ j \neq i}^{\rho} \frac{x_j y_j}{y_i} & \text{else } \rho > 1 \end{cases} = \begin{cases} x_i & \text{if} \quad \rho = 1 \\ x_i + \eta_i^{\pi} & \text{else } \rho > 1 \end{cases} \tag{5}$$

where η_i^{π} is the noise component due to the projection step. In expectation, $\eta_i^{\pi} < \eta_i^{\circ}$ (see Appendix C). Thus, the projection step diminishes the accumulated noise. More interestingly, the retrieved output term does not contain any Hadamard matrix. Therefore, we can recast the initial binding definition by multiplying the query vector y_i to the output of Equation 5, i.e, the binding function becomes $\mathcal{B}'(x, y) = x \odot y$. ρ bundle of the vector pairs is $\chi'_\rho = \sum_{i=1}^{\rho}(x_i \odot y_i)$ and the unbinding would be $\mathcal{B}^{*\prime}(x, y) = x \odot \frac{1}{y}$.

2.1 Initialization of HLB

For binding and unbinding operations to work, vectors need an expected value of zero. However, dividing the bound vector by the query during unbinding can cause instability if values are too close to zero. Thus, we define a Mixture of Normal Distribution (MiND) with a zero expected value but a non-zero absolute mean, as given in Equation 6, where $\mathcal{U}$ is the Uniform distribution. Half the elements are sampled from a normal distribution with mean $-\mu$, and the other half from a distribution with mean μ, yielding a vector with a zero mean and absolute mean of μ. The properties of the vectors sampled from a MiND distribution are given in Properties 2.1.

$$\Omega(\mu, 1/d) = \begin{cases} \mathcal{N}(-\mu, 1/d) & \text{if} \quad \mathcal{U}(0,1) > 0.5 \\ \mathcal{N}(\ \mu, 1/d) & \text{else } \mathcal{U}(0,1) \leqslant 0.5 \end{cases} \tag{6}$$

Properties 2.1 (Initialization Properties)**.** *Let $x \in \mathbb{R}^d$ sampled from $\Omega(\mu, 1/d)$ holds the following properties.* $\mathrm{E}[x] = 0,\ \mathrm{E}[|x|] = \mu$*, and* $\|x\|_2 = \sqrt{\mu^2 d}$

2.2 Similarity Augmentation

In VSAs, it is common to measure the similarity with an extracted embedding $\hat{\boldsymbol{x}}$ with some other vector $\boldsymbol{x}$ using the cosine similarity. For our HLB, we devise a correction term leveraging Theorem 2.2.

Theorem 2.2 (ϕ – ρ Relationship)**.** *Given $x_i, y_i \sim \Omega(\mu, 1/d)\ \forall\ i \in \mathbb{N} : 1 \leqslant i \leqslant \rho$, the cosine similarity ϕ between the original x_i and retrieved vector $\hat{x}_i$ is approximately equal to the inverse square root of the number of vector pairs in a composite representation ρ given by $\phi \approx \frac{1}{\sqrt{\rho}}$.*

The proof of the theorem is provided in Appendix B. The experimental result of the $\phi - \rho$ relationship closely follows the theoretical expectation provided in Appendix E which also indicates that the approximation is valid. We know from Theorem 2.2 that the similarity score ϕ drops by the inverse square root of the number of vector pairs in a composite representation ρ. Therefore, in places where ρ is known or can be estimated from $\left\|\chi_\rho\right\|_2 \approx \mu^2\sqrt{\rho \cdot d}$ (proof in Appendix D), it can be used to update the cosine similarity multiplying the scores by $\sqrt{\rho}$. Equation 7 shows the updated similarity score where in a positive case $(+)$, ϕ would be close to $1/\sqrt{\rho}$ and in a negative case $(-)$, ϕ would be close to zero.

$$\phi' = \phi \times \sqrt{\rho} \qquad \phi'_{(+)} = \phi_{\to \frac{1}{\sqrt{\rho}}} \times \sqrt{\rho}\ \approx 1 \qquad \phi'_{(-)} = \phi_{\to 0} \times \sqrt{\rho} \quad \approx 0 \tag{7}$$

3 Empirical Results

3.1 Classical VSA Tasks

A common VSA task is, given a bundle (addition) of ρ pairs of bound vectors $\boldsymbol{s} = \sum_{i=1}^{\rho} \mathcal{B}(\boldsymbol{x}_i, \boldsymbol{y}_i)$, given a query $\boldsymbol{x}_q \in \boldsymbol{s}$, can the corresponding vector $\boldsymbol{y}_q$ be correctly retrieved from the bundle. To test this, we perform an experiment similar to one in [37]. We first create a pool P of $N = 1000$ random vectors, then sample (with replacement) p pairs of vectors for $p \in \{1, 2, \cdots, 25\}$. The pairs are bound together and added to create a composite representation $\boldsymbol{s}$. Then, we iterate through all left pairs $\boldsymbol{x}_q$ in the composite representation and attempt to retrieve the corresponding $\boldsymbol{y}_q, \forall q \in [1, p]$. A retrieval is considered correct if $\mathcal{B}^*(\boldsymbol{s}, \boldsymbol{x}_q)^\top \boldsymbol{y}_q > \mathcal{B}^*(\boldsymbol{s}, \boldsymbol{x}_q)^\top \boldsymbol{y}_j, \forall j \neq q$. The total accuracy score for the bundle is recorded, and the experiment is repeated for 50 trials. Experiments are performed to compare HRR [32], VTB [13], MAP [11], and our HLB VSAs. For each VSA, the area under the curve (AUC) of the accuracy vs. the number of bound terms plot is computed, and the results are shown in Figure 5. In general, HLB has comparable performance to HRR and VTB, and performs better than MAP.

The scenario we just considered looked at bindings of only two items together, summed of many pairs of bindings. [13] proposed addition evaluations over sequential bindings that we now consider. In the *random* case we have an initial vector $\boldsymbol{b}_0$, and for p rounds, we will modify it by a random vector $\boldsymbol{x}_t$ such that $\boldsymbol{b}_{t+1} = \mathcal{B}(\boldsymbol{b}_t, \boldsymbol{x}_t)$, after which we unbind each $\boldsymbol{x}_t$ to see how well the previous $\boldsymbol{b}_t$ is recovered. In the *auto binding* case, we use a single random vector $\boldsymbol{x}$ for all p rounds. Figure 6 shows that HLB maintains a stable magnitude regardless of the number of bound vectors in both cases. This property arises due to the properties of the distribution shown in Properties 2.1. As all components have an expected absolute value of 1, the product of all components also has an expected absolute value of 1. Thus, the norm of the binding is simply $\sqrt{d}$. Combined with Figure 1 that shows the scores are near-zero when an item is not present, HLB has significant advantages in consistency for designing VSA solutions.

3.2 Deep Learning with Hadamard-derived Linear Binding

Two recent methods that integrate HRR with deep learning are tested to further validate our approach. The details of each method are provided in Appendix G and Appendix H. In each case, we run all four VSAs and see that HLB either matches or exceeds the performance of other VSAs. In every experiment, the standard method of sampling vectors from each VSA is followed as outlined in Table 3. All the experiments are performed on a single NVIDIA TESLA PH402 GPU with 32GB memory.

3.2.1 Connectionist Symbolic Pseudo Secrets

Connectionist Symbolic Pseudo Secrets (CSPS) [3] is a pseudo-encryption method used to deploy convolutional networks on untrusted platforms by binding secrets to the inputs. More details are available in Appendix G. Experiments are performed with 5 datasets: MNIST, SVHN, CIFAR-10 (CR10), CIFAR-100 (CR100), and Mini-ImageNet (MIN). First, we look at the accuracy of each method, which is lower due to the noise of the random vector $\boldsymbol{s}$ added at test time since no secret VSA is ever reused. The results are shown in Table 1, where HLB outperforms all prior methods significantly. Notably, the MAP VSA is second best despite being one of the older VSAs, indicating its similarity to HLB in using a simple binding procedure, and thus simple gradient may be an important factor in this scenario.

Table 1: Accuracy comparison of the proposed HLB with HRR, VTB, MAP-C, and MAP-B in CSPS. The dimensions of the inputs along with the no. of classes are listed in the Dims/Labels column. The last row shows the geometric mean of the results.

Dataset	Dims/ Labels	CSPS + HRR		CSPS + VTB		CSPS + MAP-C		CSPS + MAP-B		CSPS + HLB	
		Top@1	Top@5	Top@1	Top@5	Top@1	Top@5	Top@1	Top@5	Top@1	Top@5
MNIST	$28^2/10$	98.51	–	98.44	–	98.46	–	98.40	–	**98.73**	–
SVHN	$32^2/10$	88.44	–	19.59	–	79.95	–	92.43	–	**94.53**	–
CR10	$32^2/10$	78.21	–	74.22	–	76.69	–	82.83	–	**83.81**	–
CR100	$32^2/100$	48.84	75.82	35.87	61.79	56.77	81.52	57.76	84.63	**58.82**	**87.50**
MIN	$84^2/100$	40.99	66.99	45.81	73.52	52.22	78.63	57.91	82.81	**59.48**	**83.35**
GM		67.14	71.26	47.24	67.40	70.89	80.06	75.90	83.72	**77.17**	**85.40**

However, improved accuracy is not useful in this scenario if more information is leaked. The test in this scenario, as proposed by [3], is to calculate the Adjusted Rand Index (ARI) after attempting to cluster the inputs $\boldsymbol{x}$ and the outputs $\hat{\boldsymbol{y}}$, which are available/visible to the snooping third-party. To be successful, the Adjusted Rand Index (ARI) must be near zero (indicating random label assignment) for both inputs and outputs. We use K-means, Gaussian Mixture Model (GMM), Birch [45], and HDBSCAN [8] as the clustering algorithms and specify the true number of classes to each method to maximize attacker success (information they would not know). The results can be found in Table 4, where the top rows indicate the clustering of the input $\mathcal{B}(\boldsymbol{x}, \boldsymbol{s})$, and the bottom rows the clustering of the output $\hat{\boldsymbol{y}}$. All the numbers are percentages $(\%)$, showing all methods do a good job at hiding information from the adversary (except on the MNIST dataset, which is routinely degenerate).

3.2.2 Xtreme Multi-Label Classification

Extreme Multi-label (XML) classification is a challenging task due to the large output space, which can be in the range of hundreds of thousands, contributing to the bulk of the size of a neural network. While many prior works focus on innovative strategies to cluster/make hierarchies/compress the penultimate layer [20, 21, 31, 19, 44, 23], a neuro-symbolic approach was proposed by [10]. Given K total possible classes, they assigned each class a vector $\boldsymbol{c}_k$ to be each class's representation, and the set of all classes $\boldsymbol{a} = \sum_{k=1}^{K} \boldsymbol{c}_k$. More details about the XML methodology is provided in Appendix H. The details and network sizes of [10] are followed, except we replace the original VSA with our four candidates. The network is trained on 8 datasets listed in Table 2 from [5] and evaluated using normalized discounted cumulative gain (nDCG) and propensity-scored (PS) based normalized discounted cumulative gain (PSnDCG) as suggested by [20].

The classification result in terms of nDCG and PSnDCG in all the eight datasets is presented in Table 2 where the top four datasets are comparatively easy with maximum no. of features of 5000 and no. of labels of 4000. The bottom four datasets are comparatively hard with the no. of features and labels on the scale of $100K$. The proposed HLB has attained the best results in all the datasets on both metrics. In contrast to the prior CSPS results, here we see that the performance differences between HRR, VTB, and MAP are more varied, with no clear "second-place" performer.

Table 2: XML classification results in dense label representation with HRR, VTB, MAP, and HLB in terms of nDCG and PSnDCG. The proposed HLB has attained the best nDCG and PSnDCG scores on all the datasets setting a new SOTA.

DATASET	BIBTEX		DELICIOUS		MEDIAMILL		EURLEX-4K	
METRICS	nDCG	PSnDCG	nDCG	PSnDCG	nDCG	PSnDCG	nDCG	PSnDCG
HRR	60.296	45.572	66.454	30.016	83.885	63.684	77.225	30.684
VTB	57.693	45.219	63.325	31.449	87.232	66.948	76.964	31.180
MAP-C	59.280	46.092	65.376	31.943	87.255	66.886	72.439	26.752
MAP-B	59.412	46.340	65.431	32.122	86.886	66.562	71.128	26.340
HLB	**61.741**	**48.639**	**67.821**	**32.797**	**88.064**	**67.525**	**77.868**	**31.526**
DATASET	EURLEX-4.3K		WIKI10-31K		AMAZON-13K		DELICIOUS-200K	
METRICS	nDCG	PSnDCG	nDCG	PSnDCG	nDCG	PSnDCG	nDCG	PSnDCG
HRR	84.497	38.545	81.068	9.185	93.258	49.642	44.933	6.839
VTB	84.663	38.540	78.025	9.645	92.373	49.463	44.092	6.664
MAP-C	85.472	39.233	80.203	10.027	92.013	48.686	45.373	6.862
MAP-B	85.023	38.820	80.238	10.035	92.307	48.812	45.459	6.870
HLB	**88.204**	**43.622**	**83.589**	**11.869**	**93.672**	**50.270**	**46.331**	**6.952**

4 Conclusion

In this paper, a novel linear vector symbolic architecture named HLB is presented derived from Hadamard transform. Along with an initialization condition named MiND distribution is proposed for which we proved the cosine similarity ϕ is approximately equal to the inverse square root of the no. of bundled vector pairs ρ which matches with the experimental results. The proposed HLB showed superior performance in classical VSA tasks and deep learning compared to other VSAs such as HRR, VTB, and MAP. In learning tasks, HLB is applied to CSPS and XML classification tasks. In both of the tasks, HLB has achieved the best results in terms of respective metrics in all the datasets showing a diverse potential of HLB in Neuro-symbolic AI.

References

[1] Mohammad Mahmudul Alam, Edward Raff, Stella Biderman, Tim Oates, and James Holt. Recasting self-attention with holographic reduced representations. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[2] Mohammad Mahmudul Alam, Edward Raff, and Tim Oates. Towards generalization in subitizing with neuro-symbolic loss using holographic reduced representations. *Neuro-Symbolic Learning and Reasoning in the era of Large Language Models*, 2023.

[3] Mohammad Mahmudul Alam, Edward Raff, Tim Oates, and James Holt. Deploying convolutional networks on untrusted platforms using 2d holographic reduced representations. *arXiv preprint arXiv:2206.05893*, 2022.

[4] Mohammad Mahmudul Alam, Edward Raff, Tim Oates, and James Holt. Deploying convolutional networks on untrusted platforms using 2D holographic reduced representations. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 367–393. PMLR, 17–23 Jul 2022.

[5] K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. The extreme classification repository: Multi-label datasets and code, 2016.

[6] Peter Blouw and Chris Eliasmith. A neurally plausible encoding of word order information into a semantic vector space. *35th Annual Conference of the Cognitive Science Society*, 35:1905–1910, 2013.

[7] Peter Blouw, Eugene Solodkin, Paul Thagard, and Chris Eliasmith. Concepts as semantic pointers: A framework and computational model. *Cognitive Science*, 40(5):1128–1162, 7 2016.

[8] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[9] C. Eliasmith, T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen. A large-scale model of the functioning brain. *Science*, 338(6111):1202–1205, 11 2012.

[10] Ashwinkumar Ganesan, Hang Gao, Sunil Gandhi, Edward Raff, Tim Oates, James Holt, and Mark McLean. Learning with holographic reduced representations. *Advances in neural information processing systems*, 34:25606–25620, 2021.

[11] Ross W. Gayler. Multiplicative binding, representation operators & analogy (workshop poster), 1998.

[12] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International conference on machine learning*, pages 201–210. PMLR, 2016.

[13] Jan Gosmann and Chris Eliasmith. Vector-derived transformation binding: An improved binding operation for deep symbol-like processing in neural networks. *Neural Comput.*, 31(5):849–869, 2019.

[14] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv*, 2020.

[15] Robert Guirado, Abbas Rahimi, Geethan Karunaratne, Eduard Alarcón, Abu Sebastian, and Sergi Abadal. Whype: A scale-out architecture with wireless over-the-air majority for scalable in-memory hyperdimensional computing. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 13(1):137–149, 2023.

[16] Mike Heddes, Igor Nunes, Pere VergÃ©s, Denis Kleyko, Danny Abraham, Tony Givargis, Alexandru Nicolau, and Alexander Veidenbaum. Torchhd: An open source python library to support research on hyperdimensional computing and vector symbolic architectures. *Journal of Machine Learning Research*, 24(255):1–10, 2023.

[17] Qiuyuan Huang, Paul Smolensky, Xiaodong He, Li Deng, and Dapeng Wu. Tensor product generation networks for deep nlp modeling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1263–1273. Association for Computational Linguistics, jun 2018.

[18] Mohsen Imani, Deqian Kong, Abbas Rahimi, and Tajana Rosing. Voicehd: Hyperdimensional computing for efficient speech recognition. In *2017 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–8. IEEE, nov 2017.

[19] Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, and Manik Varma. Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 528–536, 2019.

[20] Himanshu Jain, Yashoteja Prabhu, and Manik Varma. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 935–944, 2016.

[21] Ankit Jalan and Purushottam Kar. Accelerating extreme classification via adaptive feature agglomeration. *arXiv preprint arXiv:1905.11769*, 2019.

[22] Michael N. Jones and Douglas J.K. Mewhort. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114(1):1–37, 2007.

[23] Armand Joulin, Moustapha Cissé, David Grangier, Hervé Jégou, et al. Efficient softmax approximation for gpus. In *International conference on machine learning*, pages 1302–1310. PMLR, 2017.

[24] 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, 2022.

[25] Denis Kleyko, Dmitri Rachkovskij, Evgeny Osipov, and Abbas Rahimi. A survey on hyperdimensional computing aka vector symbolic architectures, part ii: Applications, cognitive models, and challenges. *ACM Comput. Surv.*, 55(9), jan 2023.

[26] 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 Comput. Surv.*, 55(6), dec 2022.

[27] Siyu Liao and Bo Yuan. Circconv: A structured convolution with low complexity. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4287–4294, Jul. 2019.

[28] Mohammad Mahmudul Alam, Edward Raff, Stella R Biderman, Tim Oates, and James Holt. Holographic global convolutional networks for long-range prediction tasks in malware detection. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 4042–4050. PMLR, 02–04 May 2024.

[29] Nicolas Menet, Michael Hersche, Geethan Karunaratne, Luca Benini, Abu Sebastian, and Abbas Rahimi. Mimonets: Multiple-input-multiple-output neural networks exploiting computation in superposition. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 39553–39565. Curran Associates, Inc., 2023.

[30] Peer Neubert, Stefan Schubert, and Peter Protzel. Learning vector symbolic architectures for reactive robot behaviours. In *Proc. of Intl. Conf. on Intelligent Robots and Systems (IROS) Workshop on Machine Learning Methods for High-Level Cognitive Capabilities in Robotics*, 2016.

[31] Alexandru Niculescu-Mizil and Ehsan Abbasnejad. Label filters for large scale multilabel classification. In *Artificial intelligence and statistics*, pages 1448–1457. PMLR, 2017.

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

[33] Rebecca Saul, Mohammad Mahmudul Alam, John Hurwitz, Edward Raff, Tim Oates, and James Holt. Lempel-ziv networks. In *Proceedings on "I Can't Believe It's Not Better! - Understanding Deep Learning Through Empirical Falsification" at NeurIPS 2022 Workshops*, volume 187 of *Proceedings of Machine Learning Research*, pages 1–11. PMLR, 03 Dec 2023.

[34] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9355–9366. PMLR, 2021.

[35] Imanol Schlag and Jürgen Schmidhuber. Learning to reason with third order tensor products. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[36] Kenny Schlegel, Peer Neubert, and Peter Protzel. A comparison of vector symbolic architectures. *arXiv*, 2020.

[37] Kenny Schlegel, Peer Neubert, and Peter Protzel. A comparison of vector symbolic architectures. *Artificial Intelligence Review*, 55(6):4523–4555, Aug 2022.

[38] Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1):159–216, 1990.

[39] Julia Steinberg and Haim Sompolinsky. Associative memory of structured knowledge. *Scientific Reports*, 12(1), December 2022.

[40] Terrence C. Stewart and Chris Eliasmith. Large-scale synthesis of functional spiking neural circuits. *Proceedings of the IEEE*, 102(5):881–898, 2014.

[41] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[42] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 3009–3016. AAAI Press, 2020.

[43] Ryosuke Yamaki, Tadahiro Taniguchi, and Daichi Mochihashi. Holographic CCG parsing. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 262–276, Toronto, Canada, July 2023. Association for Computational Linguistics.

[44] Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. *Advances in neural information processing systems*, 32, 2019.

[45] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.

A Related Work

Smolensky [38] started the VSA approach with the Tensor Product Representation (TPR), where d dimensional vectors (each representing some concept) were bound by computing an outer product. Showing distributivity ($\mathcal{B}(\boldsymbol{x}, \boldsymbol{y} + \boldsymbol{z}) = \mathcal{B}(\boldsymbol{x}, \boldsymbol{y}) + \mathcal{B}(\boldsymbol{x}, \boldsymbol{z})$) and associativity, this allowed specifying logical statements/structures [14]. However, for ρ total items to be bound together, it was impractical due to the $\mathcal{O}(d^\rho)$ complexity. [36, 26, 25] have surveyed many of the VSAs available today, but our work will focus on three specific alternatives, as outlined in Table 3. The Vector-Derived Transformation Binding (VTB) will be a primary comparison because it is one of the most recently developed VSAs, which has shown improvements in what we will call "classic" tasks, where the VSA's symbolic like properties are used to manually construct a series of binding/unbinding operations that accomplish a desired task. Note, that the VTB is unique in it is non-symmetric ($\mathcal{B}(\boldsymbol{x}, \boldsymbol{y}) \neq \mathcal{B}(\boldsymbol{y}, \boldsymbol{x})$). Ours, and most others, are symmetric.

Table 3: The binding and initialization mechanisms for our new HLB with baseline methods. HLB is related to the HRR in being derived via a similar approach, but replacing the Fourier transform $\mathcal{F}(\cdot)$ with the Hadamard transform (which simplifies out). The MAP is most similar to our approach in mechanics, but the difference in derived unbinding steps leads to dramatically different performance. The VTB is the most recently developed VSA in modern use. The matrix $V_{\boldsymbol{y}}$ of VTB is a block-diagonal matrix composed from the values of the $\boldsymbol{y}$ vector, which we refer the reader to [13] for details. The TorchHD library [16] is used for implementations of prior methods.

METHOD	BIND $\mathcal{B}(x, y)$	UNBIND $\mathcal{B}^*(x, y)$	INIT x
HRR	$\mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{x}) \odot \mathcal{F}(\boldsymbol{y}))$	$\mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{x}) \oslash \mathcal{F}(\boldsymbol{y}))$	$x_i \sim \mathcal{N}(0, 1/d)$
VTB	$V_y x$	$V_y^\top x$	$\tilde{x}_i \sim \mathcal{N}(0, 1) \rightarrow x = \tilde{\boldsymbol{x}}/\|\tilde{\boldsymbol{x}}\|_2$
MAP-C	$x \odot y$	$x \odot y$	$x_i \sim \mathcal{U}(-1, 1)$
MAP-B	$x \odot y$	$x \odot y$	$x_i \sim \{-1, 1\}$
HLB	$x \odot y$	$x \oslash y$	$x_u \sim \{\mathcal{N}(-\mu, 1/d),\ \mathcal{N}(\mu, 1/d)\}$

Next is the Holographic Reduced Representation (HRR) [32], which can be defined via the Fourier transform $\mathcal{F}(\cdot)$. One derives the inverse operation of the HRR by defining the one vector $\vec{1}$ as the identity vector and then solving $\mathcal{F}(\boldsymbol{a}^*)_i \mathcal{F}(\boldsymbol{a})_i = 1$. We will use a similar approach to deriving HLB but replacing the Fourier Transform with the Hadamard transform, making the HRR a key baseline. Last, the Multiply Add Permute (MAP) [11] is derived by taking only the diagonal of the tensor product from [38]'s TPR. This results in a surprisingly simple representation of using element-wise multiplication for both binding/unbinding, making it a key baseline. The MAP binding is also notable for its continuous (MAP-C) and binary (MAP-B) forms, which will help elucidate the importance of the difference in our unbinding step compared to the initialization avoiding values near zero. HLB differs in devising for the unbinding step, and we will later show an additional corrective term that HLB employs for ρ different items bound together, that dramatically improve performance.

Our motivation for using the Hadamard Transform comes from its parallels to the Fourier Transform (FT) used to derive the HRR and the HRR's relatively high performance. The Hadamard matrix has a simple recursive structure, making analysis tractable, and its transpose is its own inverse, which simplifies the design of the inverse function $\mathcal{B}^*$. Like the FT, WHT can be computed in log-linear time, though in our case, the derivation results in linear complexity as an added benefit. The WHT is already associative and distributive, making less work to obtain the desired properties. Finally, the WHT involves only $\{-1, 1\}$ values, avoiding numerical instability that can occur with the HRR/FT. This work shows that these motivations are well founded, as they result in a binding with comparable or improved performance in our testing.

Our interest in VSAs comes from their utility in both classical symbolic tasks and as useful priors in designing deep learning systems. In classic tasks VSAs are popular for designing power-efficient systems from a finite set of operations [15, 24, 18, 30]. HRRs, in particular, have shown biologically plausible models of human cognition [22, 6, 40, 7] and solving cognitive science tasks [9]. In deep learning the TPR has inspired many prior works in natural language processing [34, 17, 35]. To wit, The HRR operation has seen the most use in differentiable systems [43, 41, 42, 27, 29, 33, 1, 2, 28]. To study our method, we select two recent works that make heavy use of the neuro-symbolic capabilities

of HRRs. First, an Extreme Multi-Label (XML) task that uses HRRs to represent an output space of tens to hundreds of thousands of classes C in a smaller dimension $d < C$ [10], and an information privacy task that uses the HRR binding as a kind of "encrypt/decrypt" mechanism for heuristic security [4]. We will explain these methods in more detail in the experimental section.

B Similarity Augmentation Proof

Proof of Theorem 2.2. We start with the definition of cosine similarity and insert the value of $\hat{x_i}$. The step-by-step breakdown is shown in Equation 8.

$$\phi = \frac{\sum_{}^{d} x_i \cdot \hat{x_i}}{\|x_i\|_2 \cdot \|\hat{x_i}\|_2} = \frac{\sum x_i \cdot \left(x_i + \sum_{j=1,\ j\neq i}^{\rho} \frac{x_j y_j}{y_i} \right)}{\|x_i\|_2 \cdot \|x_i + \sum_{j=1,\ j\neq i}^{\rho} \frac{x_j y_j}{y_i}\|_2} = \frac{\sum x_i \cdot x_i + \sum x_i \cdot \left(\sum_{j=1,\ j\neq i}^{\rho} \frac{x_j y_j}{y_i} \right)}{\|x_i\|_2 \cdot \|x_i + \sum_{j=1,\ j\neq i}^{\rho} \frac{x_j y_j}{y_i}\|_2} \tag{8}$$

Employing Properties 2.1 we can derive that $\|x_i\|_2 = \sqrt{\sum x_i \cdot x_i} = \sqrt{\mu^2 d}$ and $\|\frac{x_j y_j}{x_i}\| = \sqrt{\mu^2 d}$. Thus, the square of the $\|x_i + \sum_{j=1,\ j\neq i}^{\rho} \frac{x_j y_j}{y_i}\|_2$ can be expressed as

$$= \|x_i\|_2^2 + \sum_{j=1,\ j\neq i}^{\rho} \left\| \frac{x_j y_j}{y_i} \right\|_2^2 + \underbrace{2 \cdot \sum x_i \left(\sum_{j=1,\ j\neq i}^{\rho} \frac{x_j y_j}{y_i} \right)}_{\alpha} + \underbrace{\sum_{}^{d} \sum_{\substack{j=1 \\ j\neq i}}^{\rho-1} \sum_{\substack{l=1 \\ l\neq j}}^{\rho-1} \frac{x_j y_j}{y_i} \cdot \frac{x_l y_l}{y_i}}_{\beta} \tag{9}$$

$$= \mu^2 d + (\rho - 1) \cdot \mu^2 d + 2\alpha + 2\beta \quad = \rho \cdot \mu^2 d + 2\alpha + 2\beta$$

Therefore, using Equation 8 and Equation 9 we can write that

$$\mathrm{E}[\phi] = \frac{\mu^2 d + \alpha}{\sqrt{\mu^2 d} \cdot \sqrt{\rho \cdot \mu^2 d + 2\alpha + 2\beta}} \approx^{1} \frac{\mu^2 d}{\sqrt{\mu^2 d} \cdot \sqrt{\rho \cdot \mu^2 d}} = \frac{\mu^2 d}{\sqrt{\rho} \cdot \mu^2 d} = \frac{1}{\sqrt{\rho}} \qquad \square$$

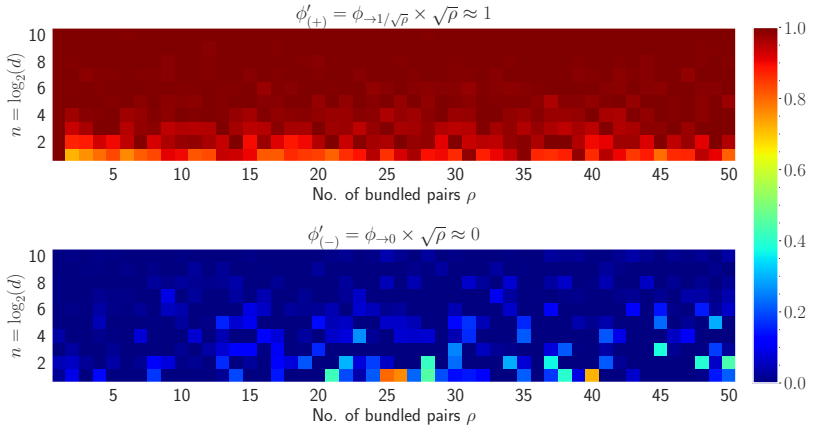

Figure 1: Empirical comparison of the corrected cosine similarity scores between $\phi'_{(+)}$ (on top) and $\phi'_{(-)}$ (on bottom) for varying n and ρ shown in heatmap. The dimension, i.e., $d = 2^n$ is varied from 2 to 1024 ($n \in \{1, 2, \cdots, 10\}$) and the number of vector pairs bundled is varied from 1 to 50. This shows that we can accurately identify when a vector $\boldsymbol{x}$ has been bound to a VSA or not when we keep track of how many pairs of terms ρ are included.

Empirical results of ϕ' for varying n and ρ are visualized and verified by a heatmap. In a composite representation $\chi'_\rho = \sum_{i=1}^{\rho} (x_i \odot y_i)$, when unbinding is applied using the query y_i, i.e., $\mathcal{B}^{*\prime}(\chi'_\rho, y_i) =$

[1]Here, α and β are the noise terms and in expectation $\mathrm{E}[\alpha] \approx 0$ and $\mathrm{E}[\beta] \approx 0$.

$\hat{x_i}$, a positive case is a similarity between x_i and $\hat{x_i}$. On the contrary, similarity between $\hat{x_i}$ and any x_j where $j \in \{1, 2, \cdots, \rho\}$ and $j \neq i$, is a negative case. Mean cosine similarity scores of 100 trials for both positive and negative cases in presented in Figure 1 where the scores for the positive cases are in the red (≈ 1) shades and the scores for the negative cases are in the blue (≈ 0) shades.

C Noise Decomposition

When a single vector pair is combined, one of the vector pairs can be exactly retrieved with the help of the other component and the inverse function, recalling the retrieved output does not contain any noise component for a single pair of vectors, i.e., $\rho = 1$. However, when more than one vector pairs are bundled, noise starts to accumulate. In this section, we will uncover the noise components accumulated with and without the projection to the inputs and analyze their impact on expectation. We first start with the noise component without the projection step η_i°.

$$\eta_i^\circ = \frac{1}{d} \cdot H(\frac{1}{Hy_i} \odot \sum_{\substack{j=1 \\ j \neq i}}^{\rho} (Hx_j \odot Hy_j) \tag{10}$$

Let, set the value of n to be 1 thus, $d = 2^n = 2$ and the number of vector pairs $\rho = 2$, i.e., $\chi_{\rho=2} = \mathcal{B}(x_1, y_1) + \mathcal{B}(x_2, y_2)$. We want to retrieve x_1 using the query y_1, thereby, the expression of η_i° is uncovered step by step for $\rho = 2$ shown in Equation 11.

$$\begin{aligned}
\underset{\rho=2}{\eta_i^\circ} &= \frac{1}{d} \cdot H(\frac{1}{Hy_1} \odot (Hx_2 \odot Hy_2)) \\
&= \frac{1}{d} \cdot \sqrt{d} \cdot H \begin{pmatrix} \frac{1}{y_1^{(0)} + y_1^{(1)}} \\ \frac{1}{y_1^{(0)} - y_1^{(1)}} \end{pmatrix} \odot \begin{pmatrix} (x_2^{(0)} + x_2^{(1)}) \cdot (y_2^{(0)} + y_2^{(1)}) \\ (x_2^{(0)} - x_2^{(1)}) \cdot (y_2^{(0)} - y_2^{(1)}) \end{pmatrix} \\
&= \frac{1}{d} \cdot d \cdot \begin{pmatrix} \frac{(x_2^{(0)} + x_2^{(1)})\,(y_2^{(0)} + y_2^{(1)})\,(y_1^{(0)} - y_1^{(1)}) + (x_2^{(0)} - x_2^{(1)})\,(y_2^{(0)} - y_2^{(1)})\,(y_1^{(0)} + y_1^{(1)})}{(y_1^{(0)} + y_1^{(1)})\,(y_1^{(0)} - y_1^{(1)})} \\ \frac{(x_2^{(0)} + x_2^{(1)})\,(y_2^{(0)} + y_2^{(1)})\,(y_1^{(0)} - y_1^{(1)}) - (x_2^{(0)} - x_2^{(1)})\,(y_2^{(0)} - y_2^{(1)})\,(y_1^{(0)} + y_1^{(1)})}{(y_1^{(0)} + y_1^{(1)})\,(y_1^{(0)} - y_1^{(1)})} \end{pmatrix} \\
&= \begin{pmatrix} \frac{\varphi_1}{\prod_{k=1}^{d} (Hy_1)_k} \\ \frac{\varphi_2}{\prod_{k=1}^{d} (Hy_1)_k} \end{pmatrix} \\
&= \frac{\mathcal{P}(x_2, y_2, y_1)}{\prod_{k=1}^{d} (Hy_1)_k}
\end{aligned} \tag{11}$$

Here, $\varphi_k \; \forall \; k \in \mathbb{N} : 1 \leqslant k \leqslant d$ are the polynomials comprises of $(x_2,\ y_2)$, and the query vector y_1. $\mathcal{P}$ is the vector of polynomials consisting of φ_k. From the noise expression, we can observe that the numerator is a polynomial and the denominator is the product of all the elements of the Hadamard transformation of the query vector. This is true for any value of n and ρ. Thus, in general, for any query y_i we can express η_i° as shown in Equation 12.

$$\eta_i^\circ = \frac{\overset{\rho}{\underset{j=1,\ j \neq i}{\mathcal{P}}} (x_j, y_j, y_i)}{\prod_{k=1}^{d} (Hy_i)_k} \tag{12}$$

The noise accumulated after applying the projection to the inputs is quite straightforward as given in Equation 13.

$$\eta_i^\pi = \frac{\sum_{j=1,\ j \neq i}^{\rho} (x_j \odot y_j)}{y_i} \tag{13}$$

Although the vectors $x_i, y_i \; \forall \; i \in \mathbb{N} : 1 \leqslant i \leqslant \rho$ are sampled from a MiND with an expected value of 0 given in Equation 6, the sample mean of x_i or y_i would be $\hat{\mu} \approx 0$ but $\hat{\mu} \neq 0$. Both the numerator of η_i° and η_i^{π} are the polynomials thus the expected value would be very close to 0. However, the expected value of the denominator of η_i° would be $\mathrm{E}[\prod_{k=1}^{d}(Hy_i)_k] = \prod_{k=1}^{d}\mathrm{E}[(Hy_i)_k] = \hat{\mu}^d$ whereas the expected value of the denominator of η_i^{π} is $\mathrm{E}[y_i] = \hat{\mu}$. Since, $\hat{\mu}^d < \hat{\mu}$, hence, in expectation $\eta_i^{\pi} < \eta_i^{\circ}$. This is also verified by an empirical study where n, i.e., the dimension $d = 2^n$ is varied along with the no. of bound vector pairs ρ and the amount of absolute mean noise in retrieval is estimated.

Figure 2 shows the heatmap visualization of the noise for both η_i° and η_i^{π} in natural log scale. The amount of noise accumulated without any projection to the inputs is much higher compared to the noise accumulation with the projection. For varying n and ρ, the maximum amount of noise accumulated when projection is applied is 7.18 and without any projection, the maximum amount of noise is 19.38. Also, most of the heatmap of η_i^{π} remains in the blue region whereas as n and ρ increase, the heatmap of η_i° moves towards the red region. Therefore, it is evident that the projection to the inputs diminishes the amount of accumulated noise with the retrieved output.

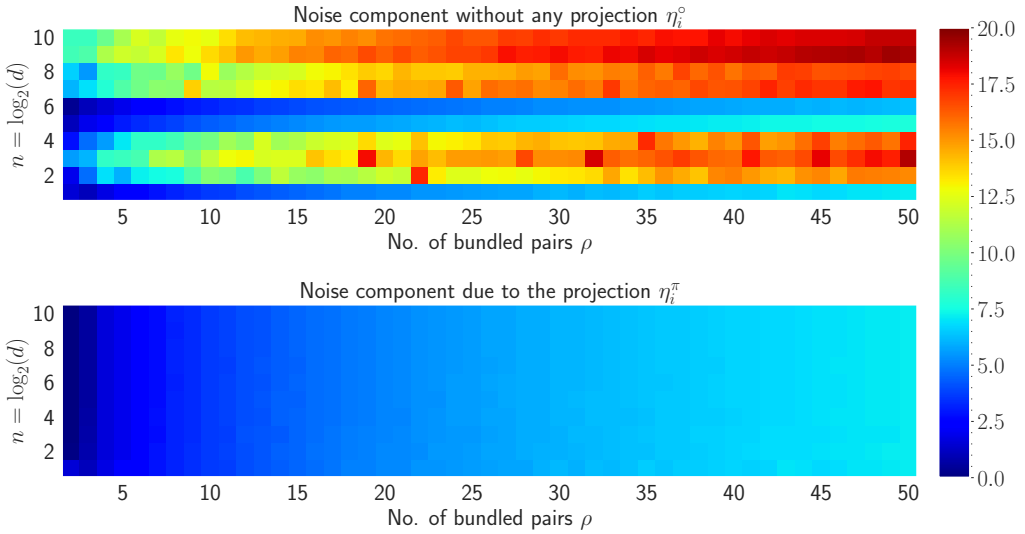

Figure 2: Heatmap of the empirical comparison of the noise components η_i° and η_i^{π} for varying n and ρ shown in natural logarithm scale. The dimension, i.e., $d = 2^n$ is varied from 2 to 1024 ($n \in \{1, 2, \cdots, 10\}$) and the number of vector pairs bundled is varied from 2 to 50.

D Norm Relation

Theorem D.1 (χ_ρ – ρ Relationship)**.** *Given $x_i, y_i \sim \Omega(\mu, 1/d) \in \mathbb{R}^d \; \forall \; i \in \mathbb{N} : 1 \leqslant i \leqslant \rho$, the norm of the composite representation χ_ρ is proportional to $\sqrt{\rho}$ and approximately equal to the $\mu^2\sqrt{\rho \cdot d}$.*

Proof of Theorem D.1. Given χ_ρ is the composite representation of the bound vectors, i.e., the summation of ρ no. of individual bound terms. First, let's compute the norm of the single bound term as shown in Equation 14.

$$
\begin{aligned}
\|\mathcal{B}(x_i, y_i)\|_2 &= \|x_i \cdot y_i\|_2 \\
&= \sqrt{(x_i^{(1)}y_i^{(1)})^2 + (x_i^{(2)}y_i^{(2)})^2 + \cdots + (x_i^{(d)}y_i^{(d)})^2} \\
&= \sqrt{(\pm\mu^2)^2 + (\pm\mu^2)^2 + \cdots + (\pm\mu^2)^2} \quad \left[E[x^{(1)}] \cdot E[y^{(1)}] = \pm\mu \cdot \pm\mu = \pm\mu^2\right] \\
&= \sqrt{\mu^4 d}
\end{aligned}
\tag{14}
$$

Now, let's expand and compute the square norm of the composite representation given in Equation 15.

$$
\begin{aligned}
\left\|\chi_\rho\right\|_2^2 &= \left\|\mathcal{B}(x_1, y_1) + \mathcal{B}(x_2, y_2) + \cdots + \mathcal{B}(x_\rho, y_\rho)\right\|_2^2 \\
&= \left\|\mathcal{B}(x_1, y_1)\right\|_2^2 + \left\|\mathcal{B}(x_2, y_2)\right\|_2^2 + \cdots + \left\|\mathcal{B}(x_\rho, y_\rho)\right\|_2^2 \; + \; \xi \\
&\quad \text{where } \xi \text{ is the rest of the terms of square expansion.} \\
&= \mu^4 d + \mu^4 d + \cdots + \mu^4 d + \xi \\
&= \rho \cdot \mu^4 d + \xi \\
\left\|\chi_\rho\right\|_2 &= \sqrt{\rho \cdot \mu^4 d + \xi} \\
&\approx \sqrt{\rho \cdot \mu^4 d} \quad [\, \xi \text{ is the noise term and discarded to make an approximation }] \\
&= \mu^2 \sqrt{\rho \cdot d} \quad \square
\end{aligned}
\tag{15}
$$

Thus, given the composite representation and the mean of the MiND distribution, we can estimate the no. of bound terms bundled together by $\rho \approx \left\|\chi_\rho\right\|_2^2 / \mu^4 d$. $\square$

Figure 3 shows the comparison between the theoretical relationship and actual experimental results where the norm of the composite representation is computed for $\mu = 0.5$ and $\rho = \{1, 2, \cdots, 200\}$. The figure indicates that the theoretical relationship aligns with the experimental results. However, as the no. of bundled pair increases, the variation in the norm increases. This is because of making the approximation by discarding ξ in Equation 15.

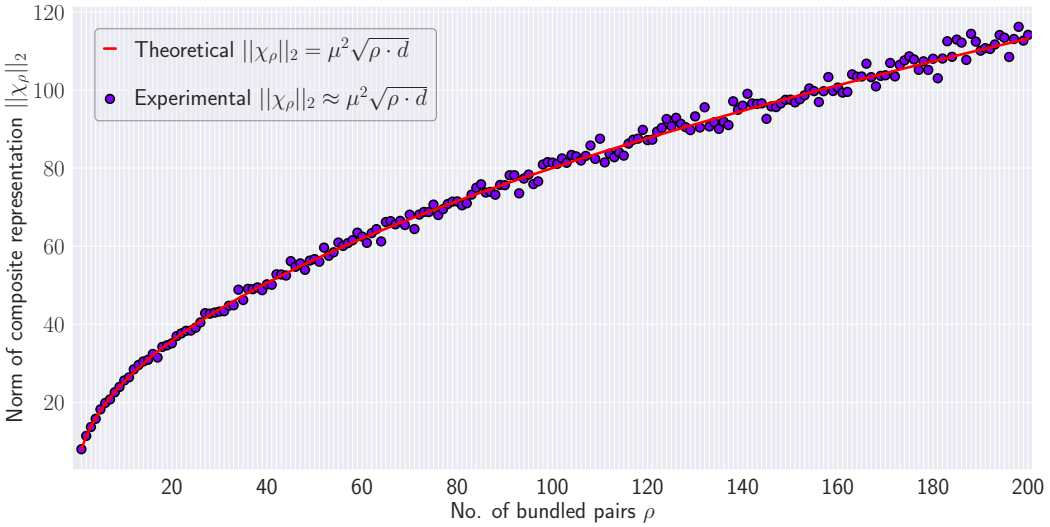

Figure 3: Comparison between the theoretical and experimental relationship of Theorem D.1. The norm of the composite representation of the bound vectors is computed for no. of bundled vectors from 1 to 200 of dimension $d = 1024$. The figure shows how the experimental value of the norm closely follows the theoretical relation between $\left\|\chi_\rho\right\|_2$ and ρ.

E Cosine Relation

Theorem 2.2 shows how the cosine similarity ϕ between the original x_i and retrieved vector $\hat{x}_i$ is approximately equal to the inverse square root of the number of vector pairs in a composite representation ρ. In this section, we will perform an empirical analysis of the theorem and compare it with the theoretical results. For $\rho = \{1, 2, \cdots, 50\}$, similarity score ϕ is calculated for vector dimension $d = 512$. Additionally, the theoretical cosine similarity score is also calculated using the value of ϕ following the theorem. Figure 4 shows the comparison between the two results where the experimental result closely follows the theoretical result. The figure also shows the standard deviation for 100 trials indicating a minute change from the actual value.

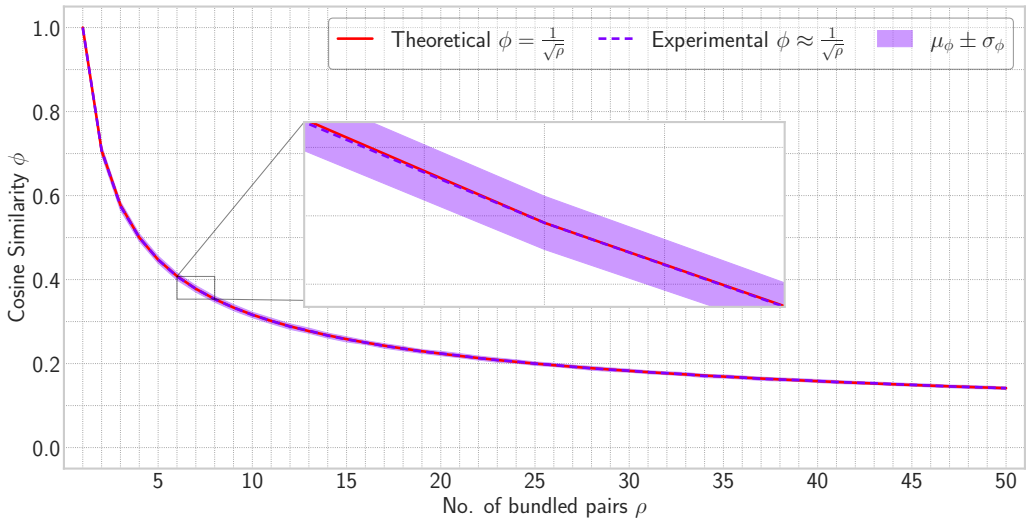

Figure 4: Comparison between the theoretical and experimental $\phi - \rho$ relationship. Vectors of dimension $d = 512$ are combined and retrieved with a varied number of vectors from 1 to 50. The zoom portion shows how closely experimental results match with the theoretical conclusion.

F Classical VSA Results

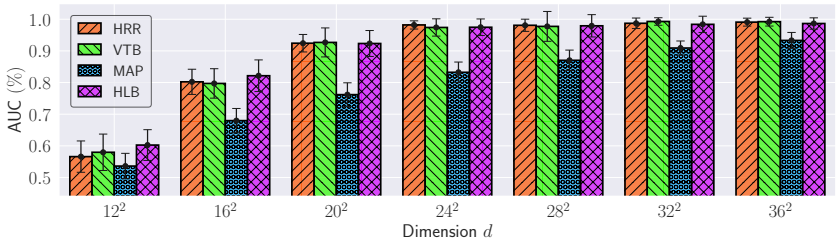

Figure 5: The area under the accuracy curve due to the change of no. of bundled pairs ρ for dimensions d. All the dimensions are chosen to be perfect squares due to the constraint of VTB.

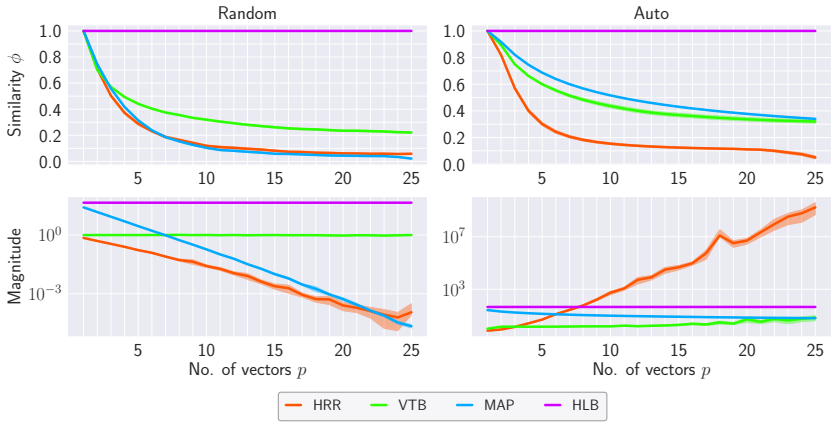

Figure 6: When repeatedly binding different random (left) or a single vector (right), HLB consistently returns the ideal similarity score of 1 for a present item (top row) and has a constant magnitude (bottom row), avoiding exploding/vanishing values.

G CSPS Details and Results

Deploying deep learning networks on a remote third-party compute environment is quite common for training and inference purposes due to the need of large computing demand. However, if the remote third party is not trusted, a user may want to obfuscate the data and the model to prevent theft. Since provable Homomorphic Encryption is too expensive [12], Connectionist Symbolic Pseudo Secrets (CSPS) [3] was proposed a fast, heuristic, pseudo-encryption. It mimics a "one-time-pad" by taking a random VSA vector $\boldsymbol{s}$ as the *secret* and binding it to the input $\boldsymbol{x}$. The value $\mathcal{B}(\boldsymbol{s}, \boldsymbol{x})$ obscures the original $\boldsymbol{x}$, and the third-party runs the bulk of the network on their platform. A result $\tilde{\boldsymbol{y}}$ is returned, and a small local network computes the final answer after unbinding with the secret $\mathcal{B}^*(\tilde{\boldsymbol{y}}, \boldsymbol{s})$. Other than changing the VSA used, we follow the same training, testing, architecture size, and validation procedure of [3].

Table 4: Clustering results of the main network inputs (top rows) and outputs (bottom rows) in terms of Adjusted Rand Index (ARI). Because CSPS is trying to hide information, scores near zero are better. Cell color corresponds to the cell absolute value, with blue indicating lower ARI and red indicating higher ARI. All numbers in percentages, and show HLB is better at information hiding.

Clustering Methods	HRR					VTB				
	MNIST	SVHN	CR10	CR100	MIN	MNIST	SVHN	CR10	CR100	MIN
K-Means	−0.02	−0.01	0.18	0.54	0.42	−0.00	−0.01	−0.01	0.02	0.00
GMM	0.01	0.00	0.09	0.61	0.44	4.67	1.37	−0.01	0.02	0.01
Birch	0.20	0.00	0.14	0.45	0.35	0.02	0.03	0.04	0.08	0.03
HDBSCAN	0.00	−0.24	1.23	0.01	0.02	0.00	0.00	0.00	0.00	0.00
K-Means	1.28	0.06	0.21	0.03	0.08	8.52	0.13	1.11	0.05	0.12
GMM	1.28	0.06	0.17	0.04	0.09	8.63	0.14	1.63	0.05	0.00
Birch	1.51	0.03	0.13	0.05	0.07	3.24	0.00	0.64	0.06	0.17
HDBSCAN	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.00
Clustering Methods	**MAP**					**HLB**				
	MNIST	SVHN	CR10	CR100	MIN	MNIST	SVHN	CR10	CR100	MIN
K-Means	0.17	0.01	0.01	0.00	0.00	0.09	0.00	0.00	0.00	0.00
GMM	3.39	−0.01	0.01	0.00	0.00	2.53	0.00	0.00	0.00	0.00
Birch	0.84	−0.00	0.00	0.01	0.00	0.83	0.00	0.00	0.01	0.00
HDBSCAN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
K-Means	15.91	0.09	0.00	0.03	0.01	13.67	−0.04	0.01	0.02	−0.00
GMM	42.43	0.11	0.00	0.03	0.00	14.96	−0.04	0.01	0.02	0.00
Birch	7.09	−0.07	−0.02	0.01	−0.00	18.44	−0.07	0.00	0.01	0.02
HDBSCAN	0.48	0.00	0.00	0.00	0.00	7.60	0.01	0.00	0.00	0.00

H XML Details

Extreme Multi Label (XML) classification is possible when only a small subset of classes exist for a given input $\boldsymbol{x}$. The VSA trick used by [10] was to define an additional "present" class $\boldsymbol{p}$ and a "missing" class $\boldsymbol{m}$. Then the target output of the network $f(\cdot)$ is itself a vector composed of two parts added together. First $\mathcal{B}(\boldsymbol{p}, \sum_k \boldsymbol{c}_k)$ represents all *present* classes, and so the sum is over a finite smaller set. Then the absent classes compute the *missing* representing $\mathcal{B}(\boldsymbol{m}, \boldsymbol{a} - \sum_k \boldsymbol{c}_k)$, which again only needs to compute over the finite set of present classes, yet represents the set of all non-present classes by exploiting the symbolic properties of the VSA.

For XML classification, we have a set of K classes that will be present for a given input, where $K \approx 10$ is the norm. Yet, there will be L total possible classes where $L \geqslant 100,000$ is quite common. Forming a normal linear layer to produce L outputs is the majority of computational work and memory use in standard XML models, and thus the target for reduction. A VSA can be used to side-step this cost, as shown by [10], by leveraging the symbolic manipulation of the outputs. First, consider the target label as a vector $\boldsymbol{s} \in \mathrm{R}^d$ such that $d \ll L$. By defining a VSA vector to represent "present" and "missing" classes as $\mathbf{p}$ and $\mathbf{m}$, where each class is given its own vector $\boldsymbol{c}_{1,\ldots,L}$, we can shift the computational complexity form $\mathcal{O}(L)$ to $\mathcal{O}(K)$ by manipulating the "missing" classes as

the compliment of the present classes as shown in Equation 16.

$$\boldsymbol{s} = \overbrace{\sum_{i \in y_i = 1} \mathcal{B}(\boldsymbol{p}, \boldsymbol{c}_i)}^{\text{Labels Present} \mathcal{O}(dK)} + \overbrace{\sum_{j \in y_j = -1} \mathcal{B}(\boldsymbol{m}, \boldsymbol{c}_j)}^{\text{Labels Absent} \mathcal{O}(dL)} = \overbrace{\mathcal{B}\left(\boldsymbol{p}, \left(\boldsymbol{a} =: \sum_{i \in y_i = 1} c_i\right)\right)}^{\text{Labels Present} \mathcal{O}(d\ K)} + \overbrace{\mathcal{B}\left(\boldsymbol{m}, \left(\boldsymbol{a} - \sum_{i \in y_i = 1} c_i\right)\right)}^{\text{Labels Absent} \mathcal{O}(dK)} \tag{16}$$

Similarly, the loss to calculate the gradient can be computed based on the network's prediction $\hat{\boldsymbol{s}}$ by taking the cosine similarity between each expected class and one cosine similarity for the representation of all missing classes. The excepted response of 1 or 0 for an item being present/absent from the VSA is used to determine if we want the similarity to be 0 (1-cos) or 1 (just cos), as shown in Equation 17.

$$loss = \overbrace{\sum_{i \in y_i = 1} (1 - \cos\left(\mathcal{B}^*(\boldsymbol{p}, \hat{\boldsymbol{s}}), \boldsymbol{c}_i\right))}^{\text{Present Classes } \mathcal{O}(d\ K)} + \overbrace{\cos\left(\mathcal{B}^*(\boldsymbol{m}, \hat{s}), \sum_{i \in y_i = 1} \boldsymbol{c}_i\right)}^{\text{Absent classes } O(d\ K)} \tag{17}$$