G-Net: A Provably Easy Construction of High-Accuracy Random Binary Neural Networks

Alireza Aghasi*
Dept. Electrical Engineering and Computer Science
Oregon State University
alireza.aghasi@oregonstate.edu

Nicholas F. Marshall
Dept. Mathematics
Oregon State University
marsnich@oregonstate.edu

Saeid Pourmand
Dept. Electrical Engineering and Computer Science
Oregon State University
pourmans@oregonstate.edu

Wyatt D Whiting
Dept. Mathematics
Oregon State University
whitinwy@oregonstate.edu

Abstract

We propose a novel randomized algorithm for constructing binary neural networks with tunable accuracy. This approach is motivated by hyperdimensional computing (HDC), which is a brain-inspired paradigm that leverages high-dimensional vector representations, offering efficient hardware implementation and robustness to model corruptions. Unlike traditional low-precision methods that use quantization, we consider binary embeddings of data as points in the hypercube equipped with the Hamming distance. We propose a novel family of floating-point neural networks, G-Nets, which are general enough to mimic standard network layers. Each floating-point G-Net has a randomized binary embedding, an embedded hyperdimensional (EHD) G-Net, that retains the accuracy of its floating-point counterparts, with theoretical guarantees, due to the concentration of measure. Empirically, our binary models match convolutional neural network accuracies and outperform prior HDC models by large margins, for example, we achieve almost 30% higher accuracy on CIFAR-10 compared to prior HDC models. G-Nets are a theoretically justified bridge between neural networks and randomized binary neural networks, opening a new direction for constructing robust binary/quantized deep learning models. Our implementation is available at https://github.com/GNet2025/GNet.

1 Introduction

We are motivated by hyperdimensional computing (HDC), a brain-inspired computational paradigm that uses high-dimensional vectors as the basic elements of computation [33]. Applications of HDC include classification and regression [7, 10, 17, 35, 39], reinforcement learning [4, 25], and dimensionality reduction [14, 24]. Recent works have also explored online learning approaches [10, 12] and advanced the theoretical understanding of HDC [26, 31, 37]. HDC methods can be efficiently implemented on hardware, making these models attractive for edge and low-energy computing [3, 5, 13, 16, 18, 19]. For readers less familiar with HDC, Section A of the Appendix offers a detailed taxonomy and broader overview of HDC methods.

For predictive tasks, an HDC pipeline typically consists of two core components: a hyperdimensional embedding module, which often carries some component of randomness, and an inference module [32]. The embedding transforms input data into high-dimensional representations—often binary

*Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

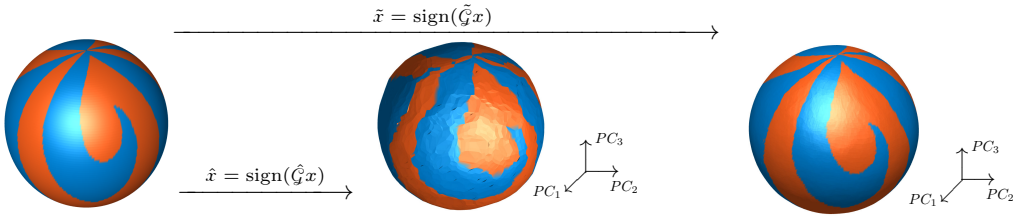
hypervectors—that retain key structural features while simplifying the data space. The inference module, commonly a lightweight classifier, operates on these embeddings and often resembles a support vector machine in decision making [20, 19]. These components are generally decoupled, and due to limited representational capacity, lack of task-specific learning—especially in the embedding stage—and the simplicity of the overall architecture, HDC models struggle to match the performance of state-of-the-art predictors. Moreover, the high-dimensional and highly structured (e.g., binary) nature of the embedding space hinders the training of more sophisticated models in that domain.

In this paper, we introduce a new class of (random) binary neural networks that operate in the hyperdimensional Hamming space without requiring training within that space. Typical approaches to training binary neural networks include using a straight-through estimator (STE) [6, 28], a continuous piecewise-defined approximation [21], a thermometer encoder [38], or quantization [9, 11] (see Section B of the Appendix for a more detailed overview of binary neural networks). In contrast, our approach leverages the geometry of the hypercube and random matrix theory. We propose a class of floating-point networks, *G-Nets*, that can be converted into randomized binary networks with accuracy guarantees derived from the concentration of measure. Central to our method is the random binary encoding $\phi : \mathbb{R}^p \to \{-1, 1\}^N$, defined as $\phi(x) = \texttt{sign}(\mathcal{G}x)$, where $\mathcal{G} \in \mathbb{R}^{N\times p}$ has i.i.d. standard normal entries—a common HDC encoding scheme [27, 13]. For large N, Grothendieck's identity (see Lemma 3.1) ensures that $\frac{1}{N}\langle\phi(x), \phi(y)\rangle \approx \frac{2}{\pi}\arcsin(\langle x, y\rangle)$ when $x, y \in \mathbb{S}^{p-1}$. Thus, inner products in $\mathbb{S}^{p-1}$ can be approximated by those in the Hamming space, up to a known nonlinearity. By absorbing this nonlinearity into the activation functions, we construct a new class of multi-layer neural networks that can be trained on source real-valued data and directly mapped along with the data to the binary hyperdimensional space, maintaining a comparable predictive performance.

Empirically, our approach circumvents the need to train expressive models directly in the binary domain, which is both high-dimensional ($N \gg p$) and structurally constrained—posing significant challenges for fitting predictive models. Theoretically, the proposed framework for constructing high-performing binary neural networks addresses key open questions in HDC research. First, we address questions about how far HDC models can be pushed to achieve accuracies on par with neural networks (asymptotically, the accuracies of the proposed hyperdimensional models converge to that of a G-Net as N increases). Second, we provide non-asymptotic results on how large the hyperdimension N needs to be for a desirably close performance to that of a G-Net. At a more abstract level, the proposed framework can be interpreted as an inexpensive access to a *meta-distribution* over binary neural networks tailored to a prediction task, where G-Net controls its mean accuracy and the hyperdimension controls the level of concentration. Sampling from this distribution without training may benefit applications such as privacy, robustness, and ensemble learning. Moreover, individual binary networks drawn from this distribution can be further refined through fine-tuning techniques.

The remainder of the paper is organized as follows. Section 2 introduces the bundle embedding problem, which concerns embedding both data and predictive models coherently, with minimal loss in performance within the embedded space. Section 3 presents the core G-Net framework as a solution to this problem and outlines architectural variations designed for varying levels of hardware compatibility. In Section 4, we provide a detailed consistency analysis between the performance of a G-Net and its hyperdimensional counterpart (EHD G-Net), including non-asymptotic, layer-wise, and network-level results showing how the choice of hyperdimension governs the discrepancy between the two. Section 5 demonstrates, under realistic conditions, how the encoding can be significantly simplified by replacing Gaussian matrices with Rademacher matrices. Finally, Section 6 presents numerical experiments that showcase the strong performance of the proposed method, followed by discussion and concluding remarks. Proofs of all theoretical results, along with additional experiments and implementation details, are provided in the Appendix.

Notations. Let $\mathbb{S}^{n-1}$ and $\mathbb{B}_2^n$ denote the unit sphere and unit Euclidean ball in $\mathbb{R}^n$, respectively. We use lowercase letters for vectors and uppercase letters for matrices. For two binary vectors $x, y \in \{-1, 1\}^n$, $\mathcal{D}_H(x, y)$ represents the normalized Hamming distance: $\mathcal{D}_H(x, y) = n^{-1}|\{i : x_i \neq y_i\}|$. For two vectors $x, y \in \mathbb{S}^{n-1}$, $\mathcal{D}_G(x, y)$ represents the normalized geodesic distance between the two vectors obtained by dividing the smaller angle between the two vectors by π. A random vector/matrix is called Gaussian (or Rademacher) if the elements are independently drawn from a standard normal distribution (or a Rademacher distribution taking values 1 and -1 with equal probability $1/2$). We use script font for random quantities (e.g., g, $\mathcal{G}$). We use teletype font for functions that operate



$x \in \mathcal{X} \subseteq \mathbb{S}^2$
$p_\theta(y \mid x)$

$\mathcal{P}_3(\hat{x}),\ \hat{x} \in \hat{\mathcal{X}} \subseteq \{-1,1\}^{100}$
$\hat{p}_\theta(y \mid \hat{x}) \approx p_\theta(y \mid x)$

$\mathcal{P}_3(\tilde{x}),\ \tilde{x} \in \tilde{\mathcal{X}} \subseteq \{-1,1\}^{1000}$
$\tilde{p}_\theta(y \mid \tilde{x}) \approx p_\theta(y \mid x)$

Figure 1: A graphical demonstration of data embedding, and bundle embedding

on matrices/vectors in an element-wise fashion, e.g., $\mathtt{sign}(x)$ is a vector with $\mathrm{sign}(x_i)$ as its i-th element.

2 Bundle Embedding Problem

Quantization simplifies data representation and computation to better suit hardware constraints. Its effectiveness is maximized when the geometric structure of the data is preserved, motivating the use of (nearly) distance-preserving mappings, often referred to as (near) isometries.

Definition 2.1. *Suppose that $\varepsilon > 0$, and let $\mathcal{X}$ and $\mathcal{Y}$ be metric spaces equipped with distances $\mathcal{D}_\mathcal{X}$ and $\mathcal{D}_\mathcal{Y}$, respectively. A mapping $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ is called an ε-near-isometry if for all $x_1, x_2 \in \mathcal{X}$: $|\mathcal{D}_\mathcal{Y}(\mathcal{A}(x_1), \mathcal{A}(x_2)) - \mathcal{D}_\mathcal{X}(x_1, x_2)| \leq \varepsilon$.*

A prominent example of such mappings is the *random binary embedding*, where real-valued vectors are converted to high-dimensional binary codes via a random linear projection followed by a quantization operator (such as a sign function). Typically, such embeddings takes the form $\mathcal{A}(x) = \mathtt{sign}(\mathcal{G}x)$, where $\mathcal{G}$ is a Gaussian matrix. It is well-understood that by increasing the height of $\mathcal{G}$, the mapping can be made arbitrarily close to an isometry, as stated below.

Proposition 2.1. *For a Gaussian matrix $\mathcal{G} \in \mathbb{R}^{N\times n}$, consider the mapping $\mathcal{A}(x) = \mathtt{sign}(\mathcal{G}x)$ from $\mathcal{X} = \mathbb{S}^{n-1}$ (equipped with a geodesic distance $\mathcal{D}_G$) to $\mathcal{Y} = \{-1,1\}^N$ (equipped with a normalized Hamming distance $\mathcal{D}_H$). Fix $\varepsilon \in (0,1)$, then there exist absolute constants c_1 and c_2 such that if $N \geq c_1 n/\varepsilon^2$, with probability exceeding $1 - \exp(-c_2 n)$, $\mathcal{A}$ is an ε-near-isometry. That is*

$$\forall x, y \in \mathbb{S}^{n-1}: \qquad \left|\mathcal{D}_H\left(\mathtt{sign}(\mathcal{G}x), \mathtt{sign}(\mathcal{G}y)\right) - \mathcal{D}_G(x,y)\right| \leq \varepsilon. \tag{1}$$

Throughout this paper, we refer to $\mathcal{X}$ as the *primal* space and $\mathcal{Y}$ as the *embedded* space. For sufficiently large N, Proposition 2.1 guarantees, with high probability, an ε-consistency between the pairwise distances measured in the primal space $\mathcal{X} = \mathbb{S}^{n-1}$ and the embedded space $\mathcal{Y} = \{-1,1\}^N$. To be able to use a similar metric on both spaces, for the same setting as Proposition 2.1, one may consider the normalized mapping $\mathcal{A}(x) = N^{-1/2}\mathtt{sign}(\mathcal{G}x)$ which will map unit vectors in $\mathbb{S}^{n-1}$ to unit vectors in $\mathbb{S}^{N-1}$. As detailed in the proof discussion of Proposition 2.1, in this case, the consistency result (1) can be reformulated as

$$\forall x, y \in \mathbb{S}^{n-1}: \qquad \left|\sin^2\left(\frac{\pi}{2}\mathcal{D}_G\left(N^{-1/2}\mathtt{sign}\left(\mathcal{G}x\right), N^{-1/2}\mathtt{sign}\left(\mathcal{G}y\right)\right)\right) - \mathcal{D}_G(x,y)\right| \leq \varepsilon, \tag{2}$$

where a geodesic metric is used on both spaces. Equation (2) reveals that no matter how small ε is, there is an intrinsic $\sin^2(\frac{\pi}{2}\cdot)$ deformation of the geodesic distance when applying $\mathcal{A}$ to the source data. To visually illustrate the effect of this embedding, Figure 1 shows the result of applying $\mathcal{A}$ to a dense set of points on $\mathbb{S}^2$, using $N = 100$ and $N = 1000$. The binary codes produced in $\{-1,1\}^N$ are then visualized in $\mathbb{R}^3$ via their first three principal components to illustrate how the representation improves as N increases. Although the spherical geometry is nearly intact for $N = 1000$, the deformation is noticeable when comparing the color labels—the coloring pattern on the right deviates from that of the original on the left.

In this paper, the *bundle embedding problem* broadly refers to the problem of coherently embedding both data and a predictive model defined on it, such that the model remains functional in the embedded space without requiring retraining. As a motivating example, suppose that a model $p_\theta(y \mid x)$ is trained on the source data $\mathcal{X}$ illustrated in Figure 1. After binary embedding, such a model is unusable due

to two main issues: first, the data format has changed from continuous to binary, and, second, there is a nonlinear distortion introduced by the embedding, which standard predictors are not equipped to correct easily. On the other hand, training a high-accuracy classifier in the embedded space that operates in binary mode is itself difficult, owing to both the discreteness of the problem and the often-large dimension N. In this case our goal is to construct predictive models that can be embedded along with the data to operate in binary mode while maintaining accuracy.

To this end, we introduce a flexible class of neural networks, whose performance remains close after the data are quantized via a random sign embedding, ensuring consistent accuracies in both the primal and embedded domains. Rewiring these models to their binary neural network counterparts is as simple as going through a random sign embedding of the primal model weights. In a hyperdimensional-computing pipeline, this approach supplies high-accuracy predictors that operate directly on the binary-represented data, obviating any training in the embedded domain.

3 G-Net Main Idea

At a high level, many standard feed-forward neural networks can be viewed as a cascade of L layers given by $y_\ell = \sigma_\ell(W_\ell y_{\ell-1})$, where W_ℓ are learnable weight matrices, y_ℓ is the output of the ℓ-th layer, and σ_ℓ is a nonlinear, typically element-wise, activation function. In this section, we discuss the bundle embedding of a class of neural networks originally operating on real-valued data, demonstrating how they can be systematically transformed to operate on binary data while maintaining accuracy controllably close to the original real-valued model. The construction is grounded in Grothendieck's identity (cf. [34, page 63]); hence, we refer to the proposed architectures as *G-Nets*.

Lemma 3.1 (Grothendieck's Identity)**.** *Let g be a standard Gaussian vector in $\mathbb{R}^p$. Then, for fixed vectors $u, v \in \mathbb{R}^p$: $\mathbb{E}\left[\operatorname{sign}(g^\top u)\operatorname{sign}(g^\top v)\right] = \frac{2}{\pi}\arcsin\left(\frac{u^\top v}{\|u\|_2\|v\|_2}\right)$.*

If $\mathcal{G} \in \mathbb{R}^{N\times p}$ is a standard Gaussian matrix with rows $g_i^\top$, then by the law of large numbers

$$u, v \in \mathbb{S}^{p-1}: \quad \frac{1}{N}\left\langle \texttt{sign}(\mathcal{G}u), \texttt{sign}(\mathcal{G}v)\right\rangle = \frac{1}{N}\sum_{i=1}^{N}\operatorname{sign}(g_i^\top u)\operatorname{sign}(g_i^\top v) \approx \frac{2}{\pi}\arcsin(\langle u, v\rangle).$$

This observation offers a new perspective on Grothendieck's Identity: through the random sign embedding, inner products in $\mathbb{S}^{p-1}$ are approximately mapped to inner products in $\{-1,1\}^N$ up to some scaling and sine distortion. More generally, when $W \in \mathbb{R}^{n\times p}$ has normalized rows (that is, $w_i^\top \in \mathbb{S}^{p-1}$, $i \in \{1,\dots,n\}$), and $y \in \mathbb{S}^{p-1}$, then $\frac{1}{N}\texttt{sign}\left(W\mathcal{G}^\top\right)\texttt{sign}\left(\mathcal{G}y\right) \approx \frac{2}{\pi}\texttt{arcsin}(Wy)$, where $\texttt{sign}$ and $\texttt{arcsin}$ functions act element-wise. Applying a suitable activation τ (later to be specified) can preserve this approximation, and for a general input $y \in \mathbb{R}^p$ deliver a neural network layer of the form

$$\sigma\left(W\frac{y}{\|y\|_2}\right) \approx \tau\left(\texttt{sign}\left(W\mathcal{G}^\top\right)\texttt{sign}\left(\mathcal{G}y\right)\right), \tag{3}$$

where $\sigma = \tau \circ \frac{2N}{\pi}\texttt{arcsin}$. Observe that the left-hand side of (3) operates on inner products of real-valued vectors, while the main right-hand side operation is inner products between their binary embeddings. The activations differ between the two sides, since σ further carries the nonlinear distortion introduced by Grothendieck's identity. If the discrepancy between the two sides of (3) is controllably small (specifically, by picking a large N), then cascading multiple layers—embedded via independent Gaussian matrices $\mathcal{G}_\ell$—forms a neural network operating in the embedded regime, with performance that closely matches its primal counterpart. The above construction and conversion can be extended to L layers as follows.

Definition 3.1 (G-Net)**.** *Let $n_0, \dots, n_L$ be a sequence of positive integers, and τ be an activation function. Suppose that W_ℓ is an $n_\ell \times n_{\ell-1}$ matrix whose rows are ℓ_2-normalized for $\ell = 1,\dots,L$, and let an input vector $y_0 = x \in \mathbb{R}^{n_0}$ be given. We define the output y_L of an L-layer G-Net iteratively by $y_\ell = \sigma\left(W_\ell \frac{y_{\ell-1}}{\|y_{\ell-1}\|_2}\right)$, for $\ell = 1,\dots,L$, where $\sigma = \tau \circ \frac{2}{\pi}\arcsin$.*

Definition 3.2 (EHD G-Net: Embedded Hyperdimensional G-Net)**.** *Suppose a trained G-Net is given as in Definition 3.1. Fix a sequence of hyperdimensions $N_1, \dots, N_L$, and let $\mathcal{G}_\ell$ be independent Gaussian matrices of size $N_\ell \times n_{\ell-1}$. We define the output $\tilde{y}_L$ of the L-layer Hyperdimensional embedding of the given G-Net iteratively by $\tilde{y}_\ell = \tau(\operatorname{sign}(W_\ell G_\ell^\top)\operatorname{sign}(G_\ell \tilde{y}_{\ell-1}))$, for $\ell = 1,\dots,L$.*

We propose three element-wise choices for τ. The simplest choice is the identity map, $\tau(x) = \texttt{Id}(x)$, primarily used for analysis purposes, which corresponds to the arc-sine activation unit $\sigma(x) =$

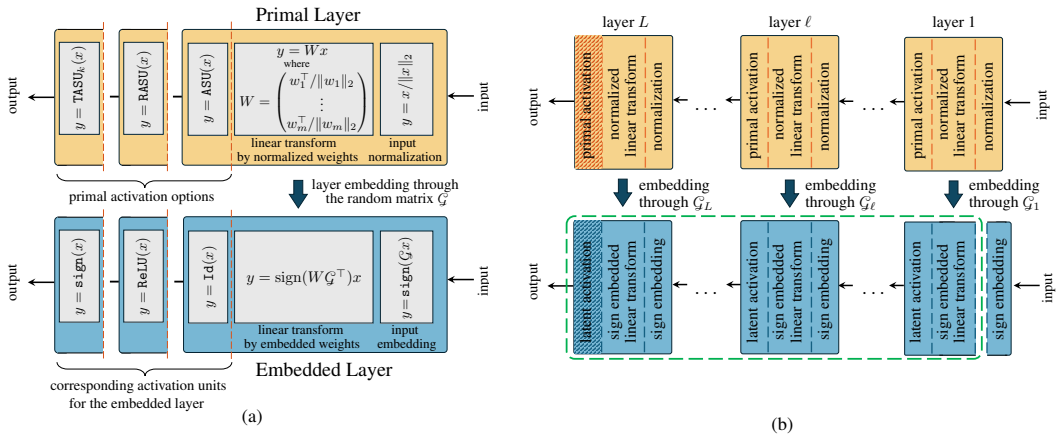


Figure 2: (a) A G-Net layer (top), and its corresponding EHD G-Net layer (bottom); the input to each block is denoted by x and the output is represented by y; (b) A G-Net and corresponding EHD G-Net constructed by cascading the proposed layer blocks

$\texttt{ASU}(x) \triangleq \frac{2}{\pi}\texttt{arcsin}(x)$ in the G-Net architecture. Additionally, we use $\tau(x) = \texttt{ReLU}(x)$, and $\tau(x) = \texttt{sign}(x)$, which lead respectively to the following G-Net activations σ:

$$\texttt{RASU}(x) \triangleq \begin{cases} \frac{2}{\pi}\arcsin(x), & x \geq 0, \\ 0, & x < 0, \end{cases} \quad \text{and} \quad \texttt{TASU}_\kappa(x) \triangleq \tanh\left(\frac{2\kappa}{\pi}\arcsin(x)\right).$$

In the last activation, we employed a smooth approximation of the `sign` function, specifically $\tau(x) = \tanh(\kappa x)$ for sufficiently large κ, to ensure that the associated G-Net activation remains smooth. The top panel of Figure 2(a) presents the architecture of a G-Net layer, and the bottom panel depicts the corresponding EHD G-Net layer. An L-layer G-Net and correspondingly a similar-depth EHD G-Net can be constructed through the composition of the proposed layers. Note that in Definition 3.2 and Figure 2, we omitted the normalization factors $1/N_\ell$. In intermediate layers, this factor can be absorbed by the subsequent `sign` operation. For the final layer, neglecting this normalization merely affects the output's scale, which is typically inconsequential.

The core bundle embedding process involves first training a G-Net on the original real-valued data to learn the weights W_ℓ, and subsequently converting it into an EHD G-Net by selecting sufficiently tall Gaussian matrices $\mathcal{G}_\ell$. The resulting EHD G-Net is a sequential neural network featuring binary weights $\texttt{sign}(W_\ell \mathcal{G}_\ell^\top)$ and appropriately adjusted activations, which are acquired *without training on the embedded binary data*. Once the EHD G-Net is constructed, the first sign embedding block of the first layer (i.e., $\texttt{sign}(\mathcal{G}_1 x)$) can be interpreted as the random sign embedding of the source data, and the rest of the pipeline (shown as the dashed box in the bottom panel of Figure 2(b)) can be viewed as an embedding of the primal inference module. Since HDC models typically employ a simple inference stage (see Section A of the Appendix), the framework can equivalently be viewed as an HDC pipeline with multi-stage embedding: the first $L-1$ layers constitute the embedding stack, and the final classification layer serves as the inference block. Under either interpretation, unlike many current HDC frameworks, the designs of our embedding and inference modules are tightly coupled by construction.

The primary distinctions between a G-Net layer (top panel of Figure 2(a)) and a standard feed-forward network layer are the ℓ_2-normalization applied at the input, and the enforced row-wise normalization on the learnable weight matrices W. While these constraints may initially seem to limit the flexibility and expressiveness of G-Nets, they have been shown to accelerate training and enhance the generalization accuracy [30, 22, 36], making them advantageous design choices rather than limitations. Furthermore, the choice of activation function—being the sole distinction between RASU and TASU G-Nets—directly influences the operating mode of the resulting EHD G-Nets. In RASU networks, each layer output (i.e., the right-hand side of (3)) is a non-negative integer vector, whereas in TASU layers, the output is binary, allowing their EHD G-Nets to operate entirely in binary mode.

The rest of the paper delves more deeply into a theoretical understanding of the proposed framework. We first examine how the hyperdimension N controls the approximation error between a G-Net

layer and its EHD counterpart, and how these errors accumulate when multiple layers are cascaded. Although Grothendieck's identity is naturally linked to Gaussian distributions, Section 5 shows that much simpler Rademacher embeddings offer comparable guarantees.

From a practical standpoint, the framework extends well beyond fully connected layers: convolution, batch-normalization, pooling, and similar modules can all be incorporated into a G-Net and subsequently embedded into a hyperdimensional model. In addition to addressing such details in Section N of the Appendix, we will also discuss details such as handling the bias term for all such layers, and the activation chosen for the last layer, which determines whether the resulting network behaves as a classifier or a regressor.

4 Consistency of G-Net and EHD G-Net

This section presents a non-asymptotic analysis of the consistency between a G-Net layer and its hyperdimensional embedding, providing high-probability bounds on the required value of N to achieve a target discrepancy between the primal and embedded layers. First, we analyze the RASU layer, which yields bounds analogous to those of an ASU layer.

Theorem 4.1. *Consider a RASU (or ASU) layer with input $x \in \mathbb{S}^{p-1}$ and a weight matrix $W \in \mathbb{R}^{n\times p}$ whose rows are ℓ_2-normalized. Let the layer output be given by $y = \texttt{RASU}(Wx)$ (or $y = \texttt{ASU}(Wx)$). Define the embedded output as $\tilde{y} = \texttt{ReLU}\left(\texttt{sign}(W\mathcal{G}^\top)\,\texttt{sign}(\mathcal{G}x)\right)$ (or $\tilde{y} = \texttt{Id}\left(\texttt{sign}(W\mathcal{G}^\top)\,\texttt{sign}(\mathcal{G}x)\right)$), where $\mathcal{G} \in \mathbb{R}^{N\times p}$ is a standard Gaussian matrix. Then, for any $c > 0$, with probability at least $1 - \exp(-c)$: $\left\|\frac{1}{N}\tilde{y} - y\right\|_2 \le \sqrt{2N^{-1}(c + \log 2n)n}$.*

Corollary 4.1. *To ensure that the discrepancy between a RASU/ASU layer output and its hyperdimensional embedding satisfies $\left\|\frac{1}{N}\tilde{y} - y\right\|_2 \le \varepsilon$, it is possible to choose the embedding dimension N such that*

$$N = \mathcal{O}\left(\varepsilon^{-2} n \log n\right). \tag{4}$$

A few remarks are in order. First, the $\log n$ term in (4) arises from reusing the same embedding matrix $\mathcal{G}$ across all rows of W, which induces dependencies among components of the discrepancy vector. In the discussion following the proof of Theorem 4.1 in the Appendix, we argue that if independence were enforced—e.g., by using a distinct embedding matrix $\mathcal{G}_i$ for each row w_i and computing $\tilde{y}_i = \texttt{ReLU}\left(\texttt{sign}(w_i^\top \mathcal{G}_i^\top)\texttt{sign}(\mathcal{G}_i x)\right)$—this would remove the $\log n$ factor but at a considerable cost in both computation and memory. Second, in our discrepancy analysis, we compare y to the scaled output $\frac{1}{N}\tilde{y}$ purely for normalization purposes, since $\tilde{y}$ is integer-valued by construction. As stated before, this scaling does not affect downstream computations, as $\tilde{y}$ is passed through another sign-based embedding layer in subsequent processing, rendering the $1/N$ scaling irrelevant in practice. Next, we analyze a TASU layer, where in addition to the standard discrepancy, one also needs to account for the error introduced by approximating the $\mathrm{sign}(\cdot)$ function with $\tanh(\kappa\cdot)$.

Theorem 4.2. *Consider a TASU layer with output $y = \texttt{TASU}_\kappa(Wx)$, where $x \in \mathbb{S}^{p-1}$ and $W \in \mathbb{R}^{n\times p}$ has normalized rows $w_i^\top$ ($\|w_i\|_2 = 1$ for $i = 1, \dots, n$). Define the embedded layer output as $\tilde{y} = \texttt{sign}\left(\texttt{sign}(W\mathcal{G}^\top)\,\texttt{sign}(\mathcal{G}x)\right)$, where $\mathcal{G} \in \mathbb{R}^{N\times p}$ is a standard Gaussian matrix. Assume*

$$\left|w_i^\top x\right| \ge \ell_{\min} > 0, \qquad i = 1, \dots, n. \tag{5}$$

Pick a target discrepancy $\varepsilon \le \sqrt{n}$ and set $\kappa \ge \frac{\pi}{2\ell_{\min}} \log \frac{4\sqrt{n}}{\varepsilon}$. Then, for any scalar $c > 0$, selecting $N \ge 8(c + \log 2n)n\kappa^2/\varepsilon^2$ guarantees that with probability exceeding $1 - 3\exp(-c)$: $\|y - \tilde{y}\|_2 \le \varepsilon$.

We emphasize that assumption (5) is essential for the analysis, to handle the discontinuity of the sign function. Specifically, when $w_i^\top x = 0$ for some row i, the TASU output yields $y_i = 0$, whereas the corresponding embedded output $\tilde{y}_i$ takes a binary ± 1 value as the output of a sign function. Practically, a common approach to handle such zero inputs to a sign quantizer is to assign ± 1 randomly with equal probabilities. In our framework, this behavior emerges naturally from the symmetry of the Gaussian matrix $\mathcal{G}$. Substituting the required κ into the hyperdimension bound, we find that achieving an ε-discrepancy necessitates a hyperdimension of $N = \mathcal{O}(n \log^3 n/(\varepsilon^2 \ell_{\min}^2))$, which is larger compared to that required for a RASU layer. The increased scale of $\log^2 n/\ell_{\min}^2$ reflects the cost of constructing an embedded layer that, unlike the integer-valued outputs of RASU layers, produces binary codes at the layer's output.

4.1 Network Consistency

The preceding layer-wise result motivates us to analyze the overall discrepancy of the EHD G-Net as a cascade of multiple layers. For conventional feed-forward architectures, tracking the error accumulation across the network layers is straightforward (e.g., see [1, 2]). In G-Net, however, the normalization operation complicates this tracking, requiring a more careful analysis. To streamline the discussions, while still covering the key technical steps, we focus on the base ASU G-Net case. Specifically, we consider a cascade of L sequential layers, where

$$y_\ell = \texttt{ASU}\left(W_\ell \frac{y_{\ell-1}}{\|y_{\ell-1}\|_2}\right), \qquad \tilde{y}_\ell = \texttt{Id}\left(\texttt{sign}\left(W_\ell \mathcal{G}_\ell^\top\right)\texttt{sign}\left(\mathcal{G}_\ell \tilde{y}_{\ell-1}\right)\right), \qquad \ell = 1, \dots, L. \quad (6)$$

Here $W_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ are the G-Net weights with normalized rows, and $\mathcal{G}_\ell \in \mathbb{R}^{N_\ell \times n_{\ell-1}}$ are independent Gaussian matrices used for the embedding of each layer. Since G-Net layers operate on normalized inputs, we define $\delta_\ell = \frac{\tilde{y}_\ell}{\|\tilde{y}_\ell\|_2} - \frac{y_\ell}{\|y_\ell\|_2}$ to quantify the accumulated discrepancy up to the ℓ-th layer. Clearly, since both networks are fed with identical inputs, $\delta_0 = 0$.

While normalizing the layer inputs is a common practice to improve the training and generalization of neural networks [30], to avoid edge cases, the cascade analysis demands some assumptions about the degree of nonlinear distortion each layer introduces, as will be detailed below.

Definition 4.1. *A given matrix $W \in \mathbb{R}^{n\times p}$ with normalized rows is called consistent with near isometry of $\texttt{ASU}$ in set $\mathcal{D} \subseteq \mathbb{B}_2^p$, if for all $x, y \in \mathcal{D}$:*

$$\|x-y\|_2^2 - \varepsilon \le \beta^{-1}\left\|\texttt{ASU}(Wx) - \texttt{ASU}(Wy)\right\|_2^2 \le \|x-y\|_2^2 + \varepsilon, \quad (7)$$

where $\beta > 0$ and $\varepsilon \in [0, 1)$ are constants independent of x and y (possibly dependent on n and p).

When the layers of an ASU G-Net maintain the property in (7), we can show that the network discrepancy scales linearly with the number of layers:

Theorem 4.3. *Consider the cascade of L G-Net layers and the corresponding hyperdimensional embedding as (6). Assume that each layer ℓ of the network is consistent with near isometry of $\texttt{ASU}$ in $\mathbb{S}^{n_{\ell-1}-1}$ with parameters β_ℓ and ε_ℓ, as stated in (7). Then with probability exceeding $1 - L\exp(-c)$:*

$$\|\delta_L\|_2 \le c' \sum_{\ell=1}^{L}\left(\sqrt{\frac{(c+\log n_\ell)n_\ell}{N_\ell \beta_\ell (1-\varepsilon_\ell)}} + \sqrt{\frac{\varepsilon_\ell}{1+\varepsilon_\ell}}\right),$$

where $\delta_L = \frac{\tilde{y}_L}{\|\tilde{y}_L\|_2} - \frac{y_L}{\|y_L\|_2}$, and c' is an absolute constant.

A natural question is: what conditions on the set $\mathcal{D}$ or on the matrix W are sufficient to ensure that (7) holds? One simple scenario occurs when $\mathcal{D}$ has a sufficiently small radius, so that (7) follows directly from the continuity of $\texttt{ASU}$ (the case with more restricted datasets). Beyond this rather trivial case, broadly speaking, one can show that when W is picked to be a tall generic matrix with normalized rows, (7) is likely to hold within the entire unit ball $\mathbb{B}_2^p$. To state this rigorously, let's consider $\mathcal{W} \in \mathbb{R}^{n\times p}$ of the form

$$\mathcal{W} = \begin{bmatrix} g_1/\|g_1\| \\ \vdots \\ g_n/\|g_n\| \end{bmatrix}, \quad (8)$$

where $g_i \sim \mathcal{N}(0, I_p)$ are independent standard normal vectors.

Theorem 4.4. *Consider $\mathcal{W} \in \mathbb{R}^{n\times p}$, following the construction format in (8), where $n \gtrsim p \ge 27$. Let $g_i \sim \mathcal{N}(0, I_p)$ be independent standard normal vectors. Then, for all $x, y \in \mathbb{B}_2^p$, with probability exceeding $1 - c\exp(-c'p)$: $\|x-y\|_2^2 - \varepsilon_{n,p}^l \le \beta_{n,p}^{-1}\left\|\texttt{ASU}(\mathcal{W}x) - \texttt{ASU}(\mathcal{W}y)\right\|^2 \le \|x-y\|_2^2 + \varepsilon_{n,p}^u$, where*

$$\beta_{n,p}^{-1} = \frac{\pi^2 p}{4\left(\sqrt{n}+\sqrt{p}\right)^2}, \qquad \varepsilon_{n,p}^l = c_l \frac{\sqrt{p}}{\sqrt{n}+\sqrt{p}}, \qquad \varepsilon_{n,p}^u = c_u\left(\frac{\sqrt{p}}{\sqrt{n}+\sqrt{p}} + \frac{n+p^2}{p\left(\sqrt{n}+\sqrt{p}\right)^2}\right),$$

and c, c', c_l and c_u are absolute numerical constants.

Observe that by choosing n to be a sufficiently large multiple of p, $\varepsilon_{n,p}^l$ and $\varepsilon_{n,p}^u$ in Theorem 4.4 can be made desirably small. The proof of the theorem precludes the use of concentration results for

Lipschitz functions, due to the infinitely steep slope of the arcsin function around ± 1. A technical contribution and key part of the proof is bounding a fourth-order induced norm of $\mathcal{W}$ (which does not offer independence along the columns). Clearly, W_ℓ are determined by the G-Net training algorithm, however, by covering a large class of matrices, Theorem 4.4 showcases that when the trained weight matrices are tall and "spread-out", there is a good chance that (7) holds. It is also noteworthy that wide neural networks experience only minor changes in their weight distributions during training [15] (see Section M of Appendix for more details). This provides an opportunity to control the weight distribution of the trained models. Empirically, in all the experiments performed in this paper, we randomly initialized the G-Net weights according to (8), and never observed an instance of poor consistency between a G-Net and its corresponding hyperdimensional embedding.

5 Rademacher Embedding

One promising approach to enhance the efficiency of the proposed framework for hardware implementation is to utilize random vectors that are simpler to generate and apply than Gaussian vectors. In this section, we demonstrate that even the most basic choice—a Rademacher vector—can be employed as the embedding matrix, provided that the G-Net weights are sufficiently "spread out"—a condition that often holds in practice. The following result extends Grothendieck's lemma to Rademacher vectors.

Theorem 5.1 (Approximate Grothendieck identity)**.** *Suppose that $u, v \in \mathbb{S}^{p-1}$ are unit vectors. Let $\mathfrak{r}$ be a Rademacher random vector in $\mathbb{R}^p$. Then,*

$$\left| \mathbb{E}\left[\operatorname{sign}\left(\mathfrak{r}^\top u\right) \operatorname{sign}\left(\mathfrak{r}^\top v\right)\right] - \frac{2}{\pi} \arcsin(u^\top v) \right| \leq cg\left(u, \frac{v - \langle u, v\rangle u}{\|v - \langle u, v\rangle u\|_2}\right),$$

where $c = 264$ is an absolute constant, and for w and w' in $\mathbb{S}^{p-1}$, $g(w, w') = \sum_{i=1}^p (w_i^2 + w_i'^2)^{3/2}$.

Corollary 5.1. *Consider unit vectors $u, v \in \mathbb{S}^{p-1}$ such that $\|u\|_\infty = \mathcal{O}(p^{-1/2})$, $\|v\|_\infty = \mathcal{O}(p^{-1/2})$, and there exists some constant $c > 0$ such that $|\langle u, v\rangle| \leq 1 - c$. Then*

$$\left| \mathbb{E}\left[\operatorname{sign}\left(\mathfrak{r}^\top u\right) \operatorname{sign}\left(\mathfrak{r}^\top v\right)\right] - \frac{2}{\pi} \arcsin(u^\top v) \right| = \mathcal{O}\left(p^{-1/2}\right).$$

Informally speaking, Corollary 5.1 says that Grothendieck's identity approximately holds for Rademacher vectors when the inner product factors are dispersed. The layer consistency results of Section 4 can be readily extended to this substantially simpler encoding scheme.

Proposition 5.1. *Consider a similar RASU (or ASU) layer as Theorem 4.1, where for fixed constants $C, \delta > 0$, $\|x\|_\infty \leq Cp^{-1/2}$, $\|w_i\|_\infty \leq Cp^{-1/2}$, and $|w_i^\top x| \leq 1 - \delta$ for all $i \in \{1, \ldots, n\}$. Define the embedded output as $\tilde{y} = \texttt{ReLU}\left(\texttt{sign}(W\mathcal{R}^\top)\, \texttt{sign}(\mathcal{R}x)\right)$ (or $\tilde{y} = \texttt{Id}\left(\texttt{sign}(W\mathcal{R}^\top)\, \texttt{sign}(\mathcal{R}x)\right)$), where $\mathcal{R} \in \mathbb{R}^{N\times p}$ is a Rademacher matrix. Then, with high probability: $\|\frac{1}{N}\tilde{y} - y\|_2 \leq \mathcal{O}(\sqrt{N^{-1} n \log n} + \sqrt{n/p})$.*

Proposition 5.2. *Consider a similar TASU layer as Theorem 4.2, where additionally for fixed constants $C, \delta > 0$, $\|x\|_\infty \leq Cp^{-1/2})$, $\|w_i\|_\infty \leq Cp^{-1/2}$, and $0 < \ell_{\min} \leq |w_i^\top x| \leq 1 - \delta$ for all $i \in \{1, \ldots, n\}$. Define the embedded layer output as $\tilde{y} = \texttt{sign}\left(\texttt{sign}(W\mathcal{R}^\top)\, \texttt{sign}(\mathcal{R}x)\right)$, where $\mathcal{R} \in \mathbb{R}^{N\times p}$ is a Rademacher matrix. Assume $Cp^{-1/2} \leq \ell_{min}/\pi$. Fix $\varepsilon \leq \sqrt{n}$ and set $\kappa \geq \frac{\pi}{\ell_{\min}} \log \frac{4\sqrt{n}}{\varepsilon}$. Then, picking $N = \mathcal{O}(n\kappa^2 \log n/\varepsilon^2)$ guarantees that $\|y - \tilde{y}\|_2 = \mathcal{O}(\varepsilon + \kappa\sqrt{n/p})$ with high probability.*

In both cases, we observe an additional $\sqrt{n/p}$ discrepancy term which, unlike the Gaussian case, does not vanish by increasing N. However, taking into account the fact that $\|y\|_2$ scales with $\sqrt{n}$, the above results can be interpreted as the *relative* discrepancy diminishing with increasing N and p. While Gaussian embedding offers the best theoretical results, our numerical experiments show that Rademacher embedding achieves a comparably close performance.

6 Numerical Experiments and Concluding Discussion

Among various applications of the proposed framework, a key contribution lies in enhancing the accuracy of predictive HDC models. Practical implementation details of G-Nets beyond standard

fully connected layers—such as convolutional, pooling, classification, etc—are provided in Section N of the Appendix. Also, a detailed and extended version of the numerical experiments presented in this section, along with additional accuracy and robustness analysis across other datasets, is included in Section O. Here, we selectively present classification results on MNIST, CIFAR-10, and human activity recognition (HAR-WSS [29]).

The experiments involve fitting a G-Net to the original training data, followed by evaluation in the binary hyperspace by applying the corresponding EHD G-Net to the random sign embedding of test data. Owing to the G-Net's normalization property, all experiments required at most three convolutional and two fully connected layers, enabling fast and efficient training in the primal space, while still achieving strong baseline accuracies for the G-Net. Panels (a-c) of Figure 3 report the average test accuracies of EHD G-Net and other HDC methods. The reported hyperdimension N represents the average N_ℓ across G-Net layers and the dimension used in other methods. For each N, the conversion of the reference G-Net to an EHD G-Net was repeated multiple times with different random matrices; the same number of repetitions was applied to other HDC techniques. The plots show the resulting mean accuracies along with ± 1 standard deviation.

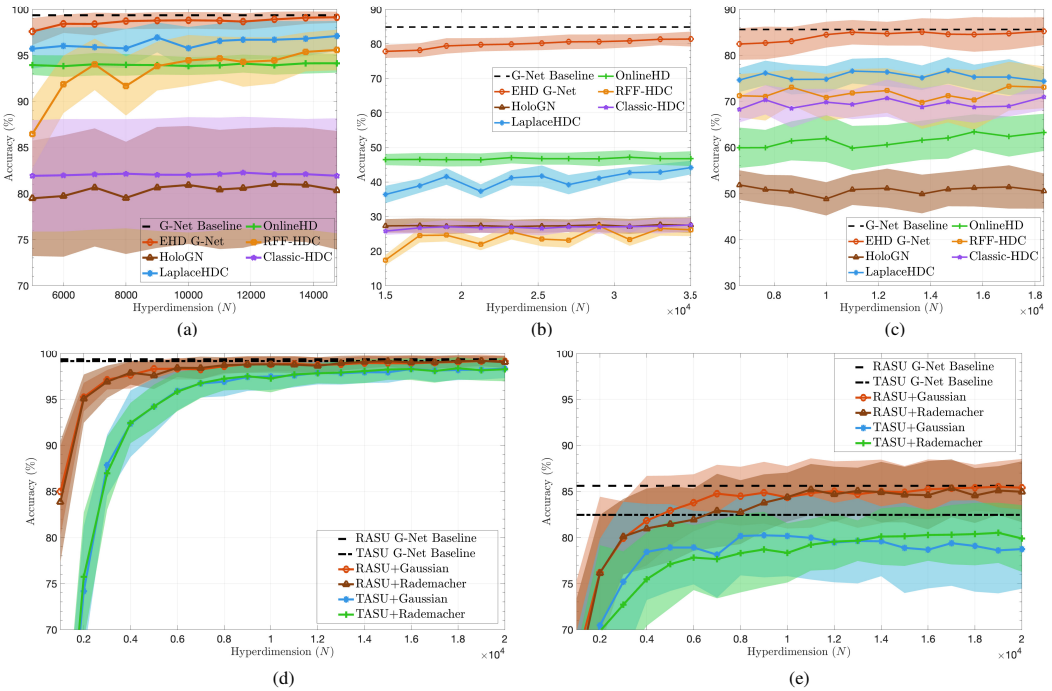


Figure 3: (a–c) Comparison of Rademacher RASU G-Net with other HDC methods (classic HDC [8], HoloGN [23], Laplace HDC [26], OnlineHD [10] and RFF-HDC[37]) on MNIST, CIFAR-10, and WSS (left to right); (d,e) performance of different G-Nets on MNIST (left) and WSS (right)

The G-Net architecture used in panels (a–c) is a RASU network. Although Gaussian embedding yields slightly better results, we report EHD G-Net accuracies with Rademacher embedding due to its hardware efficiency. One observes that without training a large binary model—and by training only a compact network in the primal space followed by inexpensive binary encoding—we achieve classifiers that outperform state-of-the-art HDC models by a significant margin. For example, HDC accuracies exceed 99% on MNIST and 81% on CIFAR-10, rivaling real-valued convolutional neural networks. Accuracy can be even further improved by employing larger G-Nets and higher hyperdimensions, as EHD G-Net performance asymptotically approaches that of the original G-Net with increasing N, as evidenced in the plots. Panels (d,e) present G-Net and EHD G-Net accuracies for both RASU and TASU networks, as well as for Gaussian and Rademacher embeddings. While RASU networks tend to yield higher accuracies, TASU remains competitive—for instance, achieving 98.4% on MNIST. As theory suggests, TASU-based EHD G-Nets converge more slowly to their G-Net baselines. These plots also reinforce the earlier statement regarding the close performance of Gaussian and Rademacher embeddings. Additional experiments and discussion are provided in Section O of the Appendix.

Relevance to Other Binary Neural Network Designs The proposed framework is rooted in hyperdimensional computing: it constructs binary, high-dimensional models without heavy training pipelines. Leveraging random matrix theory, we derive sample-complexity and concentration guarantees for the resulting embeddings. This contrasts with conventional BNN training, which poses a challenging discrete optimization problem and is therefore commonly addressed with heuristics that offer limited theoretical guarantees. G-Net sidesteps this difficulty by learning a continuous, low-dimensional generator whose sign embedding yields the binary model. Furthermore, thanks to the randomized nature of the process, a single trained G-Net can instantiate combinatorially many EHD G-Nets, providing architectural diversity that can be exploited for privacy and security on edge devices. In the extended experiments (Appendix, Section O), we also observe that BNNs initialized via an EHD G-Net attain peak accuracy within a few epochs, whereas BNNs trained from scratch typically require substantially many more epochs.

Computational Cost of EHD G-Net versus a Floating Point G-Net At the hardware level, any floating-point network can be viewed as a "binary" system by interpreting each weight as a bit sequence manipulated by floating-point operators. Unlike the EHD G-Net, however, these bit sequences contain significance structure, and the induced operations are substantially more complex. In Appendix Section P, we provide a mathematical comparison of computational cost, both time and memory, between EHD G-Net and its floating-point G-Net counterpart. We show that the binary encoding not only reduces computational complexity via a simpler representation, but also confers markedly higher robustness. More specifically, EHD G-Nets are significantly more resilient to bit flips than the corresponding floating-point models where weights are viewed as bit sequences.

In conclusion, this paper introduces a new class of randomized binary neural networks formed by inexpensive sampling from a meta-distribution with controllable mean and variance. As a primary application, the proposed framework demonstrates that hyperdimensional computing can serve as a practical and competitive computational paradigm, achieving performance comparable to conventional deep learning methods, if multi-layer encoding or inference pipelines are considered. Additionally, the paper advances the theoretical foundations of HDC by establishing connections between hyperdimension size and model performance. Moreover, this paper opens several promising directions for future research. One avenue involves fine-tuning and compressing the theoretically grounded models introduced here, including the possibility of directly training EHD G-Nets through easy optimization techniques. Another direction is expanding the theoretical understanding of the Rademacher-based embeddings of the hypersphere to the hypercube we introduced. Additionally, exploring the robustness of the proposed randomized binary neural networks—especially under adversarial conditions—remains a compelling avenue.

In terms of limitations, the proposed framework is primarily motivated by HDC applications where high-dimensional models are typical. Accordingly, EHD G-Nets—derived without training in the binary space—are also high-dimensional, which could be seen as a limitation in certain use cases. Moreover, the network analysis in Section 4.1 focuses on the base ASU activations to streamline the theoretical development. While the cascade analysis of TASU and RASU layers involves many similar steps, a comprehensive treatment requires additional analysis, which is deferred to a more extended presentation.

References

[1] Alireza Aghasi, Afshin Abdi, Nam Nguyen, and Justin Romberg. Net-trim: Convex pruning of deep neural networks with performance guarantee. *Advances in neural information processing systems*, 30, 2017.

[2] Alireza Aghasi, Afshin Abdi, and Justin Romberg. Fast convex pruning of deep neural networks. *SIAM Journal on Mathematics of Data Science*, 2(1):158–188, 2020.

[3] Toygun Basaklar, Yigit Tuncel, Shruti Y. Narayana, Suat Gumussoy, and Umit Y. Ogras. Hypervector design for efficient hyperdimensional computing on edge devices. In *tinyML 2021 Research Symposium*, 2021.

[4] Hanning Chen, Mariam Issa, Yang Ni, and Mohsen Imani. Darl: Distributed reconfigurable accelerator for hyperdimensional reinforcement learning. In *2022 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–9. ACM, 2022.

[5] Yu-Chuan Chuang, Cheng-Yang Chang, and An-Yeu Andy Wu. Dynamic hyperdimensional computing for improving accuracy-energy efficiency trade-offs. In *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 1–5. IEEE, 2020.

[6] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv e-print arXiv:1602.02830*, 2016.

[7] Shijin Duan, Yejia Liu, Shaolei Ren, and Xiaolin Xu. Lehdc: learning-based hyperdimensional computing classifier. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pages 1111–1116. ACM, 2022.

[8] Lulu Ge and Keshab K Parhi. Classification using hyperdimensional computing: A review. *IEEE Circuits and Systems Magazine*, 20(2):30–47, 2020.

[9] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-power computer vision*, pages 291–326. Chapman and Hall/CRC, 2022.

[10] Alejandro Hernández-Cano, Namiko Matsumoto, Eric Ping, and Mohsen Imani. OnlineHD: Robust, efficient, and single-pass online learning using hyperdimensional system. In *2021 Design, Automation, and Test in Europe Conference Exhibition (DATE)*, pages 56–61. IEEE, 2021.

[11] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *journal of machine learning research*, 18(187):1–30, 2018.

[12] Mohsen Imani, Justin Morris, Samuel Bosch, Helen Shu, Giovanni De Micheli, and Tajana Rosing. Adapthd: Adaptive efficient training for brain-inspired hyperdimensional computing. In *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2019.

[13] Mohsen Imani, Justin Morris, John Messerly, Helen Shu, Yaobang Deng, and Tajana Rosing. BRIC: Locality-based encoding for energy-efficient brain-inspired hyperdimensional computing. In *Proceedings of the 56th Annual Design Automation Conference 2019*, pages 1–6, 2019.

[14] Mohsen Imani, Sahand Salamat, Behnam Khaleghi, Mohammad Samragh, Farinaz Koushanfar, and Tajana Rosing. Sparsehd: Algorithm-hardware co-optimization for efficient high-dimensional computing. In *2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pages 190–198. IEEE, 2019.

[15] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[16] Geethan Karunaratne, Manuel Le Gallo, Giovanni Cherubini, Luca Benini, Abbas Rahimi, and Abu Sebastian. In-memory hyperdimensional computing. *Nature Electronics*, 3(6):327–337, 2020.

[17] Geethan Karunaratne, Abbas Rahimi, Manuel Le Gallo, Giovanni Cherubini, and Abu Sebastian. Real-time language recognition using hyperdimensional computing on phase-change memory array. In *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 1–1, 2021.

[18] Behnam Khaleghi, Hanyang Xu, Justin Morris, and Tajana Šimunić Rosing. tiny-HD: Ultra-efficient hyperdimensional computing engine for IoT applications. In *2021 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pages 408–413, 2021.

[19] 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 Computing Surveys*, 55(9):1–52, 2023.

[20] 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, 2022.

[21] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, Kwang-Ting Cheng, Martial Hebert, Cristian Sminchisescu, Yair Weiss, and Vittorio Ferrari. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Computer Vision - ECCV 2018*, volume 11219 of *Lecture Notes in Computer Science*, pages 747–763. Springer International Publishing AG, Switzerland, 2018.

[22] Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*, pages 382–391. Springer, 2018.

[23] Alec Xavier Manabat, Celine Rose Marcelo, Alfonso Louis Quinquito, and Anastacia Alvarez. Performance analysis of hyperdimensional computing for character recognition. In *2019 International Symposium on Multimedia and Communication Technology (ISMAC)*, pages 1–5, 2019.

[24] Justin Morris, Mohsen Imani, Samuel Bosch, Anthony Thomas, Helen Shu, and Tajana Rosing. Comphd: Efficient hyperdimensional computing using model compression. In *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–6. IEEE, 2019.

[25] Yang Ni, Mariam Issa, Danny Abraham, Mahdi Imani, Xunzhao Yin, and Mohsen Imani. Hdpg: hyperdimensional policy-based reinforcement learning for continuous control. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pages 1141–1146. ACM, 2022.

[26] Saeid Pourmand, Wyatt D. Whiting, Alireza Aghasi, and Nicholas F. Marshall. Laplace-hdc: Understanding the geometry of binary hyperdimensional computing. *The Journal of artificial intelligence research*, 82:1293–1323, 2025.

[27] DA Rachkovskij. Formation of similarity-reflecting binary vectors with random binary projections. *Cybernetics and Systems Analysis*, 51:313–323, 2015.

[28] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, Ali Farhadi, Nicu Sebe, Jiri Matas, Max Welling, and Bastian Leibe. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Computer Vision - ECCV 2016*, volume 9908 of *Lecture Notes in Computer Science*, pages 525–542. Springer International Publishing AG, Switzerland, 2016.

[29] Jorge Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. Human Activity Recognition Using Smartphones. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C54S4K.

[30] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016.

[31] Anthony Thomas, Sanjoy Dasgupta, and Tajana Rosing. A theoretical perspective on hyperdimensional computing. *Journal of Artificial Intelligence Research*, 72:215–249, October 2021.

[32] Pere Vergés, Mike Heddes, Igor Nunes, Denis Kleyko, Tony Givargis, and Alexandru Nicolau. Classification using hyperdimensional computing: A review with comparative analysis. *Artificial Intelligence Review*, 58(6):173, 2025.

[33] Pere Vergés, Mike Heddes, Igor Nunes, Denis Kleyko, Tony Givargis, and Alexandru Nicolau. Classification using hyperdimensional computing: a review with comparative analysis. *The Artificial intelligence review*, 58(6):173–, 2025.

[34] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[35] Junyao Wang, Sitao Huang, and Mohsen Imani. Disthd: A learner-aware dynamic encoding method for hyperdimensional classification. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2023.

[36] Skyler Wu, Fred Lu, Edward Raff, and James Holt. Exploring the sharpened cosine similarity. In *I Can't Believe It's Not Better Workshop: Understanding Deep Learning Through Empirical Falsification*, 2022.

[37] Tao Yu, Yichi Zhang, Zhiru Zhang, and Christopher M De Sa. Understanding hyperdimensional computing for parallel single-pass learning. *Advances in Neural Information Processing Systems*, 35:1157–1169, 2022.

[38] Yichi Zhang, Junhao Pan, Xinheng Liu, Hongzheng Chen, Deming Chen, and Zhiru Zhang. Fracbnn: Accurate and fpga-efficient binary neural networks with fractional activations. In *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 171–182, 2021.

[39] Zhuowen Zou, Yeseong Kim, Farhad Imani, Haleh Alimohamadi, Rosario Cammarota, and Mohsen Imani. Scalable edge-based hyperdimensional learning system with brain-like neural adaptation. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15. ACM, 2021.

Appendix

Contents

A An Overview of Hyperdimensional Computing

Hyperdimensional computing (HDC) is a machine learning paradigm inspired by the structure and operation of the brain. In HDC systems, data are represented by high-dimensional vectors, called hypervectors, typically on the order of hundreds or thousands of dimensions. Vector entries may be taken from any number of sets; various models have used real-valued entries [62], complex entries [63], and binary entries [26], [48]. In the latter of these cases, hyperdimensional computing schemes benefit from greatly simplified arithmetic, in contrast to traditional neural networks with float-valued arithmetic. HDC computations are also highly parallelizable and tolerant to stochastic computation environments [37], [69, 60].

An HDC scheme is principally comprised of two stages: embedding and inference. Consequently, the differences in HDC schemes primarily stem from the differences in their respective embedding and inference methods. In the embedding stage, input data x in some input space $\mathcal{X}$ is embedded into a high-dimensional vector space (hyperspace) $\mathcal{Y}$ as a hypervector through an embedding map $\phi\colon \mathcal{X} \to \mathcal{Y}$. Regardless of the embedding method, the mapping to $\mathcal{Y}$ should ensure that similar input data are represented by similar hypervectors.

For classification tasks with k output labels, the inference stage involves constructing a set of class representatives $\psi_c \in \mathcal{Y}$ for each class $c \in \{1, \ldots, k\}$. Predicting the class of an input x then reduces to comparing its embedding y with all class representatives and selecting the label corresponding to the representation most similar to y. It is worth noting that our proposed framework can be interpreted within the HDC framework, where in the EHD G-Net architecture of Figure 2(b), the first $L-1$ layers function as a multi-stage embedding pipeline, and the final classification layer operates as the inference module.

A.1 Hypervector Arithmetic

In classic HDC, with information embedded as hypervectors, one may use operations on these hypervectors to form composite hypervectors, and measure similarity of hypervectors with a similarity metric. The vector arithmetic operations often used are superposition, binding, and permutation as detailed below.

Superposition. The superposition operation $+\colon \mathcal{Y} \times \mathcal{Y} \to \mathcal{Y}$ combines hypervectors in a way that the similarity of the original operands is preserved [33]. In most cases, the superposition operation is taken as typical element-wise vector addition, motivating the choice of notation.

Binding. The binding operation $\circ\colon \mathcal{Y} \times \mathcal{Y} \to \mathcal{Y}$ associates two hypervectors into a new hypervector, which is dissimilar from both its operands. This operation is often used to combine component hypervectors into composite hypervectors, retaining some information of each its factors. In the special case of binary-valued hypervectors when binding is taken to be elementwise multiplication, the resulting vector arithmetic is particularly amenable to hardware-level implementation [26].

Permutation. The permutation operation, as the name suggests, permutes the entries of a hypervector. Denoting by Π a permutation matrix, the product Πy permutes the entries of y according to Π.

Similarity. A similarity measure is a function $\delta\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. The measure is typically used to determine to which class a hypervector belongs, e.g. by selecting

$$\text{class}(y) = \arg\max_{c \in \{1,\ldots,k\}} \delta(y, \psi_c).$$

A linear classifier is obtained by selecting $\delta(y, y') = y^\top y'$. Naturally, the choice of embedding method and similarity measure should be amenable in the sense that, $\delta(\phi(x), \phi(x'))$ correlates with the similarity of x and x' in $\mathcal{X}$.

A.2 Encoding Methods

We now briefly review several representative encoding techniques, nearly all of which incorporate some form of randomness.

Record-Based Encoding. This encoding framework applies when the feature values are discrete or can be closely approximated by discrete levels. This encoding method employs two types of hypervectors, representing the feature positions and their level values, respectively. The encoding

involves the binding and bundling of random hypervectors associated with the feature positions and level values [51]. In this framework, the position hypervectors are independent and uncorrelated, while the level hypervectors are correlated for neighboring levels. This way correlated vectors in $\mathcal{X}$ remain correlated in $\mathcal{Y}$.

N-Gram-Based Encoding. This method also applies to feature vectors with discrete levels. First level hypervectors are randomly generated. Then the encoding happens by binding level hypervectors that are rotationally permuted according to their feature position [8].

Fractional Power Encoding. This method enables the representation of continuous numerical values using complex-valued hypervectors. It begins by sampling a random basis hypervector with complex values (each component corresponds to a random phasor). A value x is encoded by applying fractional self-binding of the basis hypervector x number of times. The similarity kernel of the resulting encoding depends on the distribution generating the basis hypervector [54, 47].

Random Projection Encoding. This is also a general encoding scheme which incorporates random projections as the core operation [27]. For hyperspace $\mathcal{Y}$ of dimension N and input space $\mathcal{X} \subseteq \mathbb{R}^p$ this encoding is performed by $\phi(x) = \sigma(\mathcal{G}x)$, where $\mathcal{G} \in \mathbb{R}^{N \times p}$ has i.i.d. standard normal entries, and σ is a quantization operation such as a sign function. A bias $b \in \mathcal{Y}$ may optionally be added to the argument of σ, as in the method described in [66].

Random Fourier Feature Encoding This method proposed in [37] begins by prescribing a desired symmetric similarity matrix $M \in R^{l \times l}$ among l level hypervectors. Then, binary hypervectors are generated by computing the singular value decomposition (SVD) of $\sin(\frac{\pi}{2}M)$ to obtain left singular values U and diagonal matrix S_+, which is identical to the typical S with its negative entries set to 0, and $X \in \mathbb{R}^{n \times D}$ has i.i.d. standard normal entries. The Laplace-HDC model by Pourmand *et al.* [26] extends this work and shows that choosing M to be a Laplacian kernel is particularly amenable to this framework.

A.3 Inference Methods

In this section, we will briefly describe several HDC schemes present in recent literature. We will discuss both broad classes of architectures, as well as some selected realizations of these architectures. We emphasize the following techniques are not mutually exclusive within any particular HDC model. As the array of methods present in HDC models is quite large, we will discuss a relatively small number of selected methods. For a more complete survey of HDC methods, see [33].

Centroid Classifiers. Centroid classifiers are the simplest type of HDC classifiers [53]. In these methods, a hypervector is generated for each class by adding the hypervectors of the training data that belong to that class.

Adaptive Training. Adaptive learning techniques are able to adapt to changing data, and are suitable for real-time predictions. Essentially, they operate by updating class representatives only when the classifier would incorrectly predict the class of a training datum. The update to the class representative may be done directly proportional to the misclassified hypervector, as in the AdaptHD model [12], or by scaling the misclassified vector by some expression relating to similarity of the misclassified vector with its predicted class, as in OnlineHD [10].

Regenerative Training. Regenerative training techniques attempt to improve the selection of class representatives by passing through the data several times. In each pass, a prescribed number of entries of the class representatives carrying the least information (determined by entries with the smallest variance, tracked over the updates) are dropped and re-selected. Notable examples of HDC schemes utilizing this technique are NeuralHD [39] and DistHD [35].

Dimensionality Reduction. A number of HDC models reduce the dimensionality of the hyperspace, thereby increasing the time and memory efficiency of computation. With fewer dimensions in which to embed data, HDC models utilizing dimensionality reduction techniques often incorporate adaptive and regenerative techniques. The SparseHD [14] first trains a standard dense centroid classifier, and compares its performance against a sparsified copy of the classifier, with retraining in the case the sparsification reduces the model accuracy by too large a margin. CompHD [24] partitions hypervectors into s equally-sized components, then binding each component with another hypervector based on its position, and superimposing each component to yield a compressed model.

Optimization-Based Methods. Many HDC models involve tunable parameters, and a broad class of these methods optimize performance by jointly tuning them. In this sense, a subset of HDC approaches can be regarded as optimization-based. As an example, LeHDC [7] learns a classifier as a neural network, and binarizes its classification matrix to produce a small binary classifier.

We emphasize that these models all share the common two-step classification pipeline of embedding and inference. While classification is a prototypical task for HDC, many HDC models perform other machine learning tasks, such as data regression [50], reinforcement learning [4], [25] and multi-task learning [44, 43].

B An Overview of Binary Neural Networks

Binary neural networks (BNNs) are a class of neural networks which use binary-valued weights during operation. There is a rich existing literature on BNN architecture, for which we provide here a non-exhaustive review. Extant methods for developing binary neural networks can be broadly divided into several types. In no particular order, first are those that convert an existing pre-trained neural net operating with weights of a non-binary datatype, often float into an equivalent one operating with binary-valued weights. Second are those which train a binary-weighted network from scratch without the need for float-valued weights during any step of training or deployment. Additionally, there are several schemes which hybridize these types, using some combination of float- and binary-valued weights during training. For a more extensive review of BNN methods and architectures, see [71, 68, 73].

A notable subclass of BNN methods is the related XNOR/XOR-Nets, a class of methods which approximate the convolution operations in a network with binary-valued weights, in the former by using a combination of XNOR and bitcount, and in the latter with XOR in place of XNOR [28], [74]. In either variety, these networks train a float-valued network and update an approximate binary-valued network which is updated in parallel in every training pass [28], [57, 74], in contrast with our method of binarizing a G-Net into an EHD G-Net, which occurs as distinct steps in our pipeline. The Bi-Real Net framework proposed by Liu et al. [21], connects the float-valued activations to those of consecutive blocks to increase the representational capacity of the network [56, 59]. In this sense, layers in a Bi-Real net have a degree of cross-communication which is not present in our model. Another approach at network binarization is the Accurate-Binary-Convolution Network (ABC-Net) which converts a pre-trained CNN into a BNN by determining a basis of binary filters which, together with the appropriate weights, yield approximations of the original layer [58]. In contrast, our EHD G-Nets need only a single binary weight tensor in each layer.

C Proof of Proposition 2.1

Proposition 2.1. *For a standard Gaussian matrix $\mathcal{G} \in \mathbb{R}^{N\times n}$, consider the mapping $\mathcal{A}(x) = \texttt{sign}(\mathcal{G}x)$ from $\mathcal{X} = \mathbb{S}^{n-1}$ (equipped with a geodesic distance $\mathcal{D}_G$) to $\mathcal{Y} = \{-1,1\}^N$ (equipped with a normalized Hamming distance $\mathcal{D}_H$). Fix $\varepsilon \in (0,1)$, then there exist constants c_1 and c_2 such that if $N \geq c_1 n/\varepsilon^2$, with probability exceeding $1-\exp(-c_2 n)$, $\mathcal{A}$ is an ε-near-isometry. That is*

$$\forall x, y \in \mathbb{S}^{n-1}: \qquad \left|\mathcal{D}_{\mathcal{H}}\left(\texttt{sign}\left(\mathcal{G}x\right), \texttt{sign}\left(\mathcal{G}y\right)\right) - \mathcal{D}_G(x,y)\right| \leq \varepsilon.$$

Proof. The proof is a direct application of the following result from [61] which provides an optimal dependency on the dimensions and ε:

Lemma C.1 (Theorem 2.2 in [61]). *Suppose $\mathcal{G} \in \mathbb{R}^{N\times n}$ has independent $\mathcal{N}(0,1)$ entries. For a given a set $K \subseteq \mathbb{S}^{n-1}$, define the Gaussian width as*

$$\omega(K) = \mathbb{E}_{g\sim\mathcal{N}(0,I)}\left[\sup_{x\in K} g^\top x\right],$$

and denote cone$(K) = \{\alpha v : \alpha \geq 0, v \in K\}$. If cone$(K)$ is a subspace, then for a fixed $\varepsilon \in (0,1)$, there exist constants c_1 and c_2 such that when $N \geq c_1 \frac{\omega^2(K)}{\varepsilon^2}$, with probability exceeding $1 - \exp\left(-c_2\delta^2 N\right)$:

$$\forall x, y \in \mathbb{S}^{n-1}: \qquad \left|\mathcal{D}_{\mathcal{H}}\left(\boldsymbol{sign}\left(\mathcal{G}x\right), \boldsymbol{sign}\left(\mathcal{G}y\right)\right) - \mathcal{D}_G(x,y)\right| \leq \varepsilon. \tag{9}$$

Here angle(x,y) denotes the smaller angle between x and y.

Based on Lemma C.1, to use the result we only need to compute the Gaussian width of the unit sphere:

$$\omega\left(\mathbb{S}^{n-1}\right) = \mathbb{E}_{g\sim\mathcal{N}(0,I)}\left[\sup_{x\in\mathbb{S}^{n-1}} g^\top x\right] = \mathbb{E}_{g\sim\mathcal{N}(0,I)}\|g\|_2 = \frac{\sqrt{2}\Gamma(n/2+1/2)}{\Gamma(n/2)} \leq \sqrt{n},$$

where we used the fact that $\|g\|_2$ has a chi distribution with n degrees of freedom, and applied Gautschi's inequality. □

As an alternative presentation of Lemma C.1, one may be interested in a version of (9), which uses the same metric on both the source and encoded spaces. To that end, notice that

$$\begin{aligned}
\mathcal{D}_{\mathcal{H}}\left(\mathtt{sign}\left(\mathcal{G}x\right), \mathtt{sign}\left(\mathcal{G}y\right)\right) &= \frac{1}{4N}\left\|\mathtt{sign}\left(\mathcal{G}x\right) - \mathtt{sign}\left(\mathcal{G}y\right)\right\|^2 \\
&= \frac{1}{2} - \frac{1}{2}\left\langle \frac{1}{\sqrt{N}}\mathtt{sign}\left(\mathcal{G}x\right), \frac{1}{\sqrt{N}}\mathtt{sign}\left(\mathcal{G}y\right)\right\rangle \\
&= \frac{1}{2} - \frac{1}{2}\cos\left(\text{angle}\left(\frac{1}{\sqrt{N}}\mathtt{sign}\left(\mathcal{G}x\right), \frac{1}{\sqrt{N}}\mathtt{sign}\left(\mathcal{G}y\right)\right)\right) \\
&= \sin^2\left(\frac{1}{2}\text{angle}\left(\frac{1}{\sqrt{N}}\mathtt{sign}\left(\mathcal{G}x\right), \frac{1}{\sqrt{N}}\mathtt{sign}\left(\mathcal{G}y\right)\right)\right) \\
&= \sin^2\left(\frac{\pi}{2}\mathcal{D}_G\left(\frac{1}{\sqrt{N}}\mathtt{sign}\left(\mathcal{G}x\right), \frac{1}{\sqrt{N}}\mathtt{sign}\left(\mathcal{G}y\right)\right)\right),
\end{aligned}$$

where angle(u,v) represents the smaller angle between two vectors u and v. Therefore, an alternative presentation of the ε-near isometry in (9) is

$$\forall x, y \in \mathbb{S}^{n-1}: \qquad \left|\sin^2\left(\frac{\pi}{2}\mathcal{D}_G\left(\frac{1}{\sqrt{N}}\mathtt{sign}\left(\mathcal{G}x\right), \frac{1}{\sqrt{N}}\mathtt{sign}\left(\mathcal{G}y\right)\right)\right) - \mathcal{D}_G(x,y)\right| \leq \varepsilon,$$

where a geodesic distance is used for the source vectors x, y and their encodings $\mathtt{sign}\left(\mathcal{G}x\right)$ and $\mathtt{sign}\left(\mathcal{G}y\right)$.

D Proof of Lemma 3.1 (Grothendieck Identity)

Lemma 3.1. *Let g be a standard Gaussian vector in $\mathbb{R}^p$. Then, for fixed vectors $u, v \in \mathbb{R}^p$:*

$$\mathbb{E}\left[\text{sign}(g^\top u)\,\text{sign}(g^\top v)\right] = \frac{2}{\pi}\arcsin\left(\frac{u^\top v}{\|u\|_2\|v\|_2}\right).$$

Proof. We show that for fixed vectors $u, v \in \mathbb{S}^{p-1}$:

$$\mathbb{E}\left[\text{sign}(g^\top u)\,\text{sign}(g^\top v)\right] = \frac{2}{\pi}\arcsin\left(u^\top v\right).$$

Among different ways to show this identity, the geometric argument is the most straightforward. By the rotational invariance of the Gaussian distribution, we can restrict our attention to the two-dimensional subspace spanned by u and v. Consider the set $S = \{z : u^\top z \leq 0 \text{ and } v^\top z \geq 0\}$. If α is the angle between u and v, then the angle between the two lines that define the boundary of S, which are perpendicular to u and v, respectively, is also α, see Figure 4.

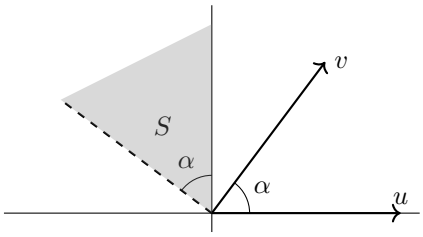


Figure 4: The set $S = \{z : u^\top z \leq 0 \text{ and } v^\top z \geq 0\}$

By similar reasoning, the set $\{z : u^\top z \geq 0 \text{ and } v^\top z \leq 0\}$ is a radial set of angle α. Thus, $\operatorname{sign}(u^\top z) = \operatorname{sign}(v^\top z)$ outside of radial sets of combined angle 2α. Thanks to the rotation invariance of the Gaussian vectors:

$$\mathbb{P}\left\{\operatorname{sign}(u^\top g) = \operatorname{sign}(v^\top g)\right\} = \frac{2\pi - 2\alpha}{2\pi} = \frac{1}{2} + \frac{\arcsin(u^\top v)}{\pi},$$

where we used the fact that $\alpha = \arccos(u^\top v) = \frac{\pi}{2} - \arcsin(u^\top v)$. Finally,

$$\begin{aligned}
\mathbb{E}\left[\operatorname{sign}(g^\top u)\operatorname{sign}(g^\top v)\right] &= \mathbb{P}\left\{\operatorname{sign}(u^\top g) = \operatorname{sign}(v^\top g)\right\} - \mathbb{P}\left\{\operatorname{sign}(u^\top g) \neq \operatorname{sign}(v^\top g)\right\} \\
&= 2\mathbb{P}\left\{\operatorname{sign}(u^\top g) = \operatorname{sign}(v^\top g)\right\} - 1 \\
&= \frac{2}{\pi}\arcsin\left(u^\top v\right).
\end{aligned}$$

□

E Proof of Theorem 4.1

Theorem 4.1. *Consider a RASU (or ASU) layer with input $x \in \mathbb{S}^{p-1}$ and a weight matrix $W \in \mathbb{R}^{n\times p}$ whose rows are ℓ_2-normalized. Let the layer output be given by $y = \texttt{RASU}(Wx)$ (or $y = \texttt{ASU}(Wx)$). Define the embedded output as $\tilde{y} = \texttt{ReLU}\left(\texttt{sign}(W\mathcal{G}^\top)\,\texttt{sign}(\mathcal{G}x)\right)$ (or $\tilde{y} = \texttt{Id}\left(\texttt{sign}(W\mathcal{G}^\top)\,\texttt{sign}(\mathcal{G}x)\right)$), where $\mathcal{G} \in \mathbb{R}^{N\times p}$ is a standard Gaussian matrix. Then, for any $c > 0$, with probability at least $1 - \exp(-c)$,*

$$\left\|\frac{1}{N}\tilde{y} - y\right\|_2 \leq \sqrt{\frac{2(c + \log 2n)n}{N}}.$$

Proof. We present the following general result, applicable to both RASU and TASU layers, with its proof provided at the end of this section.

Theorem E.2. *Consider a random matrix $\mathcal{X} \in \mathbb{R}^{n\times N}$ with entries x_{ij}, such that $a \leq x_{i,j} \leq b$ almost surely. Define a vector $e \in \mathbb{R}^n$ with entries $e_i = f(N^{-1}\sum_{j=1}^N x_{ij}) - f(\mathbb{E}[N^{-1}\sum_{j=1}^N x_{ij}])$, where f is a ρ-Lipschitz function in $[a,b]$. Then*

(a) When $\mathcal{X}$ maintains independence both along the rows and columns (i.e., independent entries), then for all $c > 0$, with probability exceeding $1 - \exp(-c)$:

$$\|e\|_2 \leq \frac{\rho(b-a)\sqrt{n}}{2\sqrt{N}} + \left(\frac{c}{c'}\right)^{1/4} \frac{\rho(b-a)n^{1/4}}{\sqrt{N}} = \mathcal{O}\left(\rho(b-a)\sqrt{\frac{n}{N}}\right),$$

where c' is a positive constant.

(b) When $\mathcal{X}$ maintains independence only along the columns, then for all $c > 0$, with probability exceeding $1 - \exp(-c)$:

$$\|e\|_2 \leq \sqrt{\frac{(c+\log 2n)n\rho^2(b-a)^2}{2N}} = \mathcal{O}\left(\rho(b-a)\sqrt{\frac{n\log n}{N}}\right).$$

We now proceed by applying this result to a RASU/ASU layer. Recall that

$$y = \texttt{RASU}(Wx), \qquad \text{and} \qquad \tilde{y} = \texttt{ReLU}\left(\texttt{sign}(W\mathcal{G}^\top)\texttt{sign}(\mathcal{G}x)\right).$$

The discrepancy between the two layers can be written as $\left\|\frac{1}{N}\tilde{y} - y\right\|^2 = \sum_{i=1}^n {e_i}^2$, where

$$e_i = \frac{1}{N}\text{ReLU}\left(\sum_{j=1}^N \text{sign}\left(g_j^\top w_i\right)\text{sign}\left({g_j}^\top x\right)\right) - \text{ReLU}\left(\frac{2}{\pi}\arcsin\left(w_i^\top x\right)\right)$$

$$= \text{ReLU}\left(\frac{1}{N}\sum_{j=1}^N \text{sign}\left(g_j^\top w_i\right)\text{sign}\left({g_j}^\top x\right)\right) - \text{ReLU}\left(\mathbb{E}\left[\frac{1}{N}\sum_{j=1}^N \text{sign}\left(g_j^\top w_i\right)\text{sign}\left({g_j}^\top x\right)\right]\right).$$

This is clearly an instance of Theorem E.2 part (b), with `ReLU` as the Lipschitz function and the quantities $\text{sign}\left(g_j^\top w_i\right)\text{sign}\left({g_j}^\top x\right)$ as x_{ij}. Since the Lipschitz constant for `ReLU` and the identity function corresponding to an `ASU` layer are both 1, and $-1 \leq \text{sign}\left(g_j^\top w_i\right)\text{sign}\left({g_j}^\top x\right) \leq 1$, this immediately implies that with probability exceeding $1 - \exp(-c)$:

$$\left\|\frac{1}{N}\tilde{y} - y\right\|_2 \leq \sqrt{\frac{2(c+\log 2n)n}{N}} = \mathcal{O}\left(\sqrt{\frac{n\log n}{N}}\right).$$

□

If the construction were modified such that the elements of e_i became independent, Theorem E.2(a) implies that the discrepancy could be reduced to $\mathcal{O}(\sqrt{n/N})$. However, achieving such independence would require employing a separate embedding matrix for each row of W, which increases the memory and computational overhead. The following example suggests that the maximum possible reduction achievable through promoting independence is limited to a factor of $\log n$.

E.1 Example

Suppose that $f(x) = \rho x$, and let x_{1j} be i.i.d. random variables for $j = 1, \ldots, N$ such that $\mathbb{P}[x_{1j} = +a] = \mathbb{P}[x_{1j} = -a] = 1/2$. Assume that $x_{ij} = x_{1j}$ for $i = 2, \ldots, n$ and $j = 1, \ldots, N$. Applying the upper bound on $\|e\|_2$ from Theorem E.2 part (b) to this example gives

$$\|e\|_2 = \mathcal{O}\left(\rho a\sqrt{\frac{n\log n}{N}}\right),$$

with high probability. For comparison, observe that

$$\mathbb{E}\|e\|_2 = \frac{\rho\sqrt{n}}{N}\left|\sum_{j=1}^N x_{1j}\right| = \Theta\left(\rho a\sqrt{\frac{n}{N}}\right),$$

as $N \to \infty$. Hence, the upper bound is only a factor of $\mathcal{O}(\sqrt{\log n})$ greater than the expected value, at least in some cases.

E.2 Proof of Theorem E.2

Proof of part (a). We start the proof by using Hoeffding's inequality stated below:

Lemma E.1 (Hoeffding's Inequality)**.** *Let $\chi_1, \ldots, \chi_N$ be independent random variables such that $a_j \leq \chi_j \leq b_j$ almost surely. Then for all $t \geq 0$:*

$$\mathbb{P}\left\{\left|\sum_{j=1}^{N} \chi_j - \mathbb{E}\left[\sum_{j=1}^{N} \chi_j\right]\right| \geq t\right\} \leq 2\exp\left(-\frac{2t^2}{\sum_{j=1}^{N}(b_j - a_j)^2}\right).$$

Applying this inequality to $y_i = N^{-1}\sum_{j=1}^{N} x_{ij}$ reveals that for all $t > 0$:

$$\mathbb{P}\left\{(y_i - \mathbb{E}[y_i])^2 \geq t\right\} \leq 2\exp\left(-\frac{2Nt}{(b-a)^2}\right).$$

A direct consequence of the Lipschitz continuity of f is that if $(f(z_1) - f(z_2))^2 \geq t$, then $(z_1 - z_2)^2 \geq t/\rho^2$, which implies

$$\begin{aligned}\mathbb{P}\left\{e_i^2 \geq t\right\} = \mathbb{P}\left\{(f(y_i) - f(\mathbb{E}[y_i]))^2 \geq t\right\} &\leq \mathbb{P}\left\{(y_i - \mathbb{E}[y_i])^2 \geq \frac{t}{\rho^2}\right\} \\ &\leq 2\exp\left(-\frac{2Nt}{\rho^2(b-a)^2}\right).\end{aligned}$$

This implies that e_i^2 is a sub-exponential random variable with sub-exponential norm

$$\|e_i^2\|_{\psi_1} = \mathcal{O}\left(\rho^2(b-a)^2 N^{-1}\right).$$

This also indicates that $\|e_i^2 - \mathbb{E}[e_i^2]\|_{\psi_1} \leq c'\rho^2(b-a)^2 N^{-1}$ (see §2.7 of [34]). Next, we will use Bernstein's inequality for the sum of sub-exponential random variables.

Lemma E.2 (Bernstein's Inequality)**.** *Let $\chi_1, \ldots, \chi_n$ be independent zero-mean sub-exponential random variables. Then for all $t \geq 0$:*

$$\mathbb{P}\left\{\sum_{i=1}^{n} \chi_i \geq t\right\} \leq \exp\left(-\tilde{c}\min\left(\frac{t^2}{\sum_{i=1}^{n}\|\chi_i\|_{\psi_1}^2}, \frac{t}{\max_{1\leq i\leq N}\|\chi_i\|_{\psi_1}}\right)\right),$$

where $\tilde{c}$ is a positive constant.

Applying Lemma E.2 to the sub-exponential random variables $e_i^2 - \mathbb{E}[e_i^2]$ gives

$$\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n} e_i^2 - \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[e_i^2\right] \geq t\right\} \leq \exp\left(-\tilde{c}n\min\left(\frac{N^2t^2}{c'^2\rho^4(b-a)^4}, \frac{Nt}{c'\rho^2(b-a)^2}\right)\right). \tag{10}$$

Setting $c_0 = \tilde{c}/c'^2$, when n is sufficiently large, picking

$$t = \sqrt{\frac{c}{c_0}}\frac{\rho^2(b-a)^2}{\sqrt{n}N}$$

in (10) guarantees that for all $c > 0$, with probability exceeding $1 - \exp(-c)$:

$$\|e\|_2^2 \leq \sum_{i=1}^{n}\mathbb{E}\left[e_i^2\right] + \sqrt{\frac{c}{c_0}}\frac{\rho^2(b-a)^2\sqrt{n}}{N},$$

or equivalently

$$\|e\|_2 \leq \sqrt{\sum_{i=1}^{n}\mathbb{E}\left[e_i^2\right]} + \left(\frac{c}{c_0}\right)^{1/4}\frac{\rho(b-a)n^{1/4}}{\sqrt{N}}. \tag{11}$$

We now proceed by bounding $\mathbb{E}[e_i^2]$ as follows:

$$\begin{aligned}\sum_{i=1}^{n}\mathbb{E}[e_i^2] &= \sum_{i=1}^{n}\mathbb{E}\left[\left(f\left(\frac{1}{N}\sum_{j=1}^{N}x_{ij}\right)-f\left(\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}[x_{ij}]\right)\right)^2\right]\\ &\le \rho^2\sum_{i=1}^{n}\mathbb{E}\left[\left(\frac{1}{N}\sum_{j=1}^{N}x_{ij}-\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}[x_{ij}]\right)^2\right]\\ &= \frac{\rho^2}{N^2}\sum_{i=1}^{n}\sum_{j=1}^{N}\mathrm{var}[x_{ij}]\\ &\le \frac{\rho^2(b-a)^2n}{4N}, \end{aligned} \tag{12}$$

where the first inequality is thanks to the Lipschitz continuity of f, and the second inequality is a direct result of the Popoviciu's inequality which states that for a random variable x, bounded between a and b, $\mathrm{var}[x] \le (b-a)^2/4$. Combining (12) and (11) proves that with probability exceeding $1-\exp(-c)$:

$$\|e\|_2 \le \frac{\rho(b-a)\sqrt{n}}{2\sqrt{N}} + \left(\frac{c}{c_0}\right)^{1/4}\frac{\rho(b-a)n^{1/4}}{\sqrt{N}} = \mathcal{O}\left(\rho(b-a)\sqrt{\frac{n}{N}}\right).$$

□

Proof of part (b). Since $\mathcal{X}$ offers independence along the columns, setting $y_i = N^{-1}\sum_{j=1}^{N}x_{ij}$, and appealing to the Hoeffding's inequality in Lemma E.1 yields

$$\forall t>0: \qquad \mathbb{P}\left\{|y_i-\mathbb{E}[y_i]|\ge t\right\} \le 2\exp\left(-\frac{2Nt^2}{(b-a)^2}\right).$$

A direct implication of the Lipschitz continuity of f is that if $|f(z_1)-f(z_2)|\ge t$, then $|z_1-z_2|\ge t/\rho$, which yields

$$\begin{aligned}\mathbb{P}\left\{e_i^2\ge\frac{t^2}{n}\right\} = \mathbb{P}\left\{|f(y_i)-f\left(\mathbb{E}[y_i]\right)|\ge\frac{t}{\sqrt{n}}\right\} &\le \mathbb{P}\left\{|y_i-\mathbb{E}[y_i]|\ge\frac{t}{\rho\sqrt{n}}\right\}\\ &\le 2\exp\left(-\frac{2Nt^2}{n\rho^2(b-a)^2}\right).\end{aligned} \tag{13}$$

For arbitrary real-valued random variables $\chi_1,\dots\chi_n$ and scalars $t_1,\dots,t_n$, one has

$$\begin{aligned}\mathbb{P}\left\{\sum_{i=1}^{n}\chi_i\ge\sum_{i=1}^{n}t_i\right\} &\le \mathbb{P}\left\{\bigcup_{i=1}^{n}\left\{\chi_i\ge t_i\right\}\right\}\\ &\le \sum_{i=1}^{N}\mathbb{P}\left\{\chi_i\ge t_i\right\},\end{aligned} \tag{14}$$

where the first inequality is valid since

$$\left\{(\chi_1,\dots,\chi_n):\sum_{i=1}^{n}\chi_i\ge\sum_{i=1}^{n}t_i\right\}\subseteq\bigcup_{i=1}^{n}\left\{\chi_i:\chi_i\ge t_i\right\},$$

and the second inequality is thanks to the union bound. An application of (14) to (13) yields

$$\begin{aligned}\mathbb{P}\left\{\|e\|_2\ge t\right\} = \mathbb{P}\left\{\|e\|_2^2\ge t^2\right\} &\le 2n\exp\left(-\frac{2Nt^2}{n\rho^2(b-a)^2}\right)\\ &= \exp\left(\log(2n)-\frac{2Nt^2}{n\rho^2(b-a)^2}\right),\end{aligned}$$

which means for all $c>0$, with probability exceeding $1-\exp(-c)$:

$$\|e\|_2 \le \sqrt{\frac{(c+\log(2n))n\rho^2(b-a)^2}{2N}} = \mathcal{O}\left(\rho(b-a)\sqrt{\frac{n\log n}{N}}\right).$$

□

F Proof of Theorem 4.2

Theorem 4.2. *Consider a TASU layer with output $y = \texttt{TASU}_\kappa(Wx)$, where $x \in \mathbb{S}^{p-1}$ and $W \in \mathbb{R}^{n\times p}$ has normalized rows $w_i^\top$ ($\|w_i\|_2 = 1$ for $i = 1, \dots, n$). Define the embedded layer output as $\tilde{y} = \texttt{sign}\left(\texttt{sign}(W\mathcal{G}^\top)\,\texttt{sign}(\mathcal{G}x)\right)$, where $\mathcal{G} \in \mathbb{R}^{N\times p}$ is a standard Gaussian matrix. Assume*

$$\left|w_i^\top x\right| \geq \ell_{\min} > 0, \qquad i = 1, \dots, n.$$

Pick a target discrepancy $\varepsilon \leq \sqrt{n}$ and set $\kappa \geq \frac{\pi}{2\ell_{\min}} \log \frac{4\sqrt{n}}{\varepsilon}$. Then, for any scalar $c > 0$, selecting $N \geq 8(c+\log 2n)n\kappa^2/\varepsilon^2$ guarantees that with probability exceeding $1-3\exp(-c)$: $\|y - \tilde{y}\|_2 \leq \varepsilon$.

Proof. Recall that

$$y = \texttt{TASU}_\kappa(Wx), \qquad \text{and} \qquad \tilde{y} = \texttt{sign}\left(\texttt{sign}(W\mathcal{G}^\top)\texttt{sign}(\mathcal{G}x)\right).$$

Denoting the discrepancy between the two layers as $e = \tilde{y} - y$, by triangle inequality we have

$$\|e\| = \|e'\|_2 + \|e''\|_2,$$

where

$$\begin{aligned} e_i' &= \tanh\left(\frac{\kappa}{N}\sum_{j=1}^N \operatorname{sign}\left(g_j^\top w_i\right)\operatorname{sign}\left(g_j{}^\top x\right)\right) - \tanh\left(\frac{2\kappa}{\pi}\arcsin\left(w_i^\top x\right)\right) \\ &= \tanh\left(\frac{\kappa}{N}\sum_{j=1}^N \operatorname{sign}\left(g_j^\top w_i\right)\operatorname{sign}\left(g_j{}^\top x\right)\right) - \tanh\left(\frac{\kappa}{N}\mathbb{E}\left[\sum_{j=1}^N \operatorname{sign}\left(g_j^\top w_i\right)\operatorname{sign}\left(g_j{}^\top x\right)\right]\right), \end{aligned}$$

and

$$e_i'' = \operatorname{sign}\left(\frac{1}{N}\sum_{j=1}^N \operatorname{sign}\left(g_j^\top w_i\right)\operatorname{sign}\left(g_j{}^\top x\right)\right) - \tanh\left(\frac{\kappa}{N}\sum_{j=1}^N \operatorname{sign}\left(g_j^\top w_i\right)\operatorname{sign}\left(g_j{}^\top x\right)\right).$$

The first term $\|e'\|_2$ can be related to Theorem E.2(b) with $f(x) = \tanh(\kappa x)$ as the Lipschitz function, for which it is easy to verify that $\rho = \kappa$. Therefore, with probability exceeding $1 - \exp(-c)$:

$$\|e'\|_2 \leq \sqrt{\frac{2(c+\log(2n))n\kappa^2}{N}}. \tag{15}$$

For the second term, using the fact

$$|\operatorname{sign}(z) - \tanh(\kappa z)| \leq 2\exp(-2\kappa|z|),$$

yields

$$|e_i''| \leq 2\exp\left(-2\kappa\left|\frac{1}{N}\sum_{j=1}^N \operatorname{sign}\left(g_j^\top w_i\right)\operatorname{sign}\left(g_j{}^\top x\right)\right|\right),$$

bounding which requires finding a lower bound for $|\frac{1}{N}\sum_{j=1}^N \operatorname{sign}\left(g_j^\top w_i\right)\operatorname{sign}\left(g_j{}^\top x\right)|$. Notice that for general u and v, the inequality $|v| - |u| > t$ implies $|u - v| > t$. Therefore, using Hoeffding's inequality

$$\begin{aligned} &\mathbb{P}\left\{\frac{2}{\pi}\left|\arcsin(w_i^\top x)\right| - \left|\frac{1}{N}\sum_{j=1}^N \operatorname{sign}\left(g_j^\top w_i\right)\operatorname{sign}\left(g_j{}^\top x\right)\right| \geq t\right\} \\ &\leq \mathbb{P}\left\{\left|\frac{1}{N}\sum_{j=1}^N \operatorname{sign}\left(g_j^\top w_i\right)\operatorname{sign}\left(g_j{}^\top x\right) - \frac{2}{\pi}\arcsin(w_i^\top x)\right| \geq t\right\} \leq 2\exp\left(-\frac{Nt^2}{2}\right), \end{aligned}$$

which implies that with probability exceeding $1-2\exp(-c)$:

$$\begin{aligned}\left|\frac{1}{N}\sum_{j=1}^{N}\operatorname{sign}\left(g_j^\top w_i\right)\operatorname{sign}\left(g_j{}^\top x\right)\right| &> \frac{2}{\pi}\left|\arcsin(w_i^\top x)\right| - \frac{\sqrt{2c}}{\sqrt{N}} \\ &\geq \frac{2}{\pi}\left|w_i^\top x\right| - \frac{\sqrt{2c}}{\sqrt{N}} \\ &\geq \frac{2}{\pi}\ell_{\min} - \frac{\sqrt{2c}}{\sqrt{N}}.\end{aligned}$$

This indicates that with probability exceeding $1-2\exp(-c)$:

$$|e_i''|^2 \leq 4\exp\left(-\frac{8\kappa}{\pi}\ell_{\min} + \kappa\sqrt{\frac{32c}{N}}\right),$$

which, after applying a union bound analogous to (14), implies that with probability at least $1-2n\exp(-c)$:

$$\|e''\|_2^2 \leq 4n\exp\left(-\frac{8\kappa}{\pi}\ell_{\min} + \kappa\sqrt{\frac{32c}{N}}\right),$$

or equivalently, with probability exceeding $1-2\exp(-c)$:

$$\|e''\|_2 \leq 2\exp\left(\log\sqrt{n} - \frac{4\kappa}{\pi}\ell_{\min} + \kappa\sqrt{\frac{8(c+\log n)}{N}}\right).$$

After combining this result with (15) we can claim that with probability exceeding $1-3\exp(-c)$:

$$\|e\|_2 \leq \sqrt{\frac{2(c+\log(2n))n\kappa^2}{N}} + 2\exp\left(\log\sqrt{n} - \frac{4\kappa}{\pi}\ell_{\min} + \kappa\sqrt{\frac{8(c+\log n)}{N}}\right).$$

Setting $N = 8(c+\log 2n)n\kappa^2/\varepsilon^2$ gives

$$\|e\|_2 \leq \frac{\varepsilon}{2} + 2\exp\left(\log\sqrt{n} - \frac{4\kappa}{\pi}\ell_{\min} + \frac{\varepsilon}{\sqrt{n}}\right). \tag{16}$$

Notice that

$$\frac{4\kappa}{\pi}\ell_{\min} \geq 2\log\frac{4\sqrt{n}}{\varepsilon} \geq \log\sqrt{n} + \frac{\varepsilon}{\sqrt{n}} - \log\frac{\varepsilon}{4},$$

where the first inequality is thanks to the assumption on κ and the second inequality is valid as long as $n \geq \varepsilon^2$. Together with (15) and (16), this implies that with probability exceeding $1-3\exp(-c)$:

$$\|e\|_2 \leq \frac{\varepsilon}{2} + 2\exp\left(\log\frac{\varepsilon}{4}\right) = \varepsilon.$$

□

G Proof of Theorem 4.3

Theorem 4.3. *Consider the cascade of L G-Net layers and the corresponding hyperdimensional embedding as (6). Assume that each layer ℓ of the network is consistent with near isometry of* ASU *in $\mathbb{S}^{n_{\ell-1}-1}$ with parameters β_ℓ and ε_ℓ, as stated in (7). Then with probability exceeding $1-L\exp(-c)$:*

$$\|\delta_L\|_2 \leq c'\sum_{\ell=1}^{L}\left(\sqrt{\frac{(c+\log n_\ell)n_\ell}{N_\ell\beta_\ell\left(1-\varepsilon_\ell\right)}} + \sqrt{\frac{\varepsilon_\ell}{1+\varepsilon_\ell}}\right),$$

where c' is an absolute constant.

G.1 Auxiliary Lemmas

To prove Theorem 4.3, we first need to state some auxiliary lemmas, which follow.

Lemma G.1. *If (7) holds then then for all $x, y \in \mathcal{D}$:*

$$x^\top y - \frac{3}{2}\varepsilon \leq \beta^{-1} \left\langle \texttt{ASU}(Wx), \texttt{ASU}(Wy) \right\rangle \leq x^\top y + \frac{3}{2}\varepsilon, \tag{17}$$

Proof. By the left-hand side of (7) we have

$$\begin{aligned}\|x\|^2 + \|y\|^2 - 2x^\top y - \varepsilon &\leq \beta^{-1} \left\|\texttt{ASU}(Wx)\right\|^2 + \beta^{-1} \left\|\texttt{ASU}(Wy)\right\|^2 - 2\beta^{-1} \left\langle \texttt{ASU}(Wx), \texttt{ASU}(Wy) \right\rangle \\ &\leq \|x\|^2 + \varepsilon + \|y\|^2 + \varepsilon - 2\beta^{-1} \left\langle \texttt{ASU}(Wx), \texttt{ASU}(Wy) \right\rangle,\end{aligned}$$

where in the second inequality we used $\beta^{-1}\|\texttt{ASU}(Wu)\|_2^2 \leq \|u\|_2^2 + \varepsilon$ twice, once by setting u to x and once to y. This implies that

$$\beta^{-1} \left\langle \texttt{ASU}(Wx), \texttt{ASU}(Wy) \right\rangle \leq x^\top y + \frac{3}{2}\varepsilon.$$

Also by the second side of the inequality we have

$$\begin{aligned}\|x\|^2 + \|y\|^2 - 2x^\top y + \varepsilon &\geq \beta^{-1} \left\|\texttt{ASU}(Wx)\right\|^2 + \beta^{-1} \left\|\texttt{ASU}(Wy)\right\|^2 - 2\beta^{-1} \left\langle \texttt{ASU}(Wx), \texttt{ASU}(Wy) \right\rangle \\ &\geq \|x\|^2 - \varepsilon + \|y\|^2 - \varepsilon - 2\beta^{-1} \left\langle \texttt{ASU}(Wx), \texttt{ASU}(Wy) \right\rangle,\end{aligned}$$

which implies that

$$\beta^{-1} \left\langle \texttt{ASU}(Wx), \texttt{ASU}(Wy) \right\rangle \geq x^\top y - \frac{3}{2}\varepsilon.$$

□

Lemma G.2. *Consider two arbitrary non-zero vectors y and $\tilde{y} \in \mathbb{R}^n$. If $\|\tilde{y} - y\|_2 \leq \varepsilon$ and $\|y\|_2 \geq \kappa > 0$, then*

$$\left\| \frac{\tilde{y}}{\|\tilde{y}\|_2} - \frac{y}{\|y\|_2} \right\|_2 \leq \frac{2\varepsilon}{\kappa}.$$

Proof. A direct implication of the assumptions is

$$\left\| \frac{\tilde{y}}{\|y\|_2} - \frac{y}{\|y\|_2} \right\|_2 \leq \frac{\varepsilon}{\kappa}. \tag{18}$$

On the other hand

$$\left\| \frac{\tilde{y}}{\|\tilde{y}\|_2} - \frac{\tilde{y}}{\|y\|_2} \right\|_2 = \left| \frac{1}{\|\tilde{y}\|_2} - \frac{1}{\|y\|_2} \right| \|\tilde{y}\|_2 = \frac{|\|y\|_2 - \|\tilde{y}\|_2|}{\|y\|_2} \leq \frac{\|y - \tilde{y}\|_2}{\|y\|_2} \leq \frac{\varepsilon}{\kappa}, \tag{19}$$

where we used the reverse triangle inequality. Now using the triangle inequality along with (18) and (19) we get

$$\left\| \frac{\tilde{y}}{\|\tilde{y}\|_2} - \frac{y}{\|y\|_2} \right\|_2 \leq \left\| \frac{\tilde{y}}{\|\tilde{y}\|_2} - \frac{\tilde{y}}{\|y\|_2} \right\|_2 + \left\| \frac{\tilde{y}}{\|y\|_2} - \frac{y}{\|y\|_2} \right\|_2 \leq \frac{2\varepsilon}{\kappa},$$

which completes the proof. □

Lemma G.3. *Consider $W \in \mathbb{R}^{n \times p}$ which is consistent with near isometry of $\texttt{ASU}$ in $\mathbb{S}^{p-1}$, as stated in (7). Then for all $x, y \in \mathbb{S}^{p-1}$:*

$$\left\| \frac{\texttt{ASU}(Wx)}{\|\texttt{ASU}(Wx)\|} - \frac{\texttt{ASU}(Wy)}{\|\texttt{ASU}(Wy)\|} \right\|_2^2 \leq \|x - y\|_2^2 + \frac{5\varepsilon}{1 + \varepsilon}. \tag{20}$$

Proof. Expanding the left-hand side of (20) gives

$$\begin{aligned}
\left\| \frac{\texttt{ASU}(Wx)}{\|\texttt{ASU}(Wx)\|} - \frac{\texttt{ASU}(Wy)}{\|\texttt{ASU}(Wy)\|} \right\|_2^2 &= \left\| \frac{\texttt{ASU}(Wx)}{\|\texttt{ASU}(Wx)\|} \right\|_2^2 + \left\| \frac{\texttt{ASU}(Wy)}{\|\texttt{ASU}(Wy)\|} \right\|_2^2 - \frac{2\left\langle \texttt{ASU}(Wx), \texttt{ASU}(Wy) \right\rangle}{\|\texttt{ASU}(Wx)\|_2 \|\texttt{ASU}(Wy)\|_2} \\
&= 2 - \frac{2\beta^{-1}\left\langle \texttt{ASU}(Wx), \texttt{ASU}(Wy) \right\rangle}{\beta^{-1/2}\|\texttt{ASU}(Wx)\|_2 \beta^{-1/2}\|\texttt{ASU}(Wy)\|_2} \\
&\leq 2 - 2\frac{x^\top y - \frac{3}{2}\varepsilon}{1+\varepsilon} \\
&= \|x-y\|_2^2 + \frac{\varepsilon}{1+\varepsilon}\left(3 + 2x^\top y\right) \\
&\leq \|x-y\|_2^2 + \frac{5\varepsilon}{1+\varepsilon},
\end{aligned}$$

where the first inequality is thanks to Lemma G.1. This completes the proof. □

G.2 Proof of the Main Theorem

Proof. We are now ready to prove Theorem 4.3. By the triangle inequality we have

$$\|\delta_\ell\|_2 \leq \left\| \frac{\tilde{y}_\ell}{\|\tilde{y}_\ell\|_2} - \frac{\texttt{RASU}\left(W_\ell \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|}\right)}{\left\|\texttt{RASU}\left(W_\ell \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|}\right)\right\|_2} \right\|_2 + \left\| \frac{\texttt{RASU}\left(W_\ell \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|}\right)}{\left\|\texttt{RASU}\left(W_\ell \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|}\right)\right\|_2} - \frac{y_\ell}{\|y_\ell\|_2} \right\|_2. \tag{21}$$

We start bounding $\|\delta_\ell\|_2$ by first bounding the first term on the right-hand side of (21). By the layer consistency we know that given $\tilde{y}_{\ell-1}$, with probability exceeding $1-p_\ell$ (where as shown in Theorem 4.1, $p_\ell = \exp(-c)$ for some desired $c > 0$):

$$\left\| \frac{1}{N_\ell}\tilde{y}_\ell - \texttt{RASU}\left(W_\ell \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|}\right) \right\|_2 = \mathcal{O}\left(\sqrt{\frac{(c+\log n_\ell)n_\ell}{N_\ell}}\right). \tag{22}$$

By the near isometry consistency of the ℓ-th layer we have

$$\left\|\texttt{RASU}\left(W_\ell \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|}\right)\right\|_2 \geq \sqrt{\beta_\ell(1-\varepsilon_\ell)}. \tag{23}$$

Using Lemma G.2, the bounds (22) and (23) imply that given $\tilde{y}_{\ell-1}$, with probability exceeding $1-p_\ell$:

$$\left\| \frac{\tilde{y}_\ell}{\|\tilde{y}_\ell\|_2} - \frac{\texttt{RASU}\left(W_\ell \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|}\right)}{\left\|\texttt{RASU}\left(W_\ell \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|}\right)\right\|_2} \right\|_2 = \mathcal{O}\left(\sqrt{\frac{(c+\log n_\ell)n_\ell}{N_\ell \beta_\ell(1-\varepsilon_\ell)}}\right). \tag{24}$$

The second term on the right-hand side of (21) can be bounded using Lemma G.3 as

$$\begin{aligned}
\left\| \frac{\texttt{RASU}\left(W_\ell \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|}\right)}{\left\|\texttt{RASU}\left(W_\ell \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|}\right)\right\|_2} - \frac{y_\ell}{\|y_\ell\|_2} \right\|_2^2 &= \left\| \frac{\texttt{RASU}\left(W_\ell \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|}\right)}{\left\|\texttt{RASU}\left(W_\ell \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|}\right)\right\|_2} - \frac{\texttt{RASU}\left(W_\ell \frac{y_{\ell-1}}{\|y_{\ell-1}\|}\right)}{\left\|\texttt{RASU}\left(W_\ell \frac{y_{\ell-1}}{\|y_{\ell-1}\|}\right)\right\|_2} \right\|_2^2 \\
&\leq \left\| \frac{\tilde{y}_{\ell-1}}{\|\tilde{y}_{\ell-1}\|_2} - \frac{y_{\ell-1}}{\|y_{\ell-1}\|_2} \right\|_2^2 + \frac{5\varepsilon_\ell}{1+\varepsilon_\ell} \\
&= \|\delta_{\ell-1}\|_2^2 + \frac{5\varepsilon_\ell}{1+\varepsilon_\ell}.
\end{aligned} \tag{25}$$

Using (25) and (24) in (21) we conclude that given $\tilde{y}_{\ell-1}$, with probability exceeding $1-p_\ell$,

$$\|\delta_\ell\|_2 \leq \|\delta_{\ell-1}\|_2 + c'\left(\sqrt{\frac{(c+\log n_\ell)n_\ell}{N_\ell\beta_\ell(1-\varepsilon_\ell)}} + \sqrt{\frac{\varepsilon_\ell}{1+\varepsilon_\ell}}\right), \qquad \ell = 1, \ldots, L, \tag{26}$$

where c' is an absolute constant and $\delta_0 = 0$. Now consider A_ℓ to be the event that (22) holds. Then we have

$$\begin{aligned}\mathbb{P}\{A_L\} &\geq \mathbb{P}\{A_{L-1}\}\,\mathbb{P}\{A_L|A_{L-1}\}\\ &\geq \mathbb{P}\{A_{L-2}\}\,\mathbb{P}\{A_L|A_{L-1}\}\,\mathbb{P}\{A_{L-1}|A_{L-2}\}\\ &\vdots\\ &\geq \mathbb{P}\{A_1\}\prod_{\ell=2}^{L}\mathbb{P}\{A_L|A_{L-1}\}\\ &= \prod_{\ell=1}^{L}(1-p_\ell).\end{aligned}$$

By the Weierstrass inequality we know if $p_\ell \in [0,1]$ for $\ell = 1,\ldots,L$, then

$$\prod_{\ell=1}^{L}(1-p_\ell) \geq 1 - \sum_{\ell=1}^{L} p_\ell.$$

Together with (26), this implies that with probability exceeding $1 - L\exp(-c)$:

$$\|\delta_L\|_2 \leq c' \sum_{\ell=1}^{L}\left(\sqrt{\frac{(c+\log n_\ell)n_\ell}{N_\ell\beta_\ell\,(1-\varepsilon_\ell)}} + \sqrt{\frac{\varepsilon_\ell}{1+\varepsilon_\ell}}\right), \tag{27}$$

where c' is an absolute constant. □

H Proof of Theorem 4.4

Theorem 4.4. *Consider $\mathcal{W} \in \mathbb{R}^{n\times p}$, where $n \gtrsim p \geq 27$, following the construction format in (8). Let $g_i \sim \mathcal{N}(0, I_p)$ be independent standard normal vectors. Then, for all $x, y \in \mathbb{B}_2^p$, with probability exceeding $1 - c\exp(-c'p)$:*

$$\|x-y\|_2^2 - \varepsilon_{n,p}^l \leq \beta_{n,p}^{-1}\,\|\texttt{ASU}(\mathcal{W}x) - \texttt{ASU}(\mathcal{W}y)\|^2 \leq \|x-y\|_2^2 + \varepsilon_{n,p}^u,$$

where

$$\beta_{n,p}^{-1} = \frac{\pi^2 p}{4\left(\sqrt{n}+\sqrt{p}\right)^2},\quad \varepsilon_{n,p}^l = c_l\frac{\sqrt{p}}{\sqrt{n}+\sqrt{p}},\quad \varepsilon_{n,p}^u = c_u\left(\frac{\sqrt{p}}{\sqrt{n}+\sqrt{p}} + \frac{n+p^2}{p\left(\sqrt{n}+\sqrt{p}\right)^2}\right),$$

and c, c', c_l and c_u are absolute numerical constants.

Proof. We present the proof as four main steps and for each step provide the related lemmas which are all proved at the end of this section.

– Step 1. Bounding the ASU Variations: Using basic calculus, one could establish the following inequality:

Lemma H.1. *Consider $x, y \in [-1,1]$. Then*

$$\begin{aligned}\frac{4}{\pi^2}(x-y)^2 \leq \left(\texttt{ASU}(x) - \texttt{ASU}(y)\right)^2 &\leq \frac{4}{\pi^2}(x-y)^2 + 2\left(1-\frac{4}{\pi^2}\right)\left(x^4+y^4\right)\\ &\leq \frac{4}{\pi^2}(x-y)^2 + \frac{6}{5}\left(x^4+y^4\right).\end{aligned}$$

Proof. See Section H.1. □

By Lemma H.1, for each row of $\mathcal{W}$ we have

$$\frac{4}{\pi^2}\left(\mathtt{w}_i^\top x - \mathtt{w}_i^\top y\right)^2 \le \left(\mathtt{ASU}(\mathtt{w}_i^\top x) - \mathtt{ASU}(\mathtt{w}_i^\top y)\right)^2 \le \frac{4}{\pi^2}(\mathtt{w}_i^\top x - \mathtt{w}_i^\top y)^2 + \frac{6}{5}\left(\left(\mathtt{w}_i^\top x\right)^4 + \left(\mathtt{w}_i^\top y\right)^4\right).$$

Summing along the rows and scaling both sides gives

$$0 \le \frac{\pi^2}{4}\left\|\mathtt{ASU}(\mathcal{W}x) - \mathtt{ASU}(\mathcal{W}y)\right\|^2 - \left\|\mathcal{W}(x-y)\right\|^2 \le \frac{3\pi^2}{10}\left(\|\mathcal{W}x\|_4^4 + \|\mathcal{W}y\|_4^4\right). \tag{28}$$

Consider $s_{\min}(\mathcal{W})$ and $s_{\max}(\mathcal{W})$ to denote the smallest and largest singular values of $\mathcal{W}$. Then for any x and y:

$$s_{\min}^2(\mathcal{W})\|x-y\|_2^2 \le \left\|\mathcal{W}(x-y)\right\|^2 \le s_{\max}^2(\mathcal{W})\|x-y\|_2^2. \tag{29}$$

On the other hand, using the notion of induced norm, for any $x \in \mathbb{B}_2^p$ one has

$$\|\mathcal{W}x\|_4^4 \le \|\mathcal{W}\|_{2\to 4}^4, \tag{30}$$

where

$$\|\mathcal{W}\|_{2\to 4} = \sup_{x:\|x\|_2\le 1} \|\mathcal{W}x\|_4.$$

Using (30) and (29) in (28) we get the following general bound for all $x, y \in \mathbb{B}_2^p$:

$$s_{\min}^2(\mathcal{W})\|x-y\|_2^2 \le \frac{\pi^2}{4}\left\|\mathtt{ASU}(\mathcal{W}x) - \mathtt{ASU}(\mathcal{W}y)\right\|^2 \le s_{\max}^2(\mathcal{W})\|x-y\|_2^2 + \frac{6\pi^2}{10}\|\mathcal{W}\|_{2\to 4}^4. \tag{31}$$

The next step is using high probability bounds for $s_{\min}(\mathcal{W})$, $s_{\max}(\mathcal{W})$ and $\|\mathcal{W}\|_{2\to 4}$ to express them in terms of the problem dimensions n and p.

– Step 2. High Probability Bounds for $s_{\min}(\mathcal{W})$ and $s_{\max}(\mathcal{W})$: In this step we will focus on bounding the singular values of $\mathcal{W}$ in (8), where $g_i \sim \mathcal{N}(0, I_p)$ are independent standard normal vectors. A central component of the proofs of this step and the next step is the following lemma, which characterizes the moment generating functions of rows of $\mathcal{W}$. Since parts (a) and (b) are closely related, they are consolidated as one result, while only part (a) is used in this step.

Lemma H.2. *Consider $x \in \mathbb{S}^{p-1}$ where $p \ge 27$, and $g = [g_1, \ldots, g_p]^\top \sim \mathcal{N}(0, I)$. Define the random variable*

$$u = \frac{\left|x^\top g\right|}{\|g\|}.$$

Then

(a) For all $\tau \in [0, 2/5]$:

$$\mathbb{E}\exp\left(\tau^2 p u^2\right) \le \exp\left(\frac{25}{4}\tau^2\right).$$

(b) For all $\tau \in [0, \log(2)/6]$:

$$\mathbb{E}\exp\left(\tau p u^4\right) \le \exp\left(\frac{13\tau}{p}\right).$$

Proof. See Section H.2. □

As a well-known result, a random variable $\mathfrak{z}$ is sub-Gaussian if there exists a constant $k > 0$ such that for all τ satisfying $|\tau| \le k^{-1}$, one has $\mathbb{E}\exp\left(\tau^2\mathfrak{z}\right) \le \exp\left(k^2\tau^2\right)$ (e.g., see Proposition 2.5.2 of [34]). Defining $u = \left|x^\top g\right|/\|g\|$, it is immediate from part (a) of Lemma H.2 that $\sqrt{p}u$ is a sub-Gaussian random variable. Furthermore, since Lemma H.2 considers $x \in \mathbb{S}^{p-1}$ as any arbitrary unit vector on the sphere, this guarantees that the normalized vector $g/\|g\|$ is generally a sub-Gaussian random vector. Notably, $g/\|g\|$ follows a uniform distribution on the unit sphere, which is guaranteed to follow a sub-Gaussian distribution (see Theorem 3.4.6 of [34]). However, in our analysis, we provide

a more detailed characterization of this property with explicit constants, which are later needed in part (b) of Lemma H.2 and the analysis in step 3.

The sub-Gaussian property of $\sqrt{p}u$ allows using the following uniform result:

Lemma H.3 (Theorem 4.6.1 of [34]). *Let $\mathcal{S}$ be an $n \times p$ matrix whose rows $s_i^\top$ are i.i.d, zero-mean, sub-gaussian isotropic random vectors in $\mathbb{R}^p$. Let $\kappa = \|s_i\|_{\psi_2}$. Then for any $t \geq 0$ and all $x \in \mathbb{S}^{p-1}$:*

$$\sqrt{n} - c\kappa^2 \left(\sqrt{p} + t\right) \leq \|\mathcal{S}x\|_2 \leq \sqrt{n} + c\kappa^2 \left(\sqrt{p} + t\right),$$

with probability at least $1 - 2\exp\left(-t^2\right)$.

Lemma H.3 provides an immediate way of uniformly bounding the singular values of $\sqrt{p}\mathcal{W}$. To that end, notice that for $g \sim \mathcal{N}(0, I_p)$:

$$\mathbb{E}\left[\frac{gg^\top}{\|g\|^2}\right] = \frac{1}{p} I,$$

which can be easily verified by symmetry for diagonal elements and negative symmetry for off-diagonal elements. This observation makes the rows of $\sqrt{p}\mathcal{W}$ isotropic. Moreover, part (a) of Lemma H.2 guarantees that the sub-Gaussian norm of $\sqrt{p}u$ is a constant, i.e., $\|\sqrt{p}u\|_{\psi_2} = \mathcal{O}(1)$. Now appealing to Lemma H.3 and setting $t = c\sqrt{p}$ guarantees that with probability exceeding $1 - 2\exp(-cp)$, for all $x \in \mathbb{S}^{p-1}$:

$$\sqrt{n} - c'\sqrt{p} \leq \|\sqrt{p}\mathcal{W}x\|_2 \leq \sqrt{n} + c'\sqrt{p},$$

or equivalently, with probability exceeding $1 - 2\exp(-cp)$:

$$\frac{\sqrt{n} - c'\sqrt{p}}{\sqrt{p}} \leq s_{\min}(\mathcal{W}) \leq s_{\max}(\mathcal{W}) \leq \frac{\sqrt{n} + c'\sqrt{p}}{\sqrt{p}}. \tag{32}$$

– Step 3. High Probability Bounds for $\|\mathcal{W}\|_{2\to 4}$: In this section we focus on bounding $\|\mathcal{W}\|_{2\to 4}$ using a covering argument. While the induced norm of random matrices with bounded and independent values has been studied in the literature [42, 49, 55], to the best of our knowledge, no such result is available for random matrices like (8) with dependence across the columns. Here using part (b) of Lemma H.2, we will provide concentration bounds on $\|\mathcal{W}\|_{2\to 4}$.

Consider a random matrix $\mathcal{W} \in \mathbb{R}^{n\times p}$ following the construction in (8). For a given $x \in \mathbb{S}^{p-1}$ we have

$$\|\mathcal{W}x\|_4^4 = \sum_{i=1}^{n} \frac{|g_i^\top x|^4}{\|g_i\|^4}.$$

For a fixed τ within the designated region of Lemma H.2 we have

$$\mathbb{E}\exp\left(\tau p\|\mathcal{W}x\|_4^4\right) \leq \exp\left(\frac{13\tau n}{p}\right).$$

By the Markov's inequality for a non-negative random variable z and all $t > 0$: $\mathbb{P}\{z \geq t\} \leq \mathbb{E}z/t$. Fixing $\tau = \log(2)/6$ and applying the Markov's inequality to $z = \exp\left(\tau p\|\mathcal{W}x\|_4^4\right)$ with $t = \exp(13\tau n/p + \gamma p)$ guarantees that for any $\gamma > 0$:

$$\mathbb{P}\left\{\tau p\|\mathcal{W}x\|_4^4 \geq \frac{13\tau n}{p} + \gamma p\right\} = \mathbb{P}\left\{p^2\|\mathcal{W}x\|_4^4 \geq 13n + \frac{6}{\log 2}\gamma p^2\right\} \leq \exp(-\gamma p). \tag{33}$$

We now proceed by applying a union bound to the tale bound above using a covering argument. The following standard result (e.g., see Corollary 4.2.13 of [34]) bounds the size of an ε-net on $\mathbb{S}^{p-1}$:

Lemma H.4. *The covering number of the unit sphere* $\mathbb{S}^{p-1} = \{x \in \mathbb{R}^p : \|x\|_2 = 1\}$ *satisfies the following for any* $\varepsilon \in (0, 1]$*:*

$$N(\mathbb{S}^{p-1}, \varepsilon) \leq \left(\frac{2}{\varepsilon} + 1\right)^p,$$

where $N(\mathbb{S}^{p-1}, \varepsilon)$ *represents an* ε*-net of* $\mathbb{S}^{p-1}$*, i.e.,*

$$\forall x \in \mathbb{S}^{p-1},\ \exists x' \in N(\mathbb{S}^{p-1}, \varepsilon):\ \ \|x - x'\| \leq \varepsilon.$$

We next relate $\|\mathcal{W}\|_{2\to 4}$ to a version of it computed on $N(\mathbb{S}^{p-1}, \varepsilon)$ by fixing $\varepsilon \in (0, 1]$ and using a standard covering argument for the norms. Let $x \in \mathbb{S}^{p-1}$ be a vector such that $\|\mathcal{W}x\|_4 = \|\mathcal{W}\|_{2\to 4}$. By the triangle inequality for any $x' \in N(\mathbb{S}^{p-1}, \varepsilon)$ we have:

$$\|\mathcal{W}x'\|_4 \geq \|\mathcal{W}x\|_4 - \|\mathcal{W}(x - x')\|_4 \geq \|\mathcal{W}\|_{2\to 4} - \|\mathcal{W}\|_{2\to 4}\|x - x'\|_2 \geq (1 - \varepsilon)\|\mathcal{W}\|_{2\to 4}.$$

After fixing $\varepsilon = 1/2$, dividing both sides of this inequality by $1 - \varepsilon$ and taking a maximum with respect to x' we get

$$\|\mathcal{W}\|_{2\to 4} \leq \frac{1}{1-\varepsilon} \sup_{x \in N(\mathbb{S}^{p-1}, \varepsilon)} \|\mathcal{W}x\|_4 = 2 \sup_{x \in N(\mathbb{S}^{p-1}, 1/2)} \|\mathcal{W}x\|_4. \tag{34}$$

Note that by Lemma H.4 the cardinality of the net can be bounded by $|N(\mathbb{S}^{p-1}, 1/2)| \leq 5^p$, which admits applying a union bound as follows:

$$\begin{aligned}
\mathbb{P}\left\{\sqrt{p}\|\mathcal{W}\|_{2\to 4} \geq 2\left(13n + \frac{6}{\log 2}\gamma p^2\right)^{1/4}\right\} &\leq \mathbb{P}\left\{\sqrt{p} \sup_{x \in N(\mathbb{S}^{p-1}, 1/2)} \|\mathcal{W}x\|_4 \geq \left(13n + \frac{6}{\log 2}\gamma p^2\right)^{1/4}\right\} \\
&\leq \sum_{x \in N(\mathbb{S}^{p-1}, 1/2)} \mathbb{P}\left\{\sqrt{p}\|\mathcal{W}x\|_4 \geq \left(13n + \frac{6}{\log 2}\gamma p^2\right)^{1/4}\right\} \\
&\leq 5^p \exp(-\gamma p) \\
&= \exp\left((\log 5 - \gamma)p\right),
\end{aligned}$$

which implies that for positive constants $\tilde{c}$ and $\tilde{c}'$, with probability exceeding $1 - \exp(-\tilde{c}'p)$ we have $p^2\|\mathcal{W}\|_{2\to 4}^4 \leq \tilde{c}(n + p^2)$, or equivalently

$$\|\mathcal{W}\|_{2\to 4}^4 \leq \tilde{c}\frac{(n + p^2)}{p^2}. \tag{35}$$

– Step 4. Combining the Results of Previous Steps: Let's define

$$\mu_{n,p} = \frac{p}{\left(\sqrt{n} + \sqrt{p}\right)^2},$$

and multiply both sides of (31) by $\mu_{n,p}$ to get

$$\mu_{n,p}s_{\min}^2(\mathcal{W})\|x - y\|_2^2 \leq \beta_{n,p}^{-1}\left\|\texttt{ASU}(\mathcal{W}x) - \texttt{ASU}(\mathcal{W}y)\right\|^2 \leq \mu_{n,p}\left(s_{\max}^2(\mathcal{W})\|x - y\|_2^2 + \frac{6\pi^2}{10}\|\mathcal{W}\|_{2\to 4}^4\right), \tag{36}$$

where

$$\beta_{n,p}^{-1} = \frac{\pi^2 p}{4\left(\sqrt{n} + \sqrt{p}\right)^2}.$$

By (32) for all $x, y \in \mathbb{B}_2^p$ and with probability exceeding $1 - 2\exp(-cp)$ the left side of (36) can be lower bounded as:

$$\begin{aligned}
\mu_{n,p}s_{\min}^2(\mathcal{W})\|x - y\|_2^2 \geq \left(\frac{\sqrt{n} - c'\sqrt{p}}{\sqrt{n} + \sqrt{p}}\right)^2\|x - y\|_2^2 &= \left(1 - \frac{(c'+1)\sqrt{p}}{\sqrt{n} + \sqrt{p}}\right)^2\|x - y\|_2^2 \\
&\geq \|x - y\|^2 - \frac{2(c'+1)\sqrt{p}}{\sqrt{n} + \sqrt{p}}\|x - y\|_2^2 \\
&\geq \|x - y\|_2^2 - \frac{8(c'+1)\sqrt{p}}{\sqrt{n} + \sqrt{p}}.
\end{aligned} \tag{37}$$

On the other hand, a combination of (35) and (32) guarantees that for all $x, y \in \mathbb{B}_2^p$, with probability exceeding $1 - \exp(-\tilde{c}'p) - 2\exp(-cp)$ the right side of (36) can be upper-bounded as

$$\begin{aligned}\mu_{n,p} s_{\max}^2(\mathcal{W})\|x-y\|_2^2 + \frac{6\pi^2\mu_{n,p}}{10}\|\mathcal{W}\|_{2\to 4}^4 &\le \left(1 + \frac{(c'-1)\sqrt{p}}{\sqrt{n}+\sqrt{p}}\right)^2 \|x-y\|_2^2 + \frac{6\tilde{c}\pi^2}{10}\frac{(n+p^2)}{p\left(\sqrt{n}+\sqrt{p}\right)^2} \\ &\le \|x-y\|_2^2 + c_u\left(\frac{\sqrt{p}}{\sqrt{n}+\sqrt{p}} + \frac{n+p^2}{p\left(\sqrt{n}+\sqrt{p}\right)^2}\right),\end{aligned} \tag{38}$$

for some positive constant c_u. In the second inequality above we used the fact that $\|x-y\|_2^2 \le 4$ and $\frac{(c'-1)\sqrt{p}}{\sqrt{n}+\sqrt{p}} < 1$ for sufficiently large n. The outcomes of (37) and (38) combined with (36) validate the claims made in Theorem 4.4, and the proof is complete.

□

H.1 Proof of Lemma H.1

Proof. Consider the following functions:

$$f_\ell(x,y) = 2\left(1 - \frac{4}{\pi^2}\right)\left(x^4+y^4\right) - \frac{4}{\pi^2}(x-y)^2 + \frac{4}{\pi^2}\left(\arcsin(x) - \arcsin(y)\right)^2,$$

and

$$f_u(x,y) = 2\left(1 - \frac{4}{\pi^2}\right)\left(x^4+y^4\right) + \frac{4}{\pi^2}(x-y)^2 - \frac{4}{\pi^2}\left(\arcsin(x) - \arcsin(y)\right)^2.$$

To prove the lemma, it suffices to show that for $(x,y) \in [-1,1]\times[-1,1]$, $f_\ell(x,y) \ge 0$ and $f_u(x,y) \ge 0$. To establish the inequality for $f_\ell(x,y)$, notice that by the Lipschitz continuity of the sin function:

$$\left(\sin(\alpha) - \sin(\beta)\right)^2 \le (\alpha-\beta)^2,$$

which by setting $\alpha = \arcsin(x)$ and $\beta = \arcsin(y)$ and multiplying both sides by $4/\pi^2$ yields

$$\frac{4}{\pi^2}\left(\arcsin(x) - \arcsin(y)\right)^2 - \frac{4}{\pi^2}(x-y)^2 \ge 0, \tag{39}$$

from which it immediately follows that $f_\ell(x,y) \ge 0$.

To show the inequality $f_u(x,y) \ge 0$, we begin by rearranging the terms of f_u and bounding them as

$$\begin{aligned} f_u(x,y) &= 2\left(1 - \frac{4}{\pi^2}\right)\left(x^4+y^4\right) + \frac{4}{\pi^2}\left(x^2+y^2\right) - \frac{4}{\pi^2}\left(\arcsin^2(x) + \arcsin^2(y)\right) \\ &\quad + 2\left(\frac{4}{\pi^2}\right)\left(\arcsin(x)\arcsin(y) - xy\right) \\ &\ge 2\left(1 - \frac{4}{\pi^2}\right)\left(x^4+y^4\right) + \frac{8}{\pi^2}(x^2+y^2) - \frac{8}{\pi^2}(\arcsin^2(x) + \arcsin^2(y)) \qquad (40) \\ &= \frac{8}{\pi^2}\left(h(x) + h(y)\right), \end{aligned}$$

where

$$h(x) = \left(\frac{\pi^2}{4} - 1\right)x^4 + x^2 - \arcsin^2(x).$$

In the derivations above, inequality (40) uses the fact

$$\left(\arcsin(x) + \arcsin(y)\right)^2 \ge (x+y)^2,$$

which is a direct implication of (39), when one uses x and $-y$ as the arguments. Thanks to symmetry in x and y, to show $f_u(x,y) \ge 0$ it suffices to show that $h(x) \ge 0$, and since h is an even function,

we can restrict the domain to $x \in [0,1]$. Notice that since $x^2 \geq \frac{4}{\pi^2}\arcsin^2(x)$:

$$\begin{aligned} h(x) &= \left(\frac{\pi^2}{4} - 1\right) x^4 + x^2 - \arcsin^2(x) \\ &\geq \left(1 - \frac{4}{\pi^2}\right) x^2 \arcsin^2(x) + x^2 - \arcsin^2(x) \\ &\geq 0, \end{aligned}$$

where the second inequality is thanks to the elementary inequality

$$\arcsin(x) \leq \frac{x}{\sqrt{1 - \left(1 - \frac{4}{\pi^2}\right) x^2}},$$

which is valid for $x \in [0,1]$ (e.g., see Theorem 1 of [41]). This completes the proof. □

H.2 Proof of Lemma H.2

H.2.1 Auxiliary Lemmas Needed to Prove Lemma H.2

To prove the lemma, we first need to state two integral bounds (Lemma H.5 and Lemma H.7) which will be later used in the main proof. Also to bound the moment generating functions, we need a tail bound which is stated in Lemma H.8 below. We first present and prove these lemmas, and then combine them to prove Lemma H.2.

Lemma H.5. *For all integers $p \geq 27$:*

$$\int_0^1 \frac{dx}{(1+x^2)^{\frac{p}{6}-1}} \leq \sqrt{\frac{6}{p-6}}.$$

Proof. Consider $k \in \left\{\frac{n}{6} : n \in \mathbb{N}\right\}$. Defining I_k as below and doing an integration by parts we get

$$\begin{aligned} I_k &:= \int_0^1 \frac{dx}{(1+x^2)^k} \\ &= \left[\frac{x}{(1+x^2)^k}\right]_0^1 + 2k \int_0^1 \frac{x^2 dx}{(1+x^2)^{k+1}} \\ &= \frac{1}{2^k} + 2k \left(\int_0^1 \frac{dx}{(1+x^2)^k} - \int_0^1 \frac{dx}{(1+x^2)^{k+1}}\right), \end{aligned} \tag{41}$$

which implies

$$I_{k+1} = \frac{1}{k2^{k+1}} + \frac{2k-1}{2k} I_k. \tag{42}$$

We now use an inductive argument and show if $I_k \leq k^{-1/2}$, then $I_{k+1} \leq (k+1)^{-1/2}$. To this end, we state the following lemma which is proved at the end of this section:

Lemma H.6. *For any $k \in \left\{\frac{n}{6} : n \geq 21,\ n \in \mathbb{N}\right\}$:*

$$\frac{1}{k2^{k+1}} < \frac{3}{10k^2\sqrt{k}} < \frac{1}{\sqrt{k+1}} - \frac{2k-1}{2k\sqrt{k}}. \tag{43}$$

Using Lemma H.6, assuming $I_k \leq k^{-1/2}$, by the recursive equation (42) we get

$$\begin{aligned} I_{k+1} &= \frac{1}{k2^{k+1}} + \frac{2k-1}{2k} I_k \\ &\leq \frac{3}{10k^2\sqrt{k}} + \frac{2k-1}{2k\sqrt{k}} \\ &\leq \frac{1}{\sqrt{k+1}}, \end{aligned} \tag{44}$$

proving the inductive step. Notice that we have unit inductive step increments in (42), while the increments in k are integer multiples of $1/6$. Hence, to establish the induction base case, it suffices to show that for some n_0, the integrals $I_{\frac{n_0}{6}}, I_{\frac{n_0+1}{6}}, \dots I_{\frac{n_0+5}{6}}$ are dominated by $\sqrt{6/n_0}, \sqrt{6/(n_0+1)}, \dots \sqrt{6/(n_0+5)}$, respectively. This can be verified for $n_0 = 6$. More specifically, $I_1 = 0.7854 < 1$, $I_{7/6} = 0.7575 < \sqrt{6/7}, \dots I_{11/6} = 0.6628 < \sqrt{6/11}$. Together with the common inductive step, this verifies that

$$\forall k \in \left\{ \frac{n}{6} : n \geq 21,\ n \in \mathbb{N} \right\} :\ \ I_k \leq k^{-1/2}. \tag{45}$$

For a given integer $p \geq 27$, setting $k = p/6 - 1$ and appealing to (45) gives

$$\int_0^1 \frac{dx}{(1+x^2)^{\frac{p}{6}-1}} \leq \sqrt{\frac{6}{p-6}},$$

which completes the proof. We are only left with providing the proof of Lemma H.6, which is presented in the sequel.

The first inequality in (43) is straightforward as the exponential expression $k2^{k+1}$ dominates the monomial $\frac{10}{3}k^2\sqrt{k}$ for sufficiently large k. It can be easily verified that the domination happens after $k \geq 21/6$. To prove the upper-bound, it suffices to show that

$$\frac{3}{10k^2} + \frac{2k-1}{2k} < \sqrt{\frac{k}{k+1}},$$

which after squaring both sides and rearranging the terms is equivalent to showing that

$$(k+1)(3+10k^2-5k)^2 < k\left(10k^2\right)^2.$$

Taking both expressions to one side and expanding the terms reveals that

$$k\left(10k^2\right)^2 - (k+1)(3+10k^2-5k)^2 = 15k^3 - 55k^2 + 21k - 9, \tag{46}$$

which is a strictly positive quantity, simply because $15k^3 + 21k$ consistently dominates $55k^2 + 9$ for every integer $k \geq 3$. This verifies the upper-bound in (43), and the proof of Lemma H.6 is complete. □

Lemma H.7. *For all integers $p \geq 27$:*

$$\int_0^1 \frac{x^2 dx}{(1+x^2)^{\frac{p}{3}-1}} \leq \frac{3^{3/2}}{2(p-6)^{3/2}},$$

Proof. Let $k \in \left\{ \frac{n}{6} : n \in \mathbb{N} \right\}$, then

$$\int_0^1 \frac{x^2 dx}{(1+x^2)^k} = \int_0^1 \frac{(1+x^2)dx}{(1+x^2)^k} - \int_0^1 \frac{dx}{(1+x^2)^k} = I_{k-1} - I_k,$$

where I_k follows the formulation in (41). By the recursive equation in (42) which is valid for $k \in \left\{ \frac{n}{6} : n \geq 21,\ n \in \mathbb{N} \right\}$ we have

$$\begin{aligned} I_{k-1} - I_k &= -\frac{1}{(k-1)2^k} + \frac{1}{2k-2} I_{k-1} \\ &= -\frac{1}{(k-1)2^k} + \frac{1}{2(k-1)\sqrt{k-1}} \\ &\leq \frac{1}{2(k-1)^{3/2}}, \end{aligned}$$

where the second inequality uses the fact $I_k \leq k^{-1/2}$ which was proved after equation (42). Setting $k = p/3 - 1$ (which still keeps k an integer multiple of $1/6$) gives

$$\int_0^1 \frac{x^2 dx}{(1+x^2)^{\frac{p}{2}-1}} \leq \frac{3^{3/2}}{2(p-6)^{3/2}}.$$

This completes the proof. □

The next lemma provides a tail bound for the random vectors of interest in Lemma H.2.

Lemma H.8. *Consider $x \in \mathbb{S}^{p-1}$ where $p \geq 3$, and $g = [g_1, \ldots, g_p]^\top \sim \mathcal{N}(0, I)$. For all $\lambda \in (0, 1]$:*

$$\mathbb{P}\left\{\frac{|x^\top g|}{\|g\|} \geq \lambda\right\} \leq \sqrt{\frac{2}{\pi(p-2)}}(1+\lambda^2)^{1-\frac{p}{2}}.$$

Proof. Since $x \in \mathbb{S}^{p-1}$ and the standard normal distribution is rotation invariant, we can set $x = e_1$ (the first canonical basis) which implies

$$\begin{aligned}
q_\lambda &= \mathbb{P}\left\{\frac{|x^\top g|}{\|g\|} \geq \lambda\right\} \\
&= \mathbb{P}\left\{\frac{|g_1|}{\|g\|} \geq \lambda\right\} \\
&= \mathbb{P}\left\{|g_1| \geq \lambda\sqrt{g_1^2 + g_2^2 \ldots + g_p^2}\right\} \\
&\leq \mathbb{P}\left\{|g_1| \geq \lambda\sqrt{g_2^2 + g_3^2 \ldots + g_p^2}\right\} \\
&= \mathbb{P}\left\{\frac{\left(\sum_{i=2}^p g_i^2\right)/(p-1)}{g_1^2} \leq \frac{1}{\lambda^2(p-1)}\right\}, \qquad (47)
\end{aligned}$$

where the inequality is thanks to the fact that

$$\left\{z \in \mathbb{R}^p : |z_1| \geq \lambda\sqrt{z_1^2 + z_2^2 \ldots + z_p^2}\right\} \subseteq \left\{z \in \mathbb{R}^p : |z_1| \geq \lambda\sqrt{z_2^2 + z_3^2 \ldots + z_p^2}\right\}.$$

Notice that the random variable $\frac{\left(\sum_{i=2}^p g_i^2\right)/(p-1)}{g_1^2}$ is the ratio of two independent chi-squared random variables each normalized by their degrees of freedom, hence follows an F-distribution with $p-1$ and 1 degrees of freedom, denoted as $\mathcal{F}_{p-1,1}$. Generally, the cumulative distribution function (CDF) of $\mathcal{F}_{d_1,d_2}$ at a given point z is evaluated as

$$\mathbb{P}\left\{\mathcal{F}_{d_1,d_2} \leq z\right\} = \frac{B\left(\frac{d_1 z}{d_1 z + d_2}; \frac{d_1}{2}, \frac{d_2}{2}\right)}{B\left(1; \frac{d_1}{2}, \frac{d_2}{2}\right)},$$

where B represents an incomplete beta function:

$$B(z; a, b) = \int_0^z t^{a-1}(1-t)^{b-1}dt.$$

Notice that the probability in (47) is basically the CDF of $\mathcal{F}_{p-1,1}$ evaluated at $\left(\lambda^2(p-1)\right)^{-1}$, therefore

$$q_\lambda \leq \mathbb{P}\left\{\mathcal{F}_{p-1,1} \leq \frac{1}{\lambda^2(p-1)}\right\} = \frac{B\left(\frac{1}{1+\lambda^2}; \frac{p-1}{2}, \frac{1}{2}\right)}{B\left(1; \frac{p-1}{2}, \frac{1}{2}\right)} = \frac{B\left(\frac{1}{1+\lambda^2}; \frac{p-1}{2}, \frac{1}{2}\right)\Gamma\left(\frac{p}{2}\right)}{\Gamma(\frac{1}{2})\Gamma\left(\frac{p-1}{2}\right)},$$

where in the last equation we used the well-known relationship between the beta function and the gamma function:

$$B(1; a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}.$$

Since $\Gamma(1/2) = \sqrt{\pi}$, and for $p \geq 1$, by the Gautschi's inequality:

$$\frac{\Gamma\left(\frac{p-1}{2}\right)}{\Gamma\left(\frac{p}{2}\right)} \geq \sqrt{\frac{2}{p}},$$

one can bound q_λ as

$$\begin{aligned} q_\lambda &\le \sqrt{\frac{p}{2\pi}} B\left(\frac{1}{1+\lambda^2}; \frac{p-1}{2}, \frac{1}{2}\right) \\ &= \sqrt{\frac{p}{2\pi}} \int_0^{\frac{1}{1+\lambda^2}} \frac{t^{\frac{p-3}{2}} dt}{\sqrt{1-t}} \\ &\le \sqrt{\frac{p}{2\pi}} \sqrt{\frac{1+\lambda^2}{\lambda^2}} \int_0^{\frac{1}{1+\lambda^2}} t^{\frac{p-3}{2}} dt \\ &= \sqrt{\frac{2p}{\pi(p-1)^2}} \frac{\left(1+\lambda^2\right)^{\frac{2-p}{2}}}{\lambda} \\ &\le \sqrt{\frac{2}{\pi(p-2)}} \frac{\left(1+\lambda^2\right)^{\frac{2-p}{2}}}{\lambda}, \end{aligned}$$

where in the second inequality we used $\sqrt{1-t} \ge \sqrt{\frac{\lambda^2}{1+\lambda^2}}$ when $0 \le t \le \frac{1}{1+\lambda^2}$. This completes the proof. □

H.2.2 Main Proof of Lemma H.2

We are now ready to use the results of the auxiliary lemmas stated above to prove Lemma H.2. Consider $x \in \mathbb{S}^{p-1}$ where $p \ge 3$, and $g = [g_1, \dots, g_p]^\top \sim \mathcal{N}(0, I)$. We define the random variable

$$u = \frac{\left|x^\top g\right|}{\|g\|}.$$

Notice that by construction $0 \le u \le 1$, which combined with the result of Lemma H.8 gives

$$\mathbb{P}\{u > \lambda\} : \begin{cases} = 0 & \lambda \ge 1 \\ \le \sqrt{\frac{2}{\pi(p-2)}} \frac{(1+\lambda^2)^{1-\frac{p}{2}}}{\lambda} & 0 < \lambda < 1 \\ = 1 & \lambda \le 0 \end{cases}. \tag{48}$$

As a result, for all $t \ge 0$ and integers $q \ge 2$:

$$\begin{aligned} \mathbb{E}\exp\left(tu^q\right) = \int_0^1 \exp\left(t\lambda^q\right) d\left(\mathbb{P}\left\{u \le \lambda\right\}\right) &= -\int_0^1 \exp\left(t\lambda^q\right) d\left(\mathbb{P}\left\{u > \lambda\right\}\right) \\ &= 1 + \int_0^1 tq\lambda^{q-1} \exp\left(t\lambda^q\right) \mathbb{P}\{u > \lambda\} d\lambda \\ &\le 1 + tq\sqrt{\frac{2}{\pi(p-2)}} \int_0^1 \lambda^{q-2} \exp\left(t\lambda^q\right) (1+\lambda^2)^{1-\frac{p}{2}} d\lambda \end{aligned} \tag{49}$$

where the third equality is thanks to integration by parts and the fact that $\mathbb{P}\{u > 1\} = 1 - \mathbb{P}\{u > 0\} = 0$.

Part (a) Proof Consider $q = 2$, then by (49) we get

$$\begin{aligned} \mathbb{E}\exp\left(tu^2\right) &\le 1 + 2t\sqrt{\frac{2}{\pi(p-2)}} \int_0^1 \exp\left(t\lambda^2\right) (1+\lambda^2)^{1-\frac{p}{2}} d\lambda \\ &\le 1 + 2t\sqrt{\frac{2}{\pi(p-2)}} \int_0^1 \left(1+\lambda^2\right)^{\frac{t}{\log 2}} (1+\lambda^2)^{1-\frac{p}{2}} d\lambda \\ &= 1 + 2t\sqrt{\frac{2}{\pi(p-2)}} \int_0^1 (1+\lambda^2)^{1+\frac{t}{\log 2}-\frac{p}{2}} d\lambda, \end{aligned} \tag{50}$$

where in the second inequality we used the fact that for all $z \in [0,1]$: $\exp(z) \le (1+z)^{\frac{1}{\log 2}}$ which implies that

$$\forall z \in [0,1],\ t \ge 0: \quad \exp(tz) \le (1+z)^{\frac{t}{\log 2}}.$$

For all $\tau \in \left[0, \sqrt{\log(2)}/\sqrt{3}\right]$, setting $t = \tau^2 p$ in (50) gives

$$\begin{aligned}\mathbb{E}\exp\left(\tau^2 p u^2\right) &\le 1 + 2\tau^2 p\sqrt{\frac{2}{\pi(p-2)}}\int_0^1 (1+\lambda^2)^{1+\frac{\tau^2 p}{\log 2}-\frac{p}{2}}d\lambda \\ &\le 1 + 2\tau^2 p\sqrt{\frac{2}{\pi(p-2)}}\int_0^1 (1+\lambda^2)^{1-\frac{p}{6}}d\lambda.\end{aligned}$$

Now using Lemma H.5, we have for all $\tau \in \left[0, \sqrt{\log(2)}/\sqrt{3}\right]$ and $p \ge 27$:

$$\begin{aligned}\mathbb{E}\exp\left(\tau^2 p u^2\right) &\le 1 + 2\tau^2 p\sqrt{\frac{12}{\pi(p-2)(p-6)}} \\ &\le 1 + \frac{25}{4}\tau^2 \\ &\le \exp\left(\frac{25}{4}\tau^2\right).\end{aligned}$$

Since $[0, 2/5] \subset \left[0, \sqrt{\log(2)}/\sqrt{3}\right]$ the advertised claim is valid.

Part (b) Proof Notice that by (49) we have

$$\mathbb{E}\exp\left(tu^4\right) \le 1 + 4t\sqrt{\frac{2}{\pi(p-2)}}\int_0^1 \lambda^2(1+\lambda^2)^{1+\frac{t}{\log 2}-\frac{p}{2}}d\lambda.$$

For all $\tau \in [0, \log(2)/6]$, setting $t = \tau p$ gives

$$\begin{aligned}\mathbb{E}\exp\left(\tau p u^4\right) &\le 1 + 4\tau p\sqrt{\frac{2}{\pi(p-2)}}\int_0^1 \lambda^2(1+\lambda^2)^{1+\frac{\tau p}{\log 2}-\frac{p}{2}}d\lambda \\ &\le 1 + 4\tau p\sqrt{\frac{2}{\pi(p-2)}}\int_0^1 \lambda^2(1+\lambda^2)^{1-\frac{p}{3}}d\lambda.\end{aligned}$$

Now using Lemma H.7 we get

$$\mathbb{E}\exp\left(\tau p u^4\right) \le 1 + 2\tau p\sqrt{\frac{54}{\pi(p-2)(p-6)^3}} \le 1 + \frac{13\tau}{p} \le \exp\left(\frac{13\tau}{p}\right).$$

I Proof of Theorem 5.1

Theorem 5.1. *Suppose that $u, v \in \mathbb{S}^{p-1}$ are unit vectors. Let $\mathfrak{r}$ be a Rademacher random vector in $\mathbb{R}^p$. Then,*

$$\left|\mathbb{E}\left[\operatorname{sign}\left(\mathfrak{r}^\top u\right)\operatorname{sign}\left(\mathfrak{r}^\top v\right)\right] - \frac{2}{\pi}\arcsin(u^\top v)\right| \le c g\left(u, \frac{v - \langle u, v\rangle u}{\|v - \langle u, v\rangle u\|}\right),$$

where $c = 264$ is an absolute constant, and for two unit vectors w and w' in $\mathbb{S}^{p-1}$:

$$g(w, w') = \sum_{i=1}^{p}(w_i^2 + w_i'^2)^{3/2}. \tag{51}$$

Proof. A main component of the proof is the following result due to Raič [67].

Theorem I.2 (Multivariate Berry-Esseen [67]). *Let $x_1, \dots, x_p$ be independent random vectors in $\mathbb{R}^d$. Assume that $\mathbb{E}x_i = 0$ for all i and that $\sum_{i=1}^p \mathrm{Cov}(x_i) = I_d$. Let $y = \sum_{i=1}^p x_i$. Then, for all convex sets $A \subseteq \mathbb{R}^d$, we have*

$$|\mathbb{P}\{y \in A\} - \mathbb{P}\{\mathfrak{z} \in A\}| \leq (42d^{1/4} + 16) \sum_{i=1}^p \mathbb{E}\|x_i\|_2^3,$$

where $\mathfrak{z}$ is a random vector with a standard normal distribution $\mathcal{N}(0, I_d)$.

Consider $\mathcal{W}$ be the subspace formed by u and v and columns of $W \in \mathbb{R}^{p\times 2}$ forming an orthonormal basis for $\mathcal{W}$. Consider $W = [w, w']$. Projecting any vector x onto $\mathcal{W}$ is done through $\mathcal{P}_{\mathcal{W}}(x) = W^\top x$. Since projection does not expand the angles we should have

$$\mathrm{sign}\left(\mathfrak{r}^\top u\right) = \mathrm{sign}\left(\mathfrak{r}^\top W W^\top u\right) = \mathrm{sign}\left(\mathfrak{r}^\top W W^\top u / \left\|W^\top u\right\|\right), \tag{52}$$

and a similar relation holds for v. Define

$$\tilde{u} := W^\top u / \left\|W^\top u\right\|, \qquad \tilde{v} := W^\top v / \left\|W^\top v\right\|.$$

The random vector $W^\top \mathfrak{r}$ can be viewed as the sum $W^\top \mathfrak{r} = \sum_{i=1}^p x_i$ for

$$x_i = \mathfrak{r}_i \begin{pmatrix} w_i \\ w_i' \end{pmatrix}, \ i = 1, \dots, p,$$

where the i subscript denotes the i-th element of each vector. Note that by construction x_i are independent, $\mathbb{E}x_i = 0$ and

$$\sum_{i=1}^p \mathrm{Cov}(x_i) = \mathbb{E}\left[W^\top \mathfrak{r}\mathfrak{r}^\top W\right] = W^\top \mathbb{E}\left[\mathfrak{r}\mathfrak{r}^\top\right] W = W^\top I_n W = I_2,$$

which indicates that x_i meet the conditions stated in Theorem I.2. We now have

$$\begin{aligned}\mathbb{E}\left[\mathrm{sign}\left(\mathfrak{r}^\top u\right)\mathrm{sign}\left(\mathfrak{r}^\top v\right)\right] &= \mathbb{E}\left[\mathrm{sign}\left(\mathfrak{r}^\top W\tilde{u}\right)\mathrm{sign}\left(\mathfrak{r}^\top W\tilde{v}\right)\right] \\ &= 2\mathbb{P}\left\{\mathfrak{r}^\top W\tilde{u} > 0, \mathfrak{r}^\top W\tilde{v} > 0\right\} - 2\mathbb{P}\left\{\mathfrak{r}^\top W\tilde{u} > 0, \mathfrak{r}^\top W\tilde{v} < 0\right\},\end{aligned}$$

where the first equality is thanks to (52) and the second equality is a simple expansion of the expectation, using the fact that $\mathrm{sign}(0) = 0$ and due to symmetry of $\mathfrak{r}$:

$$\begin{aligned}\mathbb{P}\left\{\mathfrak{r}^\top W\tilde{u} > 0, \mathfrak{r}^\top W\tilde{v} > 0\right\} &= \mathbb{P}\left\{\mathfrak{r}^\top W\tilde{u} < 0, \mathfrak{r}^\top W\tilde{v} < 0\right\}, \\ \mathbb{P}\left\{\mathfrak{r}^\top W\tilde{u} > 0, \mathfrak{r}^\top W\tilde{v} < 0\right\} &= \mathbb{P}\left\{\mathfrak{r}^\top W\tilde{u} < 0, \mathfrak{r}^\top W\tilde{v} > 0\right\}.\end{aligned}$$

For $\mathfrak{z} \sim \mathcal{N}(0, I_2)$, we have

$$\mathbb{P}\left\{\mathfrak{z}^\top\tilde{u} > 0, \ \pm\mathfrak{z}^\top\tilde{v} > 0\right\} = \frac{1}{4} \pm \frac{\arcsin(\tilde{u}^\top\tilde{v})}{2\pi} = \frac{1}{4} \pm \frac{\arcsin(u^\top v)}{2\pi},$$

where the second equality holds since the angle between $\tilde{u}$ and $\tilde{v}$ is the same as that of u and v. Applying the triangle inequality and twice appealing to Theorem I.2 we get

$$\begin{aligned}&\left|\mathbb{E}\left[\mathrm{sign}\left(\mathfrak{r}^\top u\right)\mathrm{sign}\left(\mathfrak{r}^\top v\right)\right] - \frac{2}{\pi}\arcsin(u^\top v)\right| \\ &\qquad \leq 2\left|\mathbb{P}\left\{\mathfrak{r}^\top W\tilde{u} > 0, \mathfrak{r}^\top W\tilde{v} > 0\right\} - \mathbb{P}\left\{\mathfrak{z}^\top\tilde{u} > 0, \mathfrak{z}^\top\tilde{v} > 0\right\}\right| \\ &\qquad\quad + 2\left|\mathbb{P}\left\{\mathfrak{r}^\top W\tilde{u} > 0, \mathfrak{r}^\top W\tilde{v} < 0\right\} - \mathbb{P}\left\{\mathfrak{z}^\top\tilde{u} > 0, \mathfrak{z}^\top\tilde{v} < 0\right\}\right| \\ &\qquad \leq 4\left(42(2)^{1/4} + 16\right)\sum_{i=1}^p \mathbb{E}\|x_i\|_2^3 \\ &\qquad \leq 264\sum_{i=1}^p \mathbb{E}\|x_i\|_2^3 \\ &\qquad = 264\sum_{i=1}^p \left(w_i^2 + w_i'^2\right)^{3/2}.\end{aligned} \tag{53}$$

Since this inequality holds for all selections of w and w', we can specifically pick the ones offered by the Gram–Schmidt procedure as

$$w = u, \qquad w' = \frac{v - \langle u, v\rangle u}{\|v - \langle u, v\rangle u\|}, \tag{54}$$

which implies that

$$\left|\mathbb{E}\left[\operatorname{sign}\left(\mathfrak{r}^\top u\right)\operatorname{sign}\left(\mathfrak{r}^\top v\right)\right] - \frac{2}{\pi}\arcsin(u^\top v)\right| \leq Cg\left(u, \frac{v - \langle u, v\rangle u}{\|v - \langle u, v\rangle u\|}\right). \tag{55}$$

In the sequel we show that the Gram-Schmidt selection is already tight and the right-hand side expression cannot be made any tighter. For this purpose, assume that we pick a different W matrix with columns $\tilde{w}$ and $\tilde{w}'$. Since $\tilde{w}$ and $\tilde{w}'$ must reside within the same plane as w and w' in (54), and still meet the orthonormality conditions, we must have

$$\begin{aligned}\tilde{w} &= \sin(\alpha)w + \cos(\alpha)w' \\ \tilde{w}' &= -\cos(\alpha)w + \sin(\alpha)w'\end{aligned}$$

for some $\alpha \in [0, 2\pi)$. Now, notice that

$$\begin{aligned} g(\tilde{w}, \tilde{w}') &= \sum_{i=1}^{p}\left(\tilde{w}_i^2 + \tilde{w}_i'^2\right)^{3/2} \\ &= \sum_{i=1}^{p}\left(\left(\sin(\alpha)w_i + \cos(\alpha)w_i'\right)^2 + \left(-\cos(\alpha)w_i + \sin(\alpha)w_i'\right)^2\right)^{3/2} \\ &= \sum_{i=1}^{p}\left(w_i^2 + w_i'^2\right)^{3/2} \\ &= g(w, w'), \end{aligned}$$

which indicates that the right-hand side of the inequality (55) is oblivious to the selection of W, and all such selections present an identical right-hand side. □

J Proof of Corollary 5.1

Corollary 5.1. *Consider unit vectors $u, v \in \mathbb{S}^{p-1}$ such that $\|u\|_\infty = \mathcal{O}(p^{-1/2})$, $\|v\|_\infty = \mathcal{O}(p^{-1/2})$, and there exists some constant $c > 0$ such that $|\langle u, v\rangle| \leq 1 - c$. Then*

$$\left|\mathbb{E}\left[\operatorname{sign}\left(\mathfrak{r}^\top u\right)\operatorname{sign}\left(\mathfrak{r}^\top v\right)\right] - \frac{2}{\pi}\arcsin(u^\top v)\right| = \mathcal{O}\left(p^{-1/2}\right).$$

Proof. We only need to show that $g(w, w') = \mathcal{O}(p^{-1/2})$ for w and w' picked as (54) and g defined as (51). Clearly, by construction $\|w\|_\infty = \mathcal{O}(p^{-1/2})$. On the other hand

$$\begin{aligned} \|v - \langle u, v\rangle u\|_\infty &\leq \|v\|_\infty + |\langle u, v\rangle|\|u\|_\infty \\ &\leq \|v\|_\infty + \|u\|_2\|v\|_2\|u\|_\infty \\ &= \|v\|_\infty + \|u\|_\infty \\ &= \mathcal{O}(p^{-1/2}). \end{aligned} \tag{56}$$

Moreover,

$$\begin{aligned} \|v - \langle u, v\rangle u\|_2 &\geq \|v\|_2 - |\langle u, v\rangle|\|u\|_2 \\ &= 1 - |\langle u, v\rangle| \\ &\geq c. \end{aligned} \tag{57}$$

Combining (56) and (57) implies $\|w'\|_\infty = \mathcal{O}(p^{-1/2})$, and therefore

$$\begin{aligned} g(w, w') &= \sum_{i=1}^{p} (w_i^2 + {w'_i}^2)^{3/2} \\ &\leq \max_{i=1,\dots,p} \left\{ \left\| \begin{pmatrix} w_i \\ w'_i \end{pmatrix} \right\| \right\} \sum_{i=1}^{p} (w_i^2 + {w'_i}^2) \\ &= 2 \max_{i=1,\dots,p} \left\{ \left\| \begin{pmatrix} w_i \\ w'_i \end{pmatrix} \right\| \right\} \\ &= \mathcal{O}(p^{-1/2}). \end{aligned}$$

□

K Proof of Proposition 5.1

Proposition 5.1. *Consider a similar RASU (or ASU) layer as Theorem 4.1, where for fixed constants $C, \delta > 0$, $\|x\|_\infty \leq Cp^{-1/2}$, $\|w_i\|_\infty \leq Cp^{-1/2}$, and $|w_i^\top x| \leq 1 - \delta$ for all $i \in \{1, \dots, n\}$. Define the embedded output as $\tilde{y} = \texttt{ReLU}\left(\texttt{sign}(W\mathcal{R}^\top)\, \texttt{sign}(\mathcal{R}x)\right)$ (or $\tilde{y} = \texttt{Id}\left(\texttt{sign}(W\mathcal{R}^\top)\, \texttt{sign}(\mathcal{R}x)\right)$), where $\mathcal{R} \in \mathbb{R}^{N\times p}$ is a Rademacher matrix. Then, with high probability:*

$$\left\| \frac{1}{N}\tilde{y} - y \right\|_2 \leq \mathcal{O}\left(\sqrt{\frac{n \log n}{N}} + \sqrt{\frac{n}{p}} \right).$$

Proof. The main difference between Theorem 4.1 and Proposition 5.1 is that the Gaussian matrix $\mathcal{G} \in \mathbb{R}^{N\times p}$ is replaced by a Radamacher matrix $\mathcal{R} \in \mathbb{R}^{N\times p}$. Recall that $g_j^\top$ denotes the j-th row of $\mathcal{G}$ and note that g_j enters the proofs of Theorem 4.1 through the random variables

$$\zeta_{ij} = \operatorname{sign}\left(g_j^\top w_i\right) \operatorname{sign}\left({g_j}^\top x\right). \tag{58}$$

These proofs rely on three properties of these random variables:

(P1) $-1 \leq \zeta_{ij} \leq 1$ for all $(i, j) \in \{1, \dots, n\} \times \{1, \dots, p\}$,

(P2) For fixed $i \in \{1, \dots, n\}$, the random variables $\zeta_{i1}, \dots, \zeta_{ip}$ are i.i.d.,

(P3) The expected value $\mathbb{E}\zeta_{ij} = \frac{2}{\pi} \arcsin(w_i^\top x)$.

When we replace the rows $g_j^\top$ of $\mathcal{G}$ with the rows $\mathfrak{r}_j^\top$ of $\mathcal{R}$, and define

$$\tilde{\zeta}_{ij} = \operatorname{sign}\left(\mathfrak{r}_j^\top w_i\right) \operatorname{sign}\left({\mathfrak{r}_j}^\top x\right), \tag{59}$$

properties (P1) and (P2) continue to hold for $\tilde{\zeta}_{ij}$, and by Corollary 5.1, property (P3) approximately holds

$$\mathbb{E}(\tilde{\zeta}_{ij}) = \frac{2}{\pi} \arcsin(w_i^\top x) + \mathcal{O}(p^{-1/2}). \tag{60}$$

If we define $y'_i = \operatorname{ReLU}(\mathbb{E}(\tilde{\zeta}_{i1}))$ (or $y'_i = \mathbb{E}(\tilde{\zeta}_{i1})$ for the `ASU` case) for $i \in \{1, \dots, n\}$, then the proof of Theorem 4.1 implies that for any $c > 0$,

$$\left\| \frac{1}{N}\tilde{y} - y' \right\|_2 \leq \sqrt{\frac{2(c + \log 2n)n}{N}},$$

with probability at least $1 - \exp(-c)$. Using (60) and the triangle inequality we get

$$\left\| \frac{1}{N}\tilde{y} - y \right\|_2 \leq \mathcal{O}\left(\sqrt{\frac{n \log n}{N}} + \sqrt{\frac{n}{p}} \right) \tag{61}$$

with high probability, which completes the proof. □

L Proof of Proposition 5.2

Proposition 5.2. *Consider a similar TASU layer as Theorem 4.2, where additionally for fixed constants $C, \delta > 0$, $\|x\|_\infty \le Cp^{-1/2})$, $\|w_i\|_\infty \le Cp^{-1/2}$, and $0 < \ell_{\min} \le |w_i^\top x| \le 1 - \delta$ for all $i \in \{1, \dots, n\}$. Define the embedded layer output as $\tilde{y} = \boldsymbol{sign}\big(\boldsymbol{sign}(W\mathcal{R}^\top)\, \boldsymbol{sign}(\mathcal{R}x)\big)$, where $\mathcal{R} \in \mathbb{R}^{N\times p}$ is a Rademacher matrix. Assume $Cp^{-1/2} \le \ell_{min}/\pi$. Fix $\varepsilon \le \sqrt{n}$ and set $\kappa \ge \frac{\pi}{\ell_{\min}} \log \frac{4\sqrt{n}}{\varepsilon}$. Then, picking $N = \mathcal{O}(n\kappa^2 \log n/\varepsilon^2)$ guarantees that $\|y - \tilde{y}\|_2 = \mathcal{O}(\varepsilon + \kappa\sqrt{n/p})$ with high probability.*

Proof. The proof is similar to that of Theorem 4.2 and we outline the main steps. The discrepancy between the two layers can be bounded as $\|e\|_2 = \|\tilde{y} - y\|_2 \le \|e'\|_2 + \|e''\|_2$, where following (59) as the notation:

$$e_i' = \tanh\left(\frac{\kappa}{N}\sum_{j=1}^N \tilde{\delta}_{ij}\right) - \tanh\left(\frac{2\kappa}{\pi}\arcsin\left(w_i^\top x\right)\right),$$

and

$$e_i'' = \operatorname{sign}\left(\frac{1}{N}\sum_{j=1}^N \tilde{\delta}_{ij}\right) - \tanh\left(\frac{\kappa}{N}\sum_{j=1}^N \tilde{\delta}_{ij}\right).$$

The term e' in the proof of Theorem 4.2 can be handled in a similar way as in the proof of Theorem 5.1, to indicate that with probability exceeding $1 - \exp(-c)$:

$$\|e'\|_2 \le \sqrt{\frac{2(c + \log(2n))n\kappa^2}{N}} + C\sqrt{\frac{\kappa^2 n}{p}},$$

where the additional term $\sqrt{\kappa^2 n/p}$ comes from the Lipschitz continuity of $\tanh(\kappa\cdot)$ and a triangle inequality similar to (60) and (61).

For the second term we similarly have

$$|e_i''| \le 2\exp\left(-2\kappa\left|\frac{1}{N}\sum_{j=1}^N \tilde{\delta}_{ij}\right|\right).$$

Applying Hoeffding's inequality and the triangle inequality guarantees that with probability exceeding $1 - 2\exp(-c)$:

$$\left|\frac{1}{N}\sum_{j=1}^N \tilde{\delta}_{ij}\right| > \frac{2}{\pi}\ell_{\min} - \frac{\sqrt{2c}}{\sqrt{N}} - \frac{C}{\sqrt{p}}.$$

This indicates that with probability exceeding $1 - 2\exp(-c)$:

$$\|e''\|_2 \le 2\exp\left(\log\sqrt{n} - \frac{4\kappa}{\pi}\ell_{\min} + \kappa\sqrt{\frac{8(c + \log n)}{N}} + 2\frac{C\kappa}{\sqrt{p}}\right).$$

Setting $N = 8(c + \log 2n)n\kappa^2/\varepsilon^2$, guarantees with probability exceeding $1 - 3\exp(-c)$:

$$\|e\|_2 \le \frac{\varepsilon}{2} + C\sqrt{\frac{\kappa^2 n}{p}} + 2\exp\left(\log\sqrt{n} - \frac{4\kappa}{\pi}\ell_{\min} + \frac{\varepsilon}{\sqrt{n}} + 2\frac{C\kappa}{\sqrt{p}}\right).$$

If we pick κ in a way that

$$\frac{4\kappa}{\pi}\ell_{\min} \ge 2\log\frac{4\sqrt{n}}{\varepsilon} + 2\frac{C\kappa}{\sqrt{p}}, \tag{62}$$

then by a similar argument as before we will have

$$\frac{4\kappa}{\pi}\ell_{\min} \geq 2\log\frac{4\sqrt{n}}{\varepsilon} + 2\frac{C\kappa}{\sqrt{p}} \geq \log\sqrt{n} + \frac{\varepsilon}{\sqrt{n}} + 2\frac{C\kappa}{\sqrt{p}} - \log\frac{\varepsilon}{4},$$

and subsequently

$$\|e\|_2 \leq \varepsilon + C\sqrt{\frac{\kappa^2 n}{p}}.$$

It only remains to see that when $Cp^{-1/2} \leq \ell_{\min}/\pi$ and $\kappa \geq \frac{\pi}{\ell_{\min}}\log\frac{4\sqrt{n}}{\varepsilon}$, condition (62) is met. □

M Minor Distribution Shift of Wide Neural Networks

It is well-known that in wide neural networks, the weight distribution shifts only slightly over the course of training [15]. In this section, we present a numerical experiment to demonstrate this property. For this purpose, we train a fully connected G-Net on MNIST data with a wide first layer (width 1024) initialized using row-normalized Gaussian weights. During training, we monitor the evolution of the layer's weights and report their empirical distributions at epochs 0, 1, 10, and 25, as shown in Figure 5.

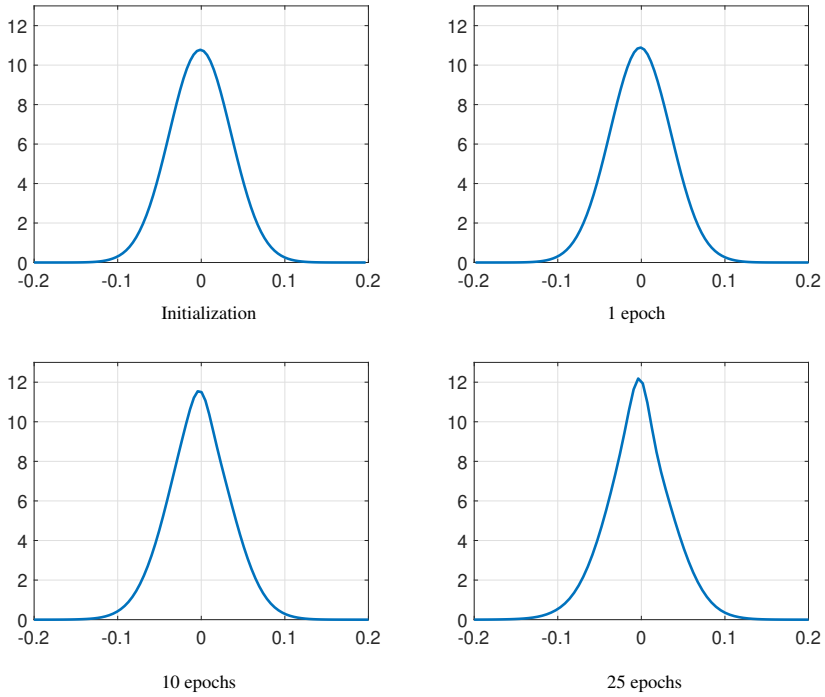


Figure 5: Minor weight distribution shift for wide layers: the distribution of the weights of a fully connected G-Net Layer at epochs 0, 1, 10, and 25

N G-Net Implementation Details

In this section, we discuss key implementation details of G-Nets, including the handling of fully connected layers, convolutional layers, general linear layers, and classification versus regression output layers. We also present alternative strategies for incorporating bias in linear layers to enhance the concentration properties of the EHD G-Net.

Fully-Connected Layers. As discussed in the paper, when $W \in \mathbb{R}^{n\times p}$ maintains ℓ_2-normalized rows, $x \in \mathbb{S}^{p-1}$ and $\mathcal{G} \in \mathbb{R}^{N\times p}$ is a Gaussian matrix, then by Grothendieck's identity and the law of large numbers:

$$\frac{1}{N}\texttt{sign}\left(W\mathcal{G}^\top\right)\texttt{sign}\left(\mathcal{G}x\right) \xrightarrow[N\to\infty]{} \frac{2}{\pi}\texttt{arcsin}(Wx). \tag{63}$$

For finite N, we refer to the left-hand side of (63) as the EHD representation of the right-hand side expression. When W and x do not maintain the normalization property, the right-hand side argument needs to be normalized properly. More specifically, consider general $W \in \mathbb{R}^{n\times p}$, $x \in \mathbb{R}^p$, and a Gaussian matrix $\mathcal{G} \in \mathbb{R}^{N\times p}$, structured as follows:

$$W = \begin{pmatrix} w_1^\top \\ \vdots \\ w_n^\top \end{pmatrix}, \qquad \mathcal{G} = \begin{pmatrix} g_1^\top \\ \vdots \\ g_N^\top \end{pmatrix}.$$

Then we can construct a diagonal matrix to perform the normalization on Wx as

$$\frac{2}{\pi}\texttt{arcsin}\left(D_{W,x}^{-\frac{1}{2}}Wx\right) \xrightarrow{\text{EHD}} \frac{1}{N}\sum_{i=1}^{N}\text{sign}\left(\langle g_i, x\rangle\right)\texttt{sign}\left(Wg_i\right), \tag{64}$$

where

$$D_{W,x} = \|x\|_2^2\,\text{diag}\left(\|w_1\|_2^2, \ldots, \|w_n\|_2^2\right), \tag{65}$$

and $\text{diag}(\alpha_1, \ldots, \alpha_n)$ is a diagonal matrix with elements $\alpha_1, \ldots, \alpha_n$. Basically, the left-hand side expression of (64) is what is implemented in a fully-connected layer of G-Net.

Smooth Approximation of the Norm. When training a G-Net, the rows of W need to be normalized by their ℓ_2 norm. Basically, the i-th row of W takes the form $w_i^\top/\|w_i\|_2$. To maintain a bounded gradient flow during the training, we use the smooth approximation

$$\|w_i\|_2 \approx \sqrt{\|w_i\|_2^2 + \varepsilon},$$

where $\varepsilon > 0$ is a small constant.

A Different Handling of the Bias for Faster Concentration. For an input $x \in \mathbb{R}^p$, the output of a fully-connected layer is normally in the form $Wx + b$ where $W \in \mathbb{R}^{n\times p}$ and $b \in \mathbb{R}^n$ are learnable weight and bias parameters. One could think of absorbing the bias in the weight matrix as

$$Wx + b = \tilde{W}\tilde{x}, \quad \text{where}: \tilde{W} = \begin{bmatrix} W & b \end{bmatrix}, \ \tilde{x} = \begin{bmatrix} x \\ 1 \end{bmatrix},$$

and view the problem as an instance of the aforementioned fully-connected case. However, as discussed in the theoretical results, especially for Rademacher embedding, having the weights evenly spread out helps with a better concentration of EHD G-Net. Since the scales of b and W after training can differ significantly, an alternative approach to introducing a bias term in linear layers—aimed at promoting a more balanced distribution of weights—is to use a fully connected layer of the form

$$W(x + c1_p),$$

where $W \in \mathbb{R}^{n\times p}$ and $c \in \mathbb{R}$ is a learnable scalar. By trading off a negligible degree of freedom, this formulation eliminates the issue of uneven scaling between weight and bias parameters. This idea naturally extends to other linear layers and simply involves adding a learnable constant to the input.

General Linear Layers. The implementation scheme described above for fully connected layers can be extended to general linear layers, which may include layers such as batch normalization, average pooling, or cascades of multiple linear transformations. Consider a more general linear operator $\mathcal{T}_W : \mathcal{V} \to \mathbb{R}^n$, where $\mathcal{V} \subseteq \mathbb{R}^{p_1\times p_2\times\cdots\times p_v}$, and W denotes the parameters associated with the operator $\mathcal{T}_W$. Let e_j denote the j-th canonical basis vector in $\mathbb{R}^n$. Then the j-th component of the output of $\mathcal{T}_W$ can be expressed as

$$\langle \mathcal{T}_W(X), e_j\rangle = \langle X, \mathcal{T}_W^*(e_j)\rangle, \tag{66}$$

where $\mathcal{T}_W^*$ is the Hermitian adjoint of $\mathcal{T}_W$. According to (66), one can interpret $\|\mathcal{T}_W^*(e_j)\|_{HS}$, the Hilbert–Schmidt norm of $\mathcal{T}_W^*(e_j)$, as the norm of the j-th "row" of the operator $\mathcal{T}_W$. Letting $\mathcal{G}_i$, for $i = 1, \ldots, N$, be i.i.d. Gaussian tensors of the same shape as X, a generalization of (64) takes the form

$$\frac{2}{\pi}\texttt{arcsin}\left(d_{W,X}^{\odot -1} \odot \mathcal{T}_W(X)\right) \xrightarrow{\text{EHD}} \frac{1}{N}\sum_{i=1}^{N} \text{sign}\left(\langle \mathcal{G}_i, X\rangle\right) \texttt{sign}\left(\mathcal{T}_W\left(\mathcal{G}_i\right)\right), \tag{67}$$

where $\odot$ denotes a Hadamard product, $d_{W,X}^{\odot -1}$ denotes the Hadamard inverse[2] of the vector $d_{W,X}$, and

$$d_{W,X} = \|x\|_{HS} \begin{pmatrix} \|\mathcal{T}_W^*(e_1)\|_{HS} \\ \vdots \\ \|\mathcal{T}_W^*(e_n)\|_{HS} \end{pmatrix}.$$

Equation (67) shows that, once the Hermitian adjoint of a linear operator is available, normalizing the operator and constructing the hyperdimensional embedding become straightforward tasks. For a simpler notation of the canonical basis, the range of $\mathcal{T}_W$ is considered to be $\mathbb{R}^n$. However, this framework readily generalizes to multi-dimensional output arrays, in which case $d_{W,X}$ is replaced by $D_{W,X}$, a tensor with the same dimensions as $\mathcal{T}_W(X)$.

Convolutional Layers. Although convolutional layers are linear and can, in principle, be addressed using the formulation from the previous section, we introduce a specialized approach tailored for a more compact representation and improved concentration properties. For notational simplicity, we focus on one-dimensional convolution with a single output channel; the generalization to higher dimensions and multiple channels is straightforward. Consider $x \in \mathbb{R}^p$ as an input array, $w \in \mathbb{R}^n$ a convolutional filter, and let $*$ denote a vector-wise convolution (or cross-correlation). The convolution $w * x$ is a vector of length m, where m depends on input parameters such as p, n, padding, and stride. Each vector-wise convolution $w * x$ can be expressed as a matrix-vector product:

$$w * x = \begin{pmatrix} w^\top P_1^\top \\ \vdots \\ w^\top P_m^\top \end{pmatrix} x, \tag{68}$$

where $P_k \in \{0,1\}^{p \times n}$, $k \in \{1, \ldots, m\}$, circularly shift the elements of w. While the explicit construction of P_k is not required, the row norms of the matrix in (68) can be efficiently obtained via a simple convolution operation as follows:

$$\begin{pmatrix} \|P_1 w\|_2^2 \\ \vdots \\ \|P_m w\|_2^2 \end{pmatrix} = (w \odot w) * 1_p,$$

where 1_p denotes a length-p vector of all ones. Following the base equation in (64), for $g_i \sim \mathcal{N}(0, I_p)$, the embedding takes the following form

$$\frac{2}{\pi}\texttt{arcsin}\left(D_{w,x}^{-\frac{1}{2}}(w * x)\right) \xrightarrow{\text{EHD}} \frac{1}{N}\sum_{i=1}^{N} \text{sign}\left(\langle g_i, x\rangle\right) \texttt{sign}\left(w * g_i\right), \tag{69}$$

where

$$D_{w,x} = \|x\|_2^2 \,\text{diag}\left((w \odot w) * 1_p\right).$$

One can use the commutative property of the convolution and write

$$w * x = x * w = \begin{pmatrix} x^\top P_1 \\ \vdots \\ x^\top P_m \end{pmatrix} w,$$

which for $g_i \sim \mathcal{N}(0, I_n)$ offers the following alternative embedding scheme

$$\frac{2}{\pi}\texttt{arcsin}\left(D_{w,x}^{-\frac{1}{2}}(w * x)\right) \xrightarrow{\text{EHD}} \frac{1}{N}\sum_{i=1}^{N} \text{sign}\left(\langle g_i, w\rangle\right) \texttt{sign}\left(x * g_i\right), \tag{70}$$

[2]The Hadamard inverse of a vector d is acquired by inverting each element of d, i.e., $[d^{\odot -1}]_j = 1/d_j$.

where

$$D_{w,x} = \|w\|_2^2 \operatorname{diag}\left((x \odot x) * 1_n\right).$$

In convolutional layers, it is typically the case that $n < p$, which makes (70) preferable over (69) due to its faster concentration with increasing N.

For multi-channel input signals, consider an input vector of length p with n_c channels, and learnable weights of length n corresponding to each input channel. We use the notation

$$X = [x_1, \ldots, x_{n_c}] \in \mathbb{R}^{p \times n_c}, \qquad W = [w_1, \ldots, w_{n_c}] \in \mathbb{R}^{n \times n_c}.$$

In this case, the output of the convolutional layer is a vector of length m which is computed via

$$W \circledast X = \sum_{j=1}^{n_c} w_j * x_j.$$

For independent Gaussian matrices $\mathcal{G}_i \in \mathbb{R}^{n \times n_c}$, a generalization of (70) to the multi-channel input case is offered through

$$\frac{2}{\pi}\texttt{arcsin}\left(D_{W,X}^{-\frac{1}{2}}\left(W \circledast X\right)\right) \xrightarrow{\text{EHD}} \frac{1}{N}\sum_{i=1}^{N} \operatorname{sign}\left(\langle \mathcal{G}_i, W\rangle\right) \texttt{sign}\left(X \circledast \mathcal{G}_i\right), \tag{71}$$

where

$$\begin{aligned} D_{W,X} &= \left(\sum_{j=1}^{n_c} \|w_j\|_2^2\right) \operatorname{diag}\left(\left(\sum_{j=1}^{n_c} x_j \odot x_j\right) * 1_n\right) \\ &= \|W\|_F^2 \operatorname{diag}\left((X \odot X) \circledast 1_{n \times n_c}\right). \end{aligned}$$

Network Output Layer. In this section, we describe how to configure the network's output layer to function either as a regressor or a classifier.

For regression, we can model the final layer as

$$y_L \;=\; \texttt{ASU}\left(D_{W_L, y_{L-1}}^{-\frac{1}{2}} W_L y_{L-1}\right),$$

where $D_{W_L, y_{L-1}}$ is given by (65), and the subsequent hyper-dimensional embedding follows the standard procedure. To employ this structure, the only requirement is to rescale the response variable to the interval $\left[-\frac{2}{\pi}, \frac{2}{\pi}\right]$ so that its range is consistent with y_L above.

For classification G-Nets, we can still use a standard soft-max activation as

$$y_L \;=\; \texttt{SoftMax}\left(D_{W_L, y_{L-1}}^{-\frac{1}{2}} W_L y_{L-1}\right). \tag{72}$$

Since the ASU function is monotonic and preserves the position of the dominant component in an array, it is omitted from (72), yet a standard hyperdimensional embedding as the right-hand side of (65) can be used. The EHD G-Net predicted label is determined by identifying the index of the maximum absolute component in the output vector.

O Extended Numerical Experiments

In this section, we conduct numerical experiments on several datasets commonly used as benchmarks in HDC classification tasks [32], focusing on more challenging vision datasets such as MNIST, FashionMNIST, and CIFAR10. We also include two human-activity recognition datasets, HAR-WSS (walking, sitting, standing) [65] and Epilepsy [70]; an automotive dataset, Ford-A [40]; a natural-language dataset, AG News [72]; and a time-series dataset, Fault Detection-A [64]. A key characteristic of all datasets selected for evaluation is that, unlike simpler HDC datasets such as European Languages [52] and ISOLET [45], which achieve high accuracy with linear classifiers like support vector machines (SVMs), the chosen benchmarks are not easily addressed by such models. In fact, many HDC methods struggle to perform well on these datasets. To assess the effectiveness of

the proposed framework, we compare its performance against several established and recent methods, including classic HDC [8], HoloGN [23], Laplace HDC [26], OnlineHD [10], and RFF-HDC [37].

Table 1 summarizes the datasets used, including the number of input features (n_0), output classes (n_L), and the corresponding G-Net architectures. We aimed to vary the layer configurations across datasets while maintaining simplicity and reproducibility. For instance, on the FashionMNIST dataset, we employed a single convolutional layer to stress-test G-Net under a shallower architecture. A demo of G-Net training/EHD conversion, along with all experiments reported in this section, is available at https://github.com/GNet2025/GNet.

Table 1: Dataset and G-Net architecture specifications. In layer descriptions: CN denotes a convolutional layer (input channels, filters, output channels), EM an embedding layer (vocabulary size, number of outputs), FC a fully connected layer (number of outputs), and CL a classification layer (number of classes)

MNIST	**FashionMNIST**	**Ford-A**	**WSS**	**Epilepsy**	**CIFAR10**	**AG News**	**Fault Detection-A**
Data Specifications							
$n_0 = 28\times 28$ $n_L = 10$	$n_0 = 28\times 28$ $n_L = 10$	$n_0 = 500$ $n_L = 2$	$n_0 = 206\times 3$ $n_L = 6$	$n_0 = 206\times 3$ $n_L = 4$	$n_0 = 32\times 32\times 3$ $n_L = 10$	$n_0 = (400, 512)$ $n_L = 4$	$n_0 = 5120$ $n_L = 3$
G-Net Specifications							
CN(1,32,3) CN(32,64,5) FC(512) CL(10)	CN(1,32,5) FC(512) CL(10)	CN(1,16,15) CN(16,16,15) CN(16,25,13) CL(2)	CN(3,64,11) CN(64,48,7) CL(6)	CN(3,64,11) CN(64,48,7) CL(4)	CN(3,32,5) CN(32,64,3) FC(512) CL(10)	EM(44120,512) CN(512,64,{5,6,7})$\times$ 2 CL(4)	CN(1, 3, 64, 9) CN(64, 48, 9) CL(3)

The first set of experiments involves fitting a G-Net to the original training data, followed by evaluation in the binary hyperspace by applying the corresponding EHD G-Net to the hyperdimensional embedding of test data. The G-Net training is quick—about ten epochs on MNIST require roughly 48 seconds on a desktop computer equipped with a GeForce RTX 4090 GPU. The EHD conversion of the same G-Net takes less than 0.5 seconds, and the inference time for each embedded sample is in the order of milliseconds. Figure 6 reports the average test accuracies of EHD G-Net and other HDC methods. The reported hyperdimension N represents the average N_ℓ across G-Net layers and the dimension used in other methods. For each N, the conversion of the reference G-Net to an EHD G-Net was repeated multiple times with different random matrices; the same number of repetitions was applied to other HDC techniques. The plots show the resulting mean accuracies along with ± 1 standard deviation.

The G-Net architecture used in Figure 6 is a RASU network, and the results, both for Gaussian and Rademacher embeddings are reported. While Gaussian embedding yields slightly better results, the accuracies of the two embedding methods are very close. One observes that without training a large binary model—and by training only a compact network in the primal space followed by inexpensive binary encoding—we achieve classifiers that outperform state-of-the-art HDC models by a significant margin. For example, HDC accuracies exceed 99.2% on MNIST, 81% on CIFAR-10, and 91% on FashionMNIST, rivaling real-valued convolutional neural networks. The reported accuracies can be even further improved by employing larger G-Nets and higher hyperdimensions, as EHD G-Net performance asymptotically approaches that of the original G-Net with increasing N.

We extend our study of the first four datasets under the same experimental setup described above. While Figure 6 presents the performance of RASU G-Nets with both Rademacher and Gaussian embeddings, Figure 7 provides a more comprehensive comparison across a broader range of hyperdimensions for both activation types. Specifically, it presents G-Net and EHD G-Net accuracies of RASU and TASU networks, evaluated with Rademacher and Gaussian embeddings. As expected, RASU networks generally attain higher G-Net accuracies, with faster EHD G-Net concentration. However, TASU still remains competitive—achieving, for instance, 98.4% accuracy on MNIST—although, as theory predicts, TASU-based EHD G-Nets converge more slowly to their G-Net baselines. This modest accuracy gap is the trade-off for a fully binary inference pipeline. Notably, the same constraint also gives TASU networks greater robustness to model perturbations, as will be discussed below.

As discussed earlier, the proposed framework develops a base G-Net model, from which combinatorially many BNNs can be instantiated. Although the frameworks and objectives differ, it is natural to ask how our approach relates to existing BNN methods. To this end, Figure 8 compares G-Net

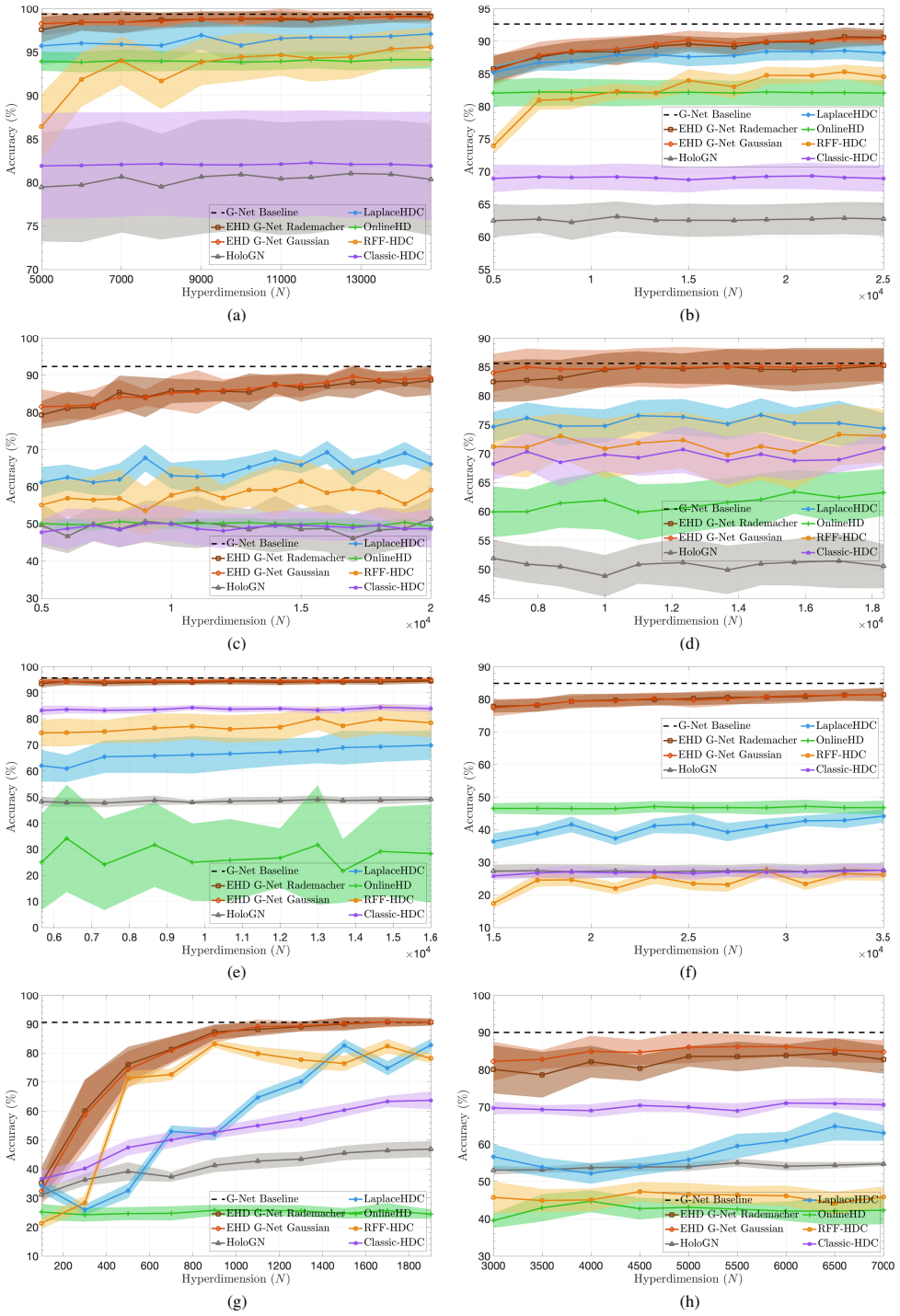


Figure 6: Comparison of Gaussian/Rademacher RASU G-Net with other HDC methods on different datasets (a) MNIST, (b) FashionMNIST, (c) Ford-A, (d) WSS, (e) Epilepsy, (f) CIFAR10, (g) AG News, (h) Fault Detection-A

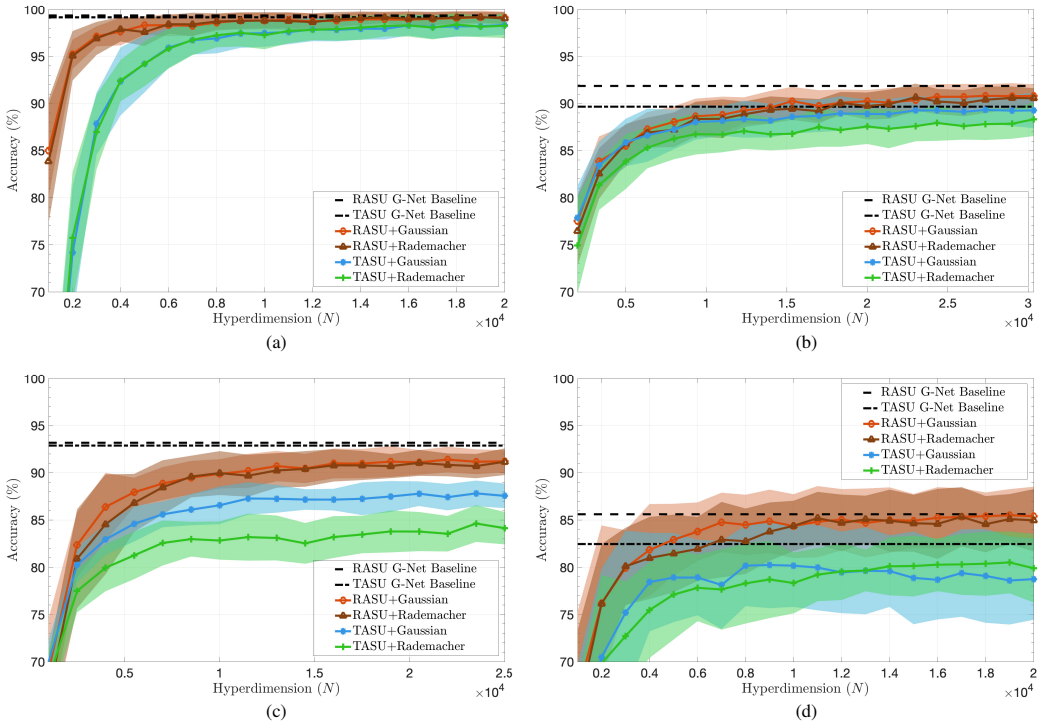


Figure 7: Performance of TASU versus RASU networks for Gaussian and Rademacher Embedding: (a) MNIST, (b) FashionMNIST, (c) Ford-A, (d) WSS

with XNOR-Net [28] and BinaryConnect [46]. A direct, fair comparison with hybrid designs such as Bi-Real Net [21] is not feasible, as those models interleave binary and floating-point operations. To conduct a fair comparison, we first train a fully connected G-Net with two hidden layers (each of width 256) on FashionMNIST and then form a binary EHD G-Net using a hyperdimensional embedding of size $N = 1000$. We subsequently fine-tune this model with either BNN technique and benchmark it against a similar BNN initialized with random weights. Figure 8 shows that EHD-initialized BNNs start with substantially higher accuracy and attain their peak within a few epochs, whereas standard BNNs typically require many more epochs to reach comparable accuracy. From an optimization standpoint, G-Net allows performing a major part of the optimization in a smaller, continuous space, without the challenge of working with binary decision variables. G-Net accuracy serves as a baseline for the binary model, indicating how far its performance can be pushed in the course of training.

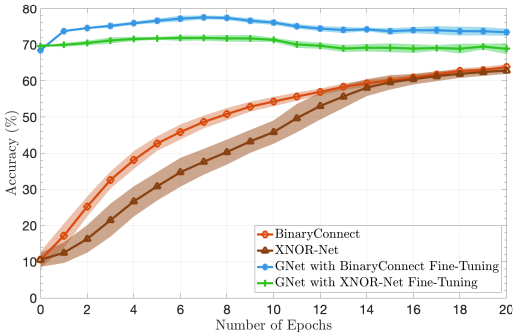


Figure 8: Comparison with BNN Methods

Hyperdimensional models are known for their resilience to model corruption, and our G-Net variants are no exception. To quantify this robustness, we stress-test RASU and TASU EHD G-Nets by randomly flipping a fraction of binary weights in every layer. For instance, applying a 10% flip rate to the MNIST model randomly inverts 10% of the weights in each of its four layers. Figure 9 plots the resulting accuracy degradation versus corruption level, with confidence regions created by repeating the experiments twenty times. Notably, TASU networks lose accuracy more gradually than their RASU counterparts. This stems from TASU's architecture: each TASU output is the sign of an inner product between two binary vectors, a quantity less sensitive to individual bit flips than the zero-thresholded binary inner products used in RASU layers. As a result, even after flipping 35% of the EHD G-Net weights, the TASU model on MNIST still maintains over 95% accuracy.

Because each HDC method employs its own hyperdimensional embedding and inference pipeline, the weight-flipping corruption described above cannot be applied uniformly across other HDC

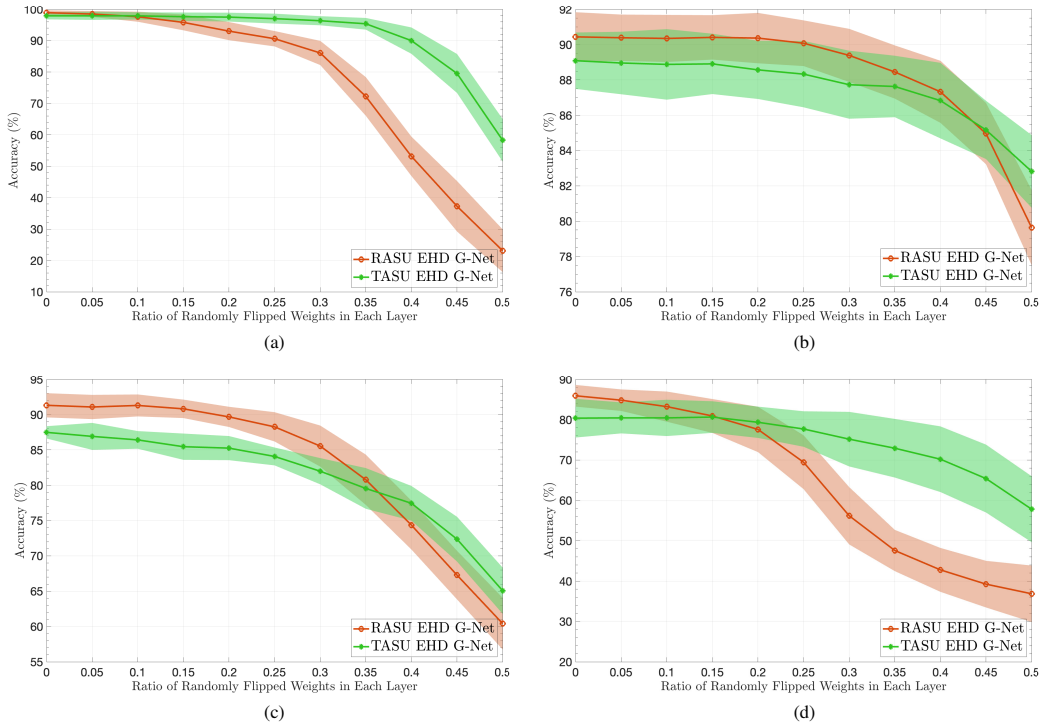


Figure 9: Robustness of TASU and RASU networks against random bit flips across all layers: (a) MNIST, (b) FashionMNIST, (c) Ford-A, (d) WSS

frameworks. Instead, we observe a shared step across all models: each generates a hypervector that is subsequently fed into an inference block for prediction. To apply a common corruption pattern across all the HDC models in our comparison study, we fix the hyperdimension and corrupt the hypervectors by flipping a specified fraction of bits immediately after the embedding stage of each method. The noisy hypervector is then fed to the inference block for prediction, and the mean accuracy across multiple experiments is reported. For EHD G-Nets, bit flips are injected right after the first random-sign embedding, and the corrupted embedding is propagated through the network. Figure 10 plots accuracy degradation for the various HDC models under this hypervector corruption. The TASU and RASU networks had a less noticeable performance gap under this test, and only TASU accuracies are reported in these plots. We attribute the superior robustness of EHD G-Nets to their tightly integrated, binary, multi-stage inference pipeline—a feature largely absent from existing HDC architectures.

P Computational Cost of EHD G-Net versus a Floating Point G-Net

In this section, we mathematically compare the computational cost, in terms of time and memory, of EHD G-Net versus the corresponding floating-point G-Net. In a sense, any floating-point network can be considered as a binary neural network by viewing the floating-point weights as bit sequences, which are operated on by floating-point arithmetic operators. However, unlike the EHD G-Net, these bit sequences would have significant bits, and the operations on these bit sequences are complicated. In contrast, the EHD G-Net uses simple operations and has no significant bits; consequently, the EHD G-Net architecture is robust to random bit flips, see Figure 9. On the other hand, floating-point numbers have significant bits (such as those encoding the sign and exponent of a floating-point number, which significantly change the output when changed). Thus, the EHD G-Net architecture is more suitable for noisy computing environments, such as low-power computing.

It is still instructive to study the different computational costs, in terms of memory and time, of each method. For simplicity, we restrict our attention to a single fully-connected EHD G-Net layer using a Rademacher matrix and the corresponding floating G-Net layer.

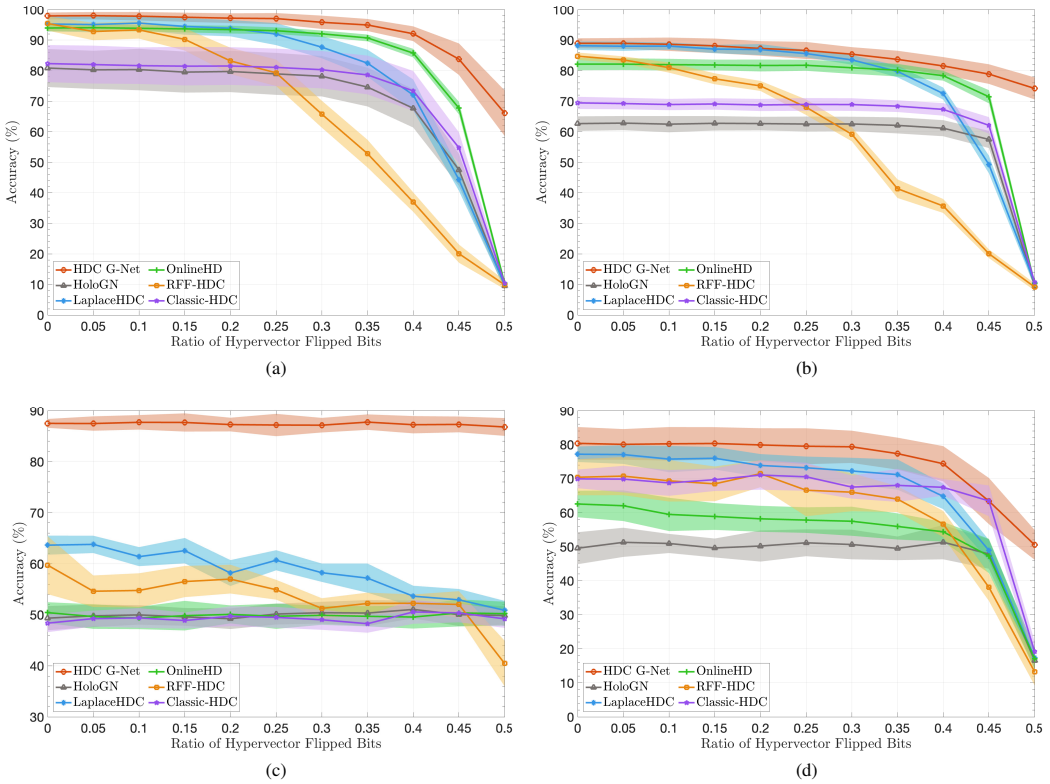


Figure 10: Comparing the robustness of an EHD G-Net with other HDC methods: (a) MNIST, (b) FashionMNIST, (c) Ford-A, (d) WSS

Fix a hyperdimension N, and the number of bits used in the floating-point representation of numbers F (for example, $F = 64$ for double precision floating point numbers). Let n and m be the input and output dimensions, respectively. Assume that $x \in \mathbb{R}^n$, $W \in \mathbb{R}^{m \times n}$, and $\mathcal{R} \in \{-1, 1\}^{N \times n}$ the hyper-dimension is N. Consider the EHD G-Net Layer that performs the map

$$x \mapsto \texttt{sign}(W\mathcal{R}^\top)\texttt{sign}(\mathcal{R}x)$$

and the corresponding floating-point G-Net layer

$$x \mapsto \frac{2}{\pi}\texttt{arcsin}(Wx).$$

The EDH G-Net must store $\texttt{sign}(W\mathcal{R}^\top)$ which has dimensions $m \times N$ and $\mathcal{R}$, which has dimensions $N \times n$. The entries of these matrices can be encoded in binary, so the total memory required to store the EHD G-Net layer is

$$\text{memory}_{\text{EHD G-Net}} = (m + n)N \quad \text{bits.}$$

The floating point (FP) G-Net involves storing the $m \times n$ matrix W whose entries each require F bits to encode, for a total of

$$\text{memory}_{\text{FP G-Net}} = (m \cdot n)F \quad \text{bits.}$$

Next, we consider the computational cost in terms of time. The EHD G-Net layer starts by applying $\mathcal{R}$ to x and then applying the sign function. This involves an XOR of sign bits, adding the resulting integers, and taking the sign of the result. Let t_{XOR} and $t_{\mathbb{Z},+}$ denote the time to compute the XOR and add integers, respectively. Finally, let t_{sign} be the time to take the sign. After that, we need to compute t_{XOR} of Nm entries, and then sum Nm binary value, which can be done with popcount. Let t_{popcount} denote the time for popcount (summing binary values). In summary, we have

$$\text{time}_{\text{EHD G-Net}} = \mathcal{O}\Big((t_{\text{xor}} + t_{\text{popcount}})(Nm) + (t_{\text{xor}} + t_{\mathbb{Z},+})(Nn) + t_{\text{sign}}(N)\Big).$$

Next, consider the floating-point G-Net layer. Let t_{arcsin} be the cost to apply $2/\pi \arcsin$, $t_{\mathbb{R},+}$ be the time to add floating point numbers, and $t_{\mathbb{R},\times}$ be the time to multiply floating point numbers. Then, we have

$$\text{time}_{\text{FP G-Net}} = \mathcal{O}\Big((m)t_{\text{arcsin}} + (m \cdot n)(t_{\mathbb{R},+} + t_{\mathbb{R},\times})\Big).$$

If N and F are fixed, along with the time cost of all operations, then the computational complexity of EHD G-Net is $\mathcal{O}(m+n)$ in terms of memory and time complexity compared to $\mathcal{O}(m \cdot n)$ for floating point G-Net. However, even when m and n are small, we emphasize that EHD G-Net offers an advantage in terms of robustness to random bit flips. Figure 11 replicates the first row of Figure 9 where we previously showed that randomly flipping the binary weights of an EHD G-Net causes a slow drop of accuracy. However, if instead one considers randomly flipping bit sequences on a floating point G-Net, the accuracy drops much faster, as demonstrated in Figure 11.

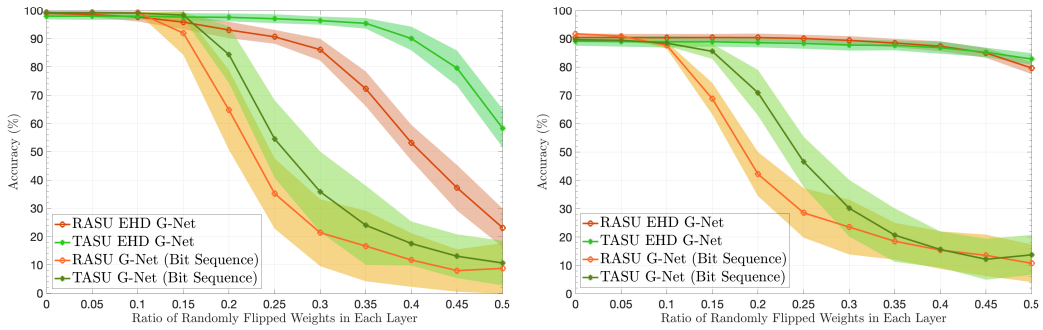


Figure 11: Robustness of TASU/RASU networks against random flips of EHD G-Net weights, and bit sequences of the corresponding G-Net: (a) MNIST, (b) FashionMNIST

Appendix References

[40] A. Bagnall. Dataset: FordA. https://www.timeseriesclassification.com/description.php?Dataset=FordA.

[41] Yogesh J Bagul and Ramkrishna M Dhaigude. Simple efficient bounds for arcsine and arctangent functions. *Research Square Preprint*, 2021.

[42] Grahame Bennett, Victor Goodman, and Charles Newman. Norms of random matrices. *Pacific Journal of Mathematics*, 59(2):359–365, 1975.

[43] Cheng-Yang Chang, Yu-Chuan Chuang, En-Jui Chang, and An-Yeu Andy Wu. Multa-hdc: A multi-task learning framework for hyperdimensional computing. *IEEE transactions on computers*, 70(8):1269–1284, 2021.

[44] Cheng-Yang Chang, Yu-Chuan Chuang, and An-Yeu Andy Wu. Ip-hdc: Information-preserved hyperdimensional computing for multi-task learning. In *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 1–6. IEEE, 2020.

[45] Ron Cole, Yeshwant Muthusamy, and Mark Fanty. *The ISOLET spoken letter database*. Oregon Graduate Institute of Science and Technology, Department of Computer . . . , 1990.

[46] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. *Advances in neural information processing systems*, 28, 2015.

[47] E. Paxon Frady, Denis Kleyko, Christopher J. Kymn, Bruno A. Olshausen, Friedrich T. Sommer, Murat Okandan, and James B. Aimone. Computing on functions using randomized vector representations (in brief). In *Neuro-Inspired Computational Elements Conference*, pages 115–122. ACM, 2022.

[48] Ross W. Gayler. Multiplicative binding, representation operators, and analogy. *Advances in Analogy Research: Integration of Theory and Data from the Cognitive, Computational, and Neural Sciences*, pages 1–4, 1998.

[49] Olivier Guédon, Aicke Hinrichs, Alexander E Litvak, and Joscha Prochno. On the expectation of operator norms of random matrices. In *Geometric Aspects of Functional Analysis: Israel Seminar (GAFA) 2014–2016*, pages 151–162. Springer, 2017.

[50] Alejandro Hernandez-Cano, Cheng Zhuo, Xunzhao Yin, and Mohsen Imani. Reghd: Robust and efficient regression in hyper-dimensional learning system. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 7–12. IEEE, 2021.

[51] Mohsen Imani, Samuel Bosch, Sohum Datta, Sharadhi Ramakrishna, Sahand Salamat, Jan M Rabaey, and Tajana Rosing. Quanthd: A quantization framework for hyperdimensional computing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(10):2268–2278, 2019.

[52] Aditya Joshi, Johan T Halseth, and Pentti Kanerva. Language geometry using random indexing. In *International Symposium on Quantum Interaction*, pages 265–274. Springer, 2016.

[53] Denis Kleyko, Evgeny Osipov, Nikolaos Papakonstantinou, Valeriy Vyatkin, and Arash Mousavi. Fault detection in the hyperspace: Towards intelligent automation systems. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pages 1219–1224. IEEE, 2015.

[54] B. Komer. *Biologically Inspired Spatial Representation*. PhD thesis, University of Waterloo, 2020.

[55] Rafal Latala and Marta Strzelecka. Operator $\ell_p \to \ell_q$ norms of gaussian matrices, 2025.

[56] Duy H. Le and Tuan V. Pham. Improving bi-real net with block-wise quantization and multiple-steps binarization on activation. In *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–6. IEEE, 2020.

[57] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE transaction on neural networks and learning systems*, 33(12):6999–7019, 2022.

[58] Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. *Advances in neural information processing systems*, 30, 2017.

[59] Zechun Liu, Wenhan Luo, Baoyuan Wu, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Binarizing deep network towards real-network performance. *International journal of computer vision*, 128(1):202–219, 2020.

[60] Alisha Menon, Daniel Sun, Melvin Aristio, Harrison Liew, Kyoungtae Lee, and Jan M. Rabaey. A highly energy-efficient hyperdimensional computing processor for wearable multi-modal classification. In *2021 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2021.

[61] Samet Oymak and Ben Recht. Near-optimal bounds for binary embeddings of arbitrary sets. *arXiv preprint arXiv:1512.04433*, 2015.

[62] T.A. Plate. Holographic reduced representations. *IEEE transactions on neural networks*, 6(3):623–641, 1995.

[63] Tony A Plate. *Holographic Reduced Representation: Distributed representation for cognitive structures*, volume 150. CSLI Publications, 2003.

[64] Paul Rabich. Dataset: Fault Detection-A. https://www.timeseriesclassification.com/description.php?Dataset=FaultDetectionA.

[65] Paul Rabich. Dataset: Walkingsittingstanding. https://www.timeseriesclassification.com/description.php?Dataset=WalkingSittingStanding.

[66] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[67] Martin Raič. A multivariate berry–esseen theorem with explicit constants. *Bernoulli*, 25(4A), November 2019.

[68] Ratshih Sayed, Haytham Azmi, Heba Shawkey, A. H. Khalil, and Mohamed Refky. A systematic literature review on binary neural networks. *IEEE access*, 11:1–1, 2023.

[69] Pere Vergés, Mike Heddes, Igor Nunes, Tony Givargis, and Alexandru Nicolau. Hdcc: A hyperdimensional computing compiler for classification on embedded systems and high-performance computing, 2023.

[70] Jose Ramon Villar. Dataset: Epilepsy. https://www.timeseriesclassification.com/description.php?Dataset=Epilepsy.

[71] Chunyu Yuan and Sos S. Agaian. A comprehensive review of binary neural network. *The Artificial intelligence review*, 56(11):12949–13013, 2023.

[72] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

[73] Wenyu Zhao, Teli Ma, Xuan Gong, Baochang Zhang, and David Doermann. A review of recent advances of binary neural networks for edge computing. *IEEE journal on miniaturization for air and space systems*, 2(1):25–35, 2021.

[74] Shien Zhu, Luan H. K. Duong, and Weichen Liu. Xor-net: An efficient computation pipeline for binary neural network inference on edge devices. In *Proceedings - International Conference on Parallel and Distributed Systems*, pages 124–131. IEEE, 2020.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Supporting theoretical results are stated in Sections 3, 4, and 5, while experiments results are in Section 6.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Each result is precisely stated with all assumptions, and detailed proofs of each theorem and support lemmas and references are provided in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The setup of the experimental results is detailed in the paper and Appendix. A link to a GitHub repository reproducing all the results is included in the abstract. All data sets used for testing are publicly available and referenced appropriately.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code that reproduces all figures and results is linked in the abstract as a public GitHub repository. The Appendix provides the experimental details.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The training and test details are specified in Section O of the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experiment results in Section 6 and Section O of the Appendix include appropriate error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Information about the computation resources used for the experiments is included in Section O of the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This work consists of fundamental research into random binary neural networks and binary hyperdimensional computing models, and has no direct positive or negative societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This work consists of fundamental research into random binary neural networks and binary hyperdimensional computing model and poses no such misuse risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All data sets and code used are publicly available for scientific use, and are appropriately referenced.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code accompanying the paper is documented with instructions about installation and execution steps.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Neither crowdsourcing nor human subjects is involved in the research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Neither crowdsourcing nor human subjects is involved in the research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The research does not involve any LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.