frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 17 September 2025
DOI 10.3389/frai.2025.1637105

OPEN ACCESS

EDITED BY
P. Balasubramaniam,
The Gandhigram Rural Institute, India

REVIEWED BY
Peter Sutor,
University of Maryland, College Park,
United States
Mary Alexandria Kelly,
Carleton University, Canada

*CORRESPONDENCE
Calvin Yeung
chyeung2@uci.edu

RECEIVED 28 May 2025
ACCEPTED 25 August 2025
PUBLISHED 17 September 2025

CITATION
Yeung C, Errahmouni Barkam H, Zou Z, Yun S, Bastian ND and Imani M (2025) Lipschitz-based robustness estimation for hyperdimensional learning.
Front. Artif. Intell. 8:1637105.
doi: 10.3389/frai.2025.1637105

Lipschitz-based robustness estimation for hyperdimensional learning

Calvin Yeung[1]*, Hamza Errahmouni Barkam[1], Zhuowen Zou[1], Sanggeon Yun[1], Nathaniel D. Bastian[2] and Mohsen Imani[1]

[1]Department of Computer Science, University of California, Irvine, Irvine, CA, United States, [2]Department of Electrical Engineering & Computer Science, United States Military Academy, West Point, NY, United States

With the adoption of machine learning models in various practical domains, there is a growing need for evaluating and increasing model robustness. Hyperdimensional computing (HDC) is a neurosymbolic computational paradigm that represents symbols as high dimensional vectors and symbolic operations as vector operations, seamlessly interfacing between neuro- and symbolic components of a model. However, there is a notable gap in HDC research regarding the robustness of HDC models to input perturbations. This study presents a novel theoretical framework tailored to evaluate the robustness of hyperdimensional classifiers against perturbations in the input space. In particular, our proposed measure of robustness gives a theoretical upper bound for the magnitude of noise a model can tolerate without changing its prediction for any given data point. We also propose a method to enhance the robustness of the model based on our proposed measure of robustness. Our approach introduces several methods to calculate model robustness as a function of the specific dataset and type of hyperdimensional encoding used. The results show that the average robustness of HDC models increases under the proposed optimization scheme while maintaining accuracy by varying the variance of the Gaussian distribution used to encode hypervectors. The practical effectiveness of our proposed measure of robustness is also demonstrated.

KEYWORDS

hyperdimensional computing, vector symbolic architectures, robustness, adversarial attacks, classification

1 Introduction

With the adoption of machine learning models in various practical domains, there is a growing need for methods that evaluate and increase model robustness, as models may be susceptible to noise, whether in the model representation or in the model input, due to various reasons such as adversarial attacks or a noisy environment or hardware. For this reason, there has been a plethora of empirical and theoretical studies on the robustness of deep learning models (Cisse et al., 2017; Wong and Kolter, 2018a; Raghunathan et al., 2018; Cohen et al., 2019; Wong and Kolter, 2018b; Zhang et al., 2019).

However, deep learning methods remain difficult to interpret and have difficulty performing symbolic reasoning. Neurosymbolic methods address these gaps by integrating neural networks with a symbolic component. Hyperdimensional computing (HDC) has emerged as a promising neurosymbolic computational paradigm, capable of both machine learning and cognitive reasoning tasks. In HDC, symbols are represented as high-dimensional vectors called hypervectors. Symbolic operations thus correspond to vector operations such as bundling and binding. The vector representation of symbols in

HDC provides a natural interface with deep networks. As summarized in Figure 1, HDC encodes data as hypervectors (Figure 1A) and manipulates them with symbolic operations such as bundling and binding (Figure 1B), yielding models that are robust and interpretable (Figure 1C).

HDC has been successfully applied to a variety of machine learning tasks, including classification and regression, and has shown to have performance comparable to neural networks on simple tasks while achieving higher noise tolerance to perturbations in model representations, higher transparency, lower power consumption, and intrinsic one- or few-shot learning capabilities (Kleyko et al., 2023; Kymn et al., 2024; Hernández-Cano et al., 2021; Poduval et al., 2022; Yeung et al., 2025). Despite the success of HDC in machine learning tasks, most work in applying HDC to machine learning has focused on model performance in terms of accuracy metrics. While there have been theoretical studies on the robustness of HDC models to noise in hyperdimensional space (Thomas et al., 2022), there is a lack of theoretical work on that in the input space.

This study aims to fill this gap by providing theoretically supported robustness measures in HDC classifiers. In particular, in this work, we focus on the robustness of a model to perturbations on the input space. To this end, we use a Lipschitz-based approach to derive robustness estimates for HDC classifiers. Our work provides a first-of-its-kind theoretical framework for evaluating model robustness to input perturbations in HDC and a novel method for learning encodings that are more robust. Our results demonstrate that our method can be used effectively to derive robustness estimates for HDC classifiers and that our method for learning more robust encodings can improve the robustness of these models. This study makes several key contributions: (1) We propose a first-of-its-kind method that characterizes theoretical per-point robustness for HDC classifiers based on Lipschitz continuity; (2) We introduce two methods for estimating the Lipschitz constant, with one that gives more liberal estimates and another that gives more conservative estimates; (3) We provide a general framework for learning a robust encoding based on the estimates of robustness we have derived.

2 Related work

In the deep neural network (DNN) literature, there is a body of work that explores the kind of robustness we are concerned with in this study, i.e., the robustness of classifiers to input perturbations from an adversarial perspective (Cisse et al., 2017; Wong and Kolter, 2018a; Raghunathan et al., 2018; Cohen et al., 2019; Wong and Kolter, 2018b; Zhang et al., 2019). In particular, multiple studies have explored the certified robustness of classifiers, which provides theoretical estimates for the robustness of DNNs to norm-bounded perturbations in the input. Cisse et al. (2017) give an upper bound for the generalization bound on the loss function that accounts for adversarial examples with perturbations up to a given magnitude in terms of the Lipschitz constant of the neural network and proposes Parseval regularization, which constrains the Lipschitz constant of each hidden layer to be less than one, to increase the robustness of the DNN.

In the HDC literature, there have been more studies investigating the robustness of HDC models to perturbations in some parts of the model representation (Thomas et al., 2022; Zhang et al., 2021; Rahimi et al., 2016; Matsui et al., 2023). Zhang et al. (2021) explore the robustness of HDC models to errors in associative memory by injecting errors into class hypervector representations and measuring the degradation of model accuracy. Thomas et al. (2022) give HDC a theoretical treatment and explore the robustness of HDC models to perturbations in hyperspace for decoding and learning tasks but do not provide a treatment for perturbations in the input space.

To the best of our knowledge, our work is the first of its kind to investigate the theoretical robustness of HDC classifiers to perturbations in the input space. At a high level, it is inspired

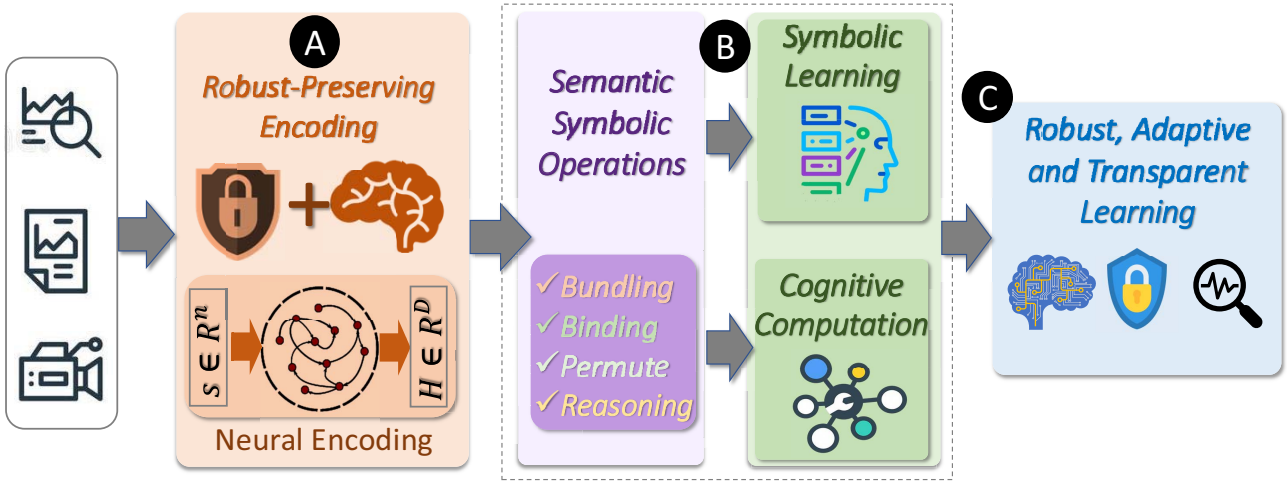

FIGURE 1
HDC encodes data and symbols as hypervectors **(A)** which can be manipulated using vector operations in a symbolic manner **(B)**, providing robustness and interpretability **(C)**.

by works in certified defenses in DNNs but is explicitly catered to HDC, using HDC theory in the analysis.

3 Background

HDC, also known as vector symbolic architecture, is a computing framework based on properties of high dimensional vectors. The fundamental unit in HDC is a high dimensional vector, also called a hypervector. A hypervector $\mathcal{H}$ lives in some hyperspace H, e.g., $\mathbb{R}^D$ for D large, and, together with some operations in hyperspace, form an algebra over vectors. Generally, there are two types of hypervectors: (1) base hypervectors, which are generated stochastically, e.g., $\mathcal{H} \sim \mathcal{N}(0, I)$; and (2) composite hypervectors, which are created by combining hypervectors using a variety of operations. These hypervectors can be compared via a similarity operation $\delta(\mathcal{H}_1, \mathcal{H}_2)$. In this work, we are mainly concerned with the inner product as a similarity measure. Generally, basic hypervectors are generated to be mutually dissimilar; i.e., quasi-orthogonal.

The three main operations in HDC, namely, bundling, binding, and permutation, can be characterized by how it affects the similarity of hypervectors. Bundling, denoted as $+$, is typically implemented as element-wise addition. If $\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2$, then both $\mathcal{H}_1$ and $\mathcal{H}_2$ are similar to $\mathcal{H}$. From a cognitive perspective, it can be interpreted as memorization. Binding, denoted as $*$, is typically implemented as element-wise multiplication. If $\mathcal{H} = \mathcal{H}_1 * \mathcal{H}_2$, then $\mathcal{H}$ is dissimilar to both $\mathcal{H}_1$ and $\mathcal{H}_2$. Binding also has the important property of similarity preservation in the sense that for some hypervector $\mathcal{V}$, $\delta(\mathcal{V} * \mathcal{H}_1, \mathcal{V} * \mathcal{H}_2) \simeq \delta(\mathcal{H}_1, \mathcal{H}_2)$. From a cognitive perspective, it can be interpreted as association. Permutation, denoted as ρ, is typically implemented as a rotation of vector elements. Generally, $\delta(\rho(\mathcal{H}), \mathcal{H}) \simeq 0$. Permutation is usually used to encode order in sequences.

It is important to note that the description above of HDC is general; there are various specific realizations of HDC with the above properties. The HDC framework gives several benefits, including robustness to noise in hyperspace, transparency, and parallelization.

3.1 Hyperdimensional learning

In this subsection, we discuss learning in HDC, focusing on classification. To adapt HDC to the task of learning from some dataset $\mathcal{D} \subset U$, where U is the input space, we must define an encoding $\phi : U \to H$ that preserves some notion of similarity in the input space. Thus, given some input $x, y \in U$, $\phi(x), \phi(y)$ are their corresponding hypervector, and $\phi(x)$ is similar to $\phi(y)$ if and only if x is similar to y.

Suppose our dataset $\mathcal{D}$ consists of m classes $C_1, C_2, ..., C_m$, where $C_i = \{x_1^{(i)}, x_2^{(i)}, ..., x_{N_i}^{(i)}\}$ for $1 \leq i \leq m$. Based on these classes, We can define a class hypervector $\phi(C_i)$[1] for $1 \leq i \leq m$. A simple way to form a class hypervector is to simply bundle all hypervectors corresponding to elements in the class; i.e., $\phi(C_i) = \sum_{x \in C_i} \phi(x)$ for $1 \leq i \leq m$. There are various other ways of forming class hypervectors which have the general form of a weighted sum $\phi(C_i) = \sum_{k=1}^{N_i} \gamma_k^{(i)} \phi(x_k^{(i)})$, where $\gamma_k^{(i)} \in \mathbb{R}$ for all $1 \leq k \leq N_i$, for all $1 \leq i \leq m$.

Given some $q \in U$ that we wish to classify, we compare the similarity between its corresponding hypervector $\phi(q)$ and all class hypervectors $\phi(C_1), \phi(C_2), ..., \phi(C_m)$. The class whose corresponding class hypervector has the highest similarity to $\phi(q)$ is designated as the predicted class for q. Thus, the encoding scheme ϕ, dataset $\mathcal{D}$, method of aggregating class hypervectors, and similarity measure $\delta(\cdot, \cdot)$, together define a classification model based on HDC.

4 Estimating hyperdimensional robustness

4.1 Preliminary concepts and definitions

Before we discuss the robustness of such models, we first introduce the Random Fourier Feature (RFF) Rahimi and Recht (2007) encoding ϕ which is generally useful in the context of learning in HDC as it is an approximation of kernel methods. The RFF encoding is a map $\phi : \mathbb{R}^n \to \mathcal{C}^D$, with $\phi(x) = e^{iMx}$, where each row $M_{i,:} \sim p$ for some distribution p. As noted in the supplements, HDC is a general computation framework with various implementations. In this case, the resulting hypervectors mapped to by the RFF encoding operate under the Fourier Holographic Reduced Representation (FHRR) model of HDC, as FHRR base hypervectors are of the form $e^{i\theta}$, where θ is a column vector such that $\theta_i \sim p$. Thus, using the language of HDC defined in the supplements, assuming each entry $M_{ij} \sim p$, we may also represent the mapping as $\phi(x) = \mathcal{H}_1^{x_1} * \mathcal{H}_2^{x_2} * ... * \mathcal{H}_n^{x_n}$, where $\mathcal{H}_i = e^{iM_{:,i}}$ is an FHRR base hypervector and $M_{:,i}$ is the i-th column of M. The similarity measure we use is the real component of the inner product defined on $\mathcal{C}^D$ which we denote as $\langle \phi(x), \phi(y) \rangle = \Re[\phi(x)^T \phi(y)^*]$, where $\phi(y)^*$ is the complex conjugate of $\phi(y)$. Rahimi and Recht (2007) show, following from Bochner's theorem, that $\langle \phi(x), \phi(y) \rangle / D \approx k(x - y)$, where k is a shift-invariant kernel that is the Fourier transform of distribution p.

Next, before presenting our methods of estimating a model's robustness, we must precisely define what it means.

Definition 1 ((ϵ, q)-Robustness). A classifier f is (ϵ, q)-robust if $f(q) = f(q + \omega)$ for all ω such that $\|\omega\| \leq \epsilon$. Alternatively, to denote the dependence of ϵ on q, we write ϵ_q.

Note that our definition of (ϵ, q)-robustness is given with respect to some choice of norm $\|\cdot\|$.

The concept of (ϵ, q)-robustness is a notion of robustness that is *per data point*. It is easy to see that ϵ_q is the shortest distance from q to a decision boundary of f in the norm of choice. However, there are several practical considerations we have to make:

1. For even input spaces of moderately high dimensions, computing ϵ_q is in general infeasible.
2. In practice, we do not care about the robustness of a model for a single data point. Instead, we care about the robustness of a model for a dataset.

1 Note that we are overloading our notation here, as $C_i \notin U$.

To address point 1, we will estimate ϵ_q using tractable methods. As we shall see in later subsections, there are various ways of estimating ϵ_q that have various levels of complexity and conservativeness. To address point 2, during the evaluation of our methods, rather than considering ϵ_q on a point-by-point basis, we will instead consider $\mathbb{E}_{x\sim\mathcal{D}}[\epsilon_x]$, where $\mathcal{D}$ is the dataset distribution.

In the following subsections, for simplicity, we will consider the binary classification case, although it is easy to extend our results to multi-class classification.

4.2 Linear approximation approach

Suppose we have two classes C_1 and C_2. We denote their corresponding class hypervectors as $\phi(C_1)$ and $\phi(C_2)$, respectively. Suppose we use the inner product as our similarity measure. Let f be the corresponding classifier; i.e., $f(x) = 1$ if $\langle\phi(C_1), \phi(x)\rangle \geq \langle\phi(C_2), \phi(x)\rangle$ and $f(x) = 2$ otherwise. We define the following function:

$$r(x) = \langle\phi(C_1) - \phi(C_2), \phi(x)\rangle. \tag{1}$$

It is evident that $r(x) = 0$ corresponds to the decision boundary of f. Thus, a simple way of estimating the distance of some input q to the decision boundary is to take the linear approximation of r at q and compute its distance to zero. That is,

$$r(q) + \nabla r(q)^T x = 0. \tag{2}$$

Rearranging terms, we get

$$r(q) = -\nabla r(q)^T x. \tag{3}$$

Taking absolute values and applying the Cauchy-Schwarz inequality, we get

$$|r(q)| = |\nabla r(q)^T x| \leq \|\nabla r(q)\|\|x\|. \tag{4}$$

Thus, we get the estimate

$$\epsilon_q \approx \frac{|r(q)|}{\|\nabla r(q)\|}. \tag{5}$$

There are various issues with this approach. First, there is no guarantee that the resulting ϵ_q computed this way satisfies the conditions for (ϵ, q)-robustness. Second, if q is close to some point in $C_1 \cup C_2$, this estimate of ϵ_q is quite likely to be a drastic overestimate as the gradient at those points tends to be close to zero. In this sense, it is a liberal estimate. So, this estimate is only useful for q's close to the decision boundary, which is not very useful.

4.3 Lipschitz-based approach

For some applications, it is important to have a strong theoretical estimate for (ϵ, q)-robustness. The approaches explored in this subsection can achieve this in the ideal case. Central to these approaches is the concept of Lipschitz continuity.

Definition 2 (Lipschitz continuity). A function $f : X \to Y$ is Lipschitz continuous if there is some $L > 0$ such that $\|f(x) - f(y)\|_Y \leq L\|x - y\|_X$, where $\|\cdot\|_X$ and $\|\cdot\|_Y$ are distance measures for X and Y, respectively. The smallest such L is called the Lipschitz constant.

Thus, if a function is Lipschitz continuous, there is a bound on how fast it can change. If we are able to find some L for the function r defined in Equation 1, we can get an estimate for ϵ_q that satisfies (ϵ, q)-robustness. This is formalized by the following proposition.

Proposition 1. If L satisfies the Lipschtiz condition for r, then f is (ϵ, q)-robust for $\epsilon = |r(q)|/L$.

Proof. Suppose L satisfies the Lipschtiz condition for r. Let q be some input query and ω some noise added to q, with $\|\omega\| \leq \epsilon$. Then,

$$|r(q + \omega) - r(q)| \leq L\|\omega\| \leq |r(q)|. \tag{6}$$

Suppose f is not (ϵ, q)-robust. Then, there is some ω with $\|\omega\|_2 \leq \epsilon$ such that $r(q + \omega) \geq 0$ and $r(q) < 0$ or $r(q + \omega) < 0$ and $r(q) \geq 0$. It follows that

$$|r(q + \omega) - r(q)| > |r(q)|, \tag{7}$$

which is a contradiction, thus proving our proposition.

Thus, with this result, our goal now is to find some L satisfying the Lipschitz condition for r to estimate ϵ_q. Assuming r is differentiable, there is a basic result for finding L that follows from the Mean Value Theorem:

Proposition 2. If r is differentiable, then $L = \sup_x \|\nabla r(x)\|$ is its Lipschitz constant.

Taking $L = \sup_x \|\nabla r(x)\|$ is the best we can do, in the sense that it gives us the largest possible estimate for $\epsilon_q = |r(q)|/L$ that is (ϵ_q, q)-robust using our Lipschitz-based approach. This gives us the estimate

$$\epsilon_q \approx \frac{|r(q)|}{\sup_x \|\nabla r(x)\|}. \tag{8}$$

Unfortunately, in practice, solving for the global maximum of $\|\nabla r(x)\|$ is intractable. The best we can do is to solve for local maxima. Thus, the estimate of ϵ_q we obtain in this way does not guarantee (ϵ_q, q)-robustness, but it at least should not give drastic overestimates as in Equation 5. In this way, Equation 8 is a more conservative estimate.

4.4 Conservative Lipschitz-based approach

For this approach, we derive an expression for L that can be more easily computed via an easier optimization problem compared to the previous $L = \sup_x \|\nabla r(x)\|$. The result is given in the following proposition:

Proposition 3. $L = \alpha\|\phi(C_1) - \phi(C_2)\|_H$ satisfies the Lipschitz condition for r, where α satisfies the Lipschitz condition for the encoding ϕ. $\|\cdot\|_H$ refers to the norm defined by the inner product $\langle\cdot,\cdot\rangle$.

Proof. For any inputs x, y,

$$\begin{aligned}\frac{|r(x)-r(y)|}{\|x-y\|} &= \frac{|\langle\phi(C_1)-\phi(C_2),\phi(x)-\phi(y)\rangle|}{\|x-y\|}\\ &\leq \frac{\|\phi(C_1)-\phi(C_2)\|_H\|\phi(x)-\phi(y)\|_H}{\|x-y\|}\\ &\leq \alpha\|\phi(C_1)-\phi(C_2)\|_H\end{aligned}$$

This gives us the estimate

$$\epsilon_q \approx \frac{|r(q)|}{\alpha\|\phi(C_1)-\phi(C_2)\|_H}. \tag{9}$$

Now, we have delegated the problem to computing some α that satisfies the Lipschitz condition for ϕ. Note that Proposition 2 implies that r is scalar-valued, which is not the case for ϕ. Thus, we cannot use it to compute α. Instead, we will compute α on a case-by-case basis for each encoding ϕ.

We show how one can find α for any shift-invariant encoding where encoded hypervectors are of constant length; i.e., any encoding ϕ where $\langle\phi(x_1),\phi(y_1)\rangle = \langle\phi(x_2),\phi(y_2)\rangle$ if $x_1 - y_1 = x_2 - y_2$ and $\|\phi(x)\|_H = K$ for some K, for all x.

Proposition 4. Suppose ϕ is shift-invariant and $\|\phi(x)\|_H = K$ for some K, for all x. Then, α satisfies the Lipschitz condition for ϕ, where

$$\alpha = \sup_x \frac{\sqrt{2(K^2-\langle\phi(x),\phi(0)\rangle)}}{\|x\|}$$

Proof. We want to find some α such that

$$\|\phi(x)-\phi(y)\|_H \leq \alpha\|x-y\|. \tag{10}$$

Note that

$$\begin{aligned}\|\phi(x)-\phi(y)\|_H &= \sqrt{\langle\phi(x)-\phi(y),\phi(x)-\phi(y)\rangle}\\ &= \sqrt{\|\phi(x)\|_H^2+\|\phi(y)\|_H^2-2\langle\phi(x),\phi(y)\rangle}\\ &= \sqrt{2(K^2-\langle\phi(x),\phi(y)\rangle)}.\end{aligned}$$

By shift invariance, we have

$$\frac{\sqrt{2(K^2-\langle\phi(x),\phi(0)\rangle)}}{\|x\|} \leq \alpha.$$

So, it is clear that

$$\alpha = \sup_x \frac{\sqrt{2(K^2-\langle\phi(x),\phi(0)\rangle)}}{\|x\|}$$

satisfies the Lipschitz condition.

Compared to the optimization problem in the previous subsection given in Proposition 2, which depends on both the encoding and the dataset, the optimization problem here depends only on the encoding. For shift-invariant encodings where $\langle\phi(x),\phi(0)\rangle$ is approximately symmetric about the origin, it follows that the global maximum if it exists, should be close to the origin. This fact indicates that this optimization problem is an easier one.

While one generally cannot guarantee convergence to the global maximum, it is likely that an optimization scheme can get rather close. Thus, the resulting approximation of ϵ_q given in Equation 9, loosely speaking, is close to (ϵ_q, q)-robust. Of course, while we get this stronger estimate compared to the approximation in Equation 8, this estimate is comparatively more conservative.

5 Learning a robust hyperdimensional encoding

We discuss how the results of the previous section can be used to learn a robust encoding. An encoding ϕ generally depends on some set of parameters M sampled from a distribution p. An example of this is the RFF encoding $\phi(x) = e^{iMx}$, where each row $M_j \sim p$.

Thus, we can characterize a family of encodings by parameterizing the encoding itself ϕ as well as the distribution p_θ from which random samples M are drawn. We denote such dependencies via the notation $\phi_\eta(\cdot;\eta,M)$. We also denote this dependence in ϵ_q by writing $\epsilon_q(\eta,M)$. Let $\bar{\epsilon}(\eta,M)$ be the average estimated robustness $\epsilon_q(\eta,M)$ over the dataset, i.e., $\mathbb{E}_{x\sim D}[\epsilon_x(\eta,M)]$. Thus, our goal is to find parameters η^*,θ^* such that

$$\eta^*,\theta^* = \arg\max_{\eta,\theta} \mathbb{E}_{M\sim p_\theta}[\bar{\epsilon}(\eta,M)]. \tag{11}$$

To compute the gradient $\nabla_\eta \mathbb{E}_{M\sim p_\theta}[\bar{\epsilon}(\eta,M)]$, we can simply do a Monte-Carlo estimate:

$$\nabla_\eta \mathbb{E}_{M\sim p_\theta}[\bar{\epsilon}(\eta,M)] = \mathbb{E}_{M\sim p_\theta}[\nabla_\eta\bar{\epsilon}(\eta,M)]. \tag{12}$$

We make a simplifying assumption that $M \sim p_\theta$ is equivalent to $M = f_\theta(Z)$, where $Z \sim N(0,I)$. This gives us the gradient

$$\nabla_\theta \mathbb{E}_{Z\sim N(0,I)}[\bar{\epsilon}(\eta,f_\theta(Z))] = \mathbb{E}_{Z\sim N(0,I)}[\nabla_\theta\bar{\epsilon}(\eta,f_\theta(Z))]. \tag{13}$$

6 Results

We apply our theoretical results above to both synthetic and real datasets. In the synthetic case, we compare the per point robustness ϵ_q to the actual distance to the decision boundary and show that our Lipschitz methods satisfy (ϵ_q, q)-robustness. In the real case, we apply our methodology for kernel learning and demonstrate that the average robustness increases under our optimization scheme. We also test the effectiveness of average robustness $\bar{\epsilon}$ as a measure of robustness by plotting the degradation in accuracy of the HDC models with different levels of average robustness when noise of increasing magnitude is added to data points.

6.1 Results on synthetic dataset

We test our theoretical results for both one- and two-dimensional synthetic datasets. Each dataset consists of two

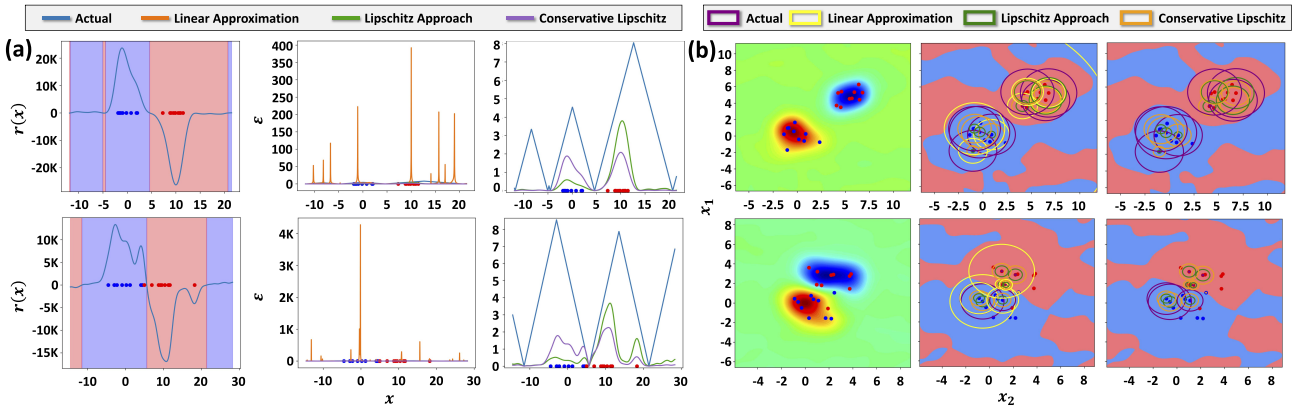

FIGURE 2
(a) Synthetic 1D dataset with two classes generated by sampling from Gaussian distributions. **(b)** Synthetic 2D dataset with two classes generated by sampling from Gaussian distributions. *Top row:* class samples are drawn from Gaussian distributions with low overlap. *Bottom row:* class samples are drawn from Gaussian distributions with higher overlap, leading to data that is not linearly separable.

classes, each of which is generated via sampling from a Gaussian distribution.

Figure 2 visualizes (Figure 2a) 1D and (Figure 2b) 2D synthetic datasets alongside the decision function $r(x)$, the decision regions, as well as the real and estimated robustness values; i.e., distance to the decision boundary. In the 1D case, this is visualized as the height of the plotted functions, and in the 2D case, as the radii of the circles. As can be seen, method 1 of computing ϵ_q does not give a good estimate of robustness as it tends drastically overestimate the distance to the decision boundary. Both Lipschitz-based methods give estimates that are generally less than the actual distance to the decision boundary, satisfying (ϵ, q)-robustness. In addition, we see that method 2 gives a better estimate in the sense that it is closer to the actual distance to the decision boundary, which corroborates with our theory above.

6.2 Results on visual data

We use a binary classification version of MNIST where the dataset contains images of numbers 1 and 2. To encode each data point in the dataset, we use an RFF encoding with dimension 10,000, with random parameters $M \sim N(0, \sigma I)$. Under our kernel learning framework, this can be expressed as $M = f_\sigma(Z) = \sigma Z$ where $Z \sim N(0, I)$. Using the methodology described in the section above, we optimize σ to maximize the average robustness. Figure 3D shows that the average robustness increases over the number of iterations of the optimization process. We do this for both methods 2 and 3 of computing robustness described above.

At each iteration in the optimization process, we get a measure of average robustness based on methods 2 or 3. In addition, we compute the corresponding train and test accuracy of the model at that point. We visualize this in Figure 3A, which plots the tradeoff between train and test accuracy and robustness computed using methods 2 and 3. The train accuracy clearly falls as robustness increases. However, increasing robustness increases test accuracy up to a certain point, which then decreases as in the case of train accuracy. Thus, we may think of robustness as a kind of regularization.

For models of different levels of average robustness, we plot its degradation in train and test accuracy as we add increasing magnitudes of noise to the data in the input space. We visualize the results in Figures 3B, C. As average robustness increases, the degradation in train accuracy decreases more slowly as noise of increasing magnitude is added to the input space. In the case of test accuracy, while there is no clear trend as in the case of train accuracy, we do see that there is greater variance in accuracy across different magnitudes of noise for models with lower average robustness, while the quality is nearly constant for the model with the highest robustness.

7 Discussion

7.1 Extending beyond the RFF encoding

While we have illustrated our method of computing robustness estimates using HDC classifiers using the RFF encoding, our method extends to all HDC encoders that are Lipschitz-continuous. This result is highlighted in Proposition 3, which delegates the estimation of the robustness to estimating the Lipschitz constant of the HDC encoder of choice. It is important to note that this result requires the HDC encoder to be Lipschitz-continuous, which is a rather loose assumption, applying not only to differentiable encoders but also to non-differentiable ones. We outlined a method to estimate the Lipschitz constant for shift-invariant encoders (e.g., RFF) in Proposition 4 as a specific example, but it is possible to derive similar results for other HDC schemes, such as binary splatter codes (BSCs) (Kanerva, 1997), multiply add permute (MAP) (Gayler, 1998), matrix binding of additive terms (MBAT) (Gallant and Okaywe, 2015), and generalized holographic reduced representations (GHRRs) (Yeung et al., 2024). Investigating ways to estimate the Lipschitz constant for such encoders will be left for future work.

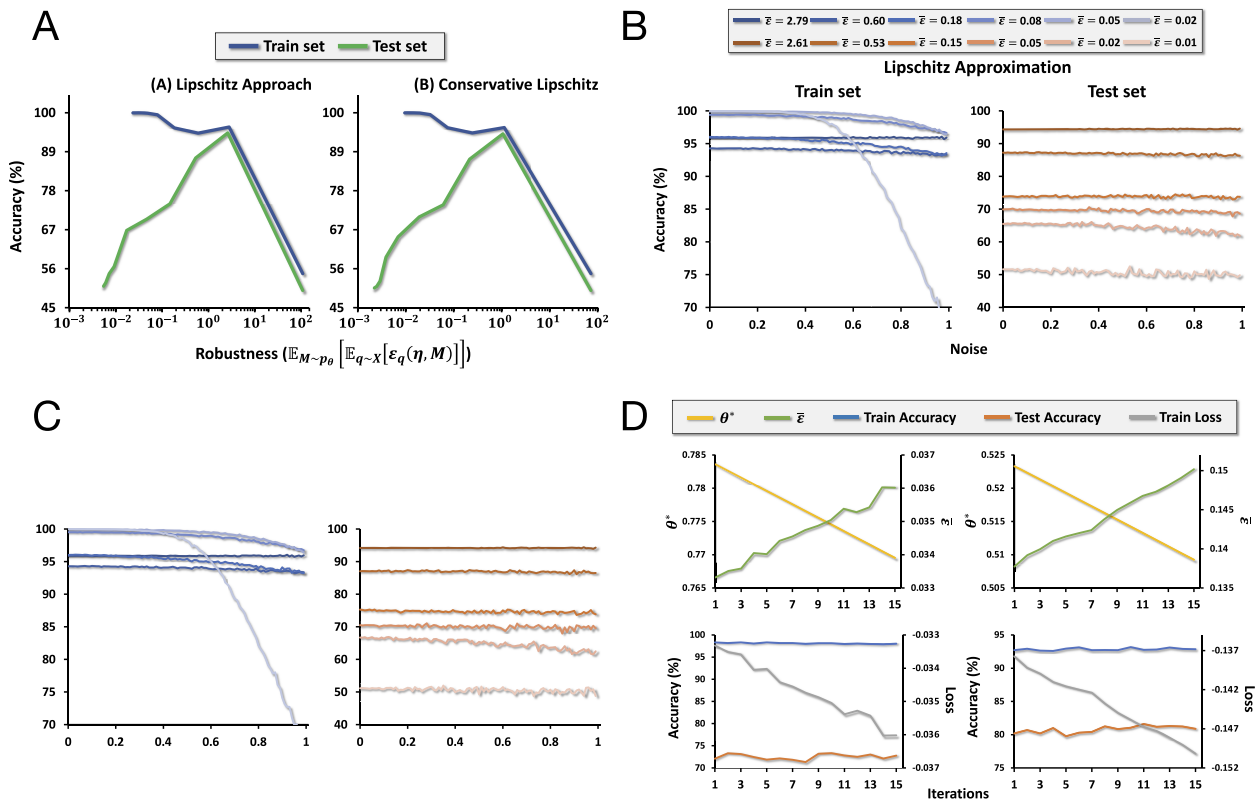

FIGURE 3
(A) The tradeoff between accuracy and robustness. Left: the tradeoff when accuracy is computed using method 2. Right: the tradeoff when the accuracy is computed using method 3. **(B)** A plot of the accuracy as increasing noise is added to the input space for models of different levels of average robustness computed using method 2. **(C)** A plot of the accuracy as increasing noise is added to the input space for models of different levels of average robustness computed using method 3. **(D)** Plots of loss, the length-scale parameter θ, average robustness, and accuracy over training iterations as we optimize for average robustness.

7.2 Quantized setting and neuro-vector-symbolic pipelines

As HDC is commonly considered within a quantized setting due to its plethora of hardware applications, we discuss a direction in which to extend our method to this particular setting. As noted above, our derived robustness estimate depends on the encoder's Lipschitz-continuity. In the quantized setting (e.g., using BSC or MAP schemes), core symbolic operations (e.g., XOR binding and fixed permutations) are Hamming isometries, so the dominant contribution to the Lipschitz constant comes from the encoder and any subsequent quantizer, which we can model as $e = q \circ f$, where f maps inputs to a continuous hypervector and q performs quantization. When f is L_f-Lipschitz and q is non-expansive (deterministically or in expectation) under Hamming distance, one obtains a usable bound $L_e \leq L_q L_f$, allowing our analysis to carry over naturally to this setting.

This decomposition aligns with established "translation" pipelines that move signals from neural or symbolic spaces into hypervector space. In particular, Mitrokhin et al. (2020) map learned embeddings to binary hypervectors and classify via binding/bundling, while Sutor et al. (2022) (HD-Glue) convert penultimate-layer signals from heterogeneous neural networks into hypervectors and aggregate them. Both follow a similar structural pattern of $e = q \circ f$: a continuous feature extractor (f) followed by a mapping to a discrete symbolic representation (q).

This approach also highlights a possible way of extending our method to translate to a neuro-vector-symbolic pipeline. While our theoretical results in this work are restricted to classifiers that operate entirely within the hypervector space (i.e., purely HDC-based), extending our results to guarantees to neuro-vector-symbolic pipelines that prepend a neural embedding requires accounting for the neural component's Lipschitz constant. Writing $f = h \circ g$ with g a neural embedding and h a continuous HDC encoder gives us $L_e \leq L_q L_h L_g$ so finding the Lipschitz constant of the resulting encoder reduces to bounding L_g in addition to L_h and L_q. Practical methods for controlling or estimating L_g are well studied (e.g., operator-norm bounds and spectral constraints at the layer level, orthogonal/Parseval parameterizations, and related Lipschitz-controlled architectures) and can be composed to produce conservative global bounds.

8 Conclusion

In summary, this study presents a first-of-its-kind theoretical framework for evaluating the robustness of hyperdimensional

classifiers based on Lipschitz continuity, which can then be used as an optimization objective for learning more robust encodings. Our experimental results demonstrate the effectiveness of our approach. We believe our work can lead to the development of more robust and reliable hyperdimensional computing models and pave the way for further research in this area.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

CY: Writing – original draft, Software, Writing – review & editing, Investigation, Methodology, Formal analysis, Validation, Conceptualization, Visualization. HE: Visualization, Writing – review & editing, Writing – original draft. ZZ: Writing – review & editing, Conceptualization. SY: Writing – review & editing, Visualization, Software. NB: Writing – review & editing. MI: Project administration, Supervision, Writing – review & editing, Funding acquisition.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This study was supported in part by the DARPA Young Faculty Award, the National Science Foundation (NSF) under Grants No. 2127780, No. 2319198, No. 2321840, No. 2312517, and No. 2235472, the Semiconductor Research Corporation (SRC), the Office of Naval Research through the Young Investigator Program Award, and Grants No. N00014-21-1-2225 and No. N00014-22-1-2067, the U.S. Army Combat Capabilities Development Command Army Research Laboratory under Support Agreement No. USMA 21050, and DARPA under Support Agreement No. USMA 23004. Additionally, support was provided by the Air Force Office of Scientific Research under Award No. FA9550-22-1-0253, along with generous gifts from Xilinx and Cisco. The funders were not involved in the study design, collection, analysis, interpretation of data, the writing of this article, or the decision to submit it for publication.

Acknowledgments

We would like to thank Nathan McDonald from the Air Force Research Laboratory for his valuable feedback.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Author disclaimer

The opinions, findings, and conclusions expressed in this study are those of the authors and do not reflect the position of the United States Department of Defense or the United States Government.

References

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. (2017). "Parseval networks: Improving robustness to adversarial examples," in *Proceedings of the 34th International Conference on Machine Learning*, eds. D. Precup, and Y. W. Teh (PMLR), 854–863.

Cohen, J., Rosenfeld, E., and Kolter, Z. (2019). "Certified adversarial robustness via randomized smoothing," in *Proceedings of the 36th International Conference on Machine Learning*, eds. K. Chaudhuri, and R. Salakhutdinov (New York: PMLR), 1310–1320.

Gallant, S. I., and Okaywe, T. W. (2015). Representing objects, relations, and sequences. *arXiv* [preprint] arXiv:1501.07627. doi: 10.48550/arXiv.1501.07627

Gayler, R. W. (1998). *Multiplicative Binding, Representation Operators & Analogy*.

Hernández-Cano, A., Matsumoto, N., Ping, E., and Imani, M. (2021). "OnlineHD: robust, efficient, and single-pass online learning using hyperdimensional system," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 56–61.

Kanerva, P. (1997). "Fully distributed representation" in *Real World Computing Symposium (RWC)* (Tokyo), 358–365.

Kleyko, D., Rachkovskij, D. A., Osipov, E., and Rahimi, A. (2023). A survey on hyperdimensional computing aka vector symbolic architectures, part I: models and data transformations. *ACM Comput. Surv.* 55, 1–40.doi: 10.1145/3538531

Kymn, C. J., Mazelet, S., Ng, A., Kleyko, D., and Olshausen, B. A. (2024). "Compositional factorization of visual scenes with convolutional sparse coding and

resonator networks," in *2024 Neuro Inspired Computational Elements Conference* (NICE) (IEEE), 1–9. doi: 10.1109/nice61972.2024.10549719

Matsui, C., Kobayashi, E., Misawa, N., and Takeuchi, K. (2023). Comprehensive analysis on error-robustness of fefet computation-in-memory for hyperdimensional computing. *Jap. J. Appl. Phys.* 62:SC1053.doi: 10.35848/1347-4065/acb1b8

Mitrokhin, A., Sutor, P., Summers-Stay, D., Fermüller, C., and Aloimonos, Y. (2020). Symbolic representation and learning with hyperdimensional computing. *Front. Robot. AI* 7:63. doi: 10.3389/frobt.2020.00063

Poduval, P., Alimohamadi, H., Zakeri, A., Imani, F., Najafi, M. H., Givargis, T., and Imani, M. (2022). GrapHD: graph-based hyperdimensional memorization for brain-like cognitive learning. *Front. Neurosci.* 16:757125. doi: 10.3389/fnins.2022.757125

Raghunathan, A., Steinhardt, J., and Liang, P. (2018). Certified defenses against adversarial examples. *arXiv* [preprint] arXiv:1801.09344. doi: 10.48550/arXiv.1801.09344

Rahimi, A., Kanerva, P., and Rabaey, J. M. (2016). "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design* (New York, NY: ACM), 64–69.

Rahimi, A., and Recht, B. (2007). "Random features for large-scale kernel machines," in *Proceedings of the 20th International Conference on Neural Information Processing Systems, NIPS'07* (Red Hook, NY: Curran Associates Inc), 1177–1184.

Sutor, P., Yuan, D., Summers-Stay, D., Fermuller, C., and Aloimonos, Y. (2022). Gluing neural networks symbolically through hyperdimensional computing. *arXiv preprint* arXiv:2205.15534.

Thomas, A., Dasgupta, S., and Rosing, T. (2022). A theoretical perspective on hyperdimensional computing. *J. Artif. Int. Res.* 72:215–249. doi: 10.1613/jair.1.12664

Wong, E., and Kolter, Z. (2018a). "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *Proceedings of the 35th International Conference on Machine Learning*, eds. J. Dy, and A. Krause (New York: PMLR), 5286–5295.

Wong, E., and Kolter, Z. (2018b). "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning* (New York: PMLR), 5286–5295.

Yeung, C., Zou, Z., Bastian, N. D., and Imani, M. (2025). Cognitive map formation under uncertainty via local prediction learning. *Intellig. Syst. Appl.* 27:200551. doi: 10.1016/j.iswa.2025.200551

Yeung, C., Zou, Z., and Imani, M. (2024). Generalized holographic reduced representations. *arXiv preprint* arXiv:2405.09689.

Zhang, H., Chen, H., Xiao, C., Gowal, S., Stanforth, R., Li, B., Boning, D., and Hsieh, C.-J. (2019). Towards stable and efficient training of verifiably robust neural networks. *arXiv* [preprint] arXiv:1906.06316. doi: 10.48550/arXiv.1906.06316

Zhang, S., Wang, R., Zhang, J. J., Rahimi, A., and Jiao, X. (2021). "Assessing robustness of hyperdimensional computing against errors in associative memory: (invited paper)," in *2021 IEEE 32nd International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, 211–217. doi: 10.1109/ASAP52443.2021.00039