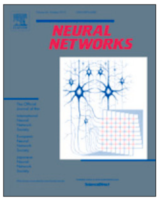
Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



2021 Special Issue on AI and Brain Science: Brain-inspired AI

Hierarchical binding in convolutional neural networks: Making adversarial attacks geometrically challenging

Niels Leadholm [*], Simon Stringer

The Oxford Centre for Theoretical Neuroscience and Artificial Intelligence, University of Oxford, Radcliffe Observatory Quarter, Oxford, OX2 6GG, United Kingdom

ARTICLE INFO

Article history:
Available online 21 July 2022

Keywords:
Adversarial examples
Robust representations
Feature binding
Robust vision

ABSTRACT

We approach the issue of robust machine vision by presenting a novel deep-learning architecture, inspired by work in theoretical neuroscience on how the primate brain performs visual feature binding. Feature binding describes how separately represented features are encoded in a relationally meaningful way, such as an edge composing part of the larger contour of an object. We propose that the absence of such representations from current models might partly explain their vulnerability to small, often humanly-imperceptible distortions known as adversarial examples. It has been proposed that adversarial examples are a result of 'off-manifold' perturbations of images. Our novel architecture is designed to approximate hierarchical feature binding, providing explicit representations in these otherwise vulnerable directions. Having introduced these representations into convolutional neural networks, we provide empirical evidence of enhanced robustness against a broad range of L_0, L_2 and L_∞ attacks, particularly in the black-box setting. While we eventually report that the model remains vulnerable to a sufficiently powerful attacker (i.e. the defense can be broken), we demonstrate that our main results cannot be accounted for by trivial, false robustness (gradient masking). Analysis of the representational geometry of our architectures shows a positive relationship between hierarchical binding, expanded manifolds, and robustness. Through hyperparameter manipulation, we find evidence that robustness emerges through the preservation of general low-level information alongside more abstract features, rather than by capturing which specific low-level features drove the abstract representation. Finally, we propose how hierarchical binding relates to the observation that, under appropriate viewing conditions, humans show sensitivity to adversarial examples.

1. Introduction

Adversarial examples are images modified by small (L_p norm constrained) perturbations that cause machine vision systems to catastrophically misclassify objects (Szegedy et al., 2014). Since their discovery, various efforts have been made at both explaining their existence, and conferring resistance to them (Gilmer et al., 2018; Goodfellow et al., 2015; Ilyas et al., 2019; Madry et al., 2018; Schott et al., 2019; Stutz et al., 2019). This includes arguments that adversarial examples represent perturbations of the input off of the data manifold (Khoury & Hadfield-Menell, 2018; Stutz et al., 2019; Tanay & Griffin, 2016), and in particular that the vulnerability of a model is associated with the presence of many such directions being available to the attacker (Khoury & Hadfield-Menell, 2018).

Both biological and artificial means of object recognition typically propose that object classes can be effectively described by a lower-dimensional manifold than the input (e.g. pixel) space. Such a data manifold describes a region where samples are concentrated, and the dimensions of the manifold represent class-preserving changes to the object representation (such as increasing its size, or rotating it). Moving between object manifolds would generally correspond to low-density regions with potentially no associated object class, such as an image of white noise (Bengio et al., 2013; DiCarlo & Cox, 2007).

We argue that part of the phenomenon of adversarial examples is that architectural choices often assume object manifolds that are too low-dimensional. While this is sufficient for classification on a standard data-set, where samples off of the data manifold are sparse, many adversarial examples can exploit this assumption by moving orthogonal to the concentrated subspace (Khoury & Hadfield-Menell, 2018; Stutz et al., 2019; Tanay & Griffin, 2016). Given that humans are (in most cases by definition) robust, this implies that adversarial examples exist *on* the manifolds that underlie human classification of objects, and that these are therefore of higher dimension. One approach is therefore to augment the data manifold, such as by adding noise during

[*] Corresponding author.
E-mail address: niels.leadholm@seh.ox.ac.uk (N. Leadholm).

https://doi.org/10.1016/j.neunet.2022.07.003

training (Ford et al., 2019; Rusak et al., 2020; Zantedeschi et al., 2017), in order to more faithfully sample from the data manifold of images that a human would classify as a particular object. We argue however that typical convolutional neural network (CNN) architectures are constrained in their ability to practically model the spectrum of variation that an object can undergo. In particular, typical architectures result in the loss of low-level spatial details in order to promote invariance and linear separability of classes. We propose that a model that maintains standard classification abilities while modeling objects more faithfully in high-dimensional space should enable greater robustness. This representational expressiveness of a model can be measured as the extent of an object's manifold in the neural state space of the model, the neural manifold.

Assuming a higher-dimensional manifold underlies human perception, and that a model can capture this, many adversaries could be viewed as on-manifold adversarial examples, after which robustness is a case of standard generalization (Gilmer et al., 2018; Stutz et al., 2019). We introduce a novel architecture that, through hierarchical feature binding (defined below), captures low-level features *alongside* lower-dimensional, invariant representations. The difficulty then is learning a useful decision boundary and sampling sufficiently in this high-dimensional space. For this reason, we complement our approach with regularization and data-augmentation with noise. The geometric intuition for our approach is summarized in Fig. 1. We note that our defense operates not by guaranteeing the absence of any vulnerable regions in the decision boundary, but by reducing the number of directions in which such regions are found. Indeed, in our discussion we suggest that, given a sufficiently powerful attack method, the network is as vulnerable as a model without our defense (i.e. the defense can be broken). However, our improvements to the classifier's decision boundary provides robustness against a variety of methods for generating adversarial attacks, including several black-box methods that rely on these vulnerable directions being present, and which represent the primary methods that could be plausibly leveraged against humans.

Many proposed adversarial defenses have shortcomings or can be entirely circumnavigated under the appropriate attack, a notorious issue in the literature that has grown increasingly apparent, and a reality affecting the vast majority of published defenses (Croce & Hein, 2020; Tramèr, Carlini et al., 2020). One of the only methods known to consistently improve robustness independent of the attack-method is adversarial training (Madry et al., 2018), or variants of it (Gowal et al., 2020), although it can suffer from issues such as over-fitting to particular distance metrics used in training (Laidlaw et al., 2021; Schott et al., 2019), and a loss in classification accuracy on clean, unaltered data (Tsipras et al., 2019).

Despite the seemingly intractable nature of adversarial examples, humans are, by definition, robust to them under normal conditions. Similar to the approach taken by recent defenses on breaking certain attacks (Xiao et al., 2020), rather than defending against any possible attacker, we demonstrate a defense method with utility against a variety of black-box attacks, including naive brute-force attacks based on noise. Our defense is notable due to its (a) connection to the growing geometric view of adversarial examples, (b) ability to generalize across multiple distance metrics, a stumbling block for many adversarial defenses (Laidlaw et al., 2021; Schott et al., 2019) and (c) relevance to experimental results on the perception of adversarial examples in humans. Importantly, the connection we establish between our approach and the geometry of a neural network's representations lends the defense to future extensions that might enable models that are fundamentally robust to all attackers.

Our contributions are as follows:

- We present a mechanism to preserve and utilize information about which lower-level features play an important role in driving a CNN's higher-level, abstract representation at a succeeding layer. Our novel architecture is loosely inspired by work predicting the existence of neurons in the primate brain that encode the hierarchical binding relationships between visual features at different spatial scales (Eguchi et al., 2018), and our findings offer a possible explanation for the sensitivity of humans to adversarial examples under appropriate viewing conditions.
- We present empirical results showing the robustness of these augmented networks to a variety of adversarial attacks, following the use of techniques to ensure a useful decision boundary. These results are complemented with findings from a brute-force noise attack that support the connection between the observed robustness and our geometric framing of the defense.
- We analyze the geometric properties of our networks' representations, finding that the neural manifolds in hierarchical binding networks are of higher dimension and radius, quantified by the manifold extent. We further observe that, while neither necessary nor sufficient for resistance to adversarial attacks, greater neural manifold extent is associated with increased adversarial robustness.
- We additionally use iterative adjustments of a key hyperparameter (the γ-proportion) to demonstrate that the observed robustness is primarily a consequence of the low-level representations that hierarchical binding preserves, rather than the explicit encoding of binding relations. Hierarchical binding in the primate brain might thereby serve a variety of computational functions, with one of its benefits being robustness through the preservation of low-level information.

2. Related work

Binding Feature binding describes the ability of the visual brain to perceive, represent and reason about the relationships between separately encoded features of objects (such as the color and shape that jointly describe a yellow triangle) (Gray, 1999; Treisman, 1998; Von Der Malsburg, 1999). For example, when we look at an alphabetical letter T, we can see the vertical and horizontal bars that comprise the letter as distinct elements, as well as the fact that these constituent elements are part of the letter *T* itself. Such *hierarchical binding* captures the causal relations between multiple scales of abstraction, e.g. that a particular vertical bar feature is part of a T and not an L nearby (Eguchi et al., 2018; Treisman, 1996). Low-level features carry meaningful spatial information due to their small receptive fields and high dimension, and hierarchical binding can thereby encode class-preserving transformations of the object — note in our example that it is not simply the abstract concept of a vertical bar that is bound to the T, but the specific representation localized in space. We emphasize that this framing does not discount the importance of invariant features; rather the goal is to capture a continuum of abstraction jointly. Previous work has been done in encoding binding-like representations (Bear et al., 2020; Burgess et al., 2019; Greff et al., 2016; Locatello et al., 2020; Reichert & Serre, 2014; Schlag et al., 2019; Whittington et al., 2020), albeit using different mechanisms to those discussed here, and without an investigation of its relevance to adversarial robustness.

Eguchi et al. (2018) proposed a mechanism by which the brain might capture hierarchical binding, encoding the relations between low-level and high-level features throughout the visual processing stream. An example of such a possible binding

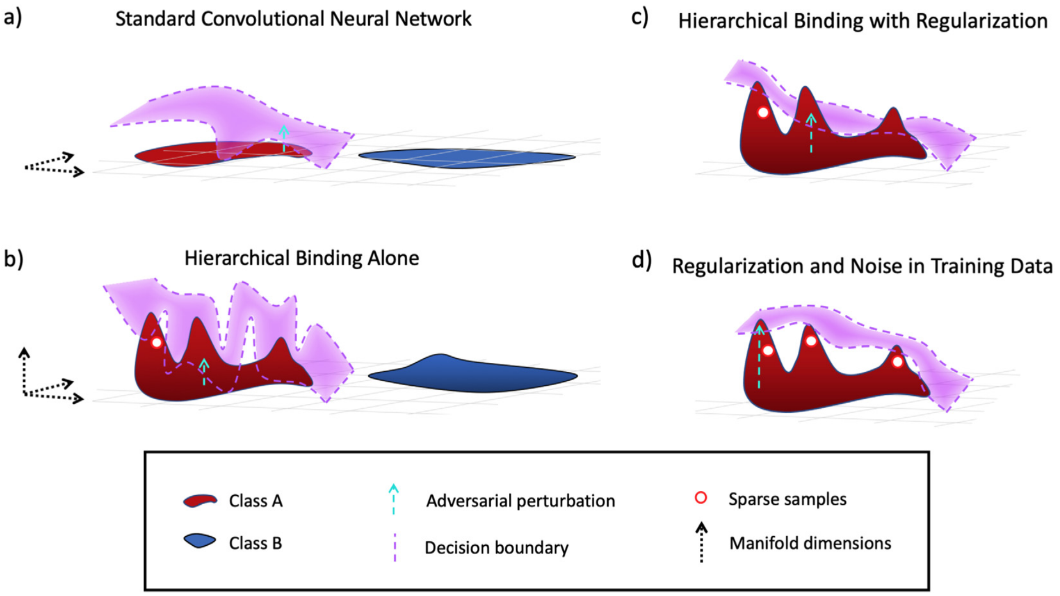

a)
Standard Convolutional Neural Network
b)
Hierarchical Binding Alone
c)
Hierarchical Binding with Regularization
d)
Regularization and Noise in Training Data
Class A
Class B
Adversarial perturbation
Decision boundary
Sparse samples
Manifold dimensions

Fig. 1. *The effect of hierarchical binding on the decision boundary.* Red and blue represent two different object manifolds (e.g. cats and dogs). Adversarial perturbations (light-blue arrows) for the red class move the input beyond the decision boundary into a region where it is classified as blue. (a) A common approach for classification is to assume low-dimensional representations of objects are sufficient, as these support linear decision boundaries that accurately separate the objects. Unfortunately, the learned decision boundary can be unpredictable off the manifold. Given the high-dimensional embedding space (e.g. pixel-space), there may be many such directions vulnerable to small perturbations. Here, an idealized 2D manifold is used to depict the data distribution typically assumed when designing model architectures. (b) We argue that there are additional, class-preserving dimensions of variation to the manifold, where the shown manifold represents all examples a human would classify as a given object (here depicted as a 3D solid). We suggest that these dimensions are impractical to model with typical convolutional neural network (CNN) architectures. Adding hierarchical binding enables the network to explicitly represent these features alongside the more abstract dimensions, potentially aligning the representation more closely with the basis of human perception. Despite this modification, the sparsity of samples in high dimensions (particularly in standard data-sets) means that further steps are required for a robust decision boundary. (c) Introducing regularization such as label smoothing means that even sparse data points can inform a more useful decision boundary. (d) Complementing label smoothing with noise during training helps sample more densely from the manifold underlying human perception, providing a more robust decision boundary against a variety of adversarial attacks. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

representation that has been identified in neurophysiological experiments is border-ownership cells, which have been identified in the V1, V2, and V4 regions of the primate visual cortex. These cells have a small classical receptive field, similar to bar/edge detecting 'simple cells'. Unlike a classical simple cell however, their response is modulated by what side of an object they form an edge of. In particular, border-ownership cells encode when a particular low-level feature is part of a specific side of a high-level object, rather than simply that a low-level feature is present (Zhou et al., 2000). Eguchi et al. (2018) predicted that the temporal coincidence detection afforded by the spike-timing of biological neurons and the lateral and top-down connectivity observed in the brain would be essential for implementing this binding mechanism. In particular, these properties could enable spiking neurons or assemblies of spiking neurons to fire if and only if a particular low-level feature was causally driving a higher-level representation. Rather than simulate these biological elements, we use a non-local algorithm to approximate such representations at the computational level (Marr, 1982) within an adapted CNN architecture. We also use a simplified form of the full hierarchical binding formulation proposed in Eguchi et al. (2018) in order to constrain the computational cost of the architecture, enabling us to explore the significance of the proposed representations for robust object classification.

It is clear that many aspects of primate vision rely primarily on abstract, low-dimensional representations, which is consistent with our proposal here, and that vision can be impoverished e.g. outside of attention. What we are describing here however is the primate perception of "objects along with their detailed features" (Lu et al., 2018). For example, even in the periphery, humans are sensitive to low-level image changes when these impact scene-like content as opposed to textures (Wallis et al., 2019). Perceptual processing in primates accesses such low/mid-level information, such as local border ownership (Kim & Feldman, 2009). Thus the representations we explore are consistent with "vision with scrutiny", rather than coarse object recognition (Hochstein & Ahissar, 2002).

Preserving Low-Level Information in Computer Vision Many methods exist to preserve low-level information in deep neural networks (He et al., 2016; Huang et al., 2017; Jacobsen et al., 2018; Ronneberger et al., 2015; Srivastava et al., 2015), spatially enrich feature representations (Hinton et al., 2018; Sabour et al., 2017) or encourage the encoding of additional factors of variation (Cheung et al., 2015). Our architecture is novel in that it captures which low-level neurons causally drove the network's high-level representation, and explicitly encodes such information as layers in their own right for classification. This explicit encoding is important for down-stream read-out of the representations. In contrast, architectures using skip connections typically combine information from low and high level layers in an additive operation (He et al., 2016); this can obscure the respective contributions of features and makes classifier read-out of the low-level details more challenging. Furthermore, our architecture specifically combines low-level features with a high spatial resolution (i.e. cross-wise dimension) alongside more abstract representations with coarser spatial information. Where skip connection-architectures concatenate features, such a preservation of spatially detailed and coarse features is not performed (Huang et al., 2017). In summary, our approach is related to the motivation for (and biological evidence of) disentangled representations (Higgins et al., 2021) — our architecture is biased so as to "disentangle as many factors as possible, discarding as little information about the data as is practical" (Bengio et al., 2013). This motivation is important as disentangled representations are associated with more sample-efficient learning (van Steenkiste et al., 2019), while learning a robust decision boundary is a sample inefficient problem (Gilmer et al., 2018; Khoury & Hadfield-Menell, 2018).

Adversarial Examples While there is a large literature on adversarial examples (see e.g. Yuan et al. (2019) for a review), we focus on those papers that are most relevant to the current work. The concept of adversarial examples as manifold failures has inspired several defenses (Jalal et al., 2017; Jang et al., 2020; Samangouei et al., 2018; Schott et al., 2019; Song et al., 2018). Stutz et al. (2019) showed that typical adversarial examples move orthogonal to the data's low-dimensional manifold, and Khoury and Hadfield-Menell (2018) provided evidence in synthetic data-sets that a greater number of directions normal to the data manifold is associated with increasing vulnerability. This appears to be because the decision boundary can be arbitrary off of the manifold (Khoury & Hadfield-Menell, 2018), and may indeed lie very close to it (Tanay & Griffin, 2016).

Related to our defense, previous work on extracting hierarchical interpretations for the predictions of neural networks has shown that these interpretations themselves can be resistant to adversarial attacks (Singh et al., 2019), although this work did not relate to the issue of robust object classification.

3. Model description

CNN Fundamentals A typical CNN architecture consists of convolution operations and a down-sampling operation such as max-pooling (see Lecun et al. (2015) for an overview). Convolutions apply a linear transformation at local regions across space, where this weighted sum serves as a feature detector, such as for the presence of an edge or an ear. If the activity within this local region and the learned weights match well, then a large activation will be output, suggesting the presence of that feature. Using the assumption that features in one part of visual space may also appear at another, the same convolutions are applied repeatedly across space, making efficient re-use of parameters. Several different convolutions can be applied at any given level of the network in order to detect multiple different features.

Max-pooling serves to provide the model with translation invariance, by taking the largest activation within a given receptive field, and up-projecting that activation value alone. When applied to the output of feature detectors, this can be thought of as evaluating, for example, if an ear was present anywhere in a given region of the image, without any concern for its precise location. Subsequent feature detectors can take advantage of this invariance by applying a convolution to the outputs of max-pooled values; this convolution (say for detecting the presence of a cat), will be less sensitive to the spatial particularities of the features of the cat, and should in principle be more capable of generalization.

Fig. 2a includes a demonstration of what a typical convolution and max-pooling operation might look like. Fig. 3a shows how a typical CNN architecture implements successive convolution and max-pooling operations, alongside our additional operations of unpooling and gradient unpooling (introduced below). While the successive use of convolutions and max-pooling are not without their limitations (see e.g. comparisons of CNNs to bag-of-feature detectors (Brendel & Bethge, 2019), and their insensitivity to global shape (Baker et al., 2018)), they capture the key principles that are believed to operate in the primate brain — that is feature detection and invariance.

Implementing Hierarchical Binding To capture which low-level features causally drove max-pooled representations, we use the operation known as unpooling. Maxpooling determines the maximum value of each type of feature over a local spatial region as a way of achieving a location invariant representation of that feature. However, such an operation loses potentially useful information about the location of that feature within the spatial region. Unpooling is used here to recover and preserve such location information, by the addition of a sub-layer of unpooled feature activations. The unpooling operation identifies which locations had maximal activations (i.e. survived max-pooling), and preserves these activations at their locations in the new sub-layer of unpooled feature activations, while the activations for that feature which did not take maximal values are set to zero (Badrinarayanan et al., 2017; Zeiler & Fergus, 2014). This procedure is illustrated in the top right of Fig. 2a. A modified version of unpooling, termed 'ratio unpooling', has previously been employed in a mixed bottom-up and top-down network as a means of preserving spatial information (Xu et al., 2019). Importantly however, this modified form of unpooling does not capture what lower-level features causally drove higher-level features (i.e. hierarchical binding), consistent with this not being the motivation of the authors.

To capture which simple features contributed to abstract representations, we introduce what we term 'gradient unpooling' (Fig. 2a). Capturing how every feature in a lower layer (denoted L^k) contributes to a feature in a succeeding layer (L^{k+1}) preserves full information about these hierarchical binding relations (Eguchi et al., 2018), but is computationally unappealing due to the number of required binding representations. This is particularly problematic if this information is fed into a fully-connected layer for learning decision boundaries in a classifier. As such, we propose a computationally more feasible approach where we treat the high-level representation as distributed, and assume that only one object is ever represented in this distributed activity at a time (a valid assumption for the data-sets we use). Our aim is then to capture what low-level features contributed to this distributed representation.

To do so, we assume that the partial derivative of a high-level activation within a hierarchical network, taken w.r.t. to a low-level activation, carries some information about the significance of the contribution of that low-level feature. This is therefore our primary signal for causal influence, and this principle of using gradients is an established approach to attributing the significance of a feature, for example where the partial derivative of a class score is taken w.r.t. pixel inputs (Simonyan et al., 2014). As we are interested in the contribution of the low-level feature neuron to the distributed representation in the higher layer, we use the sum of the partial derivatives across the high-level layer. Finally, as the scalar value of a low-level neuron's activation provides information about its presence in the image (and therefore its likelihood of having contributed to higher-level representations), we include this in the final gradient unpooling values.

We now describe gradient unpooling in detail. Our first aim is to use the proposed gradients to approximate the significance of a low-level unit a_i with the measure m_i. Specifically let $a_i^{(k)}$ be the activation of an ith unit in the lower-level layer $L^{(k)}$, and $a_j^{(k+n)}$ in the higher-level layer $L^{(k+n)}$, where $n \geq 1$. The measure units m_i form a tensor of the same dimension as $L^{(k)}$, where each unit's value is given by:

$$m_i = \sum_j \frac{\partial a_j^{(k+n)}}{\partial a_i^{(k)}} \tag{1}$$

This sum is taken over the activations $a_j^{(k+n)}$ in $L^{(k+n)}$. As the gradient is taken over the entire max-pooling layer, a proportion γ of the largest gradients are then selected with the intent of capturing the most important driving neurons. Here γ is a hyperparameter between 0 and 1. The winning gradients are used

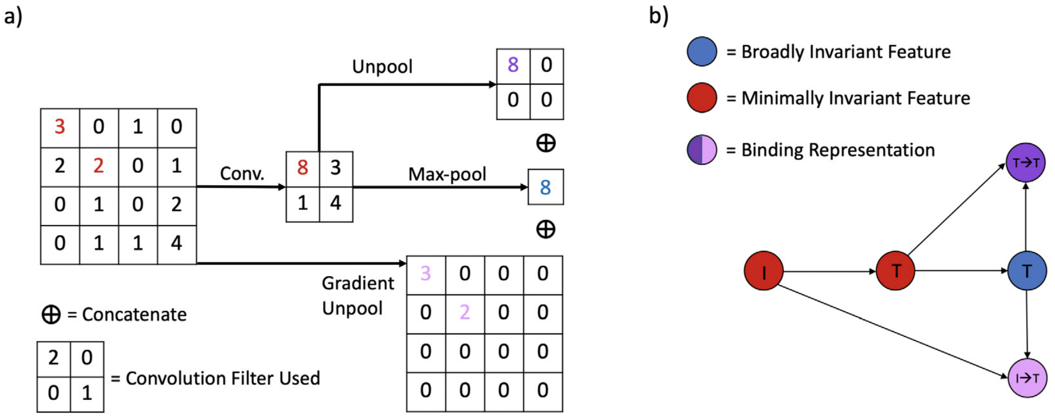


Fig. 2. *Implementing hierarchical binding in a convolutional neural network.* (a) 'Conv.' is a convolution operation with stride of 2 (that is, the weight matrix is shifted by 2 units after every convolution). Our depicted representation of gradient unpooling is simplified for the sake of intuition, as the max-pooling layer in the figure consists of only a single neuron; in reality, we take the gradient of each low-level activation w.r.t. the entire max-pooled layer and only use the proportion γ of the largest gradients to apply a Boolean mask to the activations. (b) A toy diagram to demonstrate the connection to hierarchical binding. The desire is to capture which low-level features (such as a vertical bar or a minimally invariant 'T' neuron) causally drove the more invariant representation of a 'T'. Arrows indicate that a node participated in activating a representation, rather than simply the existence of a connection. These hierarchical binding representations are then made available, alongside the invariant representations, to higher layers.

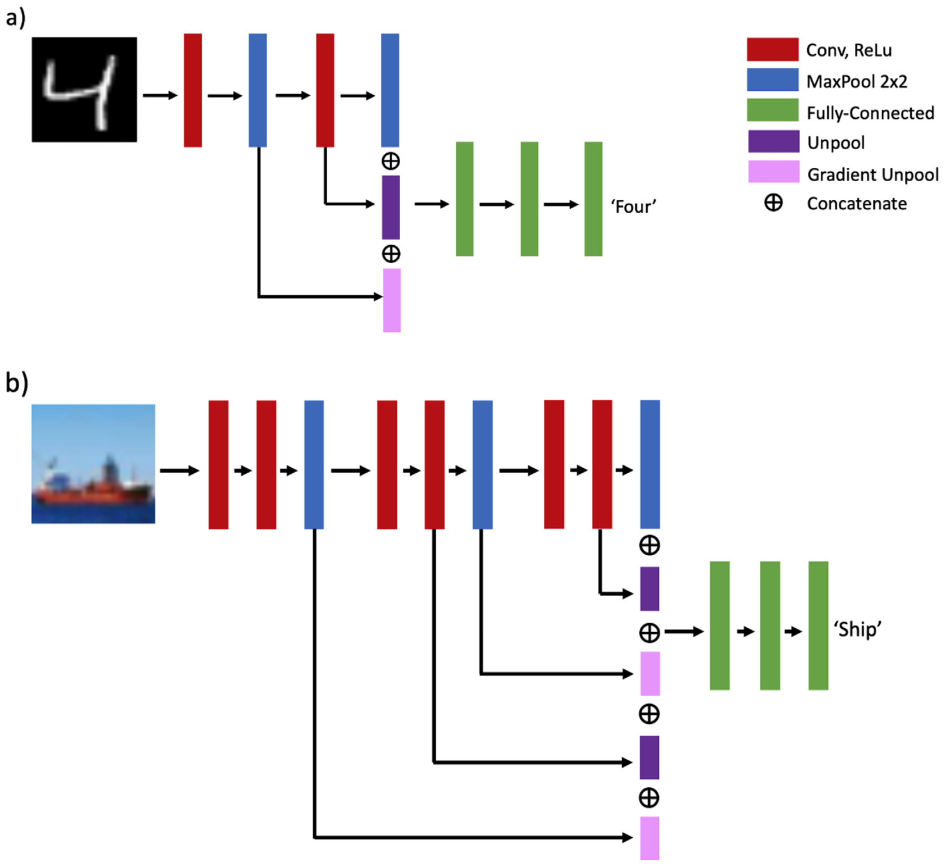


Fig. 3. *Integration of unpooling and gradient unpooling in a hierarchical binding CNN.* The diagram shows the architectures used for (a) MNIST and Fashion-MNIST, based on the LeNet-5 CNN and (b) CIFAR-10, based on a VGG-style CNN. Note that as it is not necessary to use unpooling or gradient-unpooling representations from every layer, deeper architectures can select intermittent representations to use for unpooling or gradient-unpooling, avoiding an excessive growth in parameters.

to generate a Boolean mask applied to the activations of the low-level layer. Denoting our approximation of a layer k unit's causal role with $c_i^{(k)}$, the final sub-layer of gradient-unpooling values are then given by

$$c_i^{(k)} = \begin{cases} a_i^{(k)}, \text{ where } m_i \in \text{top-proportion}_\gamma(m) \\ 0, \text{otherwise} \end{cases} \quad (2)$$

The sub-layer $c^{(k)}$ is then up-projected alongside the unpooling sub-layer. Intuitively, a neuron might have a large activation, but if many of the high-level neurons with which it shares a large weight did not survive max-pooling (i.e. it did not successfully drive them), then its summed gradient m_i is likely to be smaller than that of other neurons. The unit is thus unlikely to survive the γ cut-off and have its activation preserved in the gradient unpooling sub-layer. On the other hand, a neuron with large learned weights to the higher-level features that survived max-pooling is more likely to survive the γ cut-off. Up-projecting the activation value itself then provides some additional measure of how likely the low-level neuron was to have genuinely contributed to the abstract representation in the higher layer.

Our approach seeks to capture the low-level neurons that contributed to the distributed representation in the higher layer, although it is only an approximation of the actual causal relations between them. We also explored alternative mathematical formulations, such as including the activity of the low-level neuron in deriving the Boolean mask. That is, using

$$m_i = a_i^{(k)} \times \sum_j \frac{\partial a_j^{(k+n)}}{\partial a_i^{(k)}} \tag{3}$$

in place of Eq. (1). This offered comparable performance, but was more difficult to implement in the libraries we leveraged, and therefore Eq. (1) was used. Better measures of the importance of a low-level neuron to a higher-level representation exist (see e.g. Dhamdhere et al. (2019)), but we use our method due to its computational efficiency (Simonyan et al., 2014). In our networks, we concatenate the results of unpooling and gradient-unpooling along-side the max-pooled activations in the feed-forward stream. This serves to provide both invariant and spatially detailed representations jointly.

In Fig. 2b, we show how these operations relate to hierarchical feature binding, as described in Eguchi et al. (2018) and Isbister et al. (2018). Note that while these prior works proposed capturing every binding relationship between successive layers k and $k+n$, we seek only to preserve information about how low-level neurons have participated in driving distributed representations in the higher layer. As we demonstrate later, this additional information appears sufficient to enable a more robust object representation when targeted by adversarial attacks.

Model Architectures Both the unpooling and gradient unpooling computations can be introduced into standard CNN architectures, potentially at multiple levels. This work covers its use for CNNs used on the MNIST (Lecun et al., 1998), Fashion-MNIST (FMNIST) (Xiao et al., 2017), and CIFAR-10 (Krizhevsky, 2009) data-sets, which consist of hand-written digits, grey-scale images of clothing, and color images of objects such as horses and airplanes respectively. The CNNs used for MNIST and FMNIST are based on the LeNet-5 architecture (Lecun et al., 1998). In our Hierarchical Binding CNN (HBCNN), the LeNet-5 architecture is augmented with one unpooling and one gradient unpooling sub-layer (Fig. 3a). The models used for CIFAR-10 are based on a deeper CNN using a VGG-like architecture (Simonyan & Zisserman, 2015) (Fig. 3b). For each architectural variant (including the control models without hierarchical binding), hyperparameter tuning for adversarial robustness was performed on a cross-validation data-set (10k examples held out from the original training data-set). To provide an unbiased measure of the effect on robustness, 30 randomly generated networks for each architectural variant were then trained on the full training data-set and evaluated on the test data-set, with the median performance reported in all following results. Further model details are provided in Appendix A.

Regularization Regularization is a means of preventing a network from over-fitting data with a highly complex decision boundary that generalizes poorly to data not seen during training. To perform regularization, we introduce label-smoothing to some of our models, a method that reduces over-confident predictions by replacing the typical one-hot label vector (where the ground-truth class label is associated with a probability of 1, and all other class labels a probability of 0) with a 'one-warm' vector (Pereyra et al., 2019; Szegedy et al., 2016). Specifically, consider the probability distribution of the ground truth labels for a class y where $p(y) = 1$ and $p(k) = 0$ for all other classes $k \neq y$. To perform label smoothing, a smoothing parameter δ and the number of classes K are used to assign the probabilities for incorrect classes as

$$p(k \neq y) = \frac{\delta}{K} \tag{4}$$

The target probability of the correct label is assigned as

$$p(y) = 1 - \delta + \frac{\delta}{K} \tag{5}$$

Thus if the label smoothing parameter was set to 0.1 (the value we use in all our models), the distribution over a total of 10 labels for the first class would change from $[1.0, 0.0, \ldots, 0.0]$ to $[0.91, 0.01, \ldots, 0.01]$. Label smoothing therefore models the concept that the data labels may contain errors, and that the system should be discouraged from creating an overly complex decision boundary to classify these (Goodfellow et al., 2016).

We use label smoothing as it is a widely implemented regularization technique for enhancing adversarial robustness (Chen et al., 2021; Pang et al., 2021; Shafahi et al., 2019; Summers & Dinneen, 2019; Warde-Farley & Goodfellow, 2016). We note, however, that recent research has identified that it can, under certain circumstances, contribute to gradient masking (Lee et al., 2021). This result appears to depend on the precise training details and hyperparameters in question (Chen et al., 2021; Pang et al., 2021), generally being associated with, for example, larger values of label smoothing (Lee et al., 2021). Nevertheless, it is clear that label smoothing can also contribute to genuine increases in adversarial robustness to gradient-free attacks (Chen et al., 2021; Pang et al., 2021). This is consistent with its original motivating use in the literature, namely regularization to improve the quality of a network's predictions. Similarly, subsequent work has established connections between label smoothing and both other forms of regularization such as weight decay, and improved decision boundaries (Lukasik et al., 2020). We note that in our work, we use a small value of label smoothing (0.1), and take several steps to examine for the prevalence of gradient masking (as advised in e.g. Pang et al. (2021)), highlighting it wherever any evidence exists.

With the exception of the 'vanilla' model described later, we also use weight decay in the VGG models to further regularize the decision boundaries. This form of regularization punishes large weights in the network by adding a penalty term to the cost function of the classifier based on the sum of the squares of the network's weights (Ng, 2004). Details on the use of weight decay in the VGG networks is provided in Appendix A.

4. Methods

4.1. Adversarial attacks

Adversarial attacks are a means of producing images that are misclassified by machine vision systems, with the aim that the new image is as close to the original as possible. As measuring how large the modification is perceived to be by a human is challenging, optimization methods are generally used to create an adversarial image that minimizes a measured distance between the original image and the adversary. This distance is typically quantified by the L_0, L_2 or L_∞ norm. In one threat setting, the attack method has extensive access to the model being attacked (such as the gradients of its outputs w.r.t the pixels of the input), in which case it is termed a 'white-box attack'. In settings where the optimization process has limited knowledge about the model being attacked (such as only being able to query how it classifies any given image), the attack is known as a 'black-box' method (Brendel et al., 2018).

Inspired by the thorough evaluation of adversarial robustness in Schott et al. (2019), we evaluate our model against a broad range of black-box and white-box methods, covering L_0, L_2 and L_∞ norm measured attacks, described below. All attacks were evaluated using FoolBox v2.4 (Rauber et al., 2017). As in Schott et al. (2019), our main result is the median distance of adversaries, as this is less affected by outliers than the mean, and unlike when reporting accuracy, is not vulnerable to over-fitting on an arbitrary threshold. For completeness, we also report the accuracy against adversaries bounded by a particular perturbation threshold ϵ. All results presented are based on adversaries generated from a subset of 512 images from the test data-sets. Below, we describe the main attacks used and their core intuition; in Appendix B, we provide additional details such as hyperparameters for the attacks used.

Gradient-Based Attacks In typical training, one performs gradient descent of the loss w.r.t to the model's weights, where the loss is the term that measures the mismatch of the model's predictions to the ground-truth labels. Intuitively, a basic gradient-based attack can use knowledge of the model to perform gradient *ascent* of the loss with respect to the input *pixels*. As a result, pixels can be modified so as to cause the model to misclassify inputs, with the assumption that the modification to the pixels (if sufficiently small) will not have changed the ground-truth label as determined by a human. The Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) uses this principle to construct adversarial examples as follows. Let $J(\theta, \mathbf{x}, y)$ be the cost function used to train the classifier with parameters θ, input $\mathbf{x}$, and targets y. Defining $\boldsymbol{\eta}$ as the adversarial perturbation added to an image, and scaling with parameter ϵ, then one can generate an adversarial perturbation as follows:

$$\boldsymbol{\eta} = \epsilon(\text{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y))) \tag{6}$$

where the adversarial image is then provided by

$$\tilde{\mathbf{x}} = \boldsymbol{\eta} + \mathbf{x} \tag{7}$$

Given the privileged access to the model's gradients these methods have, these are a form of white-box attacks. Various methods have been developed within this class that can be used to minimize either the L_2 or L_∞ norm of the adversarial perturbation; as in the Schott et al. (2019) evaluation protocol, we use the Fast Gradient Method (FGM), FGSM (Goodfellow et al., 2015), L_2 and L_∞ Basic Iterative Method (BIM) (Kurakin et al., 2019), L_2 and L_∞ DeepFool (Moosavi-Dezfooli et al., 2016), and Momentum Iterative Method (MIM) (Dong et al., 2018). For MNIST and FMNIST, these attacks are also repeated with numerically estimated gradients, using the same method leveraged in Schott et al. (2019). Unlike in their study, we also include Projected Gradient Descent (PGD, closely related to BIM) for MNIST, using multiple random starts (Madry et al., 2018).

Decision-Based Attacks These rely only on the decision output of the network, and are therefore a form of black-box attack. A particularly powerful method is the Boundary Attack (Brendel et al., 2018); intuitively, an image is first perturbed by noise until it is misclassified, after which the Boundary Attack iteratively moves the adversary closer to the original image while ensuring it remains misclassified. By taking small steps, it can treat the decision boundary as approximately linear and move along it. Note that the Boundary Attack therefore requires that at a sufficiently small step size, the decision boundary behaves linearly. If this does not hold, then it can fail to operate as intended, which we later discuss as an issue when it is leveraged against our own HBCNN model.

Another decision-based attack is the Pointwise attack. This method adds salt-and-pepper noise until an image is misclassified, then returns as many pixels to their original values as possible, whilst ensuring the image remains misclassified (Schott et al., 2019).

Finally, one can simply add increasing levels of noise (such as salt-and-pepper, uniform, or Gaussian noise) until an image is misclassified. As in Schott et al. (2019), we include the Boundary Attack, their L_0 and L_2 Pointwise Attack, the Salt&Pepper Noise attack and the Gaussian Noise attack. We also include the Blended Uniform noise attack, which blends the image with uniform noise until the input is misclassified (Rauber et al., 2017).

Transfer Attacks Transfer attacks leverage the empirical observation that adversaries generated for one network (called the 'surrogate') can often transfer to other networks, even those with different architectures. The intuition is that the decision boundaries of different systems trained to perform the same task often align, and thus share vulnerable decision regions (Papernot et al., 2016; Tramèr et al., 2017). Robustness to transfer attacks as well as decision-based attacks provides evidence that a model's robustness is not simply a result of 'gradient masking'. In gradient masking, some aspect of the network makes the gradients inaccessible or otherwise less useful for carrying out the optimization steps performed by gradient-based attacks. Often unintentionally introduced by architectural modifications, gradient masking is a common but less desirable method of adversarial robustness, given its ineffectiveness against alternative methods of generating adversaries, and thus the false sense of security it provides (Athalye et al., 2018; Papernot et al., 2017).

In keeping with the above, transfer attacks are most often used to determine whether a particular model displays gradient masking. The presence of such gradient masking can result in the telltale finding that transfer attacks are more effective than white-box methods. However, transfer attacks are weaker than white-box attacks when no gradient masking exists, and they are less effective when it is difficult to craft attacks on the surrogate models used to generate the transfer attacks. They thus they do not represent a gold-standard measure for a model's absolute robustness (Tramèr, Carlini et al., 2020).

In spite of this, robustness to transfer attacks represents its own interesting research question (Tramèr et al., 2017). In particular, our defense is proposed to operate by reducing the number of directions in which the model has sub-optimal decision boundaries, and in which it is therefore vulnerable to adversaries. Transfer attacks exploit the shared dimensions of models with vulnerable decision boundaries (Tramèr et al., 2017), and as such, robustness to transfer attacks would support the geometric motivation behind our defense. Furthermore, as we elaborate in our discussion, transfer attacks are the only method that has been leveraged against humans, and therefore the nature of models robust to transfer attacks (even when vulnerable to white-box attacks), is relevant to understanding the settings under which humans may display sensitivity to adversarial examples. For this reason, transfer attacks form an emphasis of our results, and we take steps to leverage a particularly broad transfer attack designed to identify multiple adversarial directions.

To develop this broad transfer attack, our main addition to the assessment protocol in Schott et al. (2019) is that we derive transfer attacks from multiple surrogate networks, including ones with our proposed architecture. This is to ensure that transfer attack robustness is not simply a result of the HBCNN having an exotic architecture, or being difficult to generate transfers from Tramèr, Carlini et al. (2020), while still being fundamentally vulnerable. Specifically, adversaries are generated using the FGSM, L_∞ BIM, FGM, and L_2 BIM attacks. For FMNIST and CIFAR-10, these are leveraged against both a standard surrogate network and one augmented with binding. For MNIST, we use a particularly sophisticated transfer attack, where we use surrogates that represent multiple versions of the target architecture, as well as the architecture it is being compared to for robustness. For each one of these variants, we use two independent surrogates, such that for any given target architecture, we generate transfer attacks from eight surrogate models. In addition to these, we generate transfer attacks from a ninth surrogate model that has been adversarially trained (note this is a different model from the one we evaluate later), as adversarially trained networks can often produce powerful transfer attacks (Carlini et al., 2019). When attacking a network, all of these adversarial candidates (18 per distance metric) are leveraged. Using the same line-search from Schott et al. (2019), images are iteratively perturbed from the baseline image until they are misclassified, and the minimally perturbed, successfully adversarial transfer image is used in all distance and accuracy measures. Further details on our surrogate models are provided in Appendix B.

4.2. Geometrical measures

Measuring Manifold Geometry A manifold in neural state space represents the set of all points of activity that correspond to the same object, where the activity is determined in response to variations of the input, such as different object details, changes to scale, etc. (Chung et al., 2018). Note the distinction to the data manifold, i.e. the manifold describing the concentration of example images in the embedding (e.g. pixel) space. To distinguish the two, we will refer to the first as the neural manifold. We have argued that the data manifold corresponding to all objects a human would classify as a given object is higher dimensional than often assumed, and that an architecture which can more faithfully represent the additional dimensions of variation (i.e. with a more extensive neural manifold) could support a more robust decision boundary. We are therefore interested in quantifying the geometrical properties of the neural manifolds in the networks we study. To do so, we apply the techniques developed in Chung et al. (2018) and Cohen et al. (2020). This work used statistical mechanical theory to develop a novel measure of the *extent* of a neural manifold, a measure influenced by both a manifold's dimension and radius.

In brief, these measures are developed from the structure of the hyperplane that separates manifolds in binary classification. Comparing any given manifold (e.g. the neural responses to cats) against another, there will be a particular point (the 'anchor point') on the cat manifold that will uniquely define a separating hyperplane to the other manifold. Multiple anchor points will exist for a given manifold to define its separation from other objects, and statistical measures can then be applied to this set of anchor points to estimate geometric properties. Of relevance to our work, the authors derived a measure of manifold radius (R_M), dimension (D_M), and extent ($R_M\sqrt{D_M}$). R_M captures the total variance of the anchor points, normalized by the average distance between manifold centers, while D_M captures the spread of the anchor points along the different manifold axes. Finally, $R_M\sqrt{D_M}$ combines these measures to characterize the manifold's total extent, and we use this quantity as our primary measure of manifold geometry.

To define these concretely, a manifold in neural state space can be defined as having $D+1$ dimensions; one coordinate defines the center of the manifold, while the other coordinates define the axes of variation. This manifold is embedded in the neural activity space of dimension N, where $D < N$, and the activity response to any given image is represented by the vector $\mathbf{x}$. The bold notation of $\mathbf{x}$ indicates that it is a vector in $\mathbb{R}^N$. The set $\mathcal{S}$ is the (in this case finite) set of data sample points that define the manifold. As any given point on the μth manifold M^μ lies in the lower, $D+1$-dimensional linear sub-space, it can be parameterized as

$$\mathbf{x}^\mu(\vec{S}) = \sum_{i=1}^{D+1} S_i \mathbf{u}_i^\mu \tag{8}$$

Here, $\mathbf{u}_i^\mu$ are a set of orthonormal bases of the manifold's linear subspace, and like $\mathbf{x}$, these are N dimensional. The $D+1$ scalars S_i then represent the coordinates of $\mathbf{x}^\mu$ within the manifold's subspace, and the vector of coordinates is constrained to be in the set $\vec{S} \in \mathcal{S}$. Note that unlike the bold notation for $\mathbf{x}$ and $\mathbf{u}_i$, the arrow notation for $\vec{S}$ indicates that it is a vector in $\mathbb{R}^{D+1}$.

With these variables defined, recall that of interest is the separating hyperplane that enables classification of the objects, where there are total of P objects/manifolds. In particular, the aim is to identify a hyperplane that provides the maximum separation capacity, given a separating margin of κ. The separating weight vector is then defined by up to P anchor points, that is, $w = \sum_{\mu=1}^{P} \lambda_\mu y^\mu \tilde{\mathbf{x}}^\mu$, where λ_μ is a multiplier satisfying $\lambda_\mu \geq 0$, y^μ is the binary label, and $\tilde{\mathbf{x}}^\mu \in conv(M^\mu)$, where $\tilde{\mathbf{x}}^\mu$ is the anchor point (denoted by tilde) in the convex hull of M^μ.

To characterize the geometry of the anchor points, they are first projected onto a lower $D+1$-dimensional subspace (where D is determined by the number of data samples per manifold, here 250). Using this projection of each anchor point $\tilde{\mathbf{x}}$ (denoted $\tilde{S}$), one can now define the measures of manifold geometry. In particular, the effective radius and dimension are defined in terms of $\widetilde{\delta S} = (\tilde{S} - S_0)/\|S_0\|$. Here, $\widetilde{\delta S}$ represents the projection of $\tilde{\mathbf{x}}$ onto the $D+1$-dimensional subspace, relative to the manifold's center, S_0. The variation of these points is further normalized by the manifold center norm. For each measure defined below, $\langle \ldots \rangle_{\vec{T}}$ represents an average over random vectors $\vec{T}$ of dimension $D+1$, and with i.i.d normally distributed components $T_i \sim \mathcal{N}(0, 1)$. This statistical component is important, as the anchor points depend on the random orientations of the other object manifolds, and hence on $\vec{T}$. Bringing this together, the manifold radius is defined as the total variance of the normalized anchor points:

$$R_M^2 = \langle \|\widetilde{\delta S}(\vec{T})\|^2 \rangle_{\vec{T}} \tag{9}$$

The effective dimension, D_M, is defined as the angular spread between $\overrightarrow{\delta T} = \vec{T} - T_0$ (where T_0 is the projection of $\vec{T}$ onto S_0) and the anchor point $\widetilde{\delta S}(T)$:

$$D_M = \langle (\overrightarrow{\delta T} \cdot \widehat{\delta S}(\vec{T}))^2 \rangle_{\vec{T}} \tag{10}$$

Here $\widehat{\delta S}$ is a unit vector in the direction of $\widetilde{\delta S}$. Note that $D_M \leq D$. As noted earlier, the manifold extent is then given by $R_M\sqrt{D_M}$. Readers interested in additional mathematical details of these measures, including the determination of a given manifold's anchor points, are advised to refer to Chung et al. (2018) and Cohen et al. (2020).

We sample a total of 200 $\vec{T}$ in our analyses, as advised in the code-base of the implementation used — https://github.com/schung039/neural_manifolds_replicaMFT (Stephenson et al., 2019), and we use a margin value of $\kappa = 1$ for determining anchor points.

Eigenspectrum Analysis We also apply the eigenspectrum analysis described in Stringer et al. (2019) and Nassar et al. (2020). Specifically, given the covariance matrix $\sum$ of a network's layer, we analyze its eigenspectrum, denoted by the descending eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$. The covariance matrix is derived from activity sampled across multiple image presentations (here 750 examples per class in the MNIST test set). We then performed a linear fit to the eigenspectrum in log–log space, as described in Stringer et al. (2019), and implemented in the code-base for Nassar et al. (2020) available at https://github.com/josuenassar/power_law. This linear fit provides an estimate of the power-law exponent, α, which quantifies the decay of the eigenvalues (i.e. amount of described variance) as a function of the eigenvalue's rank. As we discuss further in our results, Stringer et al. (2019) observed that neural activity in mouse visual cortex approximately followed a power law such that the nth eigenvalue (principal component) scaled as $1/n$, i.e. $\alpha = 1$.

As in Stringer et al. (2019), we limit the range of eigenvalues used for this fit; in our case, we limit the eigenvalues to 10 to 350, as our baseline architecture has a total of 400 eigenvalues, while the lower bound of 10 was used in Stringer et al. (2019), as it was above this eigenvalue that they observed the power-law.

5. Experiments

In the following section, we begin by demonstrating that simply introducing hierarchical binding is insufficient for an improved decision boundary, but that additional steps of regularization (through label smoothing), and noise in the training data provide a more robust model. After describing the settings in which we observe improved robustness, we use additional analyses to clarify the role that the expanded manifold due to hierarchical binding plays in this observed effect.

5.1. Hierarchical binding alone

Our opening assumption is that the true manifold representing objects is higher dimensional than typically assumed when designing CNNs, and that capturing this would improve robustness. In particular, the HBCNN architecture affords the possibility of learning better decision boundaries along additional dimensions of variation. Importantly however, the data in a typical data-set such as MNIST will often be concentrated on a lower-dimensional manifold than is likely to underlie human perception. Furthermore, decision boundaries in high dimension are challenging to learn due to the sampling complexity (i.e. the chance of sampling along a particular dimension becomes vanishingly small, and so the amount of information available to learn the decision boundary in these dimensions is limited). We begin by discussing our results in the context of the HBCNN (LeNet-5 variant) leveraged on the MNIST data-set.

Given the above, what then is the effect of introducing binding representations without any additional modifications? Given the sparse sampling in the high-dimensional space, we would expect the augmented model's decision boundary to have limited, if any improvement (Fig. 1b), as it attempts to fit to the few data-points available. These decision boundaries however have a chance of covering directions that are otherwise vulnerable to transfer attacks, and consistent with this, we observe some enhanced robustness to L_2 transfer attacks (Table 1). This is notable, as the inclusion of hierarchical binding alone appears to cause minimal, if any, gradient masking (demonstrated by the effectiveness of the gradient-based PGD attack in Fig. 4). As such, there is nothing to prevent the creation of effective transfer attacks from our surrogates (Tramèr, Carlini et al., 2020). This suggests that across the variety of directions that an image can be perturbed to fool a classifier (including our own model), these are on average *more* effective against an undefended model.

In addition to the enhanced L_2 transfer attack resistance, some enhanced robustness to white-box attacks that use estimated gradients is seen (Table 1), likely as a result of a less smooth loss-landscape caused by the gradient unpooling and unpooling operations, similar to the effect in Xiao et al. (2020). Unlike in Xiao et al. (2020) however, the effects are minimal and sufficiently small so as to not impact white-box attacks with direct access to gradients. The thresholding that our gradient unpooling and unpooling operations introduce also increases resistance to the Boundary Attack, although this is only due to a failure of its ability to operate in its normal regime, rather than an indication of genuinely enhanced robustness. In particular, as the Boundary Attack attempts to navigate the decision boundary, it frequently becomes stuck in local minima and encounters regions where it must decrease its step size for the boundary to behave linearly.

The above results demonstrate an improvement on L_2 transfer attacks, and that elements of the architecture can inadvertently compromise the optimization techniques of some attacks. However, it was also predicted that, in the absence of additional samples, the model would be unable to learn a particularly useful decision boundary. Consistent with this, the trend is that hierarchical binding alone does not enhance robustness across the majority of attacks, and in fact some attacks are more successful against the LeNet-5 variant of the HBCNN than against an unaltered LeNet-5 model (Table 1); clearly the decision boundaries are not yet very useful and can in fact be worse than those of a standard model.

5.2. Hierarchical binding with regularization

To improve the decision boundary, we regularize with label-smoothing. Without such regularization, there is a risk that the model will over-fit to the sparse samples in the high-dimensional space, and fail to form a good approximation of the true decision boundary that would generalize to unseen data. Thus its introduction should promote a more representative (and thereby robust) decision boundary (Fig. 1c). We also apply label-smoothing to the standard CNN so as to ensure a fair control. As expected, we observe an improvement on a range of attacks for both architectures (Table 1), although much of this improvement could also be accounted for by moderate gradient masking. The HBCNN also appears more vulnerable to certain attacks following the addition of label smoothing; without noisy training data (introduced in the next section), the regularized decision boundary can sit close to clean examples and be worse than that of a standard model.

5.3. Hierarchical binding with regularization and noisy training data

To better sample the high-dimensional data manifold corresponding to human perception, we introduce Gaussian ($\sigma = 0.3$) and salt-and-pepper (120/784 pixels perturbed) noise during training (Fig. 1d). This is motivated by previous work where training with noise has been associated with increased robustness to adversarial attacks, measured either by the average adversary distance or the accuracy at a fixed-threshold perturbation (Ford et al., 2019; Rusak et al., 2020; Zantedeschi et al., 2017). For both the control model and the HBCNN, we also increase the dimension of the fully connected layers to 256 and 128, and train for longer, as cross-validation data suggested this better modeled the more complex data-set in both cases, and further enhanced robustness.

As shown in Table 2, this creates a network ('HBCNN+S+N' in the table) with enhanced robustness to virtually all attacks relative to the main control model. This result supports the notion that introducing hierarchical binding can support developing a more robust model.

As we do not perform adversarial training in our models, we cannot guarantee that any particular model will have developed a robust decision boundary. As such, we observe that our models follow a distribution in robustness, and in addition to the tabular results, we present distortion curves (Fig. 5) and histograms (Fig. 6) of model robustness for several key attacks. Fig. 5 in particular reveals that the primary robustness benefit of the

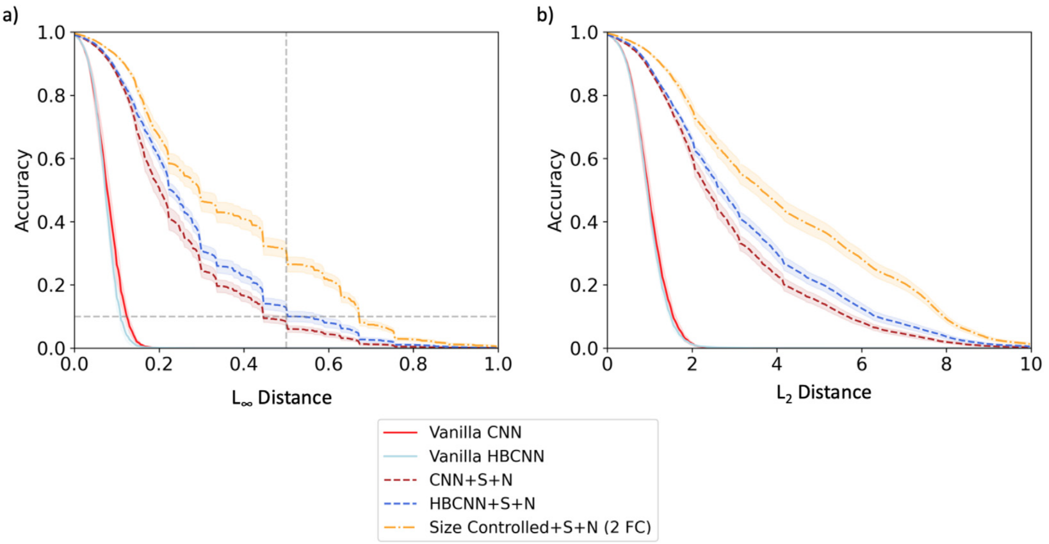


Fig. 4. *Distortion-accuracy curves for the vanilla and fully augmented (label smoothing and noisy training data) models.* The *y*-axis shows the accuracy of the model as the allowable distortion for the attack increases. (a) L_∞ PGD attack, (b) L_2 PGD attack. Accuracy above chance (horizontal gray line) on the L_∞ PGD attack beyond a distortion of 0.5 (vertical gray line) provides evidence of gradient masking. S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training. Shaded areas indicate 95% confidence interval of the mean across the 30 sample networks for each architecture.

Table 1
MNIST results (Part 1).

	Vanilla-CNN	Vanilla-HBCNN	CNN+S	HBCNN+S
Clean accuracy	99.25	99.13	99.17%	99.08%
L_2-metric ($\epsilon = 1.5$)				
Transfer ■	2.4 (84%)	2.6 (86%)	2.8 (85%)	2.8 (87%)
Uniform Noise ■	9.0 (99%)	8.7 (99%)	7.3 (98%)	7.1 (99%)
Gaussian Noise ■	7.1 (98%)	5.9 (98%)	6.8 (98%)	5.5 (98%)
Boundary ■	1.6 (57%)	7.2 (98%)	1.5 (50%)	5.3 (96%)
Pointwise ■	3.4 (96%)	2.9 (93%)	3.4 (95%)	2.7 (93%)
FGM	3.0 (82%)	2.8 (78%)	8.1 (89%)	7.8 (91%)
FGM w/GE	3.3 (82%)	6.6 (91%)	7.9 (89%)	10.1 (95%)
DeepFool	1.4 (46%)	1.4 (39%)	4.5 (82%)	5.5 (84%)
DeepFool w/GE	1.7 (59%)	1.6 (54%)	6.5 (88%)	4.8 (86%)
BIM	1.3 (37%)	1.3 (34%)	2.4 (67%)	3.2 (73%)
BIM w/GE	1.2 (34%)	1.5 (51%)	2.4 (67%)	3.1 (77%)
PGD	1.0 (13%)	1.0 (12%)	1.3 (41%)	1.6 (54%)
All L_2	1.0 (12%)	1.0 (12%)	1.1 (29%)	1.5 (52%)
L_∞-metric ($\epsilon = 0.3$)				
Transfer ■	0.21 (13%)	0.21 (9%)	0.23 (28%)	0.23 (20%)
FGSM	0.18 (15%)	0.18 (20%)	0.46 (70%)	0.46 (73%)
FGSM w/GE	0.23 (38%)	0.40 (54%)	0.46 (72%)	0.46 (74%)
DeepFool	0.12 (0%)	0.12 (0%)	0.37 (60%)	0.43 (67%)
DeepFool w/GE	0.14 (0%)	0.13 (1%)	0.59 (75%)	0.44 (65%)
BIM	0.10 (0%)	0.10 (0%)	0.19 (37%)	0.29 (48%)
BIM w/GE	0.10 (9%)	0.11 (29%)	0.18 (35%)	0.29 (48%)
MIM	0.10 (0%)	0.10 (0%)	0.21 (37%)	0.30 (50%)
MIM w/GE	0.10 (9%)	0.13 (27%)	0.21 (37%)	0.30 (50%)
PGD	0.08 (0%)	0.08 (0%)	0.09 (6%)	0.11 (4%)
All L_∞	0.08 (0%)	0.08 (0%)	0.09 (4%)	0.11 (4%)
L_∞-metric ($\epsilon = 12$)				
Pointwise ×10 ■	9 (28%)	7 (14%)	8 (23%)	6 (6%)
Salt&Pepper Noise ■	56 (93%)	22 (73%)	48 (92%)	17 (67%)
All L_0	9 (28%)	7 (14%)	8 (23%)	6 (6%)

Results are presented from leveraging a variety of attacks against architectures of interest evaluated on the MNIST data-set; note that the main MNIST results are split across two tables (this table and Table 2) due to page-width constraints. The main numerical results shown here are the median L_p distances of a successful adversary for the different attacks (rows), provided as the median performance across 30 networks for each model condition (columns). In parentheses is the median accuracy, at a given thresholded perturbation ϵ, taken across 30 networks. The All-L_0, All-L_2 and All-L_∞ distances show the minimal adversarial distance across all attacks of that distance-metric for each image. Bold indicates the best performance between the networks with noise added to the training images; blue indicates the best performance across all networks in both tables. 'Clean' refers to the accuracy on the test data-set without adversarial perturbations; 'Vanilla' indicates the absence of the modifications (regularization with label smoothing or noise in training data) introduced later; HBCNN = Hierarchical Binding-CNN (LeNet-5 variant); S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; w/GE = with Gradient Estimation; ■ = black-box attack.

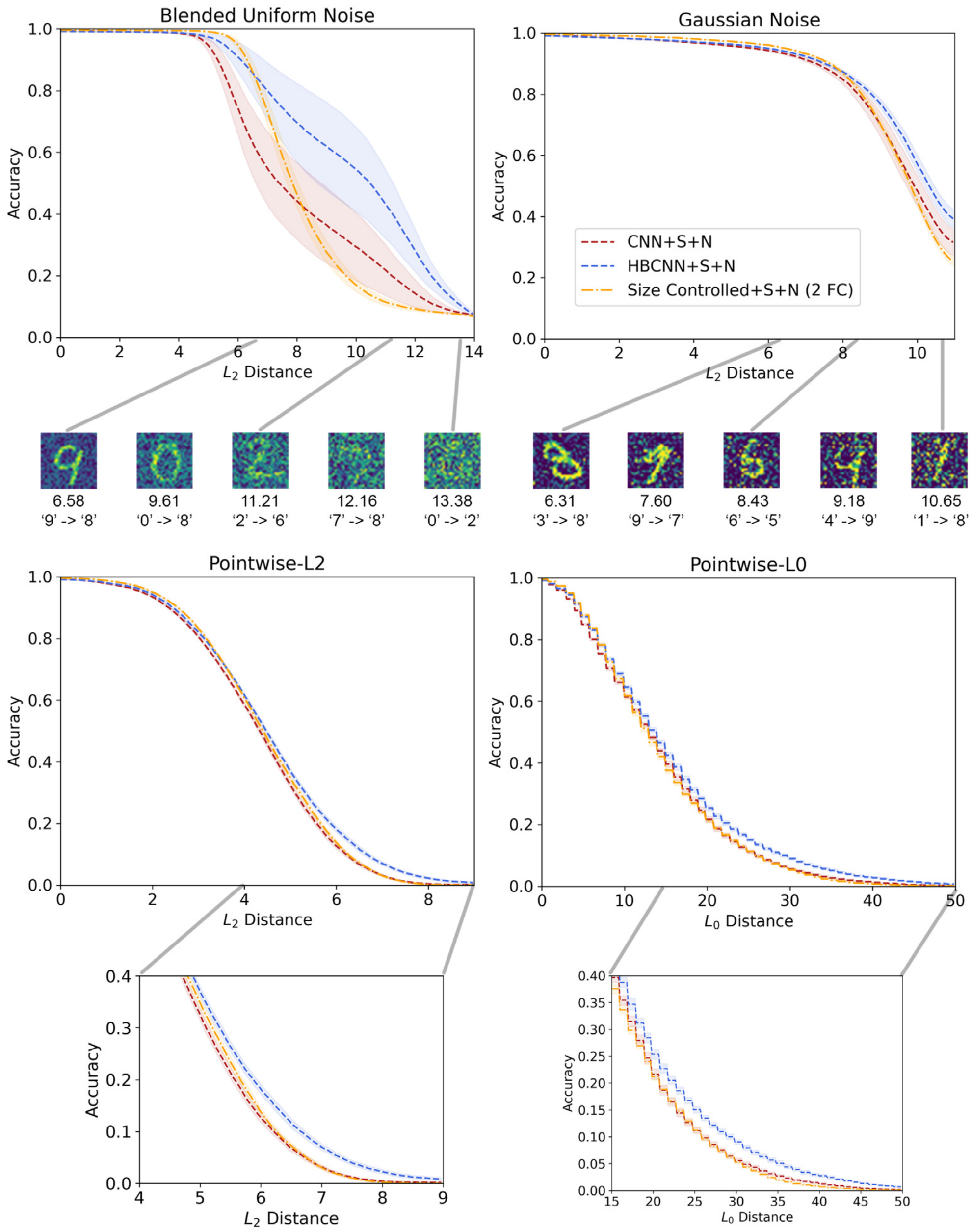


Fig. 5. *Distortion-accuracy curves for the fully augmented (label smoothing and noisy training data) models on black-box attacks.* The *y*-axis shows the accuracy of the model as the allowable distortion for the attack increases. The digits displayed under Blended Uniform and Gaussian noise indicate example adversaries that successfully fooled the HBCNN+S+N model, providing both their L_2 distance, as well as the change in classification. For the Gaussian Noise attack, the attempted additive noise does not extend beyond an L_2 distance of 11. The plots expanded from the Pointwise attacks show zoomed in views of the distortion curves. The shaded regions indicate the 95% confidence interval of the mean across the 30 sample networks for each architecture.

HBCNN+S+N emerges at higher values of distortion. We include example images that the HBCNN+S+N has misclassified at varying amounts of noise to demonstrate that the improved robustness is not associated with (implausible) super-human performance.

The robustness we observe to such a wide range of black and white-box attacks suggests that gradient masking alone cannot explain our results. Furthermore, our distortion-accuracy plot in Fig. 4 demonstrates that with an L_∞ perturbation budget of 0.5 (sufficient to change every pixel-value to gray), the accuracy of the HBCNN models reassuringly reach that of random guessing (Carlini et al., 2019). Despite this, some of our white-box attack results suggest that in both the control ('LeNet+S+N' in Table 2) and HBCNN+S+N, a degree of gradient-masking is present. This appears to be an unintended consequence of label smoothing (see our Methods), as well as the discontinuities introduced by the thresholding operation in unpooling and gradient-unpooling (Xiao et al., 2020). We emphasize that the interesting result is the broad range of enhanced robustness seen, in particular to black-box attacks, and across multiple L_p norms. With the exception of the Boundary Attack, these black-box attacks rely on sampling directions from the input to find an adversarial region, and the improved robustness points to the effectiveness of hierarchical binding in reducing the vulnerability of these directions.

We note that the transfer attack robustness, while small in magnitude, is particularly significant given that we leverage such a broad transfer attack against the MNIST-trained models, and given the relationship of transfer attacks to our opening geometric intuition. Many transfer attack evaluations in the literature use

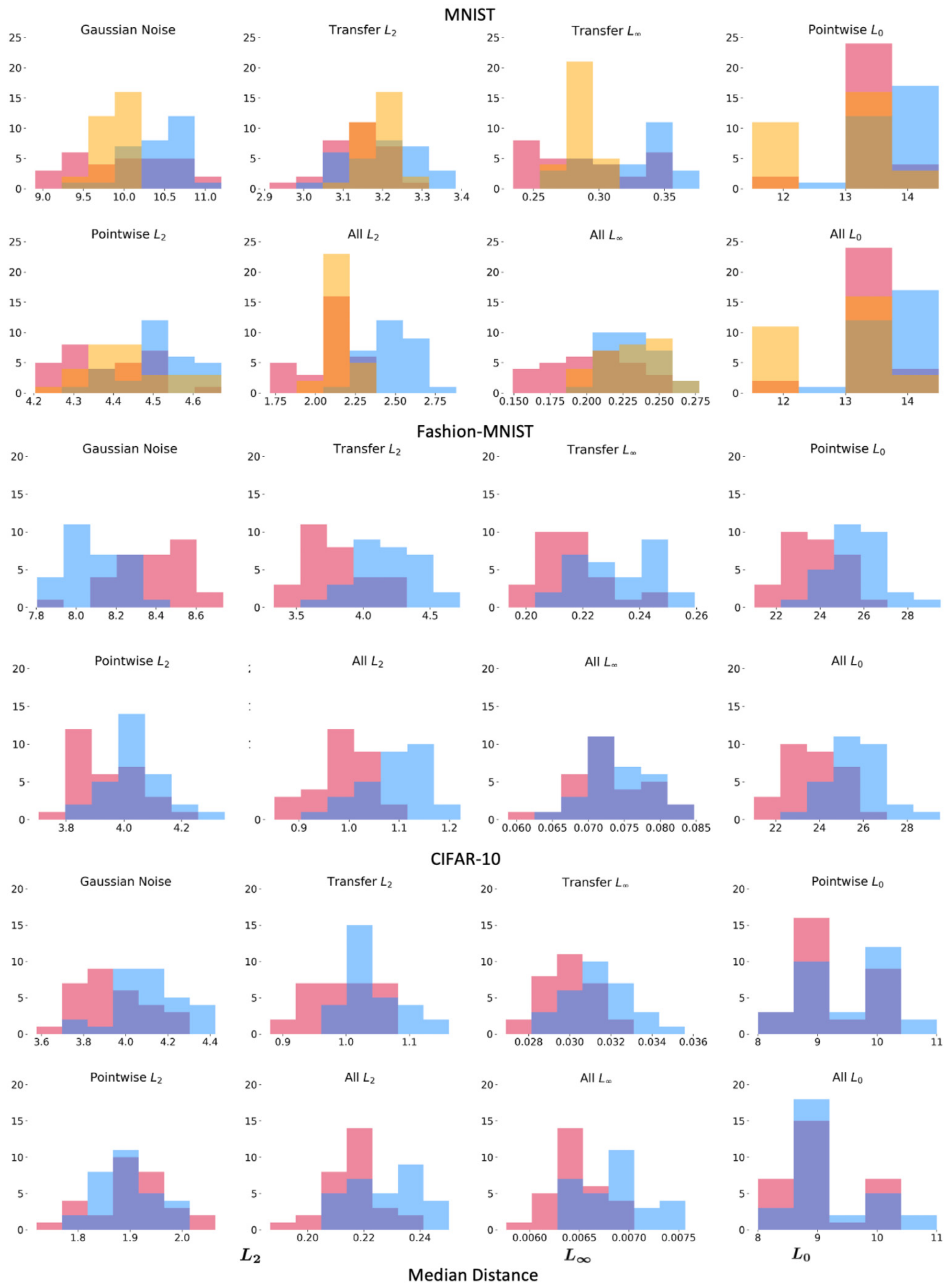


Fig. 6. *Distributions of the median performance of networks*. Shown are the distributions of the median performance of networks (30 for each condition). Blue indicates results for the HBCNN (LeNet-5 or VGG variant as appropriate for the data-set) + label smoothing (S) + noise in the training data (N); red represents the LeNet-5/VGG+S+N as appropriate for the data-set; yellow indicates the Size-controlled CNN+S+N (2 fully connected layers). The *x*-axis for each distribution indicates the distance between the original image and the adversary, using the distance measure appropriate for that attack. Note that we do not show results for every attack leveraged, but select those we deemed particularly important such as black-box attacks and the 'All' attacks evaluations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

a single model, or even one with a different architecture to generate the transfer images. We use a total of 9 surrogate models and 18 adversarial candidates for each attack, including surrogates that have been adversarially trained and independent equivalents of the architecture being attacked. Using so many candidates in our evaluation is important, as our defense is proposed to reduce the number of directions in which our HBCNN architecture and other models share vulnerable decision boundaries, which transfer attacks are otherwise designed to exploit (Tramèr et al., 2017). The observed improvement in robustness therefore lends evidence to our opening claim that, while a sufficiently powerful attacker can break our defense, these directions are more challenging to find in the HBCNN than in a standard model.

5.4. Comparisons to additional models on MNIST

To control for the possibility that the greater number of parameters in the HBCNN simply enables it to fit the noisy training data better, we also train two larger CNNs with an equivalent number of parameters to the HBCNN (results in Table 2). The first version has a single fully connected layer, as is common in

Table 2
MNIST results (Part 2).

	CNN+S+N	HBCNN +S+N	Sized-1FC +S+N	Sized-2FC +S+N	CNN+AT
Clean accuracy	99.10%	99.05%	99.43	**99.45%**	98.40%
***L_2*-metric** ($\epsilon = 1.5$)					
Transfer ■	3.1 (89%)	**3.2** (89%)	3.0 (92%)	**3.2** (92%)	3.7 (95%)
Uniform Noise ■	6.6 (99%)	**11.3** (99%)	6.2 (99%)	7.9 (100%)	8.7 (98%)
Gaussian Noise ■	10.0 (99%)	**10.5** (99%)	9.7 (99%)	9.9 (99%)	5.3 (97%)
Boundary ■	2.5 (88%)	**9.2** (98%)	2.1 (84%)	2.3 (88%)	1.4 (42%)
Pointwise ■	4.4 (96%)	**4.5** (97%)	4.2 (97%)	4.4 (97%)	1.9 (73%)
FGM	8.4 (92%)	9.7 (93%)	**9.8** (94%)	8.6 (96%)	∞ (95%)
FGM w/GE	8.2 (92%)	∞ (95%)	9.6 (94%)	8.3 (96%)	∞ (95%)
DeepFool	7.3 (91%)	8.1 (93%)	3.6 (89%)	**9.3** (96%)	9.4 (94%)
DeepFool w/GE	7.3 (93%)	7.7 (94%)	4.6 (91%)	**9.7** (96%)	9.5 (94%)
BIM	3.6 (84%)	4.0 (86%)	3.0 (85%)	**4.6** (92%)	4.9 (93%)
BIM w/GE	3.6 (84%)	4.0 (87%)	3.1 (84%)	**4.5** (92%)	4.5 (93%)
PGD	2.5 (77%)	2.8 (79%)	1.9 (71%)	**3.7** (87%)	2.8 (86%)
All *L_2*	2.1 (75%)	**2.5** (78%)	1.8 (69%)	2.1 (81%)	1.4 (39%)
***L_∞*-metric** ($\epsilon = 0.3$)					
Transfer ■	0.29 (43%)	**0.33** (60%)	0.26 (33%)	0.28 (41%)	0.39 (94%)
FGSM	0.48 (76%)	**0.62** (82%)	0.50 (76%)	0.46 (73%)	0.44 (95%)
FGSM w/GE	0.50 (78%)	**0.63** (83%)	0.51 (76%)	0.47 (79%)	∞ (95%)
DeepFool	0.83 (81%)	**1.0** (85%)	0.32 (54%)	**1.0** (94%)	0.46 (94%)
DeepFool w/GE	**1.0** (86%)	**1.0** (86%)	0.45 (73%)	**1.0** (95%)	0.71 (94%)
BIM	0.34 (56%)	0.44 (66%)	0.25 (31%)	**0.45** (63%)	0.36 (93%)
BIM w/GE	0.34 (57%)	**0.45** (67%)	0.25 (31%)	**0.45** (62%)	0.63 (93%)
MIM	0.32 (54%)	**0.42** (66%)	0.25 (33%)	0.41 (59%)	0.34 (93%)
MIM w/GE	0.35 (56%)	**0.44** (68%)	0.26 (38%)	0.42 (60%)	0.44 (94%)
PGD	0.22 (28%)	0.24 (33%)	0.15 (0%)	**0.30** (49%)	0.33 (91%)
All *L_∞*	0.20 (20%)	0.22 (25%)	0.15 (0%)	**0.23** (25%)	0.33 (91%)
***L_0*-metric** ($\epsilon = 12$)					
Pointwise ×10 ■	13 (53%)	**14** (56%)	12 (46%)	13 (52%)	4 (0%)
Salt&Pepper Noise ■	142 (97%)	135 (97%)	**156** (98%)	150 (98%)	14 (57%)
All *L_0*	13 (53%)	**14** (56%)	12 (46%)	13 (52%)	4 (0%)

Results are presented from leveraging a variety of attacks against architectures of interest evaluated on the MNIST data-set; note that the main MNIST results are split across two tables (this table and Table 1) due to page-width constraints. The main numerical results shown here are the median L_p distances of a successful adversary for the different attacks (rows), provided as the median performance across 30 networks for each model condition (columns). In parentheses is the median accuracy, at a given thresholded perturbation ϵ, taken across 30 networks. The All-L_0, All-L_2 and All-L_∞ distances show the minimal adversarial distance across all attacks of that distance-metric for each image. Bold indicates the best performance between the networks with noise added to the training images; blue indicates the best performance across all networks in both tables. 'Clean' refers to the accuracy on the test data-set without adversarial perturbations; HBCNN = Hierarchical Binding-CNN (LeNet-5 variant); Sized-*n*FC = size-controlled CNN with either $n = 1$ or 2 fully-connected layers; AT = adversarial training; S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; w/GE = with Gradient Estimation; ■ = black-box attack.

adversarially robust benchmark CNNs, such as the adversarially trained model in Madry et al. (2018). The second has two fully connected layers, like the baseline CNN and the HBCNN model. The performance of both architectures is in general not comparable, and in some cases, the larger models are more vulnerable than the smaller, baseline architecture. For the stronger, size-controlled CNN with two fully connected layers, this is the case for several important black-box attacks we evaluate, including the transfer L_∞, Boundary, and additive Gaussian noise attacks. These results indicate that arbitrarily adding parameters can increase the vulnerability of a model.

The primary exception to the enhanced vulnerability of the Size-controlled CNN+S+N (2 fully connected layers) model is the results of several gradient-based attacks, and the transfer L_2 attack. Regarding gradient based attacks, this difference is likely a consequence of a significant increase in gradient masking in the size-controlled model, an uninteresting defense. Evidence of gradient masking in this model includes that a distortion budget of $L_\infty = 0.5$ is insufficient to bring accuracy to chance levels (Fig. 4). This is significant, as the convergence plots of the PGD attacks in Fig. B.13 demonstrate that the PGD is as well optimized as possible. The existence of considerable gradient masking is further supported by the observation that the PGD attacks perform worse than the equivalent transfer attacks for this model, and the significantly better performance of the Boundary Attack in comparison to PGD (Table 2). For the transfer L_2 result, we highlight that the size-controlled models were not used as surrogate models when generating attacks for transfer (owing, in part, to the significant gradient masking), and so it is promising that the HBCNN still at least matches the robustness of the deeper size-controlled model under this setting. More importantly, in spite of this limited transfer attack against the size-controlled model in comparison to the HBCNN model, the HBCNN model is considerably more robust on transfer L_∞ attacks. In summary, these results support the inductive bias in our architecture that abstract and low-level features should be represented in their own right; it is not a question of simply adding more free parameters to attain robustness.

In terms of why the Size-controlled CNN+S+N (2 fully connected layers) model displays gradient masking, this is likely an unintended consequence of label smoothing, and its complex interactions with the properties of any specific model. We highlight that other established defenses may, in a similar manner, contribute to both a degree of genuine robustness, as well as gradient masking. For example, the well known use of adversarial training (discussed below) may contribute to gradient masking (Khoury & Hadfield-Menell, 2018; Tramèr & Boneh, 2019), which would be consistent with the finding that many black-box attacks can outperform strong gradient-based attacks against adversarially trained models (Schott et al., 2019).

We now compare our results to models that have undergone adversarial training. Adversarial training (AT) is a process of optimizing the parameters of a network given both clean training examples and adversarial images generated for that network. This process continues in an iterative fashion, with new adversaries generated as the model attempts to develop a robust decision boundary, and this remains one of the strongest methods for defending classifiers (Madry et al., 2018). We highlight that our 'HBCNN+S+N' model beats adversarial training on several black-box attacks, the All L_2 metric, and the All L_0 metric. These results are consistent with the observation that adversarial training can reduce robustness to noise (Dapello et al., 2020; Rusak et al., 2020), and black-box attacks more generally (Schott et al., 2019). We emphasize that the adversarially trained model from Madry et al. (2018) uses many more parameters; whereas our method has around 850,000 parameters, the AT model is nearly four times as large with approximately 3,275,000 trainable parameters. Despite our method achieving comparable or better robustness on several metrics using significantly fewer parameters, we also observe higher classification accuracy on the clean data-set (99.05% for our model vs 98.40% for the AT model).

We do not include the robust Analysis by Synthesis (ABS) model (Schott et al., 2019) due to the computational resources required to run it (around three orders of magnitude more time for a forward pass on our GPUs in comparison to the HBCNN). Comparing the results in Table 1 of Schott et al. (2019), with those of the 'HBCNN+S+N' model in our Table 2, our network outperforms their non-binary ABS model's state-of-the-art (SOTA) All L_2 result (median distance 2.3 for ABS vs median distance 2.5 for ours), and achieves similar results on Gaussian noise and Transfer L_∞ attacks. Our results are thus in keeping with SOTA robustness on MNIST according to the All L_2 evaluation, however we do not make any such claim due to the difficulties of a fair, head-to-head comparison, and limitations of our defense that we raise in the discussion.

5.5. Geometric measures of network representations

Manifold Extent and Robustness We have suggested that vulnerability to adversarial examples can be reduced by developing models that more faithfully represent the high-dimensional manifolds that underlie human perception. We therefore proposed an architecture with the intent of better modeling additional degrees of low-level variation. We subsequently trained with noise, in the hope of sampling more densely from the data manifold that corresponds to human representations of hand-written digits, as opposed to the more constrained data manifold found in the original MNIST training set. While these modifications were associated with enhanced robustness, it is important to determine whether our modified architecture does indeed model greater degrees of object variation, and the degree to which this is associated with robustness.

To measure the geometry of the neural manifolds in our different architectures, we apply the techniques of Chung et al. (2018) and Cohen et al. (2020). In particular, these enable us to evaluate the average dimension (D_M), radius (R_M), and extent ($R_M\sqrt{D_M}$) of the neural manifolds for the object classes. Based on our opening proposal, we would expect the neural manifolds in the hierarchical binding network's layer immediately proceeding the fully connected layers of the network to be higher dimensional than in the other architectures. Furthermore, we would expect the HBCNN to be better able to model the natural variation of objects across these dimensions, which would manifest as a larger average manifold radius. The manifold extent captures (and is proportional to) both of these metrics, and should therefore also be increased under our opening proposal. Consistent with this, we observe that all of these measures are elevated in the HBCNN relative to the other architectures in the layer immediately proceeding the fully connected parts of the networks (Fig. 7).

Furthermore, our proposal suggests that expanded neural manifolds would be associated with robustness to the measures we are most interested in, namely robustness to black-box attacks. We therefore estimate the manifold extent in the layer of activity immediately proceeding the fully connected layers, and compare this to our measures of robustness. As predicted, across all of these attacks, and importantly *across all three metrics of L_2, L_∞, and L_0 adversaries*, we observe a positive relationship between manifold extent and robustness (Fig. 8). This provides evidence that, not only does the introduction of hierarchical binding expand the neural manifold, but that this modification is associated with the improvement in robustness observed in the HBCNN.

It is also worth highlighting that, although expanding the manifold is associated with a more robust decision boundary, it does not actually guarantee the robustness of said boundary. For example, it is clear from our results that even models with small manifold extents can occasionally be robust. Our results therefore do not support that expanded manifolds are necessary, nor sufficient, for robustness in the setting that we explore. Importantly however, there is a clear association between models with expanded manifolds and robustness.

It is also interesting to note that, comparing the two size-controlled CNN architectures, the network with two-fully connected layers appears less constrained to rapidly minimize its manifold radius, dimension, and extent across the layers (Fig. 7), and this might partly explain some of its greater robustness. This is noteworthy, as many CNNs in the adversarial examples literature, such as the robust model in Madry et al. (2018), or the baseline control models in Nassar et al. (2020) and Schott et al. (2019), use a single fully connected layer.

As an additional point of interest, it is worth examining where the adversarially trained model from Madry et al. (2018) sits in these evaluations. With only one sample, and robustness measures that are significant outliers, we mark the location of the model on our scatter plots with a vertical gray line (Fig. 8). We note that the manifold extent is above virtually all of the non-HBCNN models, a finding that is consistent with the positive relationship we observe between manifold extent and robustness.

Power Law Scaling of Variance Following on from the above analysis, it is interesting to compare our proposal to another possible connection between the nature of neural manifolds and robustness. In Stringer et al. (2019), it was observed that the neural responses in mouse visual cortex were high-dimensional, and that the variance of the nth dimension scaled according to the power law $n^{-\alpha}$, where $\alpha \approx 1$. Stringer et al. (2019) proposed that the brain maintains a high-dimensional representation of the input, while ensuring that it decays sufficiently quickly that the underlying manifold is smooth. If the representation decayed more slowly, then the manifold would be non-differentiable, which could manifest in sudden changes in responses given small shifts in the input space. For this reason, they proposed that a power law decay with $\alpha \approx 1$ could support robustness, including to adversarial examples. In Nassar et al. (2020), this proposal was directly tested in deep neural networks by implementing a form of regularization that explicitly promoted the eigenspectra of neural responses to satisfy $\alpha = 1$. They observed that the introduction of such regularization promoted robustness to both white-box attacks and general noise corruption, suggesting that this could partially explain the robustness of biological vision.

It is reasonable to wonder whether the introduction of hierarchical binding results in a similar change as the regularization term used in Nassar et al. (2020). If it does, this would be a welcome outcome, as the regularization term in Nassar et al. (2020)

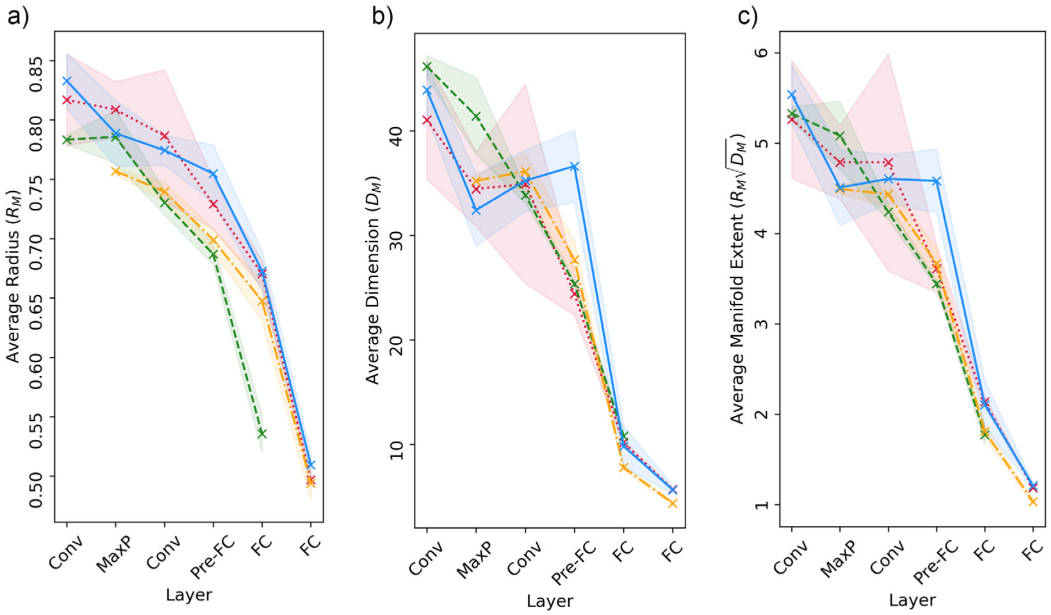


Fig. 7. *Manifold properties as a function of layer.* Presented is the average neural manifold (a) radius, (b) dimension, and (c) extent as a function of layer for the architectures we evaluate on MNIST. Note that the Size Controlled+S+N (2 FC) model has no result for its first layer, owing to insufficient GPU memory for determining this. Values represent the mean across 5 sample networks, while shaded regions indicate the 95% confidence interval. Conv = post convolution and ReLU operation; MaxP = post max-pool operation; FC = post fully-connected weights and ReLU operation. For the non-binding models, the Pre-FC layer is equivalent to MaxP. S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; (n FC) refers to the number of fully connected layers in the size-controlled models.

is computationally intensive to implement, which limited their approach to MNIST. Furthermore, it is not clear how this power law spectrum would manifest naturally in the brain (i.e. without artificially punishing representations with a regularization term).

We therefore used the same approach applied in Stringer et al. (2019) to fit an α exponent to the eigenspectrum of each of our networks, using the layer of activity immediately proceeding the fully connected layers. Some example eigenspectra are provided in Fig. 9. Interestingly, the α exponent is indeed closer to 1 in the HBCNN (mean 1.27 $\pm$ 0.02, 95% CI) vs. the baseline LeNet model (mean 1.57$\pm$0.01, 95% CI), or (included for comparison), the baseline CNN architecture used in Nassar et al. (2020) (1.46, result from a single model). However, this is not unique to the HBCNN, and in general, the α exponent appears, if anything, closer for our two size-controlled models, the size-controlled CNN with a single fully-connected layer (mean 1.24 $\pm$ 0.02, 95% CI), and with two fully connected layers (mean 1.245 $\pm$ 0.005, 95% CI). Consistent with this, we do not observe an obvious relationship across the architecture types between the α exponent and any of our primary black-box robustness metrics (Fig. 10). While some of the scatter plots, such as for the Gaussian noise attack, suggest possible correlations within a given architecture, these do not appear to be in consistent directions. In summary, and unlike the measure of manifold extent related to our opening proposal, there is no evidence that a power-law spectrum might explain the robustness changes seen across our architectural variants.

We noted earlier that the adversarially trained model in general has an elevated manifold extent compared to typical networks. It is interesting to therefore wonder where the estimated α exponent of the adversarially trained model from Madry et al. (2018) sits. Curiously, we find that the model actually has a *larger* α exponent than any of the other models (Fig. 10), where we once again mark the adversarially trained model with a gray line. This may seem counter-intuitive given the result in Nassar et al. (2020) suggesting that standard architectures have larger α exponents, and that regularizing them to fall closer to 1 improves robustness. It is, however, consistent with the original proposal in Stringer et al. (2019) that a larger α exponent would result in a smoother (and therefore a potentially more robust) manifold, albeit at a potential cost in the efficiency of the neural code.

5.6. Understanding the contribution of hierarchical binding

We have argued that (a) the binding representations would enable robustness by augmenting the low-dimensional representations already present in a typical CNN architecture, (b) these representations are meaningful because they capture the low-level features that causally drive higher-level features, and (c) hierarchical binding reduces the number of directions in which the decision boundary is vulnerable to potential attacks. In the following, we provide results exploring these claims.

To address (a) and (b), we first perform an analysis where we vary a key hyperparameter, the γ proportion, in our MNIST models. While our main results in the previous section supported point (c), effects from inadvertent gradient masking, which can also affect the successful creation of transfer attacks (Tramèr, Carlini et al., 2020), makes unequivocally demonstrating this effect challenging. We therefore also perform an analysis with a magnitude-constrained Gaussian attack, demonstrating that hierarchical binding significantly reduces the probability of sampling and identifying a region with an incorrect decision boundary.

5.6.1. The importance of binding dimension and causality

The γ hyperparameter in our model determines the proportion of largest gradients that are used to mask the low-level feature activations that will be up-projected. We implemented this as a means of approximating which of these low-level features were most important in driving the higher representation, with the prediction that the most causally important neurons would be the most useful to augment the other-wise low-dimensional manifold. To explore this, we systematically vary γ, and observe the effect on the robustness of a network. This analysis demonstrates:

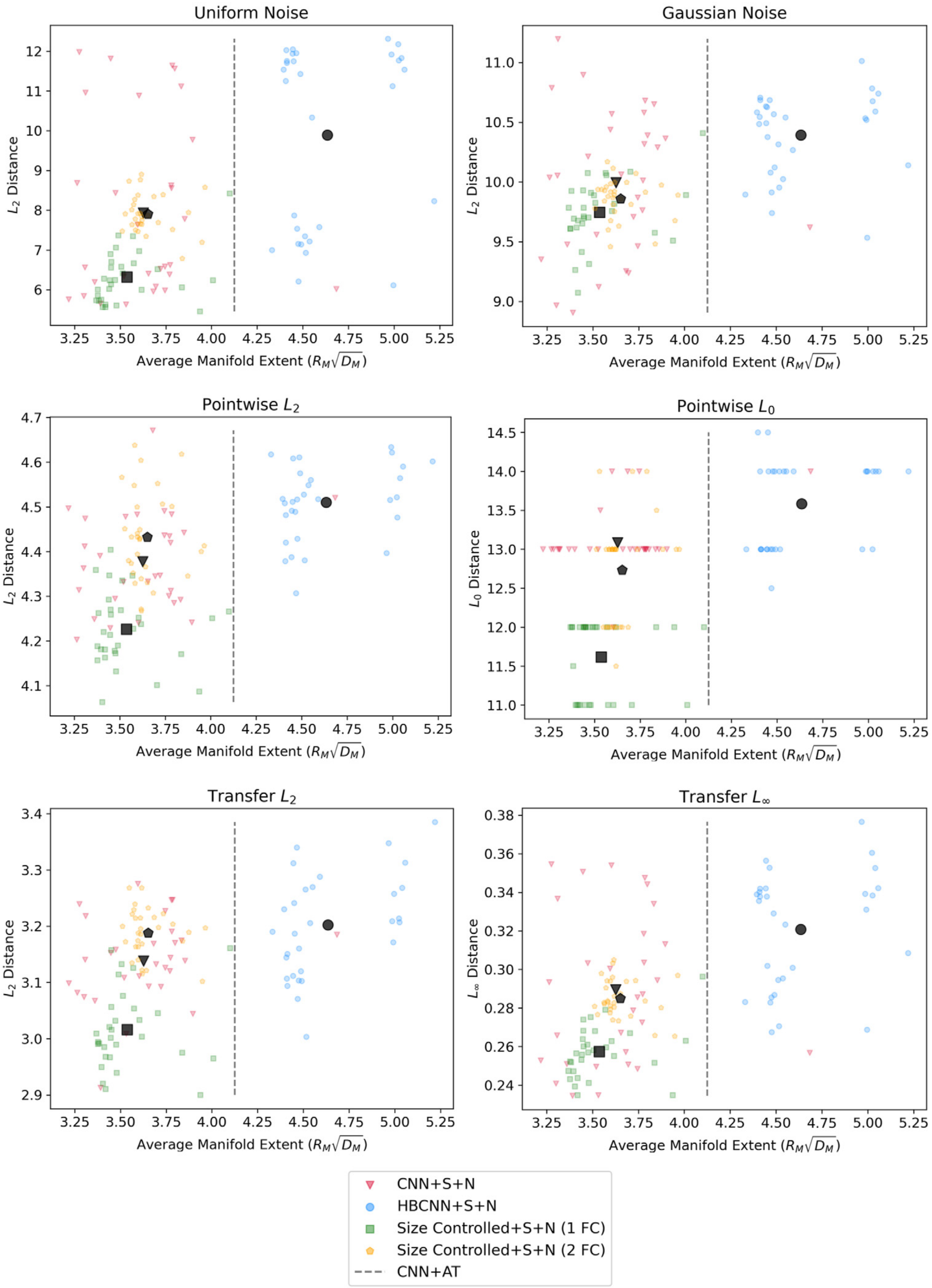


Fig. 8. *Robustness to black-box attacks vs. manifold extent.* The robustness of individual models as a function of their average manifold extent ($R_M\sqrt{D_M}$). Manifold extent is determined for the layer of activity immediately proceeding the fully connected layers. The black markers indicate the mean value for a given architecture type. The dashed gray-line indicates the $R_M\sqrt{D_M}$ value of the adversarially trained model from Madry et al. (2018), which is not displayed as a point due to its significant outlier values for robustness. S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; (n FC) refers to the number of fully connected layers in the size-controlled models.

- The benefit of increasing the dimension (i.e. number) of low-level binding representations available. More information about the low-level features alongside the abstract, high-level features is generally associated with greater robustness.
- A benefit even without masking (i.e. up-projecting all low-level activations of a layer rather than selecting only a proportion). Thus hierarchical binding's benefit appears to primarily be from the preservation of low-level information alongside high-level features, rather than specifically encoding which low-level features causally drove the high-level representation. The result also provides further evidence against gradient masking as a main effect.
- The benefit of the selected binding representations having played a larger causal role in the representation at the higher level. If only a sub-proportion of low-level features are up-projected, it is generally better to up-project the most causally significant representations.

Fig. 11 shows that, as the proportion of largest gradients used is increased (using larger values of γ), the network shows a rapid rise in its robustness, presumably as the key causally important dimensions are captured. If γ is too small (e.g. $\gamma = 0.1$), one

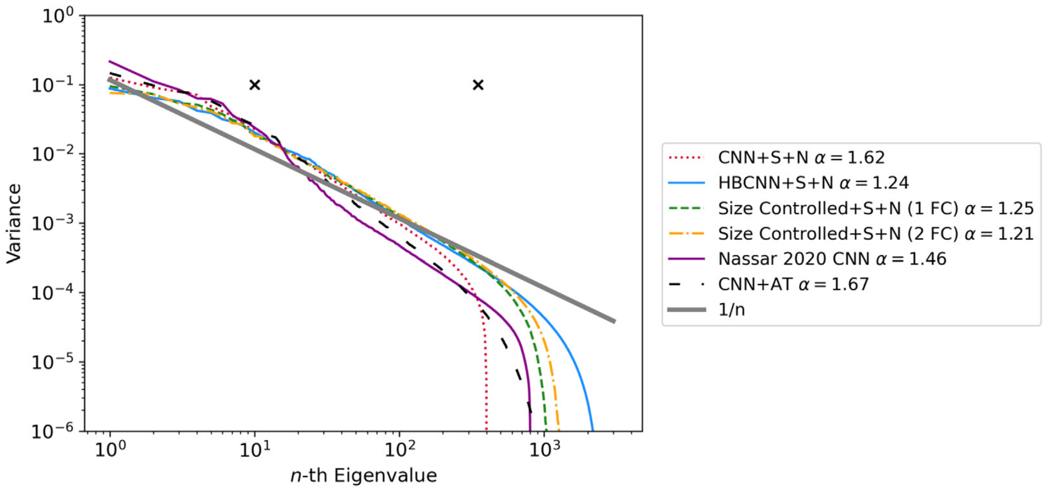


Fig. 9. *Sample eigenspectra for median models trained on MNIST.* The eigenspectra of sample models, where the model with the median (or nearest to median) α value for that architecture is selected for plotting. Also shown are the eigenspectra for the adversarially trained model (Madry et al., 2018), and a CNN with the same architecture and hyperparameters as the baseline model in Nassar et al. (2020), for comparison. The '×' symbols indicate the range of ranked eigenvalues used for estimating the α exponent (10, 350), where the total number of eigenvalues in the CNN+S+N model is 400. The plotted eigenvalues are normalized by the total variance. S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; (n FC) refers to the number of fully connected layers in the size-controlled models.

would expect that not all of the important low-level features are being included and up-projected, and as such there are fewer representations available for informing decision boundaries along the high-dimensional manifold. With an even smaller value ($\gamma = 0.01$), it appears that the representations may be too labile to be useful, and that they can in fact be harmful to robustness. Interestingly, there is a rapid rise as γ increases, but this benefit then largely plateaus. This result suggests that while the primate brain might implement hierarchical feature binding for a variety of computational purposes, we observe that the primary influence this has on robustness is from the preservation of low-level features along-side more abstract features, rather than the additional encoding of which low-level features causally drove more abstract representations. This finding is consistent with our proposed connection between hierarchical binding, manifold geometry, and robustness, where the preservation of low-level features alone should provide a more expressive neural manifold.

When γ reaches 1.0, we are simply up-projecting all the low-level features alongside those of max-pooling (i.e. without any masking). The benefit in spite of no masking procedure demonstrates that our main effect cannot be explained by our gradient-unpooling operation introducing discontinuities as has been found in high-sparsity regimes of k-Winner-Take-All (Xiao et al., 2020). While we noted above that a higher γ appears useful for several attacks, this is not universally the case, and in particular a higher γ drastically reduces the robustness of the classifier to the Boundary Attack by making the decision boundary more linear and amenable to attack. We tuned γ in our networks for robustness across the range of attacks leveraged, while capturing the notion of sparseness embodied in our opening, biological motivation. In the case of MNIST, results from the cross-validation data suggested that a value of 0.4 was optimal.

Fig. 11 also shows that using the same algorithm for generating the gradient unpooling representations, but with the γ-*smallest* gradients to derive the binding information instead, generally confers less benefit until virtually all activations are up-projected. This supports the notion that our implementation of gradient-unpooling is capturing some information about which low-level representations are most useful for supporting robust decision boundaries.

Note that we do not claim that the implementation we use is necessarily the best possible for preventing adversarial attacks, and machine-learning focused approaches might want to deviate from our sparse choice of γ. It may be for example that a network using the largest gradients with $\gamma = 1.0$, or a network using the smallest gradients and $\gamma = 0.9$ might be better for certain attacks, such as the Pointwise attack. Rather we aim to demonstrate that our biologically motivated process of up-projecting a sparse representation of the hierarchical binding features confers robustness, and that this generally performs better when the most important (by our approximate measure) features are up-projected, rather than the least important. The noisiness of the trends in Fig. 11 may partly relate to the ultimate failure of our gradient unpooling operation to perfectly capture which low-level neurons were causally important to the more abstract representation.

Note that we do not show results for the γ-smallest gradients on the L_∞ transfer attack, for which additional surrogates would be required, nor any results from the L_2 transfer attack, where the trend is more noisy, and interestingly, the unpooling layer (here ablated) appears to play an important role in robustness. Unpooling was ablated in these experiments to remove the confound of any sparsity in the up-projected representation, and the results suggest that *gradient* unpooling can be sufficient for preventing many of the vulnerable regions leveraged in attacks such as Gaussian and uniform noise. We comment again on the significance of this ablation in the next sub-section.

5.6.2. Hierarchical binding reduces the probability of finding vulnerable decision regions

To provide additional evidence that the decision boundary of the model augmented with hierarchical binding is fundamentally more challenging to attack in a black-box setting, i.e. that it is more difficult to identify vulnerable regions, we leverage a magnitude constrained attack with Gaussian noise. Unlike the additive Gaussian Noise Attack used in our main results which iteratively increases the amount of noise until misclassification is successful, we now add noise sampled from a fixed distribution ($\mu = 0$, $\sigma = 0.35$, i.e. Gaussian noise of greater magnitude than used during training). This attack is repeated up to 50,000 times for each image, and in Fig. 12, we show the proportion of images that have been misclassified as the number of attempts increases.

As we have noted, our method cannot guarantee the absence of vulnerable regions in the decision boundary, however we

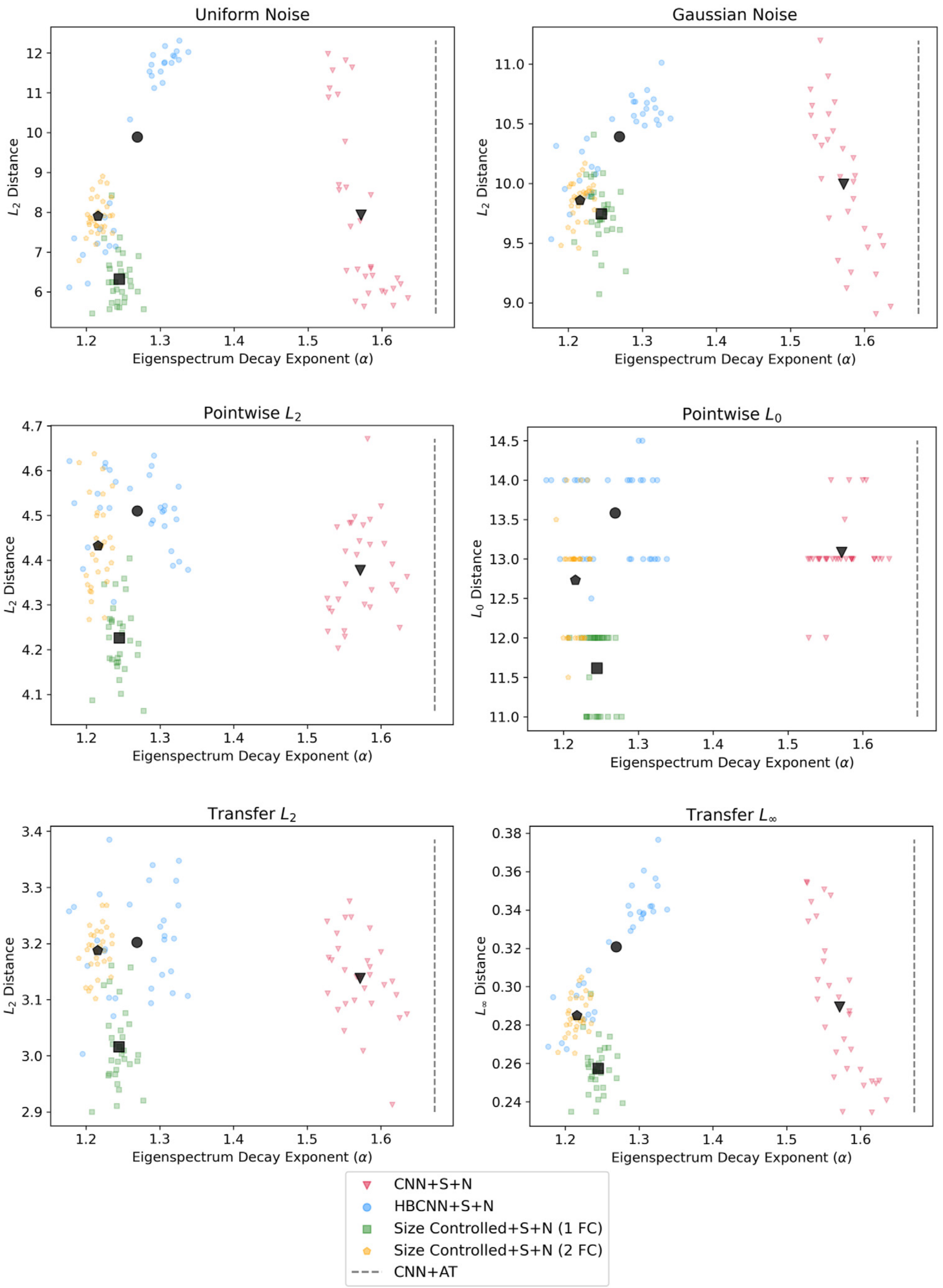


Fig. 10. *Robustness to black-box attacks vs. α exponent fit to eigenspectrum decay.* The robustness of individual models as a function of their fitted exponent (α) to the eigenspectrum, where Stringer et al. (2019) observed $\alpha \approx 1$. α is estimated for the layer of activity immediately proceeding the fully connected layers. The black markers indicate the mean value for a given architecture type. The dashed gray-line (far right) indicates the α value of the adversarially trained model from Madry et al. (2018), which is not displayed as a point due to its significant outlier values for robustness. S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; (n FC) refers to the number of fully connected layers in the size-controlled models.

would expect these regions to be sampled randomly with very low probability. Whether using a method such as a transfer attack, the Pointwise attack, or indeed brute force noise, one is less likely to arrive in these regions for the hierarchical binding model owing to its improved decision boundary. Consistent with this, we observe that it is much harder to find such regions through random sampling in the HBCNN+S+N than the standard CNN+S+N or size-controlled models (Fig. 12). It is interesting to note that, consistent with our earlier results, the size-controlled CNNs appear to be more vulnerable than the standard CNN+S+N. This finding supports that naively adding parameters, rather than striving for representations that reflect the manifold of the underlying data, can be harmful for robustness. It is once again noteworthy that this is particularly the case for the shallower size-controlled CNN with only one hidden layer, despite this being a common motif in models assessed (Nassar et al., 2020; Schott et al., 2019) or designed (Madry et al., 2018) for adversarial robustness.

Fig. 12 also shows the results of two ablations. During and after training, we either always set to 0 the activations of the representations conveying more invariant, abstract information (max-pooling and unpooling, the former after gradient unpooling has been performed), or more low-level information (gradient

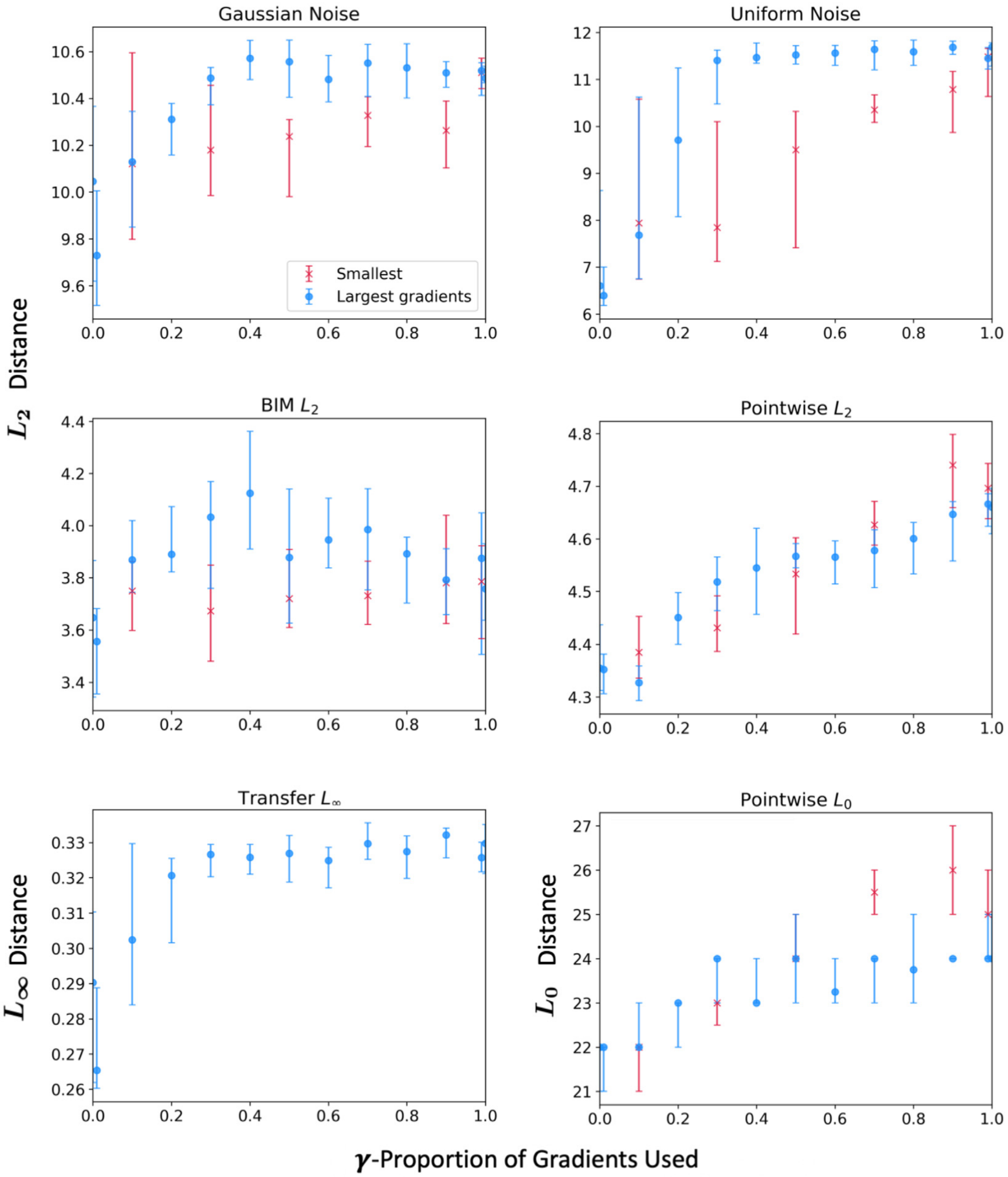


Fig. 11. *The effect of hierarchical binding dimension and causal role on robustness.* We systematically vary the γ-proportion of gradients used to mask the gradient unpooling representations along the x-axis; results are shown from using both the largest (blue) and smallest (red) γ gradients. 0.0 is equivalent to the standard CNN control model we study, while 1.0 means no masking is applied and all low-level activations are up-projected. The y-axis represents the adversarial perturbation, measured using the appropriate L_p norm for that attack. Each point represents the median distance of a successful adversary, provided as the median performance across 30 networks trained on MNIST without unpooling. Error bars show the 95% confidence interval of the median. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

unpooling). Gradient unpooling appears to provide the majority of the benefit for Gaussian noise leveraged in this manner, while unpooling alone offers little benefit alongside max-pooling. On the other hand, we noted from the ablated networks in the previous section that unpooling appears important for transfer L_2 robustness. We suspect that the directions which unpooling helps the network cover are exceedingly rare to sample randomly, and as such, we see limited benefits in Fig. 12; indeed we know from our main results that there are *many* vulnerable regions within the space that we sample in this sub-section, and yet clearly these are very difficult to find with brute-force noise.

Finally, the ablations in Fig. 12 show that gradient unpooling is itself not sufficient to explain the benefit for robustness. If it is not paired with the more abstract representations provided by unpooling and max-pooling, then the network performs worse than the standard CNN+S+N. Once again, this supports our opening motivation of combining these representations for a robust model, rather than gradient-unpooling e.g. introducing an unusual non-linearity that is the source of robustness.

5.7. Performance on other data-sets

To determine whether the observed robustness generalizes to a more complex setting, we apply the same LeNet-5 based architectures to FMNIST, albeit with the gradient-unpooling parameter $\gamma = 0.3$ rather than 0.4, and implement a VGG style HBCNN for the CIFAR-10 data-set with $\gamma = 0.1$ (model details in Appendix A). Broadly speaking, the model architectures and training protocol for FMNIST is the same as that used for MNIST,

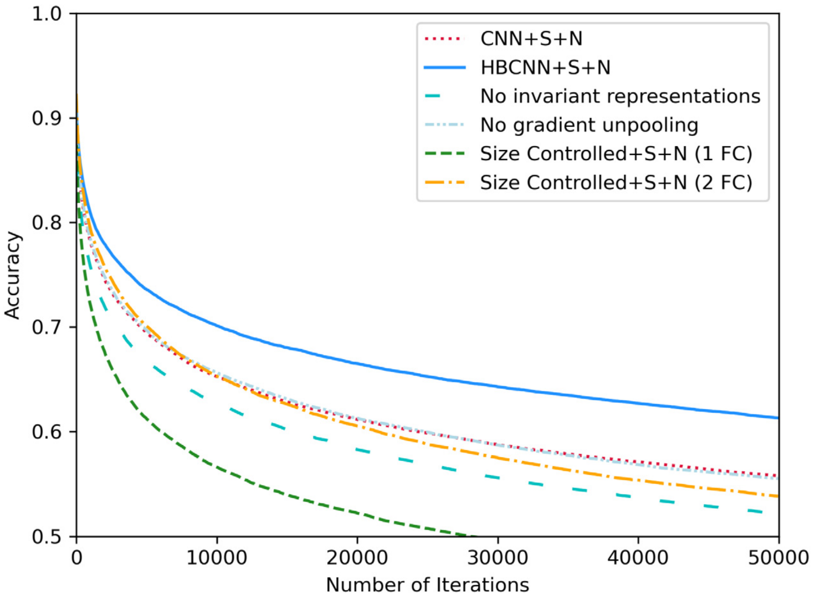


Fig. 12. *Leveraging repeated Gaussian noise of constant standard deviation.* We show the proportion of correctly classified images across 30 networks as a function of the number of attempted additions of Gaussian noise ($\mu = 0$, $\sigma = 0.35$) to cause misclassification. The 'No invariant representations' model is the HBCNN+S+N model trained and evaluated with the max-pooling and unpooling layers ablated. The 'No gradient unpooling' model is the HBCNN+S+N model trained and evaluated with the gradient-unpooling layer ablated. S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; (n FC) refers to the number of fully connected layers in the size-controlled models.

including the amount of Gaussian and salt-and-pepper noise. In order to preserve clean classification accuracy on CIFAR-10, our noise-augmented training regime uses Gaussian noise an order of magnitude smaller ($\sigma = 0.03$ vs 0.3), and no salt-and-pepper noise. With the exception of the 'vanilla' model described in the results table, we also use weight decay (Ng, 2004) in the VGG models to further regularize the decision boundaries. We note that the optimal weight-decay value was tuned for robustness for both the control and the binding-augmented models separately. Given our observation on MNIST that the size-controlled architectures generally resulted in greater vulnerability to adversarial attacks on meaningful metrics, we do not include any in these analyses.

Our results for FMNIST and CIFAR-10 are presented in Tables 3 and 4 respectively. While there are a few attacks where we fail to generalize the effect of the HBCNN being stronger than the robust control on FMNIST (Gaussian noise, Pointwise L_2, DeepFool attacks, All L_∞) and CIFAR-10 (Pointwise L_2, MIM, All L_0), the trend of enhanced robustness above the control model and across multiple attacks is observed. As such, it appears that some of the benefits of hierarchical binding can generalize to different architectures and data-sets, although the current approach does not result in as strong an effect as we observe in MNIST. Once again, we pair our tabular results with distributions in Fig. 6 of model performances across several key attacks.

Interestingly, the HBCNN (LeNet-5 variant) surpasses the clean accuracy of the control model on FMNIST (where the control is the 'CNN+S+N' model in Table 3). This is not a fair comparison, as the HBCNN has more parameters, but it supports the proposal that when binding representations are included, adversarial robustness becomes a question of on-manifold generalization error, making robustness compatible with clean classification accuracy (Stutz et al., 2019), rather than an orthogonal objective. Finally, we note that on CIFAR-10, while the observed improvement is broad-spectrum, it does not match the highly established method of adversarial training.

6. Discussion

We have demonstrated the implementation of a novel CNN architecture, inspired by recent work in theoretical neuroscience (Eguchi et al., 2018; Isbister et al., 2018). This architecture approximates hierarchical binding representations by capturing the causal relations between lower-level features and the higher-level representation of an object. Within the framework of adversarial examples as off-manifold perturbations, we presented empirical evidence of enhanced robustness to a variety of attack methods, following the introduction of these representations. We measured geometric properties of the representations in our HBCNN and standard architectures, observing a relationship between manifold extent and robustness, consistent with our opening premise. The additional analysis of a brute-force noise attack complemented our main results to further demonstrate that the observed robustness aligns with our geometric view of the defense. Further below, we discuss how hierarchical binding representations could help explain the apparent sensitivity of humans to adversarial examples under specific experimental conditions.

One interesting finding of our work is that, while hierarchical binding was implemented as a biologically-inspired defense, the results in Section 5.6.1 suggest that the primary observed benefit was due to the preservation of low-level features alongside more abstract, high-level features. As we discussed under our Methods, we designed our architecture to implement a simplified version of the hierarchical binding proposed in Eguchi et al. (2018); without such constraints, the model would face significant computational challenges owing to the number of features. It may be that future architectures can more faithfully capture this proposed form of hierarchical binding. Under such a setting, the precise preservation of hierarchical binding relations might provide additional benefits to robustness beyond the preservation of low-level information achieved by the currently presented model.

A key strength of the proposed approach is the broad range of attack types against which adversarial robustness is enhanced, particularly to black-box attacks across L_0, L_∞ and L_2 metrics. Many adversarial defenses result in a trade-off in robustness on

Table 3
Fashion-MNIST results.

	Vanilla-CNN	CNN+S+N	HBCNN+S+N	CNN+AT
Clean accuracy	90.87%	87.38%	**88.26%**	88.46%
L_2-metric ($\epsilon = 1.5$)				
Transfer ■	1.7 (55%)	3.7 (78%)	**4.2** (81%)	3.1 (80%)
Uniform Noise ■	3.4 (78%)	5.1 (87%)	**5.6** (87%)	3.9 (89%)
Gaussian Noise ■	2.9 (76%)	**8.4** (86%)	8.1 (86%)	3.9 (87%)
Boundary ■	0.4 (7%)	1.6 (52%)	**4.5** (84%)	0.9 (23%)
Pointwise ■	2.3 (70%)	**4.0** (83%)	**4.0** (82%)	1.7 (59%)
FGM	1.2 (45%)	2.5 (62%)	**2.9** (65%)	4.5 (72%)
FGM w/GE	1.2 (44%)	2.5 (62%)	**4.5** (76%)	4.8 (72%)
DeepFool	0.4 (12%)	**1.6** (52%)	**1.6** (52%)	2.3 (66%)
DeepFool w/GE	0.5 (1%)	**2.1** (58%)	1.9 (56%)	2.3 (65%)
BIM	0.3 (1%)	1.0 (36%)	**1.1** (39%)	1.8 (58%)
BIM w/GE	0.3 (8%)	1.0 (35%)	**1.3** (46%)	1.8 (57%)
All L_2	0.3 (0%)	1.0 (32%)	**1.1** (38%)	0.9 (21%)
L_∞-metric ($\epsilon = 0.1$)				
Transfer ■	0.09 (45%)	0.22 (77%)	**0.23** (78%)	0.18 (87%)
FGSM	0.04 (24%)	0.13 (57%)	**0.15** (60%)	0.24 (80%)
FGSM w/GE	0.05 (30%)	0.13 (58)	**0.16** (62%)	0.32 (80%)
DeepFool	0.03 (2%)	**0.13** (58%)	0.12 (56%)	0.26 (79%)
DeepFool w/GE	0.03 (2%)	**0.16** (62%)	0.13 (57%)	0.27 (79%)
BIM	0.02 (1%)	0.07 (37%)	**0.08** (41%)	0.15 (76%)
BIM w/GE	0.02 (9%)	0.07 (36%)	**0.08** (42%)	0.16 (76%)
MIM	0.02 (1%)	0.07 (38%)	**0.08** (41%)	0.15 (76%)
MIM w/GE	0.02 (8%)	0.08 (38%)	**0.09** (46%)	0.16 (76%)
All L_∞	0.02 (0%)	**0.07** (35%)	**0.07** (39%)	0.15 (76%)
L_0-metric ($\epsilon = 12$)				
Pointwise ■	8 (32%)	24 (69%)	**26** (68%)	4 (17%)
Salt&Pepper Noise ■	23 (63%)	**167** (85%)	133 (84%)	16 (54%)
All L_0	8 (32%)	24 (69%)	**26** (68%)	4 (17%)

Results are presented from leveraging a variety of attacks against architectures of interest evaluated on the Fashion-MNIST data-set. The main numerical results shown here are the median L_p distances of a successful adversary for the different attacks (rows), provided as the median performance across 30 networks for each model condition (columns). In parentheses is the median accuracy, at a given thresholded perturbation ϵ, taken across 30 networks. The All-L_0, All-L_2 and All-L_∞ distances show the minimal adversarial distance across all attacks of that distance-metric for each image. Bold indicates the best performance between the networks with noise added to the training images; blue indicates the best performance across all networks in both tables. 'Clean' refers to the accuracy on the test data-set without adversarial perturbations; 'Vanilla' indicates the absence of the modifications (regularization with label smoothing or noise in training data) introduced later; HBCNN = Hierarchical Binding-CNN (LeNet-5 variant); AT = adversarial training; S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; w/GE = with Gradient Estimation; ■ = black-box attack.

one metric (such as to L_∞ attacks) for either minimal robustness to attacks that minimize another distance metric, or even enhanced vulnerability to them (Laidlaw et al., 2021; Schott et al., 2019). Note for example, the enhanced vulnerability to L_0 attacks seen in the adversarially trained model in Table 2. Particularly in the MNIST setting, our architecture appears to defy this typical trade-off, except for low-risk attack settings that require large changes to the input, such as brute-force salt-and-pepper noise. The method also maintains much of its accuracy on clean, unmodified images, with these benefits coming at relative computational and parameter efficiency compared to other leading defenses on MNIST.

Nevertheless, we must highlight some notable limitations of this work. The proposed architecture adds a hyperparameter (the γ-proportion) that for optimal performance requires tuning, with a general trade-off for different attacks. We found that a more complex data-set appears to benefit from a smaller proportion (MNIST $\gamma = 0.4$, FMNIST $\gamma = 0.3$, CIFAR-10 $\gamma = 0.1$). Furthermore, in the setting of FMNIST and CIFAR-10, the effect size vs. the control was more subtle, and for CIFAR-10, the robustness was not comparable to adversarial training. Finally, the adversarial examples that fool the HBCNN do not appear, on average, to be more meaningful than for the robust control model (Appendix C), and as we discuss below, the model remains vulnerable to a sufficiently capable attacker. Notwithstanding these limitations, our analysis indicates that the hierarchical representations described herein could be an important feature to developing fundamentally robust models. We discuss this in further detail below.

6.1. Models that are difficult to fool, vs. models that are impossible to fool

We have noted that our proposed architecture does not guarantee the absence of vulnerable decision regions, and that a sufficiently powerful attacker may be able to find these. Consistent with this, we have conducted exploratory experiments, requiring re-implementing the architecture in a different deep-learning library (PyTorch), and version of the adversarial attack library (Foolbox 3) in order to leverage newer attacks. While these are provisional results (Foolbox 3 required untested modifications to accept a model such as ours that uses gradients at inference), they suggest that newer attacks such as the Brendel (Brendel et al., 2019) and DDN (Rony et al., 2019) attack may be able to find adversarial sub-spaces that are shared for a standard CNN+S+N and a HBCNN+S+N model, resulting in the same apparent vulnerability given a sufficiently powerful attacker.

This suggests that, given sufficient optimization, the defense can be broken; it may be harder to find vulnerable regions by chance or when leveraging directions effective against other networks, but in such a high-dimensional space, there will still be many pockets that can be found by a sufficiently powerful

Table 4
CIFAR-10 results.

	Vanilla-VGG	VGG+S+N	HBCNN+S+N	ResNet+AT
Clean accuracy	86.51%	**86.43%**	86.07%	87.25%
***L_2*-metric** ($\epsilon = 1.5$)				
Transfer ■	0.68 (26%)	0.99 (36%)	**1.03** (38%)	6.23 (80%)
Uniform Noise ■	3.18 (80%)	5.16 (85%)	**5.40** (85%)	7.64 (86%)
Gaussian Noise ■	2.31 (69%)	3.89 (80%)	**4.12** (79%)	7.50 (82%)
Boundary ■	0.27 (3%)	0.42 (5%)	**4.81** (84%)	1.11 (36%)
Pointwise ■	1.60 (53%)	**1.91** (60%)	1.88 (60%)	2.11 (62%)
FGM	0.26 (21%)	0.39 (26%)	**0.47** (30%)	1.06 (42%)
DeepFool	0.17 (0%)	0.44 (14%)	**0.49** (14%)	0.97 (35%)
BIM	0.14 (1%)	0.22 (1%)	**0.23** (2%)	0.66 (21%)
All *L_2*	0.14 (0%)	0.22 (0%)	**0.23** (1%)	0.66 (20%)
***L_∞*-metric** ($\epsilon = 8/255$)				
Transfer ■	0.19 (34%)	0.030 (48%)	**0.031** (50%)	0.153 (82%)
FGSM	0.07 (17%)	0.010 (26%)	**0.012** (30%)	0.037 (54%)
DeepFool	0.005 (1%)	0.013 (28%)	**0.015** (28%)	0.039 (56%)
BIM	0.004 (1%)	0.006 (4%)	**0.007** (5%)	0.028 (46%)
MIM	0.004 (1%)	**0.007** (4%)	**0.007** (6%)	0.029 (47%)
All *L_∞*	0.004 (0%)	0.006 (3%)	**0.007** (5%)	0.028 (46%)
***L_0*-metric** ($\epsilon = 12$)				
Pointwise ■	7 (35%)	9 (42%)	**10** (43%)	11 (45%)
Salt&Pepper Noise ■	15 (53%)	**21** (57%)	**21** (57%)	27 (57%)
All *L_0*	7 (34%)	**9** (42%)	**9** (43%)	11 (45%)

Results are presented from leveraging a variety of attacks against architectures of interest evaluated on the CIFAR-10 data-set. The main numerical results shown here are the median L_p distances of a successful adversary for the different attacks (rows), provided as the median performance across 30 networks for each model condition (columns). In parentheses is the median accuracy, at a given thresholded perturbation ϵ, taken across 30 networks. The All-L_0, All-L_2 and All-L_∞ distances show the minimal adversarial distance across all attacks of that distance-metric for each image. Bold indicates the best performance between the networks with noise added to the training images; blue indicates the best performance across all networks in both tables. 'Clean' refers to the accuracy on the test data-set without adversarial perturbations; 'Vanilla' indicates the absence of the modifications (regularization with label smoothing or noise in training data) introduced later; HBCNN = Hierarchical Binding-CNN (VGG variant); AT = adversarial training; S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; w/GE = with Gradient Estimation; ■ = black-box attack.

attack. This is consistent with the results of Gilmer et al. (2018). On a synthetic data-set of 500-dimensional spheres, even with enormous training data and sufficient expressive capacity to perfectly separate the class manifolds, a model will learn a decision boundary that significantly deviates from the ground-truth decision boundary, and is therefore vulnerable to adversarial attacks. The intuition is that if the dimensionality is sufficiently high, a model can be 'statistically' perfect (in their case 100% accuracy on 20,000,000 test examples), even if there are *many regions* where the decision boundary extends into the wrong manifold.

In short, we have presented our architecture so as to address an issue of expressive power for learning an appropriate decision boundary, but it cannot fully resolve the inherent challenges of learning these decision boundaries in high-dimensional spaces. As such, low-probability vulnerable regions clearly still exist, which sufficiently powerful attacks are capable of finding. This raises the question of what additional steps could address this persistent challenge. It is possible that unsupervised pre-training (Chen et al., 2020; Hénaff et al., 2020) would be useful, and indeed techniques based on self-supervised learning have recently been leveraged with success on improving adversarial robustness (Gowal et al., 2021).

As is clear from the results of Gilmer et al. (2018) however, more data alone is unlikely to be sufficient. A complementary approach would be to address how the model generalizes to unseen data. Typical neural networks have unfavorable properties in this regard, such as forming polytopes that classify with high probability regions extending into infinity (Hein et al., 2019). The favorable geometric behaviors of methods such as k-Nearest Neighbor have been noted before as being potentially useful, although these often come at a severe cost in expressive power (Khoury & Hadfield-Menell, 2018; Schott et al., 2019). Architectures designed to enable robust invariance, such as Capsule Networks (Sabour et al., 2017), may be important in combination with the modifications we have presented in this work.

6.2. Other methods that preserve low-level information

It is worth clarifying the relationship of our work to other methods of preserving low-level information. The importance of hierarchical binding for a more robust decision boundary is that the low-level features are represented explicitly in the late stages of the network, with their own activations and thereby unique learned weights projecting to the decision part of the network. Furthermore, this preservation of information is constructed so as to jointly capture a spectrum of feature abstraction. Thus the architecture is designed to avoid more entangled representations of detailed and abstract features as would be found in ResNets (He et al., 2016) and related architectures that can preserve all information about the input, such as i-RevNet (Jacobsen et al., 2018).

In particular, we emphasize that even with $\gamma = 1.0$, our network does not equate to current CNNs with skip connections. In networks such as ResNets, the cross-wise dimension (essentially the spatial resolution of the representation) is progressively decreased throughout the layers of the network through methods such as using convolutions and skip connections with a stride of 2. This results in transforming from a feature map of dimension such as 56 × 56 to 7 × 7 (He et al., 2016). In our network, the activity is concatenated alongside the representations in the higher layer, rather than added via a skip connection, and as such, the low-level activations maintain their cross-wise dimension/resolution. Furthermore, these up-projected representations have their own, unique learned weights projecting to the final layers of the network. Similarly, in DenseNets, max-pooling between blocks progressively decreases the feature-map dimension, resulting in lost spatial resolution. Within a block, the channel-wise concatenation of features does not capture the same motif as hierarchical binding of combining low-resolution, abstract features alongside higher resolution, low-level features, as the feature-map dimensions are constant (Huang et al., 2017). In summary, our architecture represents a novel method of combining low-level and abstract features, even when $\gamma = 1.0$.

Returning to our initial proposal and Fig. 1, the simpler read-out of the features that motivates our architecture is important for enabling sufficient representational capacity, as well as amenable features, so as to efficiently learn decision boundaries along all dimensions that constitute class-preserving transformations of the object. The results from our size-controlled CNNs support the importance of this inductive bias built into our architecture — i.e. that low and high-level representations should be represented separately while being conserved up to the decision-level of the network.

In summary, the desire is not to preserve all information in the image through an entangled representation (which deep-invertible networks do perfectly (Jacobsen et al., 2018)), but to promote a robust decision boundary by providing a classifier distinct representations of both the abstract and low-level features of an object. This is thus distinct from the general advantages of skip connections, which can provide benefits such as supporting learning in deeper architectures (He et al., 2016).

6.3. Adversarial examples and human perception

Is it possible to connect our biologically-motivated model back to the experimental literature on the robustness of primate vision? Perceptually, adversarial examples in humans might manifest as misclassifying a briefly presented object (Elsayed et al., 2018), or guessing at above chance levels what the adversarial perturbation will induce in machines (Zhou & Firestone, 2019). Such effects might be achieved by either randomly adding noise in the hope of finding an adversarial example, or leveraging transfer attacks from machine learning models (Elsayed et al., 2018).

Humans without time-constraints do not appear to be sensitive to adversarial perturbations (note e.g. recent limitations identified by Dujmović et al. (2020) regarding Zhou and Firestone (2019)), and as such the only experimental evidence for human sensitivity to adversarial examples is from Elsayed et al. (2018). They used transfer attacks created from an ensemble of CNNs to determine whether these would impact human accuracy in a two alternative forced choice task (e.g. cat vs. dog). While the effect on classification was not comparable to the dramatic shift seen in machine vision systems, they measured statistically significant drops in accuracy when humans were constrained to view adversarial images for a very short duration (around 60-70 ms) followed by masking intended to limit recurrent and top-down processing.

The CORNet-S architecture is a CNN variant that makes use of recurrent activity to better match the primate ventral stream (Kubilius et al., 2019). Together with additional modifications in the form of a front-end that better matches the processing seen in V1, Dapello et al. (2020) demonstrated its enhanced robustness above a CNN without recurrence. Without this modified front-end however, CORNet-S appears to be more vulnerable than a typical CNN without recurrence, such as AlexNet, to white-box adversarial examples, making it unclear what the specific contribution of recurrence to robustness might be. In addition, the evaluation in Dapello et al. (2020) did not include transfer attacks, the method used in Elsayed et al. (2018). The exclusion of transfer attacks may have been based on the challenges of creating an appropriate surrogate, however it remains an open question to what extent recurrence in of itself can account for the experimental results in the latter study. Similarly, in Huang et al. (2020), the value of recurrent feedback for robustness to adversarial examples was demonstrated, although the evaluation of black-box attack methods such as transfer attacks was limited to versions of their model that had undergone adversarial training. As noted in our introduction, it has been predicted that top-down and lateral activity in a spiking neural network would be needed to implement the proposed hierarchical binding algorithm in a biological system, and so the disruption of such processing provides a possible basis for the observed effect in Elsayed et al. (2018).

More generally, there have been several valuable contributions to explain human robustness to adversarial examples, but we feel these still leave important questions unanswered. Vuyyuru et al. (2020) demonstrated the effect on robustness of non-uniform retinal sampling and varying receptive field sizes with eccentricity, but their effect was specific to small perturbations, and the evaluation of black-box attacks did not assess whether transfer attack robustness was greater for the proposed model vs. an undefended one. Tadros et al. (2020) examined the effect of sleep-like algorithms on robustness, and again while this might account for some of the difference between humans and artificial systems, their method actually increased the vulnerability of the model on MNIST to the Boundary Attack (see their Table 1), with no evaluation of other black-box methods such as transfer attacks or the Pointwise attack. Nassar et al. (2020) examined whether regularizing representations to follow the $1/n$ power-law observed in Stringer et al. (2019) could explain the benefits of biological neurons displaying such characteristics. While their results suggested that this might indeed partly explain robustness in biological vision, our own results suggest that this also has limited explanatory power across diverse architectures. A similar approach proceeding Nassar et al. (2020) was taken in Li et al. (2019), where the representations of a ResNet were encouraged to align with responses from the mouse visual system. This promising work demonstrated robustness to both noise and a variety of L_p metrics, although it remained unclear what the specific mechanism was for this robustness. In summary, we believe our work helps address an explanatory gap regarding human robustness to adversarial examples, in particular to transfer attacks — the one method that has been leveraged against humans (Elsayed et al., 2018), and a threat setting where our method consistently enhances robustness. We emphasize that the absence of transfer attacks from some of the above studies does not imply incomplete evaluations, but rather highlights a unique contribution of our work in making a direct connection between a biologically inspired defense and adversarial sensitivity vs. robustness in humans.

An additional relevance of our results to the human perception of adversarial examples is the question of a robustness vs. accuracy trade-off. Some theoretical results have suggested that adversarial robustness and accuracy on unmodified, clean datasets are mutually incompatible objectives and that robustness necessarily comes at a cost to clean accuracy (Tsipras et al., 2019). This notion, however, is counter-intuitive given the observation that humans are both accurate generally, and robust to adversarial examples. We noted that our method results in either a limited drop (or in the case of Fashion-MNIST improvement) in clean classification accuracy, suggesting that these objectives are not in fact mutually exclusive. More specifically, the modest drop in clean accuracy we see on MNIST is less than for other robust models, yet our model uses either fewer parameters (cf Madry et al. (2018)), or less compute time (cf Schott et al. (2019)) than these. We have argued that when hierarchical binding captures additional dimensions that describe the true, underlying manifold of the object, a robust decision boundary becomes a case of on-manifold generalization (Gilmer et al., 2018; Stutz et al., 2019), and rapid classification can proceed as normal. In summary, our approach combines robustness (at least to attacks that can be

Table A.5
Number of trainable parameters in the primary models.

CNN (LeNet-5 variant)	# of parameters	# input/output
1st Conv.	156	1/6 c
2nd Conv.	2,416	6/16 c
1st FC	102,656	400/256 u
2nd FC	32,896	256/128 u
Output	1,290	128/10 u
Total	**139,414**	
HBCNN (LeNet-5 variant)		
1st Conv.	156	1/6 c
2nd Conv.	2,416	6/16 c
1st FC	813,312	400+1600+1176/256 u
2nd FC	32,896	256/128 u
Output	1,290	128/10 u
Total	**850,070**	
Size-controlled CNN (1 FC)		
1st Conv.	832	1/32 c
2nd Conv.	51,264	32/64 c
1st FC	819,712	1600/512 u
Output	5,632	512/10 u
Total	**877,440**	
Size-controlled CNN (2 FC)		
1st Conv.	1,404	1/54 c
2nd Conv.	145,908	54/108 c
1st FC	691,456	2700/256 u
2nd FC	32,896	256/128 u
Output	1,290	128/10 u
Total	**872,954**	
CNN+AT from Madry et al. (2018)		
1st Conv.	832	1/32 c
2nd Conv.	51,264	32/64 c
1st FC	3,212,288	3136/1024 u
Output	11,264	1024/10 u
Total	**3,275,648**	

Summary of the number of parameters that can be optimized during learning for the primary models leveraged on MNIST. Abbreviations: Conv. = convolution; FC = fully connected; c = channels; u = units; AT = adversarial training.

plausibly leveraged against humans), efficiency (both parameter and compute-wise) and persistent clean accuracy. We believe this provides additional evidence that the basis of adversarial examples and robustness that we explore is well aligned with what distinguishes vision in artificial systems and primates.

7. Conclusion

The work presented in this paper provides evidence for the hypothesis that the preservation of low and high-level visual features at different spatial scales is important for robust object recognition, whether through hierarchical feature binding in the primate brain or in artificial systems. While our strongest results are on MNIST, robustness on this simplest of image data-sets remains far from solved (Mu & Gilmer, 2019; Nassar et al., 2020; Schott et al., 2019; Tramèr, Behrmann et al., 2020). Future systems that better preserve hierarchical binding information and make use of it may be important to generalizing the observed robustness to more complex settings, such as the ImageNet (Russakovsky et al., 2015) data-set of natural images.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Simon Stringer is a shareholder in Applied AGI Limited, a company that plans to develop smart machine vision applications using a variety of convolutional neural network and spiking neural network architectures.

Acknowledgments

This work was supported by the Biotechnology and Biological Sciences Research Council (BBSRC), United Kingdom, grant number BB/M011224/1. We thank Nasir Ahmad, Andrew Saxe, James Isbister, Hannah Dury, Richard Lonsdale, and Basile Confavreux for helpful discussions and comments.

Appendix A. Additional model details

All models are implemented in TensorFlow 1.14 (Abadi et al., 2016), and used the Rectified Linear Unit (ReLU) activation function (Nair & Hinton, 2010). At the time of publication, code for creating our models and sample model check-points will be made available at https://github.com/nielsleadholm/CNN_Binding_TensorFlow

The LeNet-5 model used as the vanilla and label smoothing control for MNIST consists of two convolution (6 and 16 channels, kernel size (5, 5)) and two max-pooling layers, followed by two fully connected layers of size 120 and 84. This baseline model was initially chosen as the starting point for our HBCNN, as the limited number of channels and use of two (rather than one) fully connected layers would help limit the increase in parameter numbers as binding was introduced. When we introduce noise in training, we use two fully connected layers of dimension 256 and 128, as this offered better performance for both the LeNet-5 and HBCNN (LeNet-5 variant), consistent with this representing a more complex data-set to fit. For the HBCNN (LeNet-5 variant), the unpooling layer is applied to the last max-pooling layer, while

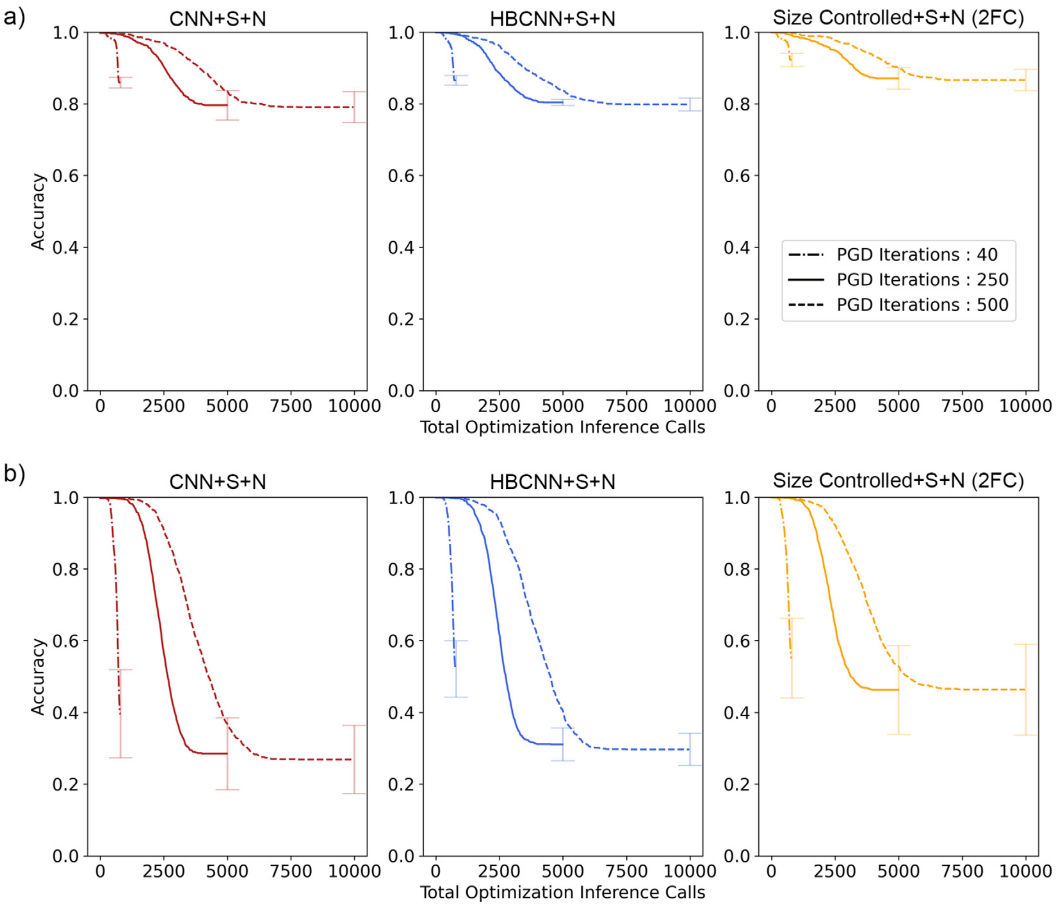


Fig. B.13. *Convergence of PGD attacks nested in binary search over hyperparameters.* The y-axis shows the accuracy of the model as the total number of inference calls increases, where accuracy is provided at a threshold of $\epsilon = 1.5$ for the L_2 PGD attack (a), and $\epsilon = 0.3$ for the L_∞ PGD attack (b). Each PGD attack is associated with 20 random starts, of which the best result for each image is used. For each random start, a binary search with $k = 20$ steps is performed to optimize the step-size and ϵ value of the PGD attack, minimizing the final perturbation of a successful adversary. Each binary search iteration is associated with a budget of PGD iterations ($n = 40$, 250, or 500). Thus, excluding the initial model queries to establish a starting point for the binary search, there are a total of $n \times k$ possible inference calls for a given optimization process. We show the convergence of the optimization process given different PGD budgets, where each setting has the flexibility to tune the hyperparameters of the attack. While increasing the possible number of PGD iterations from a typical value of 40 to 250 provides a marked benefit, doubling this to 500 does not. We therefore choose a value of 250, which also enables us to run the PGD attack across many seeds without a prohibitive computational cost. The error bars represent the 95% confidence interval across 5 random seeds.

the gradient-unpooling sub-layer is between the last max-pooling layer and the pre-convolution activations proceeding it (Fig. 3a). For FMNIST, we used the same architectures as for MNIST with the larger fully connected layers.

We also evaluate two non-binding CNNs for MNIST with an equivalent number of free parameters to our HBCNN (LeNet-5 variant). The first of these has one fully connected layer, as this is a common architectural choice in adversarial example research, such as the robust model in Madry et al. (2018), and the baseline models in Schott et al. (2019) and Nassar et al. (2020). This size-controlled model has two convolutions with channel sizes 32 and 64, and a fully connected layer of dimension 512, for a combined 877,440 parameters vs 850,070 in the HBCNN (LeNet-5 variant) — see Table A.5. The second size-controlled model has two fully connected layers, as in the case of the LeNet-5 and HBCNN (LeNet-5 variant) models. It has two convolutions with channel sizes 54 and 108, and fully connected layers of dimension 256 and 128, for a combined 872,954 parameters.

The two other architectures included in our MNIST results are the base-line CNN used in Nassar et al. (2020), and the adversarially trained CNN from Madry et al. (2018). The former consists of two convolution (16 and 32 channels, kernel size (3, 3)) and two max-pooling layers, followed by a single fully connected layer of dimension 1000. When training this network, we use the same hyperparameters as outlined in Nassar et al. (2020). The adversarially trained CNN consists of two convolution (32 and 64 channels, kernel size (5, 5)) and two max-pooling layers, followed by a single fully connected layer of dimension 1024. Unlike the main CNNs leveraged in MNIST, note that this architecture uses ‘same’ padding for both convolutions, resulting in a larger (7 × 7 rather than 5 × 5) cross-section when feeding into the fully connected layer. This contributes to the much larger number of parameters (Table A.5).

For the VGG architectures leveraged on CIFAR-10 (Fig. 3b), we used three blocks, each containing two convolutions and one max-pooling (channel dimensions 32, 32, 64, 64, 128, 128, kernel sizes (3, 3) throughout), followed by two fully connected layers of dimension 120 and 84, so as to once again constrain the growth in free parameters following the introduction of binding. For the HBCNN (VGG variant) we used two un-pooling sub-layers (corresponding to the 2nd and 3rd blocks), and two gradient unpooling sub-layers (from the 3rd max-pooling layer to the 1st and 2nd max-pooling layers respectively) (Fig. 3b).

All of our models were regularized with dropout (Srivastava et al., 2014) of 0.25 and, where used, label smoothing of 0.1. Training of a network's parameters was performed using the Adam optimizer (Kingma & Ba, 2015) with a learning rate of 0.001 (MNIST and FMNIST) and 0.0005 (CIFAR-10), and a batch-size of

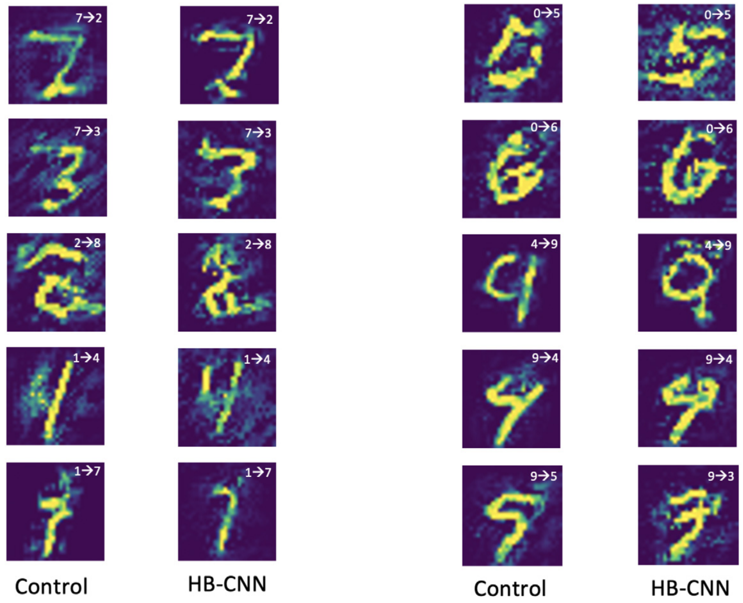


Fig. C.14. *Cherry-picked, semantically meaningful adversaries.* The annotation shows the original class followed by the prediction of the network following the adversarial perturbation. Shown are adversaries for the control (LeNet-5) and HBCNN (LeNet-5 variant) networks with label smoothing and noisy training data.

128. Training was performed for 45 epochs (vanilla MNIST model, and LeNet-based models with smoothing), 90 epochs (all other LeNet-based models, except for the 2-fully connected layers, size-controlled CNN, which showed considerable improvement on cross-validation data when trained for 180 epochs), and 500 epochs (all VGG models). For VGG models, we also augmented the training data with random shifts and horizontal flipping of the image.

For VGG models with smoothing and noise in the training data, we used L_2 regularization of 10^{-3} (standard) or 10^{-5} (HBCNN) for the weights from the final max-pooled layer, and 10^{-3} for the unpooling and gradient unpooling representations.

Our evaluation includes adversarially trained models for MNIST and CIFAR-10 loaded from the MadryLab Challenge repositories (https://github.com/MadryLab/mnist_challenge) and (https://github.com/MadryLab/cifar10_challenge), the latter of which is based on a Wide ResNet architecture (Zagoruyko & Komodakis, 2016). These were generated by the original authors of Madry et al. (2018), and use the adversarial training with the Projected Gradient Descent (PGD) attack described therein. An adversarially trained model for FMNIST was loaded from the repository from Croce et al. (2020) (https://github.com/max-andr/provable-robustness-max-linear-regions); these were trained using PGD attacks (40 iterations), with 50% adversarial images, and 50% clean images in each batch, for 100 epochs.

Appendix B. Hyperparameters for adversarial attacks

For PGD we used 20 random starting points for the adversary, followed by 250 iterations with an initial step-size of 0.01, selecting the best adversary of the 20 for each image. For BIM we used 10 iterations with an initial step-size of 0.05. MIM was applied with 10 iterations, an initial step-size of 0.06, and a decay factor of 1.0. For PGD, BIM, and MIM, the step-size and epsilon were automatically adapted in Foolbox using a binary search. For MNIST, we combine 10 runs of the Pointwise L-0 attack, as performed in Schott et al. (2019), taking the minimal adversary. For the Boundary attack we used 1000 iterations, a step-adaptation size of 1.5, and initial adversaries generated with the Blended Uniform Noise attack; for the Boundary attack we also evaluated the performance against one of the HBCNN (LeNet-5 variant) networks using 25,000, 100,000, and 1,000,000 iterations (tuning the step adaptation size for each), but this did not result in any notable improvement in its performance, and it remained uncompetitive against the HBCNN model in comparison to other attacks.

For Fashion-MNIST and CIFAR-10, surrogate models for creating transfer attacks were based on the respective standard and HBCNN architectures with label smoothing, but not noise, with the intent that these would be easier for the gradient-based methods to attack and generate adversaries from than the fully augmented architecture. Thus for these data-sets, each transfer attack was based on two surrogate models. For MNIST, the surrogate models for all attacks were based on : standard CNN+S, HBCNN+S, standard CNN+S+N, HBCNN+S+N, and CNN+AT models where two of each of these formed surrogates, with the exception of CNN+AT where only one was used, for a total of 9 surrogates. The two exceptions on MNIST to this was (a) when targeting the vanilla CNN and HBCNN models, where vanilla CNN and HBCNN models replaced the standard CNN+S and HBCNN+S models and (b) when targeting the ablated models for Fig. 11, where any surrogates with the hierarchical binding architecture also had unpooling removed.

Appendix C. Cherry picked examples

From among 300 MNIST adversarial examples for the 30 models of the standard CNN and HBCNN (LeNet-5 variant) networks with label smoothing and noisy training data, we selected what we thought to be (in our potentially biased view) the 10 most semantically convincing adversaries generated by the BIM L_2 attack for each model (Fig. C.14). We feel there are convincing examples for both models, with no obvious systematic difference.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., ... Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX symposium on operating systems design and implementation*.

Athalye, A., Carlini, N., & Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *35th international conference on machine learning*. MIT.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), http://dx.doi.org/10.1109/TPAMI.2016.2644615.

Baker, N., Lu, H., Erlikhman, G., & Kellman, P. J. (2018). Deep convolutional networks do not classify based on global object shape. *PLoS Computational Biology*, http://dx.doi.org/10.1371/journal.pcbi.1006613.

Bear, D. M., Fan, C., Mrowca, D., Li, Y., Alter, S., Nayebi, A., Schwartz, J., Fei-Fei, L., Wu, J., Tenenbaum, J. B., & Yamins, D. L. (2020). Learning physical graph representations from visual scenes. In *Advances in neural information processing systems, Vol. 2020-December*.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, http://dx.doi.org/10.1109/TPAMI.2013.50.

Brendel, W., & Bethge, M. (2019). Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet. In *International conference on learning representations*.

Brendel, W., Rauber, J., & Bethge, M. (2018). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *6th international conference on learning representations*.

Brendel, W., Rauber, J., Kümmerer, M., Ustyuzhaninov, I., & Bethge, M. (2019). Accurate, reliable and fast robustness evaluation. In *Advances in neural information processing systems, Vol. 32*.

Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., & Lerchner, A. (2019). MONet: Unsupervised scene decomposition and representation. arXiv preprint arXiv:1901.11390.

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., & Kurakin, A. (2019). On evaluating adversarial robustness. [ISSN: 23318422].

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th international conference on machine learning*. URL http://arxiv.org/abs/2002.05709.

Chen, T., Zhang, Z., Liu, S., Chang, S., Wang, Z., & Ima, T. (2021). Robust overfitting may be mitigated by properly learned smoothing. In *ICLR*.

Cheung, B., Livezey, J. A., Bansal, A. K., & Olshausen, B. A. (2015). Discovering hidden factors of variation in deep networks. In *3rd international conference on learning representations*.

Chung, S., Lee, D. D., & Sompolinsky, H. (2018). Classification and geometry of general perceptual manifolds. *Physical Review X*, *8*(3), http://dx.doi.org/10.1103/PhysRevX.8.031003.

Cohen, U., Chung, S. Y., Lee, D. D., & Sompolinsky, H. (2020). Separability and geometry of object manifolds in deep neural networks. *Nature Communications*, *11*(1), http://dx.doi.org/10.1038/s41467-020-14578-5.

Croce, F., Andriushchenko, M., & Hein, M. (2020). Provable robustness of relu networks via maximization of linear regions. In *AISTATS 2019 - 22nd international conference on artificial intelligence and statistics*.

Croce, F., & Hein, M. (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th international conference on machine learning*.

Dapello, J., Marques, T., Schrimpf, M., Geiger, F., Cox, D., & DiCarlo, J. (2020). Simulating a primary visual cortex at the front of CNNs improves robustness to image perturbations. In *34th conference on neural information processing systems*.

Dhamdhere, K., Yan, Q., & Sundararajan, M. (2019). How important is a neuron? In *7th international conference on learning representations*.

DiCarlo, J. J., & Cox, D. D. (2007). Untangling invariant object recognition. *Trends in Cognitive Sciences*, http://dx.doi.org/10.1016/j.tics.2007.06.010.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., & Li, J. (2018). Boosting adversarial attacks with momentum. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*. http://dx.doi.org/10.1109/CVPR.2018.00957.

Dujmović, M., Malhotra, G., & Bowers, J. S. (2020). What do adversarial images tell us about human vision? *ELife*, *9*, http://dx.doi.org/10.7554/ELIFE.55978.

Eguchi, A., Isbister, J. B., Ahmad, N., & Stringer, S. (2018). The emergence of polychronization and feature binding in a spiking neural network model of the primate ventral visual system. *Psychological Review*, http://dx.doi.org/10.1037/rev0000103.

Elsayed, G. F., Papernot, N., Shankar, S., Kurakin, A., Cheung, B., Goodfellow, I., & Sohl-Dickstein, J. (2018). Adversarial examples that fool both computer vision and time-limited humans. In *Advances in neural information processing systems, Vol. 2018-December*.

Ford, N., Gilmer, J., Carlini, N., & Cubuk, E. D. (2019). Adversarial examples are a natural consequence of test error in noise. In *36th international conference on machine learning, Vol. 2019-June*.

Gilmer, J., Metz, L., Faghri, F., Schoenholz, S. S., Raghu, M., Wattenberg, M., Goodfellow, I., & Brain, G. (2018). The relationship between high-dimensional geometry and adversarial examples. ArXiv:1801.00634.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Goodfellow, I., Schlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International conference on learning representations*.

Gowal, S., Huang, P.-S., van den Oord, A., Mann, T., & Kohli, P. (2021). Self-supervised adversarial robustness for the low-label, high-data regime. In *International conference on learning representations*.

Gowal, S., Qin, C., Uesato, J., Mann, T., & Kohli, P. (2020). Uncovering the limits of adversarial training against norm-bounded adversarial examples. ArXiv.

Gray, C. M. (1999). The temporal correlation hypothsis of visual feature integration: Still alive and well. *Neuron*, *24*, 31–47.

Greff, K., Srivastava, R. K., & Schmidhuber, J. (2016). Binding via reconstruction clustering. In *4th international conference on learning representations*. URL http://arxiv.org/abs/1511.06418.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition, Vol. 2016-December* (pp. 770–778). IEEE Computer Society, http://dx.doi.org/10.1109/CVPR.2016.90.

Hein, M., Andriushchenko, M., & Bitterwolf, J. (2019). Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition, Vol. 2019-June*. http://dx.doi.org/10.1109/CVPR.2019.00013.

Hénaff, O. J., Srinivas, A., De Fauw, J., Razavi, A., Doersch, C., Eslami, S. M. A., & Oord, A. v. d. (2020). Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the 37th international conference on machine learning*. URL http://arxiv.org/abs/1905.09272.

Higgins, I., Chang, L., Langston, V., Hassabis, D., Summerfield, C., Tsao, D., & Botvinick, M. (2021). Unsupervised deep learning identifies semantic disentanglement in single inferotemporal face patch neurons. *Nature Communications*, *12*(1), http://dx.doi.org/10.1038/s41467-021-26751-5.

Hinton, G., Sabour, S., & Frosst, N. (2018). Matrix capsules with EM routing. In *6th international conference on learning representations*.

Hochstein, S., & Ahissar, M. (2002). View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron*, *36*(5), http://dx.doi.org/10.1016/S0896-6273(02)01091-7.

Huang, Y., Gornet, J., Dai, S., Yu, Z., Nguyen, T., Tsao, D., & Anandkumar, A. (2020). Neural networks with recurrent generative feedback. In *34th conference on neural information processing systems*.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings - 30th IEEE conference on computer vision and pattern recognition*. http://dx.doi.org/10.1109/CVPR.2017.243.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). Adversarial examples are not bugs, they are features. In *Neural information processing systems*.

Isbister, J. B., Eguchi, A., Ahmad, N., Galeazzi, J., Buckley, M., & Stringer, S. (2018). A new approach to solving the feature binding problem in primate vision. *Interface Focus*, *8*.

Jacobsen, J. H., Smeulders, A., & Oyallon, E. (2018). I-RevNet: Deep invertible networks. In *6th international conference on learning representations*.

Jalal, A., Ilyas, A., Daskalakis, C., & Dimakis, A. G. (2017). The robust manifold defense: Adversarial training using generative models. arXiv preprint arXiv:1712.09196.

Jang, U., Jah, S., & Jah, S. (2020). On the need for topology-aware generative models for manifold-based defenses. In *International conference on learning representations*.

Khoury, M., & Hadfield-Menell, D. (2018). On the geometry of adversarial examples. arXiv preprint arXiv:1811.00525.

Kim, S. H., & Feldman, J. (2009). Globally inconsistent figure/ground relations induced by a negative part. *Journal of Vision*, *9*(10), http://dx.doi.org/10.1167/9.10.8.

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd international conference on learning representations*.

Krizhevsky, A. (2009). *Learning Multiple Layers of Features from Tiny Images*. Science Department, University of Toronto, Tech..

Kubilius, J., Schrimpf, M., Kar, K., Rajalingham, R., Hong, H., Majaj, N. J., Issa, E. B., Bashivan, P., Prescott-Roy, J., Schmidt, K., Nayebi, A., Bear, D., Yamins, D. L., & DiCarlo, J. J. (2019). Brain-like object recognition with high-performing shallow recurrent ANNs. In *Advances in neural information processing systems, Vol. 32*.

Kurakin, A., Goodfellow, I. J., & Bengio, S. (2019). Adversarial examples in the physical world. In *5th international conference on learning representations*.

Laidlaw, C., Singla, S., & Feizi, S. (2021). Perceptual adversarial robustness: Defense against unseen threat models. In *International conference on learning representations*.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE* (November), (pp. 1–46). http://dx.doi.org/10.1109/5.726791.

Lee, H., Bae, H., & Yoon, S. (2021). Gradient masking of label smoothing in adversarial robustness. *IEEE Access*, *9*, http://dx.doi.org/10.1109/ACCESS.2020.3048120.

Li, Z., Brendel, W., Walker, E. Y., Cobos, E., Muhammad, T., Reimer, J., Bethge, M., Sinz, F. H., Pitkow, X., & Tolias, A. S. (2019). Learning from brains how to regularize machines. In *Advances in neural information processing systems, Vol. 32*.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., & Kipf, T. (2020). Object-centric learning with slot attention. In *34th conference on neural information processing systems*. URL http://arxiv.org/abs/2006.15055.

Lu, Y., Yin, J., Chen, Z., Gong, H., Liu, Y., Qian, L., Li, X., Liu, R., Andolina, I. M., & Wang, W. (2018). Revealing detail along the visual hierarchy: Neural clustering preserves acuity from V1 to V4. *Neuron*, *98*(2), http://dx.doi.org/10.1016/j.neuron.2018.03.009.

Lukasik, M., Bhojanapalli, S., Menon, A., & Kumar, S. (2020). Does label smoothing mitigate label noise? In *Proceedings of the 37th international conference on machine learning*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *6th international conference on learning representations*.

Marr, D. (1982). *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. MIT Press.

Moosavi-Dezfooli, S. M., Fawzi, A., & Frossard, P. (2016). DeepFool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*. http://dx.doi.org/10.1109/CVPR.2016.282.

Mu, N., & Gilmer, J. (2019). MNIST-C: A robustness benchmark for computer vision. In *ICML 2019 workshop on uncertainty and ro- bustness in deep learning*.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *ICML 2010 - proceedings, 27th international conference on machine learning*.

Nassar, J., Sokol, P., Chung, S., Harris, K., & Park, I. (2020). On 1/n neural representation and robustness. In *34th conference on neural information processing systems*.

Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings, twenty-first international conference on machine learning*. http://dx.doi.org/10.1145/1015330.1015435.

Pang, T., Yang, X., Dong, Y., Su, H., & Zhu, J. (2021). Bag of tricks for adversarial training. In *ICLR*.

Papernot, N., McDaniel, P. D., & Goodfellow, I. J. (2016). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on asia conference on computer and communications security*. http://dx.doi.org/10.1145/3052973.3053009.

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., & Hinton, G. (2019). Regularizing neural networks by penalizing confident output distributions. In *5th international conference on learning representations*.

Rauber, J., Brendel, W., & Bethge, M. (2017). Foolbox: A python toolbox to benchmark the robustness of machine learning models. In *Reliable machine learning in the wild workshop, 34th international conference on machine learning*. URL http://arxiv.org/abs/1707.04131.

Reichert, D. P., & Serre, T. (2014). Neuronal synchrony in complex-valued deep networks. In *2nd international conference on learning representations*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*. http://dx.doi.org/10.1007/978-3-319-24574-4{_}28.

Rony, J., Hafemann, L. G., Oliveira, L. S., Ben Ayed, I., Sabourin, R., & Granger, E. (2019). Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition, Vol. 2019-June*. http://dx.doi.org/10.1109/CVPR.2019.00445.

Rusak, E., Schott, L., Zimmermann, R. S., Bitterwolf, J., Bringmann, O., Bethge, M., & Brendel, W. (2020). A simple way to make neural networks robust against diverse image corruptions. In *Lecture notes in computer science, Vol. 12348 LNCS*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), http://dx.doi.org/10.1007/s11263-015-0816-y.

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems*.

Samangouei, P., Kabkab, M., & Chellappa, R. (2018). Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *6th international conference on learning representations*.

Schlag, I., Smolensky, P., Fernandez, R., Jojic, N., Schmidhuber, J., & Gao, J. (2019). Enhancing the transformer with explicit relational encoding for math problem solving. arXiv preprint arXiv:1910.06611.

Schott, L., Rauber, J., Bethge, M., & Brendel, W. (2019). Towards the first adversarially robust neural network model on MNIST. In *7th international conference on learning representations*.

Shafahi, A., Ghiasi, A., Huang, F., & Goldstein, T. (2019). Label smoothing and logit squeezing: A replacement for adversarial training? arXiv preprint arXiv:1910.11585.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd international conference on learning representations*.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *3rd international conference on learning representations*.

Singh, C., Yu, B., & James Murdoch, W. (2019). Hierarchical interpretations for neural network predictions. In *7th international conference on learning representations*.

Song, Y., Nowozin, S., Kushman, N., Kim, T., & Ermon, S. (2018). PixelDefend: Leveraging generative models to understand and defend against adversarial examples. In *6th international conference on learning representations*.

Srivastava, R. K., Greff, K., & Schmidhuber, J. (2015). Training very deep networks. In *Advances in neural information processing systems*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*.

Stephenson, C., Feather, J., Padhy, S., Elibol, O., Tang, H., McDermott, J., & Chung, S. Y. (2019). Untangling in invariant speech recognition. In *Advances in neural information processing systems, Vol. 32*.

Stringer, C., Pachitariu, M., Steinmetz, N., Carandini, M., & Harris, K. D. (2019). High-dimensional geometry of population responses in visual cortex. *Nature*, *571*(7765), http://dx.doi.org/10.1038/s41586-019-1346-5.

Stutz, D., Hein, M., & Schiele, B. (2019). Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*. http://dx.doi.org/10.1109/CVPR.2019.00714.

Summers, C., & Dinneen, M. J. (2019). Improved adversarial robustness via logit regularization methods. arXiv preprint arXiv:1906.03749.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition, Vol. 2016-December*. http://dx.doi.org/10.1109/CVPR.2016.308.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. In *2nd international conference on learning representations*.

Tadros, T., Ramyaa, R., Krishnan, G. P., & Bazhenov, M. (2020). Biologically inspired sleep algorithm for increased generalization and adversarial robustnes in deep neural networks. In *ICLR*.

Tanay, T., & Griffin, L. (2016). A boundary tilting persepective on the phenomenon of adversarial examples. arXiv preprint arXiv:1608.07690.

Tramèr, F., Behrmann, J., Carlini, N., Papernot, N., & Jacobsen, J. H. (2020). Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In *37th international conference on machine learning, Vol. PartF168147-13*.

Tramèr, F., & Boneh, D. (2019). Adversarial training and robustness for multiple perturbations. In *Advances in neural information processing systems, Vol. 32*.

Tramèr, F., Carlini, N., & Brendel, W. (2020). On adaptive attacks to adversarial example defenses. ArXiv.

Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2017). The space of transferable adversarial examples. arXiv preprint arXiv:1704.03453.

Treisman, A. (1996). The binding problem. *Current Opinion in Neurobiology*, *6*(2), http://dx.doi.org/10.1016/S0959-4388(96)80070-5.

Treisman, A. (1998). Feature binding, attention and object perception. *Philosophical Transactions of the Royal Society, Series B (Biological Sciences)*, http://dx.doi.org/10.1098/rstb.1998.0284.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., & Madry, A. (2019). Robustness may be at odds with accuracy. In *ICLR*. MIT.

van Steenkiste, S., Locatello, F., Schmidhuber, J., & Bachem, O. (2019). Are disentangled representations helpful for abstract visual reasoning? In *Advances in neural information processing systems, Vol. 32*.

Von Der Malsburg, C. (1999). The what and why of binding: The modeler's perspective. *Neuron*, *24*(1), 95–104. http://dx.doi.org/10.1016/S0896-6273(00)80825-9.

Vuyyuru, M., Banburski, A., Pant, N., & Poggio, T. (2020). Biologically inspired mechanisms for adversarial robustness. In *34th conference on neural information processing systems*.

Wallis, T. S., Funke, C. M., Ecker, A. S., Gatys, L. A., Wichmann, F. A., & Bethge, M. (2019). Image content is more important than bouma's law for scene metamers. *ELife*, *8*, http://dx.doi.org/10.7554/eLife.42512.

Warde-Farley, D., & Goodfellow, I. (2016). 11 Adversarial perturbations of deep neural networks. In *Perturbations, optimization, and statistics* (p. 311).

Whittington, J. C., Muller, T. H., Mark, S., Chen, G., Barry, C., Burgess, N., & Behrens, T. E. (2020). The Tolman-Eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation. *Cell*, *183*, http://dx.doi.org/10.1016/j.cell.2020.10.024.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

Xiao, C., Zhong, P., & Zheng, C. (2020). Enhancing adversarial defense by k-winners-take-all. In *International conference on learning representations 2020*.

Xu, C., Yang, J., Lai, H., Gao, J., Shen, L., & Yan, S. (2019). UP-CNN: Un-pooling augmented convolutional neural network. *Pattern Recognition Letters*, http://dx.doi.org/10.1016/j.patrec.2017.08.007.

Yuan, X., He, P., Zhu, Q., & Li, X. (2019). Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, http://dx.doi.org/10.1109/TNNLS.2018.2886017.

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. In *British machine vision conference 2016, Vol. 2016-September*. http://dx.doi.org/10.5244/C.30.87.

Zantedeschi, V., Nicolae, M. I., & Rawat, A. (2017). Efficient defenses against adversarial atacks. In *AISec 2017 - proceedings of the 10th ACM workshop on artificial intelligence and security*. http://dx.doi.org/10.1145/3128572.3140449.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*. http://dx.doi.org/10.1007/978-3-319-10590-1-53.

Zhou, Z., & Firestone, C. (2019). Humans can decipher adversarial images. *Nature Communications*, *10*(1), http://dx.doi.org/10.1038/s41467-019-08931-6.

Zhou, H., Friedman, H. S., & von der Heydt, R. (2000). Coding of border ownership in monkey visual cortex. *The Journal of Neuroscience*, *20*(17), 6594–6611. http://dx.doi.org/10.1523/JNEUROSCI.2797-12.2013.