UNIVERSITY OF CALIFORNIA SAN DIEGO

Comparing human and machine visual perception

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Cognitive Science

by

Vijay Veerabadran

Committee in charge:

Professor Virginia R. de Sa, Chair
Professor Andrea Chiba
Professor Garrison W. Cottrell
Professor Michael C. Mozer
Professor Eran Mukamel
Professor Zhuowen Tu

2024

The Dissertation of Vijay Veerabadran is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

DEDICATION

Dedicated to those who relentlessly pursue greatness, refuse to yield, and craft their own fate, to those driven by hunger for success in this world.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

As I reflect on the journey that has been my doctoral research, I am profoundly aware of the numerous individuals whose support and guidance have been instrumental in shaping my path. At the forefront of this journey has been my advisor, Professor Virginia de Sa, whose mentorship has been nothing short of transformative. Professor de Sa not only facilitated an environment conducive to my growth as a researcher but also provided the freedom to explore problems that piqued my curiosity, paired with invaluable feedback on how to approach them. I am eternally grateful for her support, and her genuine interest in my development as a seasoned researcher.

Next, I extend my heartfelt thanks to my committee members, Professor Andrea Chiba, Professor Garrison Cottrell, Professor Eran Mukamel, Professor Michael Mozer, and Professor Zhuowen Tu. Their insights and guidance were crucial in honing my research direction, allowing me to delve deeply into a significant area of study. Their generosity in sharing time and expertise played a key role in navigating through the inevitable research hurdles.

The Cognitive Science department at UCSD has been the ideal setting for my doctoral work. Being part of such a vibrant community of researchers committed to advancing our understanding of the brain has been both inspiring and humbling. I am grateful to all the faculty, staff, and students in the department for nurturing a culture rich in curiosity and collaboration.

My time in the de Sa lab has been significantly enriched by a group of exceptionally bright and supportive colleagues. To Ollie D’ Amico, Simon Fei, Shuangquan Feng, Kueida Liao, Eric Morgan, Mahta Mousavi, Srinivas Ravishankar, Shuai Tang, and Xiaojing Xu, I owe a debt of gratitude for their friendship, support, and insightful discussions, which have greatly contributed to my work and personal growth.

Lastly, but most importantly, I must acknowledge the unwavering support of my friends and family. Their love, encouragement, and belief in my abilities have been the bedrock of my resilience and determination. Without their constant support, the journey towards achieving my academic aspirations would have been immeasurably more challenging. To them, I offer my deepest thanks and love, for being my source of strength and inspiration throughout this journey and beyond.

In closing, the journey to this point has been one of collaboration, learning, and personal growth. To everyone who has been a part of this journey, thank you. Your contributions have not only shaped this thesis but have also indelibly marked my journey as a scientist and as a person.

Chapter 2, in part, is obtained from the following material: Veerabadran, V., Raina, R., and de Sa, V. R. (2021). Bio-inspired learnable divisive normalization for anns. In *SVRHM 2021 Workshop@ NeurIPS*. Chapter 3, in part, is obtained from the following material: Veerabadran, V. and de Sa, V. R. (2020). Learning compact generalizable neural representations supporting perceptual grouping. *arXiv preprint arXiv:2006.11716*. Chapter 4, in part, is obtained from the following material: Veerabadran, V., Ravishankar, S., Tang, Y., Raina, R., and de Sa, V. R. (2023b). Adaptive recurrent vision performs zero-shot computation scaling to unseen difficulty levels. In *Thirty-seventh Conference on Neural Information Processing Systems*. Chapter 5, in part, is obtained from the following material: Veerabadran, V., Goldman, J., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., Shlens, J., Sohl-Dickstein, J., Mozer, M. C., et al. (2023a). Subtle adversarial image manipulations influence both human and machine perception. *Nature Communications*, 14(1):4933. I, the dissertation author, was the primary investigator and author of all the above listed publications.

VITA

2013-2017	Bachelor of Engineering, Anna University, Chennai
2017-2018	Research Assistant, Brown University
2018-2024	Doctor of Philosophy, University of California San Diego

PUBLICATIONS

Veerabadran, V., Ravishankar, S., Tang, Y., Raina, R., & de Sa, V. R. (2023, November). Adaptive recurrent vision performs zero-shot computation scaling to unseen difficulty levels. *In Thirty-seventh Conference on Neural Information Processing Systems.*

Veerabadran, V., Goldman, J., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., Shlens, J., Sohl-Dickstein, J., Mozer, M. C., & Elsayed, G. F. (2023). Subtle adversarial image manipulations influence both human and machine perception. *Nature Communications*, 14(1), 4933. https://doi.org/10.1038/s41467-023-40499-0

Veerabadran, V., Tang, Y., Raina, R., & de Sa, V. (2023). Cortically motivated recurrence enables visual task extrapolation. *Journal of Vision*, 23(9), 4684-4684.

Veerabadran, V., Raina, R., & De Sa, V. (2022). Bio-inspired divisive normalization improves object recognition performance in ANNs. *Journal of Vision*, 22(14), 3592-3592.

Veerabadran, V., Raina, R., & de Sa, V. R. (2021). Bio-inspired learnable divisive normalization for ANNs. In *SVRHM 2021 Workshop @ NeurIPS.*

Veerabadran, V., & de Sa, V. R. (2020). Learning compact generalizable neural representations supporting perceptual grouping. *arXiv preprint arXiv:2006.11716.*

Veerabadran, V., & de Sa, V. R. (2019). V1Net: A computational model of cortical horizontal connections. In *SVRHM 2019 Workshop @ NeurIPS.*

FIELDS OF STUDY

Major Field: Cognitive Science

ABSTRACT OF THE DISSERTATION

Comparing human and machine visual perception

by

Vijay Veerabadran

Doctor of Philosophy in Cognitive Science

University of California San Diego, 2024

Professor Virginia R. de Sa, Chair

In this dissertation, we focus on examining differences in perception between humans and computer vision models and contribute novel research methods to increase their alignment. In recent studies comparing how humans and deep neural networks used in computer vision perceive visual stimuli, we find extensive evidence on how these highly performant models' visual perception often poorly aligns with human perception. For example, these models have been shown to classify objects in a scene solely based on a small fraction of border pixels in an image (Carter et al., 2021), preferentially attend to information outside the human frequency sensitivity spectrum (Subramanian et al., 2023), and (in)famously classify images by local texture rather than by global form (Geirhos et al., 2019b). These deviations of machine vision are often

due to their overreliance on short-range features and our first set of contributions directly address this by adding lateral connections—critical for long-range spatial feature processing in biological vision—into deep neural networks. First, in Chapters 2 and 3, we introduce the bio-inspired DivNormEI and V1Net models respectively which implement feedforward and recurrent lateral connections in deep neural networks (DNNs). We show that these models develop bio-realistic orientation tuning and directly lead to robust object recognition/segmentation. We also show that recurrent lateral connections give rise to parameter-efficient contour integration (a task well-known to test long-range feature integration capacity). In Chapter 4, we introduce LocRNN, a high-performing recurrent circuit evolved from V1Net and propose combining it with Adaptive Computation Time (ACT) to learn a dynamic instance-conditional number of RNN timesteps. ACT enables LocRNN to generalize in a zero-shot manner to novel test-time difficulty levels of challenging visual path integration tasks. These chapters together highlight the effectiveness of our proposed bio-inspired design in creating human-like robustness to out-of-distribution settings. Complimentary to bio-inspired design, we also propose a new way to compare human and machine perception; advancing this area helps us better identify factors of deviation between these systems and guides us in building future neural networks with stronger alignment. In an elaborate psychophysics study described in Chapter 5, we explored how humans and deep neural networks alike, can be tricked by barely noticeable adversarial changes to images. We discuss the degree of alignment between the two visual systems and identify factors which influence this alignment. Our actionable predictions we discuss in this Chapter inspires the design of future neural network models with a goal of strengthening their alignment to human perception. We conclude this dissertation by eliciting important future directions of expansion of the research described here to build the next generation of computer vision models increasingly aligned with human vision.

Chapter 1

Introduction

Computational models of vision have served the important coupled goals of understanding biological vision and progressing towards creating machines with powerful visual capability. The seminal work by Hubel and Wiesel (1968) characterizing receptive fields in the cat striate cortex inspired the Neocognitron (Fukushima, 1980a) (a hierarchical extension of this building block) and later the LeNet model (LeCun et al., 1998) which combined Convolutional Neural Networks with gradient-based learning, and is the predecessor to most of today's high-performing modern Artificial Neural Networks (ANNs). In this dissertation, we use 'ANNs' interchangeably with Deep Neural Networks (DNNs).

Even though this approach to take inspiration from biological visual processing led to disruptive developments in the field of computer vision, we note that recent model architecture design tends to largely disregard this approach. Modern art in the field of computer vision aim to develop models that optimize performance on a wide range of computer vision tasks such as image recognition and segmentation (Xu et al., 2023; Liu et al., 2022; Dosovitskiy et al., 2020; He et al., 2016), text-generation (Brown et al., 2020; Touvron et al., 2023; Chowdhery et al., 2023) and video recognition (Wei et al., 2022; Fan et al., 2021; Feichtenhofer et al., 2019). Despite their impressive state-of-the-art performance on many vision benchmarks, we often see evidence for the growing disparity between machine vision and biological vision (Subramanian et al., 2023; Feather et al., 2023; Bommasani et al., 2021; Koh et al., 2021; Geirhos et al., 2019a).

Due to the lack of explicit structure that once used to be an integral part of computer vision models, recent approaches tend to solve for a solution that minimizes a task-specific objective function and this often leads to their underlying mechanics driven primarily by 'shortcut strategies' that minimize the objective in hand but fail to learn generalizable solutions that can handle significant changes to stimulus attributes during test time (Geirhos et al., 2020). On an even serious note, interpretability of model decisions is becoming a progressively harder challenge as these models rely heavily on cues such as texture (Geirhos et al., 2018b; Brendel and Bethge, 2019a), high-frequency information (Subramanian et al., 2023), and very small subsets of pixels in images – even just the borders (Carter et al., 2021). While these are all cues likely also used by biological vision, one must note that we do not yet have evidence for such small-scale unitary cues strongly influencing high-level human perceptual decisions and cognition.

In this dissertation, we address the development of human-like computer vision algorithms from the following two directions: (I) Brain-inspired modeling of computer vision algorithms and (II) psychophysical assessment of the human-machine perceptual gap.

Brain-inspired modeling of computer vision algorithms: First, we hypothesized that the solution to improving computer vision models' generalizability and similarity to human-like visual reasoning lies in finding and implementing explicit biologically inspired structure found in prior visual neuroscience research into modern ANN-based computer vision models. Our contribution here is a cost-effective alternative to achieving human-like robustness using giant neural networks that are trained on large amounts of data for enormous amounts of GPU-hours (Geirhos et al., 2021). We highlight the difficulty involved in this task as it requires rethinking and combining the sophisticated yet small-scale computational neuroscience models with simple and large-scale big-data based DNN models trained on massively parallel compute. We conducted several computational modeling, experimental and in-silico electrophysiology studies combined with a holistic evaluation of model performance on in-distribution and out-of-distribution stimuli to study the validity of this hypothesis.

We posit that the abovementioned divergences between human and machine vision can be traced back to the latter's over-reliance on spatially local cues, and the lack of learning global cues across longer spatial extent. In this dissertation we describe our contribution adding neurally inspired lateral connections in deep neural networks for enhancing their spatial processing ability.

In Chapter 2, we introduce DivNormEI, a bio-inspired learnable divisive normalization model for ANNs. DivNormEI is a modified extension of prior computational neuroscience models of the canonical divisive normalization operation, a type of lateral interaction ubiqitously found in the brain (Carandini and Heeger, 2012) supporting spatial contextual processing. We develop an implementation of learnable divisive normalization as a differentiable neural network layer, integrate them into high-performing deep convolutional networks and train them on large-scale supervised and self-supervised objective functions with gradient-based learning. To our knowledge, we demonstrated for the first time, the simultaneous emergence of bio-realistic orientation tuning and robustness to perceptual distortions during object recognition/segmentation in such convolutional ANNs trained with DivNormEI.

In Chapter 3, we extend DivNormEI into a model of recurrent linear/non-linear lateral connections and evaluate it on two challenging visual reasoning benchmarks. We show that ANNs with carefully designed recurrent lateral connections achieve comparable performance to feedforward ANNs of multiple orders more parameters and processing depth. This study served as a precursor to our upcoming follow-up on making ANNs with recurrence generalize their learned visual reasoning strategies to novel test-time difficulty levels. In Chapter 4, we introduce a new more performant recurrent circuit of lateral connections, LocRNN, which retains the division of neurons into one population of principal neurons and a second population of interneurons but discards the unwanted gating computations and explicit divisive normalization operations present in V1Net. We combined LocRNN (and other RNNs from prior art) with Adaptive Computation Time (Graves, 2016) to learn recurrent halting at a per-instance level. We showed for the first time that Convolutional RNNs trained with Adaptive Computation Time (referred as AdRNNs) systematically generalize like humans to novel test-time difficulty levels of contour integration tasks by scaling their computation. We also showed that AdRNNs explain human behavioral data collected for these kinds of contour integration tasks using psychophysics.

Psychophysical assessment of human-machine perceptual gap: Complementary to bio-inspired modeling, we highlight the importance of simultaneously studying similarities between human and machine visual perception as this guides us in our quest to create more human-like computer vision models. In Chapter 5, we discuss our experimental studies where we tested whether human perception is susceptible to subtle adversarial manipulations that were assumed to uniquely plague computer vision systems. We showed that even very subtle adversarial manipulations significantly bias human visual perception. We showed that the extent of human perceptual bias increases with the magnitude of the adversarial manipulation, and that there exist differences among adversarial target object categories in how strongly they influence human perception. Although this observed alignment is promising, we recognize the ripe opportunitiy to further enhance this alignment by identifying factors that influence it. This work inspires the design of future computer vision models with a goal of being strongly aligned with human perception.

Finally, we discuss open future directions of our work such as the exploration of additional crucial biological architectural and training constraints (e.g. foveation, top-down connections, local learning rules) and development of state-of-the-art encoder models of primate ventral visual responses by optimizing ANNs' representational similarity to high-resolution single-unit neural recordings of awake saccading macaques. We hypothesize that both these directions are capable of enhancing the shared feature selectivity of ANNs and human vision, which we identified in Chapter 5 as a crucial component of human-machine perception alignment. We believe that the research described in this dissertation and the abovementioned future directions together will inspire an enthusiastic participation from the community in building machine perception systems that are increasingly aligned with human perception, especially now following the increased adaptation of these models in critical aspects of the society.

Chapter 2

Bio-inspired learnable divisive normalization for ANNs

2.1 Introduction

In this chapter we introduce DivNormEI, a novel computational model of divisive normalization and lateral interactions that is an extension of prior computational neuroscience models of normalization and horizontal connections (Blakeslee and McCourt, 1999; Schwartz and Simoncelli, 2001; Grossberg and Raizada, 2000; Li, 1998). Divisive normalization has been extensively studied in the fields of neuroscience and perception; these studies have highlighted the importance of this canonical computation for several traits of biological vision such as nonlinear response properties, efficient coding, invariance with respect to specific stimulus dimensions and redundancy reduction. Here, we combine divisive normalization with linear lateral interactions to design a single circuit that we call DivNormEI. Different from prior models that were fit to explain behavioral/physiological data or used a small stimulus set, here we explore training DivNormEI's parameters in a task-driven fashion by optimizing performance on large-scale supervised object recognition and by optimizing self-supervised objective functions with gradient-based learning. We demonstrate the emergence of the ubiquitous contrast invariant tuning property in a self-supervised ANN equipped with DivNormEI. Additionally, we report our observation of improved large-scale object recognition accuracy on the ImageNet-100 dataset by virtue of the DivNormEI block. Comparing the tuning properties of our model's simple-cell equivalent neurons pre- and post-normalization, we observe the crucial role played by lateral connections and normalization to emergence of contrast invariance. We observed that a VGG-16 network pretrained on the large-scale ImageNet dataset did not show this property of contrast invariant tuning, further highlighting the specific role of normalization and lateral connections in developing this particular invariance. We hypothesize that the hierarchical incorporation of our DivNormEI will promote development of invariance to more stimulus dimensions that shall be advantageous for high-level vision tasks such as image segmentation and object detection.

2.2 Related work

Computational models of divisive normalization and lateral connections have been studied previously by the neuroscience and vision science communities. Of these models, we find Schwartz and Simoncelli (2001)'s model to be most relevant to our divisive normalization computation, where the authors developed a simple yet neurally plausible circuit for divisive normalization optimized to maximize independence of filter responses. Our model is also related to Robinson et al. (2007)'s model that performs orientation- and spatial-frequency dependent normalization of filter responses with model parameters fit to best explain a suite of brightness illusions. Our work is also related to prior computational models of lateral interactions in the primary visual cortex such as Li (1998); Grossberg and Raizada (2000). Our proposed model is an extension of these sophisticated yet small-scale models (in terms of size and stimulus exposure) to integration within large-scale gradient-trained ANNs.

We find our work to be related to Ballé et al. (2015) where a deep network integrated with learnable Generalized Divisive Normalization modules (albeit without the linear lateral connections that are present in our model) is trained to perform the task of image density modeling and compression. Our proposed work is also related to Burg et al. (2021) where the authors develop an end-to-end trainable model of divisive normalization similar to ours. Key differences between our work and the above work are: (i) our model performs linear lateral inhibition and excitation on top of learnable divisive normalization, (ii) parameters of our model are estimated with computer vision tasks such as super-resolution and image recognition, whereas Burg et al. (2021) train their model to predict V1 neuronal responses recorded from macaque primary visual cortex.

2.3 Methods

In this work, we extend prior computational models of divisive normalization and lateral interactions, and implement them within the framework of modern ANNs. We develop a learnable version of this nonlinear normalization computation along with linear excitatory and inhibitory lateral connections in a single module we call DivNormEI explained as follows.

2.3.1 DivNormEI layer: Learnable divisive normalization with lateral connections

We begin by defining a typical instantiation of divisive normalization that has been widely studied by prior art wherein, divisive normalization computes the ratio of an individual neuron's response to the summed activity of other neurons in its neighborhood. The following equation summarizes this well-studied formulation of a neuron i's normalization corresponding to an input stimulus drive x:

$$y_i(x) = \frac{y_i(x)}{\Sigma_j \mathbf{w}_j * y_j(x) + \sigma^2} \tag{2.1}$$

where j represents neighboring neurons of i and $\mathbf{w}, \sigma$ are learnable free parameters.

We now define our proposed learnable divisive normalization layer with lateral connections called DivNormEI. Let Y be an intermediate feature map – in response to stimulus drive X – that is being normalized by DivNormEI s.t. $Y \in \mathrm{d}^{h,w,c}$. c is the number of features present in Y (e.g. if Y is the conv1 feature map in a ResNet-50 network, $c = 64$) while h, and w are the spatial dimensions of Y. In our model, each neuron in feature map k at spatial location i, j receive three kinds of activity modulation from neighboring neurons with i, j at their center:

(i) divisive (or shunting) modulation that performs learned upscaling / downscaling (similar to gain control), (ii) linear excitatory modulation that positively influences activity with a learned additive operation and (iii) linear inhibitory modulation that negatively influences activity with a subtractive operation.

Once the intermediate feature map Y arrives into the DivNormEI block, the first modulatory influence described above, i.e., divisive normalization of incoming features is performed as follows:

$$\tilde{Y}_{i,j,k}(x) = \frac{Y^2_{i,j,k}(x)}{\Sigma_{m,n}\mathbf{w}^{div}_{m,n,k}Y^2_{m,n,k}(x) + \sigma^2} \tag{2.2}$$

Per the above equation, divisive normalization of neurons in feature map k is performed using the weighted summed activity of neighboring neurons (indexed by m,n) with a learnable 2-D depthwise-convolutional weight matrix $\mathbf{w}^{div}_k \in \mathrm{d}^{h_{div},w_{div}}$ that is unique to each feature map. h_{div}, w_{div} represent the spatial extent of divisive normalization. Neurons in feature map k are normalized only by their spatial neighbors in the same feature map k. This particular design choice ensures that activations in neighboring spatial locations have reduced redundancy after divisive normalization.

The divisively normalized output $\tilde{Y}_{i,j,k}$ from Eqn. 2.2 is then modulated by two opposing lateral interactions: additive excitation and subtractive inhibition. Two learnable weight matrices $\mathbf{w}^{exc} \in \mathrm{d}^{h_e,w_e,c,c}$ and $\mathbf{w}^{inh} \in \mathrm{d}^{h_i,w_i,c,c}$ compute the weighted summed activity of neighboring neurons from all feature channels for excitation and inhibition as follows (h_e, w_e and h_i, w_i denote the spatial extent of excitatory and inhibitory lateral connections):

$$E_{i,j,k}(x) = \Sigma_{m,n,o}\mathbf{w}^{exc}_{m,n,o,k} * \tilde{Y}_{m,n,o}(x) \tag{2.3}$$

$$I_{i,j,k}(x) = \Sigma_{m,n,o}\mathbf{w}^{inh}_{m,n,o,k} * \tilde{Y}_{m,n,o}(x) \tag{2.4}$$

Eqn. 3 and 4 are implemented as 2D convolutions with kernels w^{exc} and w^{inh} respectively. Linear excitation and inhibition E and I are integrated with $\tilde{Y}$ as follows to produce the normalized

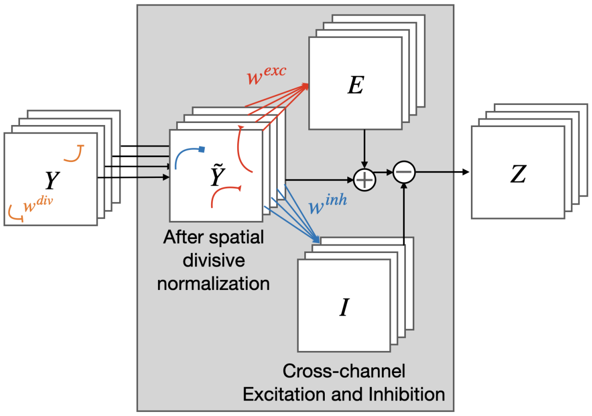

w^{exc}
E
Y
w^{div}
$\tilde{Y}$
w^{inh}
Z
After spatial divisive normalization
I
Cross-channel Excitation and Inhibition

Figure 2.1. Architecture diagram for DivNormEI. Here we demonstrate the working of DivNormEI described above with spatial divisive normalization and cross-channel linear excitation and inhibition.

output:

$$\tilde{Z}_{i,j,k}(x) = \tilde{Y}_{i,j,k} + E_{i,j,k}(x) - I_{i,j,k}(x) \tag{2.5}$$

$$Z_{i,j,k}(x) = \gamma(BN(\tilde{Z}_{i,j,k}(x))) \tag{2.6}$$

In the above Eqn. 6, BN corresponds to batch normalization (Ioffe and Szegedy, 2015) and γ represents the ReLU nonlinearity. In our experiments, we initialize $\mathbf{w}_{div}$ to be a set of c 2-D Gaussian kernels to build strong local divisive inhibition that prevent redundancies with gradually decreasing inhibition from far-off neurons. However, it is to be noted that these parameters are also trained with backpropagation along with $\mathbf{w}^{exc}$ and $\mathbf{w}^{inh}$ that are initialized randomly. Unless specified otherwise, all lateral connection weights $\mathbf{w}^{*}$ are maintained non-negative at each step of training. In subsequent sections, we discuss our experiments training ANNs with DivNormEI using self-supervised and supervised objective functions. In all our experiments, we set $h_{div} = w_{div} = 5$, $h_e = w_e = 9$ and $h_i = w_i = 7$ based on hyperparameter optimization w.r.t classification accuracy on a custom dev set containing a proportion of ImageNet-100 train images.

2.4 Results

2.4.1 Experiment 1: In-silico electrophysiology with task-driven ANN

Berkeley Segmentation Dataset 500. In this experiment, we explored the self-supervised task of image super-resolution on natural images from the Berkeley Segmentation Dataset (Arbelaez et al., 2010), referred from here onward as BSDS500. For training super-resolution models in this experiment, we sample random crops of size 48x48 pixels from the training images of BSDS500 (original image size is 321x481 pixels) and use them as the high-resolution ground truth images. Corresponding low-resolution input images are obtained by down-sampling the ground truth images by a factor of 4 to size 12x12 pixels.

Encoder-decoder architecture for super-resolution. We implemented an encoder-decoder architecture for super-resolution. The encoder contains a fixed convolution layer initialized with a Gabor filter bank and a DivNormEI layer. The Gabor filter bank contains square filters of size 15px and 21px. At each filter size, we design filters selective to 4 orientations ($\theta = 0, \pi/4, \pi/2, 3\pi/4$), 2 spatial frequencies (2 cycles per degree, 3 cycles per degree) and 4 phase values ($\phi = 0, \pi/2, \pi, 3\pi/2$). The encoder's output thus contains 64 feature maps. The decoder consists of 3 layers that upsample the encoder's output; first two layers are instantiated with transposed convolution (Zeiler and Fergus, 2014) with 64 filters each followed by batch normalization and ReLU nonlinearity. The last decoder layer is a 1x1 convolution with *tanh* nonlinearity that produces the final output in image space.

Lateral connections encourage data-driven emergence of contrast invariant tuning Simple cells in primary visual cortex of cats and primates maintain contrast-invariant orientation tuning, i.e., the orientation selective response of neurons remains roughly steady despite varying input stimulus contrast (Troyer et al., 1998; Nowak and Barone, 2009). In this study, we studied whether lateral connections and data-driven self-supervised learning can contribute to the emergence of this property. To study this hypothesis, we generated 100 sinusoidal grating stimuli that correspond to 25 grating orientations obtained at uniform intervals between 0 and π at 4 contrast levels (shown in Fig. 2.2.E). For each of the 64 feature maps in our encoder, we computed the neural tuning curves in response to the above 100 stimuli. Using stimuli at each contrast level, we computed the average of these tuning curves after ordering them such that each neuron's response to its preferred orientation stimulus was at the center of its tuning curve.

The average tuning curves **before normalization** are shown as a function of stimulus contrast in Fig.2.2.A, wherein the orientation selectivity decreases with decreasing stimulus contrast . This is similar to the average tuning curves of an ImageNet pretrained VGG-16 that we show in Fig. 2.2.B, i.e., both the self-supervised pre-normalization encoder neurons and ImageNet-pretrained VGG-16 neurons behave similarly and show a lack of contrast invariance.

On the other hand, the average tuning curve (of the same neurons in Fig.2.2.A) **after normalization using DivNormEI** as shown in Fig.2.2.C post-normalization is significantly more invariant to stimulus contrast (orientation selectivity and tuning curve variance is consistent across contrast levels). This post-normalization behavior shown in Fig. 2.2.C is similar to the reference behavior of cat primary visual cortical neurons we show in Fig. 2.2.B obtained from (Busse et al., 2009).

2.4.2 Experiment 2: DivNormEI improves object recognition accuracy on ImageNet-100

In this experiment, we evaluated the utility of our proposed DivNormEI model for the computer vision task of object recognition on the ImageNet-100 dataset (a subset of the ImageNet dataset with 100 randomly sampled classes which are also present in the validation set, standardized by Tian et al. (2020)). For fast prototyping and GPU memory constraints, we created a custom shallower variant of the VGG architecture with 9 layers of processing that we used in this experiment. We compared the following two architectures on ImageNet-100 classification: (a) Baseline-VGG9 – a 9-layer ANN with 3x3 convolution layers, max pooling and

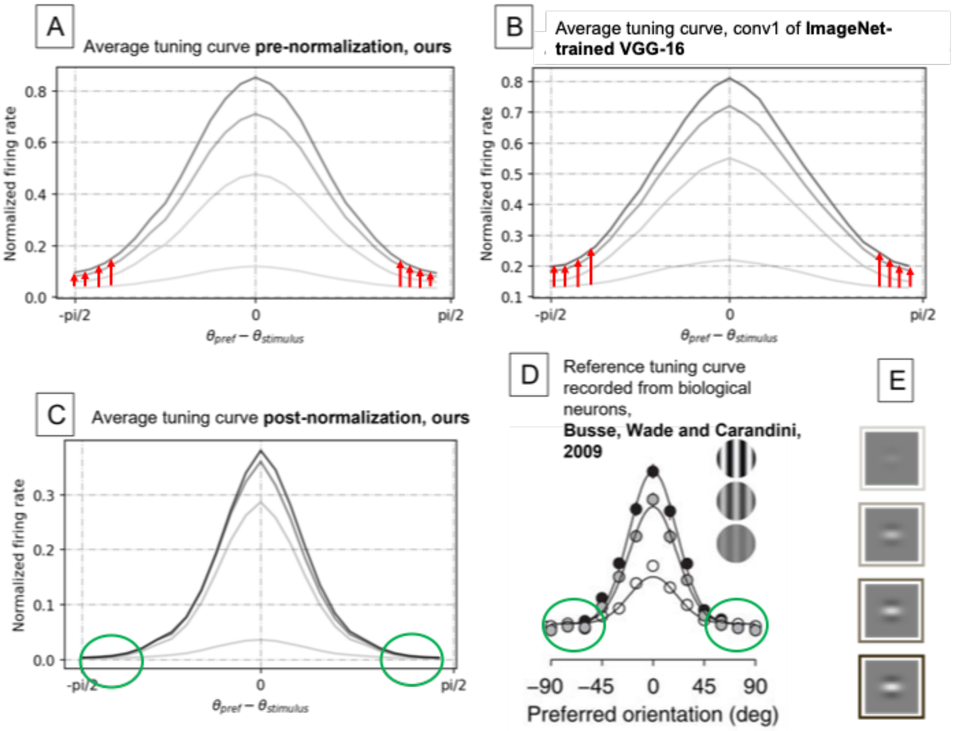


Figure 2.2. Neural tuning properties of networks pre- and post-normalization. (A) Average tuning curve of neurons in our self-supervised encoder's output before application of DivNormEI layer. (B) Average tuning curve of the conv1 neurons of the VGG-16 model (Simonyan and Zisserman, 2014). (C) Average tuning curve of our self-supervised encoder's output after DivNormEI layer. (D) Reference of the contrast invariant tuning property in cat V1 simple cells. (Busse et al., 2009) (E) Example sinusoidal grating stimulus at four contrast levels.

fully connected layers for classification, and (b) DivNorm-VGG9 – a 9-layer ANN with the same architecture as Baseline-VGG9, with the addition of an end-to-end trainable DivNormEI block at the output of the first convolution layer (output of DivNormEI block sent to subsequent 8 layers for classification). We trained two initializations of each of these models wherein the first pair of Baseline-VGG9 and DivNorm-VGG9 were initialized with the same weights, same as the second pair of Baseline-VGG9 and DivNorm-VGG9 models. **We observed that DivNorm-VGG9 models outperformed Baseline-VGG9 models on the ImageNet-100 validation set.** DivNorm-VGG9 had better sample efficiency than Baseline-VGG9, and was more accurate in classification by 1.8% (Top-1 validation accuracy of Baseline-VGG9: 73.3%, DivNorm-VGG9: **75.12%**). This observation suggests that DivNormEI is also relevant to improving the discriminative power of modern ANNs and can be integrated into further computer vision solutions like image segmentation and detection that rely on object semantics and discriminability.

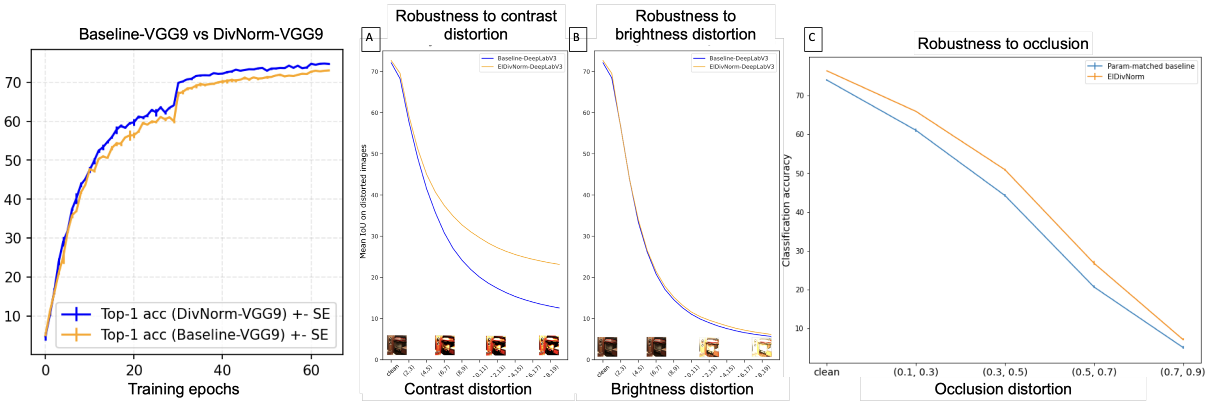
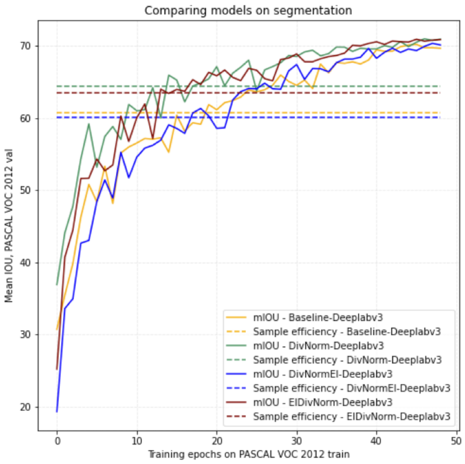
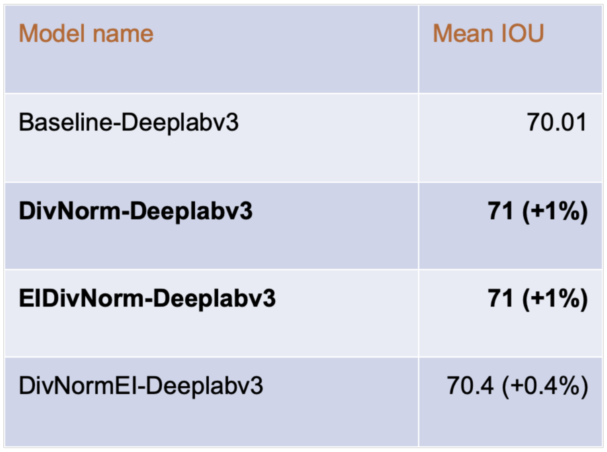


Figure 2.3. Image classification performance of Baseline-VGG9 and Divnorm-VGG9. Error bars computed over 2 random seeds. The steep accuracy increase at epoch 30 coincides with learning rate decay for both architectures

2.5 Discussion

In this chapter, we discussed DivNormEI, a novel computational model of divisive normalization and lateral interactions – both canonical computations that are ubiquitously found in biological visual neurons associated with diverse functions such as contrast normalization, redundancy reduction, and nonlinear neuronal response properties. We conducted two experiments to address (1) the emergent biological similarity from data-driven training of our model and (2) its utility in modern ANNs trained on computer vision problems. We computed the orientation tuning curves of neurons post normalization by DivNormEI and observed their response to be invariant to input stimulus contrast. This property of contrast invariant tuning is similar to that of primary visual cortical neurons. We also tested the specific role of divisive normalization in developing contrast invariant tuning; i.e., an ImageNet-pretrained VGG-16 model exposed to a million natural images still does not possess this property. We also compared two pairs of identical convolutional architectures with the difference that one network among each pair contained a DivNormEI layer after its first convolution layer on large-scale object recognition





Model name	Mean IOU
Baseline-Deeplabv3	70.01
DivNorm-Deeplabv3	**71 (+1%)**
EIDivNorm-Deeplabv3	**71 (+1%)**
DivNormEI-Deeplabv3	70.4 (+0.4%)

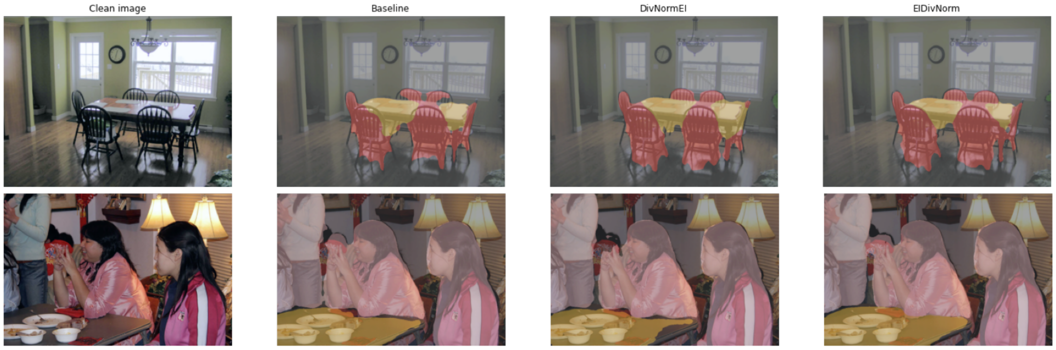


Figure 2.4. Comparing performance of baseline and DivNormEI models on robust segmentation. Ablations of DivNorm are compared in the Table shown here. DivNorm-Deeplabv3 is an ablation of the E-I linear connections. EIDivNorm-Deeplabv3 is an ablation of the ordering of linear (E-I) and nonlinear (DivNorm) operations.

from images in the ImageNet-100 dataset. We observed that the architectures with DivNormEI blocks possessed higher sample efficiency and classification accuracy compared to their *identical* baseline architectures without DivNormEI. We find this superior performance of DivNorm-architectures suggestive of the role of DivNormEI (and similar brain-inspired computations) in improving performance on computer vision tasks like image segmentation, where object discriminability is key. The studies in this part of the dissertation were limited to specific forms of lateral interaction and by application to smaller-sized deep networks due to time-limited computational constraints. However, we believe that our promising initial findings encourage further investigation of the role and implementation of divisive normalization and other relevant lateral and recurrent cortical computations in modern ANN architectures. In the next chapter, we discuss our extension of this model of feedforward lateral connections into a recurrent circuit with linear/non-linear lateral connections.

2.6 Acknowledgements

Chapter 2, in part, is obtained from the following publication. I, the dissertation author, was the primary investigator and author of this material.

Veerabadran, V., Raina, R., and de Sa, V. R. (2021). Bio-inspired learnable divisive normalization for anns. In *SVRHM 2021 Workshop@ NeurIPS*

This work was supported by: (1) Sony Faculty Innovation Award program from Sony Research, and (2) the Kavli Symposium Inspired Proposals award from the Kavli Institute of Brain and Mind.

Chapter 3

Recurrent lateral connections perform parameter-efficient contour integration

3.1 Introduction

Feedforward lateral connections are an effective model that enhances spatial processing locally. However, other visual reasoning problems such as contour integration require processing spatial dependencies of large distances in the scene. It is hypothesized that such integration of long-range spatial dependencies are mediated by lateral recurrent connections between cortical neurons (Roelfsema, 2006; Li, 1998). In this Chapter, we extend our previously proposed DivNormEI model of divisive lateral connections into a recurrent convolutional circuit. We make the following three-fold contribution: (1) We introduce V1Net – a novel bio-inspired recurrent unit based on prior research concerning the role of recurrent horizontal connections in human perceptual grouping, (2) We introduce MarkedLong, a simple contour integration task formulated as a binary image classification problem; this proposed dataset contains 800,000 distinct stimulus images with a resolution of 256 × 256 pixels, (3) We report observations from our comprehensive comparison of V1Net's performance accuracy, sample efficiency, solution compactness (defined by number of trainable parameters) and solution generalization against that of a range of feedforward and recurrent neural architectures on MarkedLong and the previously published PathFinder challenges (Linsley et al., 2018).

Our results suggest that (i) V1Net, a carefully designed recurrent unit inspired by cortical long-range horizontal connectivity matches or outperforms task performance of all our comparison models while containing a fraction of their parameter count, (ii) V1Net's superior performance compared to two relatively parameter-heavy recurrent architectures highlights the importance of explicit linear-nonlinear horizontal interactions and gain modulation, and (iii) V1Net's horizontal connections are reminiscent of reports on their structure from single-cell physiology and psychophysics (Gilbert and Wiesel, 1989; Field et al., 1993).

3.2 Related work

Neuroscience and vision scientists have used recurrent computer vision models operating on static images to model biological horizontal connections and/or re-entrant feedback connections. Typically in such models, bottom-up representations of static images are processed by a stack of identical computational blocks with weight sharing across the stack to model a nonlinear dynamical function of the bottom-up input (e.g., Liao and Poggio (2016a); Perona and Malik (1990)). This approach is in contrast to treating static images as a temporal sequence of successive pixels (Van Den Oord et al., 2016; Gregor et al., 2015; Oord et al., 2016b). Zamir et al. (2017) developed a variant of the Convolutional-LSTM architecture to introduce the intrinsic compositionality present in object features into recurrent neural representations through top-down feedback connections in their recurrent cell. CORNet-S proposed by Kubilius et al. (2018b) is a recurrent-convolutional model built through convolutional layers with residual skip-connections that tops the Brain-Score leaderboard, a benchmark evaluating object recognition performance and match to physiological recordings along the ventral visual pathway. Nayebi et al. (2018) proposed a class of recurrent computer vision models similar to CORNet-S designed through a large-scale Neural Architecture Search to jointly optimize object recognition performance and cortical resemblance. Unlike the above-mentioned models, our proposed V1Net block explicitly incorporates linear-nonlinear inhibitory and excitatory horizontal connections along with a gain control mechanism; we empirically show that this implementation-level bio-inspired design leads to fast learning of grouping routines. Horizontal Gated Recurrent Unit (hGRU), a model of cortical long-range horizontal connections proposed by Linsley et al. (2018) efficiently performs low-level perceptual grouping with high accuracy and sample efficiency compared to a range of feedforward and recurrent comparison models on their PathFinder challenge. Our proposed V1Net architecture is a computationally simpler and parameter-efficient model of horizontal connections compared to hGRU. Unlike one of hGRU's key assumptions that horizontal connections are symmetric in nature, V1Net incorporates 3 diverse cell types (and convolution kernels) which are not constrained to maintain symmetric horizontal synapses. Each of these cell types support one of three kinds of linear and nonlinear horizontal connectivity patterns. We explore key machine learning innovations that are lacking in hGRU. Among such innovations, V1Net utilizes depthwise-separable convolutions that have been shown to maintain high expressivity with a significant reduction in parameter count (Chollet, 2017; Howard et al., 2017). hGRU batch-normalizes the recurrent output state at each timestep using an independent batch normalization layer, restricting its inference operation to a fixed number of recurrent iterations that was used during training. V1Net tackles this issue by employing a single Layer Normalization operation (Ba et al., 2016) shared across timesteps.

Hasani et al. (2019) develop Surround Modulation Networks which introduce non-trainable filter banks of feedforward and lateral connections (similar to those in Robinson et al. (2007)) to DCNs, resulting in improved sample efficiency and perceptual robustness on ILSVRC (Russakovsky et al., 2015). Feedforward Atrous convolution operations are another way to encode long-range spatial dependencies as has been reported in dense prediction and autoregressive modeling studies (Yu and Koltun, 2015; Chen et al., 2018a,b; Oord et al., 2016a). Conditional random fields are a class of probabilistic graphical models which are also of relevance and have been shown to improve performance on dense prediction problems such as semantic segmentation through incorporation of global visual context (Arnab et al., 2018; Triggs and Verbeek, 2008).

3.3 Methods

3.3.1 Extending DivNormEI into a recurrent circuit - V1Net

In this section, we explain V1Net - our recurrent extension of DivNormEI discussed in Chapter 2. V1Net's recurrent horizontal connections learn to implement a version of incremental grouping by activity enhancement as proposed by Roelfsema (2006). Our experiments show that V1Net learns to build long-range horizontal interactions wherein spatially distant neurons that are part of the same object mutually excite each other along a transitive chain of local connections. This enhanced activity of neurons responding to parts of the same whole representationally implements a task-relevant grouping of bottom-up features; we discuss this observation later in this chapter.

Mathematical formulation: V1Net's gating computation (Eqn. 3.1) is identical to that of a ConvLSTM whose state updates are governed by the original LSTM update equations (Hochreiter and Schmidhuber, 1997) with fully-connected operations replaced with convolution operations. At any discrete recurrent iteration $\mathbf{t}$, consider the neuronal population activity pre-horizontal modulation to be available in the recurrent input $\mathbf{X_t} \in \mathbb{R}_{n,k,h,w}$, which is typically the output of a convolution layer in a DCN. Contextual interactions between the k^2 input feature pairs in $\mathbf{X_t}$ are governed by V1Net's horizontal connections.

The following procedure summarizes how V1Net's grouping representation is formed/updated at every recurrent iteration: The most recent neuronal response reflecting iterative grouping until iteration $\mathbf{t}$ is available in V1Net's hidden state, $\mathbf{H_t}$. Distinct convolutional filter banks with varying spatial dimensions, namely $\mathbf{W_{div}}$, $\mathbf{W_{inh}}$ and $\mathbf{W_{exc}}$, model a mapping from state $\mathbf{H_{t-1}}$ (from the previous iteration) and the bottom-up feedforward input $\mathbf{X_t}$ to the following populations of three different cell types each of which implement (1) gain control, (2) subtractive inhibition and (3) additive excitation. Their corresponding activities are stored in $\mathbf{H_t^{div}}$, $\mathbf{H_t^{inh}}$ and $\mathbf{H_t^{exc}}$ (Eqn. 3.2,3.3); each $\mathbf{H_t^{+}}$ is a 4-dimensional tensor of identical shape to $\mathbf{H_t}$ and $\mathbf{X_t}$. Excitatory influence and gain modulation are then integrated with $\mathbf{g_t}$, a nonlinear function of $\mathbf{X_t}$ and $\mathbf{H_t}$ to produce

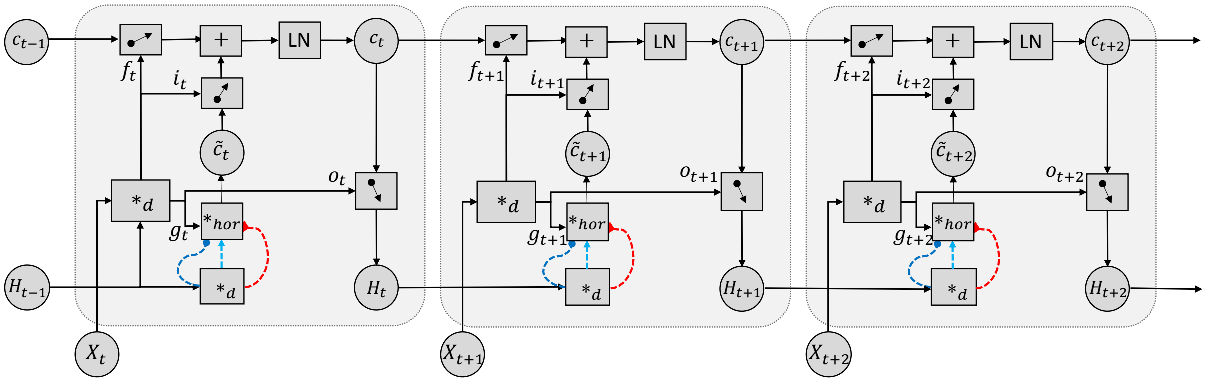


Figure 3.1. Our proposed V1Net cell architecture with recurrent linear excitatory (red), subtractive inhibitory (turquoise) and nonlinear gain modulating (navy blue) horizontal connections unrolled for three recurrent iterations. Eqn. 3.2,3.3,3.4 from Sec. 3.3.1 are encapsulated into box ($*_{hor}$) in this figure.

$\mathbf{\tilde{c}_t}$, a candidate representation of the current grouping calculation (Eqn. 3.4). $\mathbf{\tilde{c}_t}$ is mixed with the previous grouping estimate ($\mathbf{c_{t-1}}$) to produce an updated grouping estimate, $\mathbf{c_t}$. A nonlinear mapping of $\mathbf{c_t}$ following layer normalization (LN) and gating produces $\mathbf{H_t}$ (Eqn. 3.5) which contains updated grouping information for subsequent processing. The following equations summarize V1Net's horizontal connection dynamics. Depthwise separable convolutions are accompanied by additive biases which are omitted from the equations for reading convenience:

$$\begin{pmatrix} \mathbf{f_t} \\ \mathbf{i_t} \\ \mathbf{o_t} \\ \mathbf{g_t} \end{pmatrix} = \sigma(\mathbf{W_{xh}} *_d \mathbf{X_t} + \mathbf{U_{hh}} *_d \mathbf{H_{t-1}}) \tag{3.1}$$

$$\mathbf{H_t^{exc}}, \mathbf{H_t^{inh}} = \sigma(\mathbf{W_{exc}} *_d \mathbf{H_{t-1}}), \sigma(\mathbf{W_{inh}} *_d \mathbf{H_{t-1}}) \tag{3.2}$$

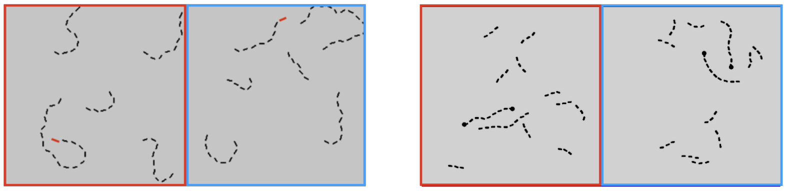
$$\mathbf{H_t^{div}} = \sigma(\mathbf{W_{div}} *_d \mathbf{H_{t-1}}) \tag{3.3}$$

$$\mathbf{\tilde{c}_t} = \mathbf{H_t^{div}} \times (\mathbf{g_t} + \mathbf{H_t^{exc}}) - \mathbf{H_t^{inh}}; \tag{3.4}$$

$$\mathbf{c_t} = \mathbf{f_t} \odot \mathbf{c}_{t-1} + \mathbf{i_t} \odot \tanh(\mathbf{\tilde{c}_t}); \mathbf{H_t} = \mathbf{o_t} \odot \gamma(\mathbf{LN}(\mathbf{c_t})) \tag{3.5}$$

Each $\mathbf{W}_.$ and $\mathbf{U}_.$ is a 2-D convolution kernel, $\sigma(.)$ and $\gamma(.)$ represent sigmoid and ReLU nonlinearities respectively; $*_d$ represents 2-D depthwise separable convolution (Jin et al., 2014; Chollet, 2017). A key property is that receptive fields of neurons in the horizontal kernels grow at every iteration of recurrence, and hence their range of horizontal connectivity scales positively with the amount of recurrent processing. The V1Net unit is shown in Fig.3.1.

3.3.2 Datasets – MarkedLong and PathFinder



(a) Samples from MarkedLong **(b)** Samples from PathFinder

Figure 3.2. Sample images from the MarkedLong (a) and PathFinder (b) datasets. Images are color-inverted for viewing convenience (keeping marked PathFinder segment red).

Dataset 1: MarkedLong contour integration benchmark We introduce the MarkedLong contour integration benchmark, a large scale 2-Alternative Forced Choice (2-AFC) task, i.e., a binary image classification problem that tests model ability to learn low-level serial grouping routines. In MarkedLong, each image quadrant contains one connected path P_i composed of l_i locally co-circular oriented segments (i.e., tangential to the same circle) of equal size. Three of these paths have a length $l_i = 12$, while the remaining path has length $l_j = 18$. One of the 54 segments of these 4 connected paths is rendered in a marker-red color, while all other segments are rendered in the foreground white color in a uniform black background. An image is classified positive if one of the segments of the long path P_j is marked red, hence the name MarkedLong; it is classified negative if one of the segments of the three short paths is marked red. Each path of an image is rendered with a multi-stage stochastic path-rendering algorithm discussed in the Supplementary Information (see Supplementary Sec. 3.3.3) to avoid the strategy of rote-memorizing instances of the dataset. The dataset consists of 800,000 RGB images of 256 × 256 pixels. We used a 75-25 training/validation split for MarkedLong experiments.

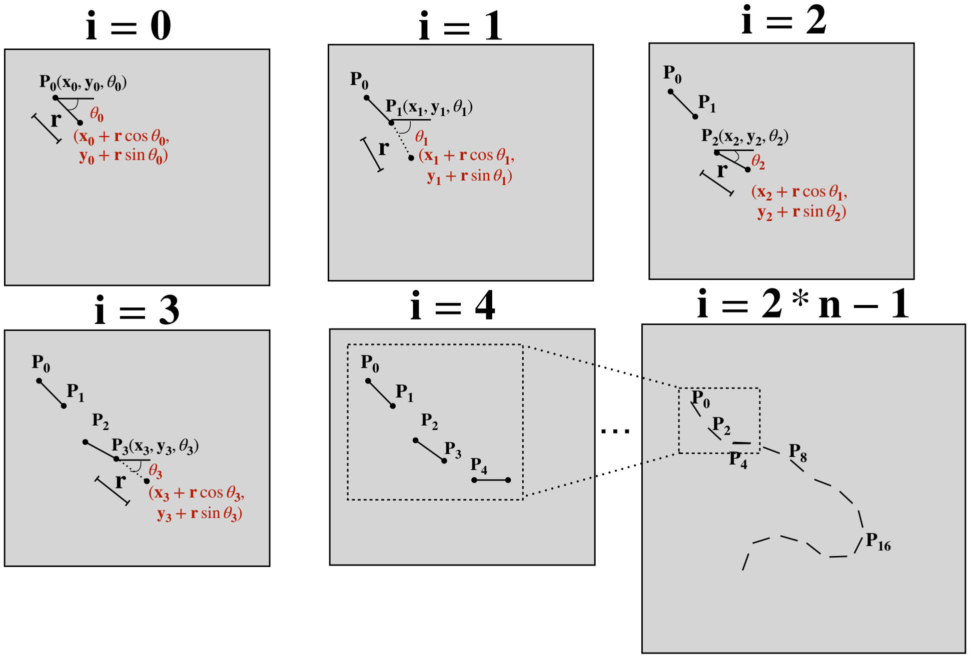
Dataset 2: PathFinder contour integration benchmark We also test the models on one version of the PathFinder challenge introduced by Linsley et al. (2018). In PathFinder, each image consists of two main connected paths P_0 and P_1 of length 9 segments as in MarkedLong. Each image also contains two circular disks each of which is placed at one of the 4 end points of P_0 and P_1. Images that contain a disk on both ends of the same path are classified as positive, and those containing a disk on endpoints of different paths are classified as negative. This dataset consists of a total of 1,000,000 RGB images of 150 × 150 pixels. We used the medium-difficulty version of pathfinder with 9-length main paths to balance task difficulty and prototyping timescale. Following the experimental settings of Linsley et al. (2018), we used a 90-10 training/validation split for our PathFinder experiments.

Both MarkedLong and PathFinder introduce short distractor paths in the background in every image to increase task difficulty via visual crowding.

3.3.3 Details on MarkedLong dataset generation

The central component to generating MarkedLong images is our stochastic path-rendering algorithm described below. As mentioned in the main paper, all MarkedLong images contain exactly one long path (18 bars long) and three short paths (12 bars long). In addition to these 4 paths, few more shorter distractor paths (6 bars long) are placed at randomly sampled locations within the image. Each one of the 4 primary paths (18- and 12-bars long) originates from a seed location sampled uniformly within one of the 4 quadrants in an image (each path seeds from a unique quadrant). Deliberately seeding paths in unique quadrants reduces the amount of intersection (or) overlap of different paths by pushing them away from each other. Originating from these seed locations, fully grown connected paths of desired lengths are developed using the following iterative algorithm.

Path rendering description: Our algorithm samples 2*n points that lie on a smooth connected curve that is locally co-circular; every alternate pair of points is then connected to form an n-length path. A seed pixel location $\mathbf{P_0}(\mathbf{x_0}, \mathbf{y_0})$ is sampled uniformly within a desired



$\mathbf{i = 0}$
$\mathbf{P_0(x_0, y_0, \theta_0)}$
θ_0
$\mathbf{r}$
$(\mathbf{x_0} + \mathbf{r}\cos\theta_0,$
$\mathbf{y_0} + \mathbf{r}\sin\theta_0)$
$\mathbf{i = 1}$
$\mathbf{P_0}$
$\mathbf{P_1(x_1, y_1, \theta_1)}$
θ_1
$\mathbf{r}$
$(\mathbf{x_1} + \mathbf{r}\cos\theta_1,$
$\mathbf{y_1} + \mathbf{r}\sin\theta_1)$
$\mathbf{i = 2}$
$\mathbf{P_0}$
$\mathbf{P_1}$
$\mathbf{P_2(x_2, y_2, \theta_2)}$
θ_2
$\mathbf{r}$
$(\mathbf{x_2} + \mathbf{r}\cos\theta_1,$
$\mathbf{y_2} + \mathbf{r}\sin\theta_2)$
$\mathbf{i = 3}$
$\mathbf{P_0}$
$\mathbf{P_1}$
$\mathbf{P_2}$
$\mathbf{P_3(x_3, y_3, \theta_3)}$
θ_3
$\mathbf{r}$
$(\mathbf{x_3} + \mathbf{r}\cos\theta_3,$
$\mathbf{y_3} + \mathbf{r}\sin\theta_3)$
$\mathbf{i = 4}$
$\mathbf{P_0}$
$\mathbf{P_1}$
$\mathbf{P_2}$
$\mathbf{P_3}$
$\mathbf{P_4}$
...
$\mathbf{i = 2 * n - 1}$
$\mathbf{P_0}$
$\mathbf{P_2}$
$\mathbf{P_4}$
$\mathbf{P_8}$
$\mathbf{P_{16}}$

Figure 3.3. Step by step illustration of MarkedLong's path generation algorithm. The shown bar sizes / distance between bars in this figure is not representative of images in our dataset, we use the most suitable settings for reader's convenience in this figure.

quadrant's boundaries as the starting point of the first bar. The orientation $\theta_\mathbf{0}$ of this bar is sampled from the uniform distribution $\theta \sim \mathbf{U}(\mathbf{0}, \pi/\mathbf{4})$. The end point of this bar, which acts as the starting point for the next iteration, is computed by projecting a vector of length $\mathbf{r}$ at an angle θ_0 to the horizontal axis, $\mathbf{P_1}(\mathbf{x_1}, \mathbf{y_1}) = (\mathbf{x_0} + \mathbf{r}\cos\theta_\mathbf{0}, \mathbf{y_0} + \mathbf{r}\sin\theta_\mathbf{0})$.

Each successive point $\mathbf{P_{i+1}}$ is updated to $[\mathbf{x_i} + \mathbf{r}\cos\theta_\mathbf{i+1}, \mathbf{y_i} + \mathbf{r}\sin\theta_\mathbf{i+1}]$, $\theta_\mathbf{i+1}$ is sampled from the uniform distribution $\theta \sim U(\theta_i - \pi/4, \theta_i + \pi/4)$. In the event of $\mathbf{P_{i+1}}$ falling outside the image boundary, the algorithm backtracks two steps to step $i-1$, and continues rendering along a different path. This iterative process stops upon reaching the exit condition, $\mathbf{i = 2 * n - 1}$. Every pair of adjacent points $\mathbf{P_{2*i}}, \mathbf{P_{2*i+1}}, \mathbf{i} \in [\mathbf{0}, \mathbf{n-1}]$ are connected using line segments at this stage to render a connected n-length path. A pictorial walkthrough of this algorithm is shown in Fig. 3.3.

Generating positive class images: We sample one quadrant $i_q^l \in [0,3]$ within which the longest path is seeded. Path rendering algorithm selects a seed within this quadrant and grows a path of length $n = 18$. A marker index i_m is sampled uniformly from $i \sim U[0, 17]$ and rendered with a line segment red in color. Following longest marked path's generation, one short path is seeded and generated in each remaining quadrant with length $n = 12$. $n_d \sim U[1, 4]$ number of distractors are then rendered within the entire image.

Generating negative class images: In order to generate negative class images, none of the bars in the longest path seeded in i_q^l are marked with a red marker bar. Instead, one of the three 12-length bars is chosen at random to be marked; a marker index i_m is sampled uniformly from $i \sim U[0, 11]$ and this bar on the chosen short path is rendered with a line segment red in color.

In total, MarkedLong consists of 400,000 positive class images and 400,000 negative class images; 100,000 images rendered in each class are flipped horizontally and vertically through online data augmentation, resulting in 400,000 images per class.

3.3.4 Implementation details of baseline models compared

We evaluate each model using the following 3 metrics: (a) **Performance accuracy**: Maximum validation accuracy measured on MarkedLong and PathFinder respectively (Fig.3.4.a,b), (b) **Sample efficiency**: Measure of how data-efficient learning is, computed as the area under the validation curve from beginning to the end of training on MarkedLong and PathFinder respectively (Fig.3.4.c,d), and (c) **Parameter efficiency**: Measured by counting the number of trainable parameters in a model.

We study 15 different models (for exact architecture, see Supplementary Sec. **??**). Broadly, we classify them as belonging to the following four families of DCNs: **(F1)** Standard Convolutional DCNs, **(F2)** Atrous-convolutional DCNs, **(F3)** Recurrent-Convolutional DCNs, and **(F4)** very deep over-parameterized DCNs.

Implementation details All standard convolutional DCNs (F1), Atrous-convolutional DCNs (F2) and Recurrent convolutional DCNs (F3) were implemented using a common three-block structure. These models consist of a standard *input block* designed as a 7x7 convolution layer with 32 filters. A subsequent block that we denote as the *intermediate block* is unique to each model and responsible for learning task-relevant grouping transformations of the input activity. Our *readout block* consisted of a Global-Average-Pooling operation followed by a 512-D fully connected layer and a 2-D output layer. Parameters were initialized with the Variance Scaling method (He et al., 2015) and optimized using the Adam optimizer (Kingma and Ba, 2015) with initial learning rate of 1e-3 on Pathfinder and 5e-4 on MarkedLong respectively, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 1e-8$. All models were implemented using the TensorFlow framework (Abadi et al., 2016) and trained on 8-core Google Cloud TPU-v3 nodes with a batch size of 128.

F1: Standard Convolutional DCNs: We compare 5 standard DCN models with varying amounts of hierarchical processing. Our comparison consists of (1) a 3-layer DCN (FF-1L), (2) a 5-layer DCN (FF-4L) and (3) a 7-layer DCN (FF-7L). To assess the impact of the number of output convolutional kernels at a given depth of processing, we evaluated (4) FF-7Lx2, a 7-layer DCN in which each intermediate block convolutional layer learns twice the number of filters as in FF-7L. Intermediate block convolution layers of (1), (2) and (3) contained a filter bank of 32 5x5 kernels, while (4) contained 64 5x5 kernels. We implemented (5) a feedforward surround-modulation CNN (Hasani et al., 2019) to assess the role of feedforward vs recurrent lateral connections in grouping. FF-SMCNN's architecture is identical to that of FF-7L, but its input convolution layer consists of fixed DoG kernels (in addition to trainable kernels) that implement convolutional surround modulation. None of these tested feedforward DCNs matched the accuracy of our proposed V1Net model on MarkedLong and had a strikingly large sample efficiency difference compared to V1Net. This observation suggests an inability of feedforward DCNs in F1 to efficiently group long paths in MarkedLong. On PathFinder, only the deep 7-layer FF-7L and FF-7L-x2 matched V1Net's performance. DCNs in F1 which, at their best come close to matching V1Net's performance, are significantly more parameter-expensive.

F2: Atrous-convolutional DCNs: An exponential growth in the receptive field size of dilated convolution operations, or atrous convolutions, have been shown to be useful for learning long-range contextual features in DCNs and should hence be beneficial for our contour integration tasks. We compare 4 different atrous convolutional DCNs with varying amount of processing depth. We implement (6) a 3-layer atrous-DCN (ATR-1L), (7) a 5-layer atrous-DCN (ATR-4L), (8) a 7-layer atrous-DCN (ATR-7L) and (9) a high parameter variant of ATR-7L with twice the convolutional filters at each intermediate block convolutional layer that we denote as ATR-7Lx2. We implemented these models by replacing the standard 2D-convolution layers in models (1,2,3,4) with atrous convolution layers (Chollet, 2017) while maintaining the kernel size as 5x5 within the intermediate block. Among these models, we note that the shallow Atrous-DCNs are significantly worse than V1Net, i.e, our experiments found Atrous Convolutions to be useful only when they were accompanied by a deep hierarchical architecture (as in ATR-7L, ATR-7Lx2). However, they struggle to learn the PathFinder challenge wherein they fail to match the sample efficiency of V1Net by a significant extent.

F3: Recurrent convolutional DCNs: We test the utility of recurrent computations by evaluating 3 recurrent-convolutional DCNs with varying inspiration and architecture. Each of these networks is designed as a 3-layer DCN with the intermediate block comprised of one of our chosen recurrent-convolutional layers with 5 iterations of recurrence. In this class, we evaluate (10) GRU-1L (Cho et al., 2014) wherein the intermediate block consists of a Convolutional-GRU cell, (11) CORNet-S wherein the intermediate block consists of the bio-inspired CORNet-S module (Kubilius et al., 2018a), (12) V1Net-1L wherein the middle layer consists of a V1Net module. Results from this family of models suggest that not all recurrent architectures perform equally. CORNet-S, inspired by the concepts of weight sharing and residual skip connections matches V1Net's accuracy and sample efficiency on PathFinder. However on MarkedLong, CORNet-S has a significantly poor sample efficiency compared to V1Net. GRU-1L matches the accuracy of V1Net on both MarkedLong and PathFinder, however, it has significantly worse sample efficiency on both problems. These models implement recurrent computations and consistently achieve high accuracies, suggesting they are learning grouping routines. However, they lack the diverse types of recurrent horizontal connections, gain modulation mechanism, and various cell-types that V1Net additionally possesses. We hypothesize this to be the reason for their sub-optimal learning speed compared to V1Net. While both GRU-1L and CORNet-S contain more parameters than V1Net, they are relatively more parameter-efficient compared to FF-7Lx2 models and those in F4 discussed next.

F4: Very deep overparameterized DCNs: We evaluate three previously published state-of-the-art DCNs which contain about $10^3\times$ more trainable parameters than a V1Net-1L model. In this class, we study the performance of (13) a variant of the VGG-16 model Simonyan and Zisserman (2014) with a batch normalization layer before every nonlinear rectification layer, (14) the latest ResNet-18 model with pre-activation He et al. (2016), and (15) the latest ResNet-50 model with pre-activation. We report experiments wherein we trained these models from scratch as we noticed that ImageNet-pretraining was inferior due to a significant domain shift. All models we tested in this family of over-parametrized DCNs showed consistently high performance on both tasks, similar to V1Net's performance.

We highlight that V1Net-1L, while being the most parameter efficient model we compared, consistently beats the accuracy and sample efficiency of all 11 models in the first three model families F1, F2 and F3 on PathFinder and MarkedLong. Negligible differences in the performance of the V1Net with those in F4 suggest that the extremely compact V1Net representations are equally as expressive as that of the evaluated very-deep state-of-the-art DCNs with just a fraction of trainable weights as is observed in Fig. 3.4.e,f. comparing model parameter efficiency.

3.4 Results

3.4.1 V1Net learns the most compact generalizable grouping routines

Although MarkedLong and PathFinder are different problems, we recognize a latent similarity between them in that generalizable solutions to either task involve similar functional aspects relating to contour integration. To investigate this, we explore the transfer learning accuracy of each model from MarkedLong to PathFinder. We trained each model on MarkedLong and finetuned the output fully-connected layers (for learning a new readout strategy) and all batch-normalization parameters (including ones in the input and intermediate blocks - to adapt to PathFinder's minibatch statistics) on PathFinder. We froze all convolution layers in the input and intermediate blocks (that we consider to implement contour grouping routines learned on MarkedLong). Our observations from this experiment show strong evidence supporting our initial hypothesis that carefully designed recurrent architectures learn compact generalizable solutions to perceptual grouping. Among the 15 models compared on this transfer learning experiment, only half the models showed promising generalization performance (Fig.3.4.g). None of the standard DCN models from F1 generalize to PathFinder; Fig.3.4.e shows that the majority of these models (except shallow 1-layer DCNs which show poor source performance) fail to generalize despite learning successful solutions to solve MarkedLong. Only the deep Atrous-DCNs from F2 show good generalization, further highlighting their dependency on parameter-heavy solutions and relatively more hierarchical processing compared to recurrent architectures. Out of the 3 tested recurrent architectures in F3, our proposed V1Net and CORNet-S generalize successfully. All overparameterized DCNs from F4 generalize well to PathFinder. We are ideally looking for compact models with high parameter efficiency and transfer accuracy which lie on the upper left-hand corner of Fig. 3.4.g., it has to be noted that V1Nets are the closest models in this region and are finetuning many fewer variables (10 variables) relative to CORNet-S (70) and DCNs in F4(200) that generalize well.

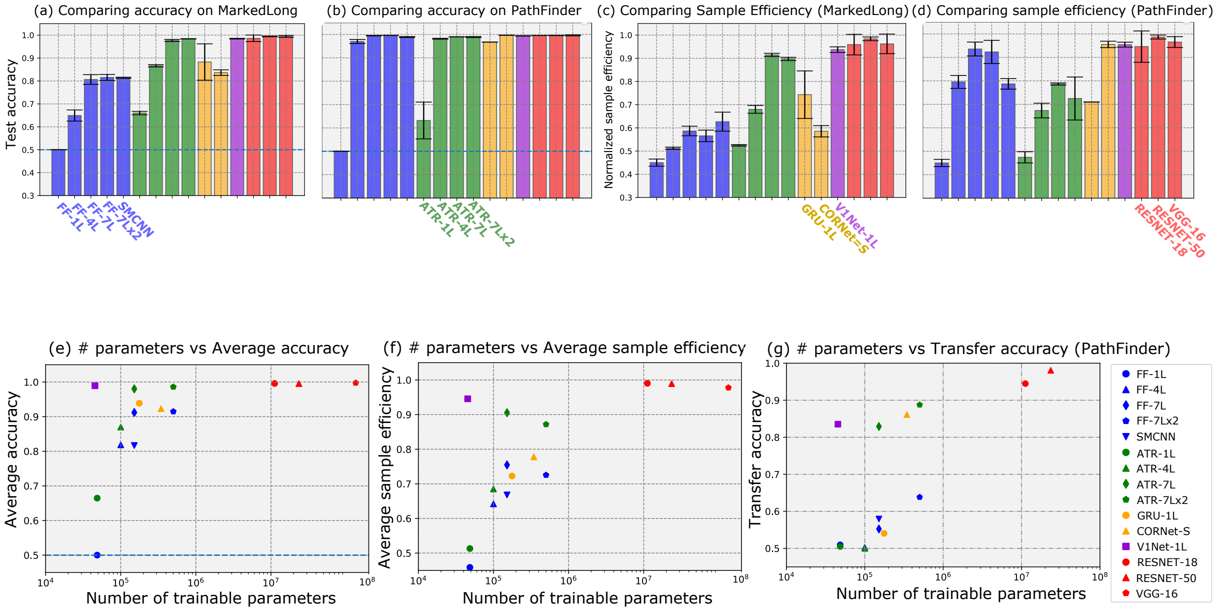
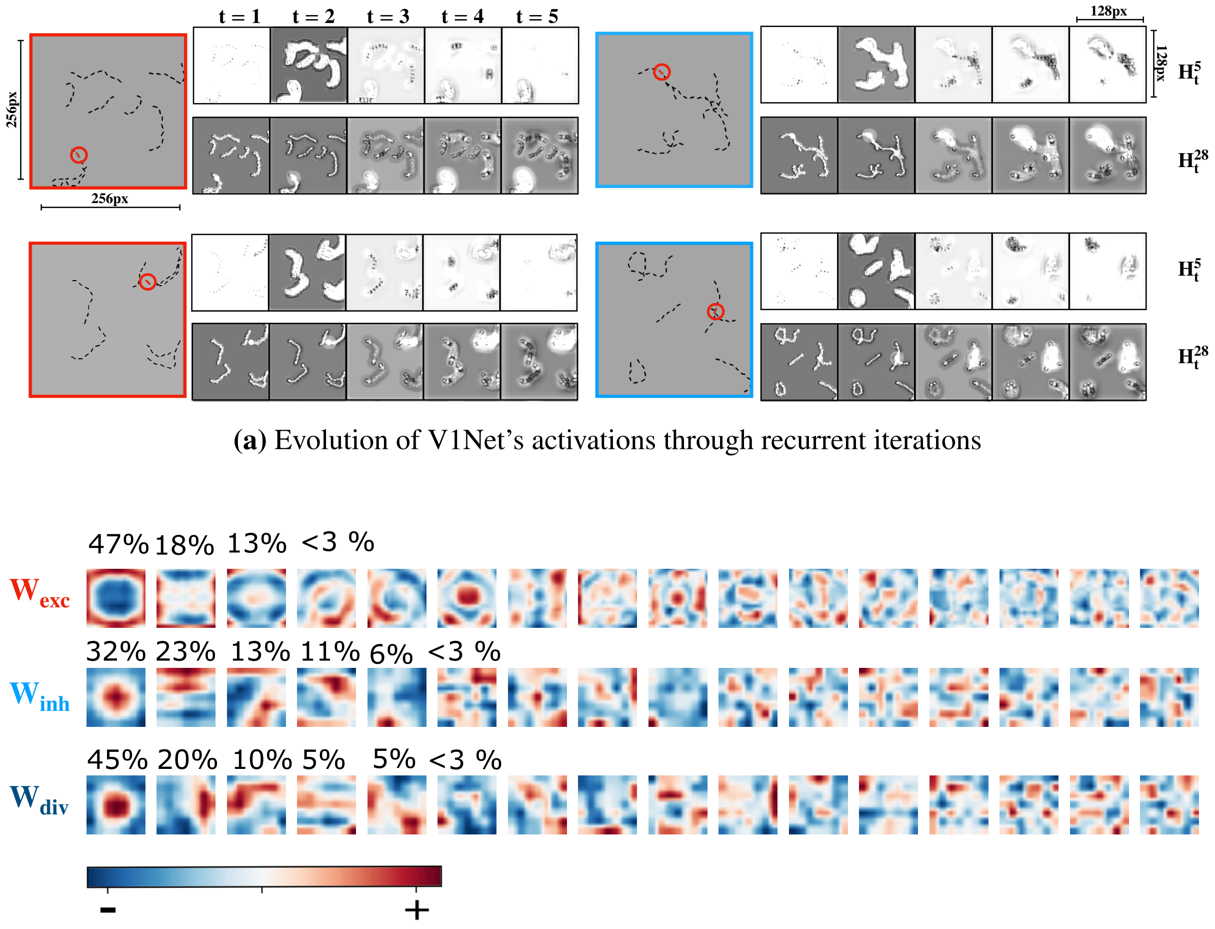


Figure 3.4. Comparing performance accuracy (a,b) and sample efficiency (c,d) on MarkedLong (a,c) & PathFinder (b,d). Parameter efficiency is computed using average accuracy (e) and sample efficiency (f) across both MarkedLong and PathFinder. Transfer accuracy of models trained on MarkedLong and finetuned on PathFinder shown in (g). Models in families F1, F2, F3 and F4 are color-coded with blue, green, (orange, V1Net with purple) and red bars. Error bars in the figures correspond to standard deviation across 3 model runs with different random seeds. Models from F1, F2 and F3 (except V1Net) tend to learn accurate sample-efficient solutions only on one of these two tasks as opposed to V1Nets and models in F4. V1Net's performance on-par with very-deep DCNs from F4 indicates their commensurate expressivity with DCNs which are three orders more parameter-heavy.

3.4.2 Interpreting V1Net's MarkedLong grouping strategy

In order to qualitatively understand V1Net's solution to MarkedLong, we visualized the temporal evolution of activation maps in $\mathbf{H_t}$, the internal state of a trained V1Net model that reflects grouping on MarkedLong. In Fig. 3.5a, we show a subset of these visualizations on a couple of positive (left column) and negative (right column) class images from MarkedLong: the $5^{th}(\mathbf{H_t^5})$ and $28^{th}(\mathbf{H_t^{28}})$ activation maps from a V1Net trained for 5 recurrent iterations. Each column of activation maps corresponds to the model's response at a particular timestep of processing. (1) Analyzing activations from $\mathbf{H_t^{28}}$ suggests that the activity along every path in the input gets progressively enhanced. Particularly, a blob of activity marking the 'red marker' is incrementally enhanced through recurrent processing to create an attention map focused on the marked path. (2) Activations from $\mathbf{H_t^5}$ are equally informative. During early recurrent steps, all paths are activated sparsely in $\mathbf{H_t^5}$. With more recurrent processing, only the marked path (bright blob in Fig. 3.5a) and longest path (dark blob) get enhanced activity consistently as they both compete and suppress the activity of all remaining paths. The top-right panel of Fig. 3.5a. is a particularly difficult example for even humans to quickly parse the grouping of the marked and long paths. It is interesting how in early recurrent steps, similar to the human visual system, V1Net groups together elements of two paths that intersect but later through further recurrent processing infers the correct part-whole ownership via long-range contextual feature integration. It then separates the long and marked path components into their respective light/dark groups. These visualizations provide an interpretation of how V1Net combines different algorithmic grouping blocks (attending to the marker, growing paths to find the longest path, etc).

We examined V1Net's learned horizontal connection patterns on MarkedLong by applying PCA to a bank of 32 excitatory ($\mathbf{W_{exc}}$), inhibitory ($\mathbf{W_{inh}}$) and divisive inhibition ($\mathbf{W_{div}}$) horizontal connection kernels each (Fig. 3.5b.). The resulting top 4 principal components explained about 80% of the total variance in each of these filter banks. These highly represented PC's – that resemble suppressive center-surround and facilitatory 'association-field' interactions – bear similar structure to biological horizontal connections in primate area V1 Gilbert and Wiesel (1989); Field et al. (1993). Further PCs are likely modeling intricate higher-order interactions.



(a) Evolution of V1Net's activations through recurrent iterations

(b) Representative horizontal connections learned in V1Net

Figure 3.5. (a) Visualizing temporal dynamics of V1Net activations for MarkedLong. Marker locations highlighted with red circles for viewing convenience. (b) Visualizing the top PCs computed from a bank of excitatory (top), inhibitory (middle) and divisive inhibition (bottom) horizontal connection kernels learned by a V1Net trained on MarkedLong sorted in decreasing order of their ratio of explained variance (mentioned in figure only for top PCs). See Sec. 3.4.2

3.4.3 Discussion

We hypothesized that recurrence-infused DCNs are capable of learning superior solutions to perceptual grouping problems involving contour integration governed by low-level Gestalt cues. We test this hypothesis through a two-pronged approach and find evidence in support of our hypothesis: A 3-layer V1Net-DCN learns the most compact, generalizable solutions that reach near-perfect accuracy solutions to both MarkedLong and PathFinder challenges with exceptional sample efficiency that is comparable only to very-deep and parameter-heavy state of the art

DCNs with three orders of magnitude more trainable parameters among a broad range of 15 feedforward and recurrent models. Visualizing the internal state of V1Net's recurrent grouping routine provides an algorithmic interpretation linking how V1Net's low-level recurrent horizontal interactions achieve the high-level computational goals of MarkedLong and PathFinder.

Our experimental results accord with prior findings from neuroscience and vision science on how the ventral visual stream performs perceptual grouping effortlessly using recurrent interactions despite their relatively shallow anatomy (compared to recent DCNs with multiple orders greater hierarchy) Lamme et al. (1998); Roelfsema (2006); Kreiman and Serre (2020). Here, we show that standard DCNs struggle to match the superior performance, sample efficiency and generalizability of the proposed recurrent V1Net-1L model on 2 contour integration tasks. On this note we believe that the utility of bio-inspired recurrent models such as V1Net should be explored in overcoming some of the recent deficiencies (where texture is biased over shape) observed in DCNs Geirhos et al. (2019b); Brendel and Bethge (2019b); Baker et al. (2018); Long and Konkle (2018); Laskar et al. (2018) by emphasizing extended contour processing through recurrent horizontal interactions Loffler (2008); Elder (2018); Elder et al. (2018).

On the other hand, we are intrigued by the superior ability of parameter-heavy feedforward state-of-the-art DCNs in performing perceptual grouping with identical performance to recurrent DCNs. Further analyses of strategies learned by these models will inform future studies of cortical processing to search for currently unexplored mechanisms achievable by nuanced feedforward computations.

In summary, our work highlights the efficacy of recurrent computations in learning highly compact, generalizable solutions to perceptual grouping problems such as the proposed MarkedLong and PathFinder challenges. Our bio-inspired V1Net recurrent unit learns the most compact generalizable solutions to the above grouping problems, and it evolves interpretable internal states reflecting the model's learned low-level grouping strategies. These characteristics of V1Net make it suitable for sample- and parameter-efficient on-device computer vision as well as computational modeling of primate cortical processing.

3.4.4 Acknowledgements

Chapter 3, in part, is obtained from the following arXiv preprint. I, the dissertation author, was the primary investigator and author of this material.

Veerabadran, V. and de Sa, V. R. (2020). Learning compact generalizable neural representations supporting perceptual grouping. *arXiv preprint arXiv:2006.11716*

We thank Drew Linsley and Thomas Serre for sharing with us the PathFinder challenge dataset. This work was supported by the Kavli Symposium Inspired Proposals award from the Kavli Institute of Brain and Mind, the Department of Cognitive Science at UC San Diego, Cloud TPU resources made available by Google through the TensorFlow Research Cloud program, Social Sciences Computing Facility at UC San Diego, Nautilus compute cluster supported by Pacific Research Platform, and the Titan V GPU donated by NVIDIA Corporation.

Chapter 4

Adaptive recurrent vision performs zero-shot computation scaling to unseen difficulty levels

4.1 Introduction

Recurrent Neural Networks (RNNs) have emerged as a powerful tool for solving machine reasoning tasks, demonstrating impressive performance across a variety of tasks that require processing of sequential inputs – see Chapter 3 for a discussion of such advantages of RNNs. While recurrent processing is valuable for time-varying inputs, it can also be useful for static inputs as recurrent networks have the ability to scale computation to varying difficulty levels by varying the number of recurrent iterations. This can be helpful when different problems from the same task family exhibit significant variations in complexity (consider mazes, for example). However, conventional RNNs such as V1Net from Chapter 3 struggle to automatically generalize across different difficulty levels due to their fixed computational graph, requiring retraining, fine-tuning, and/or human intervention to pick the number of recurrent iterations needed. This limitation hampers their practical utility in real-world scenarios, since the span of difficulty levels seen during training is finite.

Human reasoning, in contrast, is characterized by a highly adaptible use of computation - arbitrarily more computation can be used for more difficult tasks. Despite the evidence for such scaling in human computing there is limited work in the literature analyzing Adaptive RNNs (AdRNNs) that, on a sample-by-sample basis, automatically decide when to stop. Graves introduced Adaptive Computation Time (ACT), a mechanism to automatically stop computation, by learning to generate a scalar halting probability (Graves, 2016). However, similar to subsequent work (Banino et al., 2021), it was evaluated on either simple tasks such as parity checking, or language tasks. Few studies examine the training and evaluation of these AdRNNs on visual reasoning tasks, which pose unique challenges due to the redundant and high-dimensional nature of visual data. Furthermore, extrapolating to harder instances within-task has not been sufficiently studied, with Banino et al. (2021) evaluating such zero-shot performance only on the simple parity task.

In this Chapter, we study the problem of computation scaling in recurrent vision RNNs, with an emphasis on zero-shot extrapolation to harder/larger problems within the same task. The ability to handle unseen difficulty levels without fine-tuning, or human intervention, enables more robust and adaptable computer vision systems, which are crucial for real-world applications. We investigate the effectiveness of AdRNNs on two challenging, publically available, visual reasoning tasks based on curve tracing and route segmentation, namely PathFinder (introduced by Linsley et al. (2018), which we also used in Chapter 3) and Mazes (introduced by Schwarzschild et al. (2021)).

The contributions of this work are as follows:

- We combine Convolutional RNNs with an adaptive computation method of Graves (2016) producing Adaptive ConvRNNs (AdRNNs) that are capable of learning a downstream task simultaneously while also learning to scale their computational steps as per input image/task requirements.

- We introduce LocRNN, a high performing recurrent architecture inspired by prior computational models of recurrence in biological vision.

- We show that AdRNNs learn to dynamically halt processing early (or late) to solve easier (or harder) problems when the train- and test-difficulty levels are matched on complex visual reasoning problems inspired by stimuli used in prior cognitive science research.

- During test time we introduce a previously unseen harder difficulty level. We evaluate AdRNNs on these new difficulty levels and show that they zero-shot generalize to more difficult problem settings not shown during training by dynamically increasing the number of recurrent iterations at test time well beyond the number of recurrent steps used during training.

4.2 Related Work

Our work is relevant to the visual routines literature introduced by Ullman (1984) and further reviewed elaborately by Roelfsema et al. (2000). The core idea of visual routines that make it relevant to our studied question of task extrapolation is the flexible sequencing of elemental operations resulting in a dynamic computational graph, which make RNNs a natural approach to solve such tasks.

Prior research has proposed mechanisms for RNNs that learn the amount of recurrent computational steps required. Following are two particularly relevant attempts in prior literature in this area: Graves (2016) developed the Adaptive Computation Time (ACT) method to train RNNs on Natural Language Processing tasks with a sigmoidal halting unit that determines when to halt recurrent processing, and Banino et al. (2021) extend upon this pioneering work by taking a probabilistic approach to halting and introducing a geometric prior-based specification of computational budget. However, neither of these studies applied adaptive computation to vision RNNs (that are significantly trickier to optimize). Also the prior studies did not attempt to examine computation scaling under zero-shot generalization to harder difficulty levels. Additionally, Eyzaguirre and Soto (2020) developed a version of ACT applied to visual reasoning problems, however, their observations on the CLEVR dataset (in which the issues related to object diversity are highlighted by Kim et al. (2018)) do not explore generalization to new difficulty levels which is central to our results.

In terms of generalization from easy to hard vision problems, our work is relevant to Schwarzschild et al. (2021) and Bansal et al. (2022) where the authors evaluate the application of recurrent neural networks to generalize from easier to harder problems. Our work extends and differs from their intriguing study of recurrence in task extrapolation in the following ways: 1) For recurrent networks used in their study, human intervention is required to specify the number of recurrent computational steps during the testing phase. 2) Their work explores sequential task extrapolation in general with abstract problems such as Prefix Sum and solving Chess Puzzles while our work extends it to particularly focus on extrapolation in visual task learning. 3) We present evaluation on the Pathfinder challenge (Linsley et al., 2018), a relatively more large-scale visual reasoning task shown to be very challenging (Tay et al., 2020), the design of which dates back to (Jolicoeur et al., 1986). 4) Schwarzschild et al. (2021) and Bansal et al. (2022) implement only the most straightforward form of recurrence realized by weight-tying, i.e., they only evaluate ResNets with weight sharing across residual blocks (albeit with a novel training scheme in the follow up study (Bansal et al., 2022)). In addition to such recurrent ResNets, we present analyses with highly sophisticated recurrent architectures specialized for recurrent image processing. 5) We introduce LocRNN, a high performing recurrent architecture based on prior computational models of cortical recurrence.

On the design of recurrent architectures, our work is loosely related to (Eigen et al., 2013), (Pinheiro and Collobert, 2014), (Veerabadran and de Sa, 2020) and (Liao and Poggio, 2016b) which discuss the role of weight sharing in feedforward networks to produce recurrent processing. We are interested, however, in designing new specialized recurrent architectures that play a role both in human and machine vision. While there exist more such recurrent architectures informed by neuroscience such as Linsley et al. (2018); Nayebi et al. (2022), we find these architectures to be quite difficult to interpret and unstable to train (based on in-difficulty performance evaluation included in the Supplementary). Our proposed LocRNN architecture in comparison is an elegant and easy-to-interpret implementation of recurrence that is stable to train. Our work is also relevant to prior work on modeling speed-accuracy tradeoffs observed during visual perception (Spoerer et al., 2020).

4.3 Methods

4.3.1 Datasets

For evaluating the ability of various models in exhibiting task extrapolation, we curate two challenging visual reasoning tasks, Mazes and PathFinder, with instances at multiple parametric difficulty levels. Both tasks involve the visual routines of marking and curve tracing (Ullman, 1984). These datasets are inspired by prior visual psychophysics research where such tasks were used abundantly to estimate the cognitive and neural underpinnings of sequential visual processing, such as incremental grouping, structure and preference of lateral connections, etc. (Jolicoeur and Ingleton, 1991; Ullman, 1984; Li et al., 2006; Roelfsema, 2006). In the following subsections we describe the specifics of our tasks.

PathFinder challenge – curve tracing

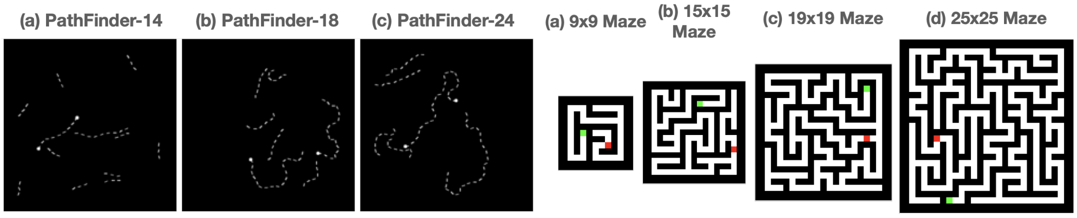


Figure 4.1. Representative examples from PathFinder and Mazes datasets. Left: (a) Positive PathFinder-9 (b) Negative PathFinder-18 (c) Positive PathFinder-24; Right: (a) 9×9 mazes, (b) 15×15 mazes, (a) 19×19 mazes, (a) 25×25 mazes

Task description: In the PathFinder task introduced by Linsley et al. (2018), models are trained to identify whether two circular disks in an input stimulus form the two ends of a locally connected path made up of small "segments". Each image consists of two main long connected paths P_0 and P_1 made up of locally aligned segments as well as shorter distractor paths. Each image also contains two circular disks which are placed at two of the 4 possible endpoints of P_0 and P_1. Images that contain a disk on both ends of the same path are classified as positive, and those containing a disk on endpoints of different paths are classified as negative. Examples are shown in Figure 4.1.

Difficulty levels: Pathfinder is designed at different difficulty levels parameterized by the length (number of segments) of the paths P_0 and P_1 mentioned above. The easiest version uses paths that are 9 segments long (PathFinder-9), while the medium and hard versions contain paths that are 14 (PathFinder-14) and 18 (PathFinder-18) segments long respectively (see example images in the Appendix). This dataset consists of a total of 800,000 RGB images at each difficulty level, each with a spatial resolution of 160 × 160 pixels. There are an equal number of positive and negative instances at each difficulty level. We use 700,000 images for training and 100,000 images as the test set from each difficulty level. We combined these datasets to create a more challenging dataset with varying levels of difficulty, which we refer to as PathFinder-Mixed. To evaluate the zero-shot difficulty extrapolation on PathFinder in Sec. 4.5.2 we generated 100,000 images each with contour lengths 21 and 24 respectively; we call these PathFinder-21 and PathFinder-24.

Evaluation criteria: Since this is a classification challenge, we use accuracy, i.e. *% correct on test-images* as the evaluation metric to rank model performance on PathFinder. Model architectures receive an input image and process it via a stack of standard convolution/recurrent-convolution layers followed by a classification two-class readout. Since this is a binary classification challenge with balanced classes, chance performance is 50% for random predictions.

Mazes challenge - route segmentation

Task description: Human beings are adept at solving mazes, a task that requires application of a similar serial grouping operation like PathFinder in order to discover connectivity from a starting point to the final destination of the maze amidst blocking walls. For evaluating model performance on solving mazes of varying difficulty, we use the publicly available version of the Mazes challenge developed by Schwarzschild et al. (2021). They implemented this Mazes challenge as a binary segmentation problem where models take $N \times N$ images of square-shaped mazes as input with three channels (RGB) with the start position, end position, permissible regions and impermissible regions marked in the image. The output produced by models is a binary segmentation of the route discovered by the model from the start to the end position.

Difficulty levels: The Mazes challenge has been designed at several difficulty levels, each difficulty level is parameterized by the size of the square grid that the maze is designed into. We use maze datasets of grid sizes 9×9 and 15×15 for training. Each dataset consists of 50,000 training images and 10,000 test images that are guaranteed to not overlap. The spatial resolution of 9×9 maze images is 24×24 pixels and that of 15×15 maze images is 36×36 pixels. Similar to PathFinder-Mixed, we combine 9×9 and 15×15 grid sizes to form Mazes-Mixed. We also constructed larger mazes with grid sizes 19×19 (with spatial resolution 44×44 pixels) and 25×25 (with spatial resolution 56×56 pixels) to evaluate a model's ability to extrapolate to difficulties not seen during training. As above with PathFinder's extrapolation evaluation, these novel difficulty mazes were only used during testing and were not in any form used in the training or hyperparameter optimization process.

Evaluation criteria: Mazes is a segmentation challenge and hence, one could potentially consider partially correct routes during evaluation (for example with average of per-pixel accuracy). However this is less strict than giving each image a single binary score reflecting if *all* pixels are labeled correctly. Evaluation criteria for mazes is hence the total *% of test-set mazes completely accurately solved* at a given difficulty level.

4.3.2 Model architectures and training

Implementations of adaptive computations evaluated on task extrapolation

In this section, we describe our choice of recurrent architecture designs that we use to study the behavior of adaptive computation and task extrapolation in deep learning. We process images (denoted as $\mathbf{X} \in \mathbb{R}^{c,h,w}$) in three stages that are common to all models we evaluate in this study. These three stages are as follows: (1) an input convolution layer (denoted as `input(.)`)

that operates on the image directly.

$$\mathbf{h_0} = \texttt{ReLU}(\texttt{input}(\mathbf{X})) \tag{4.1}$$

The output from this stage $\mathbf{h_0} \in \mathbb{R}^{d,h,w}$ is fed as input to the following recurrent block in (2).

$$\mathbf{h_t} = \mathbf{r}(\mathbf{h_{t-1}}, \mathbf{h_0}), t \in [1, t_{train}] \tag{4.2}$$

where $\mathbf{r}(.)$ is the recurrent block. t_{train} is a training hyperparameter indicating the maximum number of timesteps for unrolling $\mathbf{r}(.)$ during training. This block consists of a sequence of convolution layers applied in an iterative manner for any arbitrary number of timesteps. This is the sub-part of our model that is capable of performing adaptive computations. (The specific architecture of these convolution blocks constitute different implementations of recurrent operations described further below.) (3) a readout layer containing a block of convolution and pooling operations that produce the desired output from our network.

$$\hat{\mathbf{y}}_\mathbf{t} = \texttt{readout}(\mathbf{h_t}), t \in [1, t_{train}] \tag{4.3}$$

While we keep the input and readout layer architectures the same for all models we evaluate, their intermediate recurrent blocks have different architectures (corresponding to their respective recurrent cells). We explore three different implementations of recurrent computations which are used in the second, recurrent block of models. Our first choice as the intermediate feature processing block consists of a residual network with weight tying across all layers; we refer to this model as R-ResNet-30. Next, we study the performance of the following specialized convolutional recurrent units from prior work: horizontal convolutional GRU (hConvGRU) and its stable variant (Linsley et al., 2018, 2020) and a convolutional Gated Recurrent Unit (ConvGRU) (Ballas et al., 2015) with LayerNorm (ConvGRU does not converge in the absence of LayerNorm). Third, we design a novel recurrent cell based on prior computational models of cortical recurrent processing (Li et al., 2006); this model equips the biologically inspired design choices of long-range lateral interactions, gating, and a separate population of interneurons. This model is referred to as LocRNN and is described in the following Sec. 4.3.2. Importantly all models are matched for trainable parameters.

Combining ConvRNNs with Adaptive Computation Time (ACT)

The central theme of our work is to show that RNNs can flexibly adapt (or scale) their computation according to input requirements. We achieve this ability by combining ConvRNNs with an adaptive computation mechanism based on Graves (2016) called Adaptive Computation Time (ACT). A difference between our work and ACT is that our visual reasoning task involves static inputs (i.e. sequence of length 1) whereas Graves (2016) deals with variable-length sequences and learns adaptive processing of each token. Owing to this difference, our halting mechanism is similar to ACT applied to a 1-token input sequence.

The key idea of ACT is to introduce a separate "halting mechanism" that learns to control the number of recurrent computation steps dynamically, conditioned on each input example's processing. In addition to producing the next recurrent state, an RNN equipped with ACT also produces a scalar value called the "halting score" for each computation step. In addition to the next hidden state computed using Equation 4.2, we generate a scalar halting score p_t at each step using a learnt convolution layer ($\texttt{halt\_conv}(.)$) that is shared across timesteps:

$$p_t = \sigma(\texttt{max\_pool}(\texttt{halt\_conv}(\mathbf{h_{t-1}}))) \tag{4.4}$$

The RNN treats the cumulative sum of the halting scores up to timestep t (P_t described below) as a quantity used to determine whether processing is terminated at that timestep.

$$P_t = \sum_{t'=1}^{t} p_{t'}$$

The RNN keeps track of the accumulated halting scores and checks if the cumulative sum of the halting scores at each step reaches a predefined threshold $(1-\varepsilon)$. If the threshold is reached, the computation stops (i.e., $p_t = 0\ \forall t > t_{halt}$) where the cumulative halting score threshold is reached at timestep $t_{halt} = \min\{t : P_t >= (1-\varepsilon)\}$. The adaptive hidden state is then computed as a weighted average of the hidden states up to t_{halt}, scaled by the halting scores at each time step. The readout is then applied to this final adaptive hidden state to produce the ACT task prediction, $\hat{\mathbf{y}}_{act}$:

$$\mathbf{h}_{act} = \sum_{t=1}^{t_{halt}} p_t \cdot \mathbf{h_t} \tag{4.5}$$

$$\hat{\mathbf{y}}_{act} = \texttt{readout}(\mathbf{h}_{act}) \tag{4.6}$$

The abovementioned ConvRNN combined with ACT is trained to optimize both the downstream task loss ($||\mathbf{y} - \hat{\mathbf{y}}_{act}||_p$) and an auxiliary ‘ponder cost’ corresponding to the halting mechanism that encourages models to use the fewest number of recurrent step to process an input example. Ponder cost is computed as the cumulative halting score until timestep $t_{halt} - 1$, maximizing this term encourages (or minimizing its negative as shown below) completing the task as quickly (with as few recurrent steps) as possible. The following final objective function is optimized to train the full model containing ConvRNNs with ACT where τ is a hyperparameter.

$$\mathscr{L} = \sum_{i=0}^{i=||\mathscr{D}||} \frac{1}{||\mathscr{D}||} ||\mathbf{y}^i - \hat{\mathbf{y}}^i_{act}||_p - \tau \sum_{t=1}^{t^i_{halt}-1} p^i_t \tag{4.7}$$

Formulation of LocRNN

We note that prior work has explored the development of recurrent architectures tailored for vision. Here, we introduce a similar but highly expressive recurrent architecture designed based on a computational model of iterative contour processing in primate vision from Li (1998). This model is an ODE-based computational model of interactions between cortical columns in primate area V1 mediated by “lateral connections” local to a cortical area.

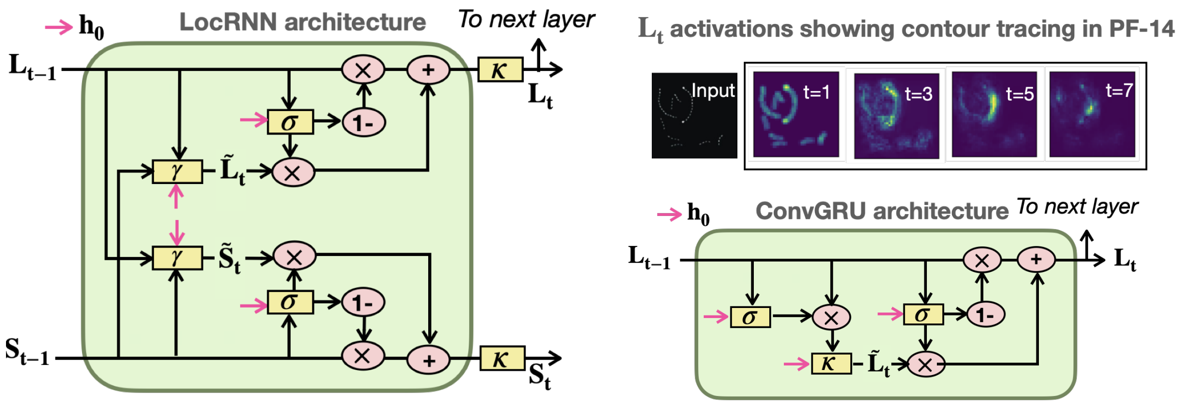


Figure 4.2. Contrasting the architectures of LocRNN (left) and ConvGRU (bottom-right). $\mathbf{h_0}$ is shown as a magenta arrow and does not change across timesteps. As illustrated also in this figure, LocRNN's interneuron activations ($\mathbf{S_t}$) are not passed to the next layer. ConvGRU on the other hand uses a single uniform neural population. (Top-right) shows a visualization of LocRNN's $\mathbf{L_t}$ activations as they perform contour tracing to solve an input image from PathFinder-14 (PF-14).

By discretizing the continuous-form ODE dynamics of processing units from Li (1998), we arrived at an interpretable and powerful set of dynamics for "LocRNN". As in ConvGRU, the effective receptive field of LocRNN output neurons increases linearly with recurrent timesteps.

Unlike ConvGRU, the LocRNN's hidden state is composed of two neural populations, **L** and **S**; the activity in these populations are referred to as $\mathbf{L_t}$ and $\mathbf{S_t}$ at timestep t respectively. These two populations are motivated by the x (excitatory) and y (inhibitory) neurons in Li (1998). However we found that restricting the signs of their weights (to reflect exclusively excitatory and inhibitory connections) led to less stable behavior. On the other hand, retaining the interneuron property (local computation not projecting out for downstream processing) of the **S** population performed better than using a single uniform population. The following equations illustrate the working of ACT and how the readout is applied in LocRNN.

$$p_t = \sigma(\texttt{max\_pool}(\texttt{halt\_conv}(\mathbf{L_{t-1}}))) \tag{4.8}$$

$$\hat{\mathbf{y}}_{act} = \texttt{readout}\left(\sum_{t=1}^{t_{halt}} p_t \cdot \mathbf{L_t}\right) \tag{4.9}$$

Initially, both $\mathbf{L_0}$ and $\mathbf{S_0}$ populations are set to a tensor of zeros with the same shape as

$\mathbf{h_0}$, a $4d$ tensor of shape (`batch_size`×`channels`×`height`×`width`).

L and S update gates $\mathbf{G_t^L}$ and $\mathbf{G_t^S}$ (same shape as $\mathbf{L}$ and $\mathbf{S}$) are computed as functions of the input and current hidden states $\mathbf{L_{t-1}}$ and $\mathbf{S_{t-1}}$ using 1x1 convolutions $\mathbf{U}_*$.

$$\mathbf{G_t^L} = \sigma(LN(\mathbf{U_L} * \mathbf{h_0}) + LN(\mathbf{U_{L \to L}} * \mathbf{L_{t-1}})) \tag{4.10}$$

$$\mathbf{G_t^S} = \sigma(LN(\mathbf{U_S} * \mathbf{h_0}) + LN(\mathbf{U_{S \to S}} * \mathbf{S_{t-1}})) \tag{4.11}$$

Each of the 4 types of lateral connections are modeled by convolution kernels $\mathbf{W_{L \to L}}$, $\mathbf{W_{L \to S}}$, $\mathbf{W_{S \to S}}$, and $\mathbf{W_{S \to L}}$ respectively of shape $d \times d \times k \times k$ where d is the dimensionality of the hidden state and k represents the kernel spatial size.

$$\mathbf{\tilde{L}_t} = \gamma(\mathbf{W_L} * \mathbf{h_0} + \mathbf{W_{L \to L}} * \mathbf{L_{t-1}} + \mathbf{W_{S \to L}} * \mathbf{S_{t-1}}) \tag{4.12}$$

$$\mathbf{\tilde{S}_t} = \gamma(\mathbf{W_S} * \mathbf{h_0} + \mathbf{W_{L \to S}} * \mathbf{L_{t-1}} + \mathbf{W_{S \to S}} * \mathbf{S_{t-1}}) \tag{4.13}$$

Once the long-range lateral influences are computed and stored in $\mathbf{\tilde{L}_t}$ and $\mathbf{\tilde{S}_t}$, these are mixed with the previous hidden states using the gates computed in Eq. 4.10 and 4.11. These hidden states are then passed on to subsequent recurrent iterations where even longer-range interactions occur (as time increases).

$$\mathbf{L_t} = \kappa(LN(\mathbf{G_t^L} \odot \mathbf{\tilde{L}_t} + (1 - \mathbf{G_t^L}) \odot \mathbf{L_{t-1}})) \tag{4.14}$$

$$\mathbf{S_t} = \kappa(LN(\mathbf{G_t^S} \odot \mathbf{\tilde{S}_t} + (1 - \mathbf{G_t^S}) \odot \mathbf{S_{t-1}})) \tag{4.15}$$

In the above equations, $LN()$ stands for Layer Normalization (Ba et al., 2016), and the nonlinearities γ and κ are both set to ReLU. One of the differences between LocRNN and ConvGRU (which we also evaluate) is the presence of the interneuron $\mathbf{S}$ population in the former.

4.4 Training and implementation details

All architectures we evaluated within a task were matched in terms of the number of parameters. The input convolutional layer's kernel size is 7 × 7. Number of channels used by the model remains unchanged across layers and is determined per-model for matching overall number of trainable parameters across models. For PathFinder, we fix the number of channels to be 32 for LocRNN, 21 for hConvGRU and ConvGRU, and 64 for ResNet-30 with a filter size of 9x9 in the intermediate recurrent layers. For Mazes, we fix the number of kernels (d) to be 128 for LocRNN, ConvGRU, hConvGRU, and 100 for ResNet-30 & R-ResNet-30 and the kernel size is fixed to be 5x5. Our readout layers for classification (PathFinder) contain a global average pooling layer followed by a fully-connected layer with output dimensionality of 1 producing the classification logit. We apply binary cross-entropy loss on the logit to train models on PathFinder. On Mazes, we use a 1 × 1 convolution with 1 output channel to produce a binary segmentation map. We use pixel-wise binary cross-entropy to train models on Mazes. These readout layers are used uniformly for all architectures evaluated.

On Mazes training minibatch size is set to 64 images (and inference batch size of 50 images) and a learning rate schedule starting with warmup followed by step learning rate decay as indicated in Schwarzschild et al. (2021) for 50 total epochs of training. On PathFinder, we set the training minibatch size to 256 images and a constant learning rate of 1e-4 for all models for a total of 20 epochs of training. All models were trained on NVIDIA RTX A6000 GPUs and implemented using PyTorch (Paszke et al., 2017).

4.5 Results

4.5.1 Adaptive RNNs scale their computation as a function of input difficulty

In this section, we report our observations from training AdRNNs on a mixture of difficulty levels from PathFinder and Mazes & evaluating them on a held-out set of images from the same mixture of difficulty levels of PathFinder and Mazes. We created one training dataset for each of these challenges by combining input samples from multiple "easy" difficulty levels. Combining input samples from multiple difficulty levels increases the diversity of computational requirements during training and helps in developing models that do not degenerate to using the same computational steps for all input samples.

As described in Section 4.3.1, we created a PathFinder training set of images and corresponding labels by sampling an equal number of images from three difficulty levels: PathFinder-9, PathFinder-14, and PathFinder-18. Similarly, we created a Mazes training set of images and ground-truth segmentation labels by sampling an equal number of images from two difficulty levels: 9x9 mazes and 15x15 mazes.

On held-out test sets that matched the difficulty level of the above-described training data, we tested whether models were able to scale their recurrent computational steps as a function of the input difficulty level. We show the results from this evaluation in Table. 4.1. **We observed that both the sophisticated AdRNNs tested (ConvGRU and LocRNN) were able to generalize to the held-out set.** We also observed that variants of horizontal GRU (Linsley et al., 2018) and Linsley et al. (2020) models generalized to the (within-difficulty) held-out set without using ACT. The simpler recurrent network without gating, weight-tied R-ResNet-30, was unable to learn on PathFinder and on Mazes highlighting the importance of specialized operations such as gating and backpropagation through time. The presence of skip connections between R-ResNet-30's recurrent blocks could still not match the expressivity of the specialized RNNs.

If ACT is working as expected, we must observe that AdRNNs that learn the task dynamically use less compute to solve easy examples and more compute (more recurrent iterations) to solve harder examples. To check for this trend, we analyzed the number of steps chosen by the model before halting for each example from the validation sets of PathFinder and Mazes; with varying contour lengths and maze sizes respectively. These results are shown by the cool colors in Figure 4.3 for PathFinder and in the Supplementary for Mazes. As is clearly observable from the trend in these Figures, **examples that we consider as harder (longer contours in PathFinder and larger mazes) are assigned a higher number of recurrent computation steps by ACT than easy examples.** Hence we show that AdRNNs obtained by combining ConvRNNs with ACT training are capable of learning both PathFinder and Mazes in addition to learning to adapt their recurrent computational steps as a function of input example difficulty. To the best of our knowledge, we are the first to show the above result for visual tasks inspired by stimuli used in prior cognitive science research.

Table 4.1. Accuracies (%) ↑ of models on the two visual reasoning tasks. Chance performance is 50% for pathfinder and 0% for Mazes.

	PathFinder-Mixed (%)	Mazes-Mixed (%)
ResNet-30	50.41	0.0
R-ResNet-30 (ACT)	49.37	0.0
Linsley et al. (2020)	50.0	78.33
hConvGRU $t_{inference} = t_{train}$	89.66	99.69
hConvGRU (stable halting)	89.65	99.54
ConvGRU (ACT)	95.26	98.4
LocRNN (ACT) (ours)	97.13	98.4

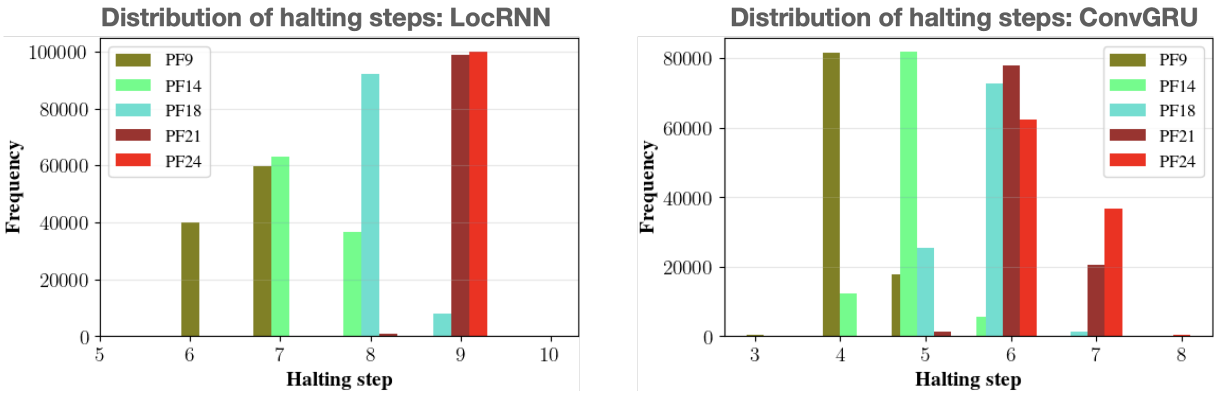


Figure 4.3. Distributions of halting steps across samples in each validation set of PathFinder. Computation appears to scale to match difficulties of the datasets for both LocRNN (left) and ConvGRU (right) models. 9-length contours typically halt after 4-6 steps, while 24-length contours can take up to 9 steps. The red bars show the distribution in the extrapolation datasets.

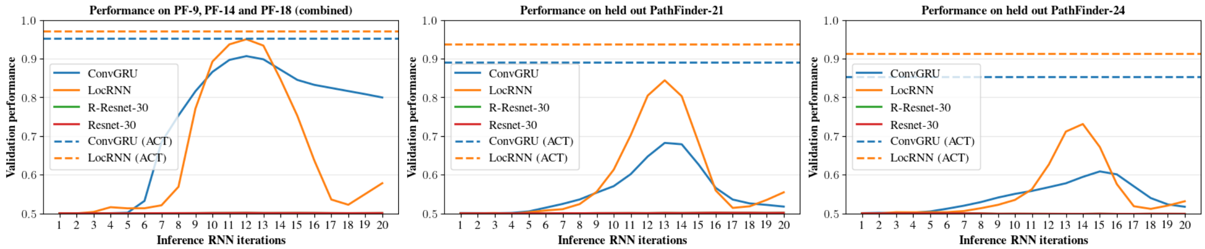


Figure 4.4. Generalization to novel test-time difficulty levels of PathFinder. Vanilla RNNs (solid lines) and AdRNNs (dashed) trained on 12 iterations (a) within-difficulty (b) extrapolating to PathFinder-21 (c) extrapolating to PathFinder-24

4.5.2 Adaptive RNNs generalize to novel difficulty levels by scaling their computation

In typical circumstances where RNNs are expected to solve instances that are more difficult, human intervention in the form of specification of the number of recurrent iterations is commonplace (Schwarzschild et al., 2022). This approach is both laborious and expensive to perform as it requires training and/or evaluation on new test sets with various settings of the number of recurrent timesteps. Additionally, this process needs to be repeated at every occasion of new incoming test data that is potentially of increased difficulty. In the previous section, we showed the ability of AdRNNs in learning to solve PathFinder and Mazes while also demonstrating that they assign computation as a function of input difficulty on a test-set with matched task difficulty level as the training set. Here we are interested in testing whether AdRNNs are able to extrapolate (by using more recurrent computational steps compared to training) in complex reasoning tasks. We constructed two additional datasets each in the PathFinder and Mazes families to test these adaptive models' ability to extrapolate, as mentioned in Section 4.3.1. While AdRNNs are trained on the training difficulty levels for a maximum of t_{train} iterations, **we evaluated them on these new (more difficult) datasets at inference with a maximum of $t_{inference} \geq t_{train}$ recurrent iterations where we refer to AdRNNs as operating in their extrapolation phase.** We report the results from this evaluation in Table. 4.2 and in Fig. 4.4.

Table 4.2. Accuracies (Mean % ± SEM) ↑ on extrapolation datasets. Chance performance is 50% for pathfinder and 0% for Mazes.
*hConvGRU training converged only on 1 out of 3 seeds (which were chosen randomly for all models here) on PathFinder, corresponding results above show this converged model's performance.

	PathFinder-21 (%)	PathFinder-24 (%)	Mazes-19 (%)	Mazes-25 (%)
ResNet-30	50.0	50.0	0.	0.
R-ResNet-30 (ACT)	50.0	50.0	0.	0.
Linsley et al. (2020)	50.0	50.0	2.93	0.01
hConvGRU $t_{inference} = t_{train}$ *	64.21	58.35	16.2 ± 2.66	5.29 ± 0.26
hConvGRU (stable halting)	50.0	50.0	50.26 ± 6.04	21.36 ± 2.82
ConvGRU (ACT)	82.63 ± 4.84	74.14 ± 6.52	75.1 ± 11.96	46.93 ± 4.2
LocRNN (ACT) (ours)	**92.89 ± 0.9**	**85.81 ± 5.57**	**86.83 ± 2.94**	**49.99 ± 4.48**

Adaptive RNNs generalize to novel difficulty levels by scaling their computation: Our evaluation of AdRNNs trained with ACT on novel harder difficulty levels shows that as expected, AdRNNs have learned to use the optimal number of recurrent computational steps required for achieving strong generalization to the novel difficulty levels on a per-image level. On PathFinder, as seen in Figure 4.3, most instances from PathFinder-21 and PathFinder-24 take 9 steps in LocRNN and up to 7 steps in ConvGRU, higher than the number of steps used for easy training difficulty examples. In Fig 4.5 we visualize the relationship between mazes' difficulty and the number of recurrent iterations used by ACT (on a per-instance level). We observe that maze difficulty (length of the ground-truth route in pixels) and the number of ACT iterations are strongly positively correlated. Notably, AdRNNs trained with ACT choose to make t_{halt} during inference on longer mazes greater than t_{halt} used on the shorter training mazes.

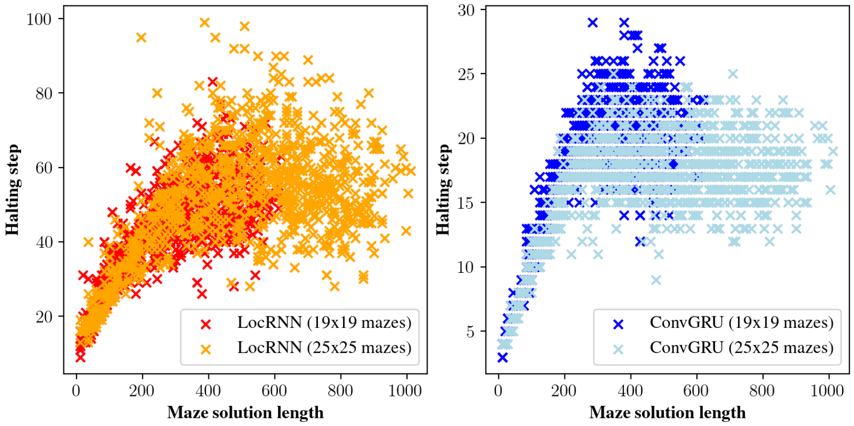


Figure 4.5. Relationship between halting step and difficulty level of Mazes for the extrapolation evaluation. Here we visualize the relationship between the halting step and solution length of mazes at a per-instance level. We note a strong positive correlation between the difficulty level (length of maze solution segmentation) and halting step used by LocRNN (left) and ConvGRU (right) AdRNNs.

4.5.3 AdRNNs outperform halting in hConvGRU based on stability of hidden-state dynamics

Using stability in hidden-state space as an alternative way to perform halting during extrapolation, we evaluate hConvGRU on extrapolation. For this method, we halted processing in hConvGRU when the mean absolute difference between two subsequent states ($||\mathbf{h_t} - \mathbf{h_{t-1}}||$) reduced to less than a tenth of the difference between the first two states (heuristic for stability). This method referred as hConvGRU (stable halting) in Tables 4.1 and 4.2 shows differences in generalization across our datasets. **AdRNNs perform better than hConvGRU (stable halting) on both datasets**; especially, stable halting completely fails to show generalization to PathFinder-21 and -24. We hypothesize this method to work suboptimally for the following reasons:

- Stability-based halting requires defining a hand-engineered heuristic on how much change in the output is considered small enough to halt. In typical segmentation tasks like Mazes, the network output for the initial few timesteps is highly stable in making nonsensical predictions (predicting all pixels as the negative class) and thus, heuristics need to identify an inflection point in the output trajectory where meaningful predictions start to emerge and stabilize. In the absence of ground truth information, one cannot pick a heuristic that generalizes to unseen data.

- Learnable halting makes less assumptions about the hidden state's properties, and hence doesn't enforce hard constraints such as stability to be satisfied by training. Some, but not all RNNs, have stable hidden states wherein the network response stops changing after reaching an attractor. Linsley et al. (2020) argue that RNNs that are expressive have an intrinsic inability to learn stable hidden states. For RNNs to be stable, their hidden state transformation needs to model a contractive mapping (Miller and Hardt, 2018; Pascanu et al., 2013). That is, the recurrent transition function F satisfies the following inequality: $||F(h_t) - F(h_{t-1})||_p < \lambda||h_t - h_{t-1}||_p$. RNNs with stable hidden states that satisfy the above inequality are quite difficult to train on challenging problems in practice. Even when stable models perform comparably to unstable models as in (Miller and Hardt, 2018), the authors show unstable models' advantages such as performance improvements in the short-time horizon and lesser vanishing gradient issues.

4.6 Discussion

The advantage of using recurrent networks for processing static inputs adaptively, particularly in order to zero-shot generalize to new difficulty levels is understudied. In this work, we show that deep convolutional networks with intermediate recurrent blocks (ConvRNNs operating on image features) can be combined with an adaptive computation technique to learn to dynamically process different input examples based on a per-instance difficulty level. We combine ConvRNNs with a learnable halting mechanism that is based on Graves (2016) to produce AdRNNs. We evaluated diverse implementations of recurrence in increasing level of sophistication/complexity, R-ResNet-30 with weight-tying, ConvGRU and hConvGRU with gating, and LocRNN with two separate populations of horizontally connected units (one of which are interneurons) and gating on two challenging visual reasoning problems, PathFinder and Mazes, which are generated at various levels of difficulty. First, we showed that only the specialized RNNs (hConvGRU) and AdRNNs (LocRNN and ConvGRU) are capable of learning PathFinder and Mazes. Second, these AdRNNs trained with ACT are learning to dynamically use less (or more) recurrent computational steps for easy (or hard) PathFinder and Maze problems respectively as discussed in Section. 4.5.1. More interestingly when the difficulty level of the test set was increased relative to training difficulty levels, AdRNNs generalized to these harder instances in a zero-shot manner by allocating more recurrent computation than was ever used during training. They also outperform stability heuristics-based adaptive recurrent networks. Our work empirically shows this hypothesized advantage of using adaptive recurrent processing on static-image tasks for the first time to the best of our knowledge.

In addition to adaptively scaling computation according to the need, the AdRNNs can also solve the problems more efficiently, choosing to stop at earlier times than the non-adaptively trained RNNs (compare halting times in Fig 4.3 with performance curves in Fig 4.4) with human arbitrarily chosen training iterations. Thus, just as computation can be scaled to unseen problem difficulties without human intervention, the training iterations required can be (better) discovered automatically.

4.7 Acknowledgements

Chapter 4, in part, is obtained from the following publication. I, the dissertation author, was the primary investigator and author of this material.

Veerabadran, V., Ravishankar, S., Tang, Y., Raina, R., and de Sa, V. R. (2023b). Adaptive recurrent vision performs zero-shot computation scaling to unseen difficulty levels. In *Thirty-seventh Conference on Neural Information Processing Systems*

We thank Garrison W. Cottrell, Michael C. Mozer, Michael L. Iuzzolino, Pradeep Shenoy and Saurabh Singh for helpful discussions and insights on this work. This work was supported by: (1) NSF CRCNS Award No: 2208362, (2) Sony Faculty Innovation Award, and (3) Kavli Symposium Inspired Proposal. We thank the Social Sciences Computing Facility at UCSD for their help with the model training and inference infrastructure.

Chapter 5

Subtle adversarial image manipulations influence both human and machine vision

5.1 Introduction

Artificial neural networks (ANNs) have produced revolutionary advances in machine intelligence, from image recognition (Krizhevsky et al., 2012) to natural language understanding (Collobert et al., 2011) to robotics (Lee et al., 2020). The inspiration for ANNs was provided by biological neural networks (BNNs) (von Neumann, 1958). For instance, convolutional ANNs adopt key characteristics of the primate visual system, including its hierarchical organization, local spatial connectivity, and approximate translation equivariance (Fukushima, 1980b; Fukushima and Miyake, 1982). The historical relationship between ANNs and BNNs has also led to ANNs being considered as a framework for understanding biological information processing. Visual representations in ANNs are strikingly similar to neural activation patterns in primate neocortex (Khaligh-Razavi and Kriegeskorte, 2014; Yamins et al., 2014; Cadieu et al., 2014; Kriegeskorte, 2015; Yamins and DiCarlo, 2016; Serre, 2019), and ANNs have been successful in accounting for a range of behavioral phenomena in human perception, learning, and memory (Berardino et al., 2017; Kar et al., 2019; Kim et al., 2021; Kubilius et al., 2016; Mozer, 1991; Rumelhart et al., 1986; Wenliang and Seitz, 2018).

However, qualitative differences exist between human and machine perception. Architecturally, human perception has capacity limitations generally avoided in machine vision systems, such as bottlenecks due to spatial attention and the drop off in visual acuity with retinal eccentricity (Jacobs and Bates, 2019). In terms of training environments, human perception is immersed in a rich multi-sensory, dynamical, three-dimensional experience, whereas standard training sets for ANNs consist of static images curated by human photographers (Jacobs and Bates, 2019). While these differences in architecture, environment, and learning procedures seem stark, they may not reflect differences in underlying knowledge and capacities, but instead constraints in manifesting the knowledge (i.e., the performance-competence distinction raised by Firestone (Firestone, 2020)). Nonetheless, these differences may have behavioral consequences. ANNs are found to be brittle relative to human perception in handling various forms of image corruption Geirhos et al. (2019a). One possible explanation for this finding is that machine perception is heavily influenced by texture whereas human perception is guided by shape (Geirhos et al., 2018a). The robustness gap between machine and human perception is narrowing as ANNs and training data sets increase in scale, reaching the point where machines surpass human robustness on some forms of input noise (Geirhos et al., 2021). Nonetheless, even as the robustness gap narrows, humans make systematically different classification errors than machines (Geirhos et al., 2021).

One particular class of ANN errors has attracted significant interest in the machine-learning community because the errors seem incongruous with human perception. These errors are produced by *adversarial image perturbations*, subtle image-specific modulations of individual pixels that are designed to alter ANN categorization to different coarse image classes (Szegedy et al., 2013; Biggio et al., 2013; Goodfellow et al., 2015), as illustrated in Figure 5.1a. This adversarial effect often transfers to ANN models trained on a different data set (Szegedy et al., 2013), with a different algorithm (Papernot et al., 2016a), or even to machine learning algorithms with fundamentally different architectures (Szegedy et al., 2013) (e.g., adversarial examples designed to fool a convolution neural network may also fool a decision tree). What makes these perturbations so remarkable is that to casual human observers, the image category is unchanged and the adversarial perturbations are interpreted—to the extent they are even noticed—

as irrelevant noise.

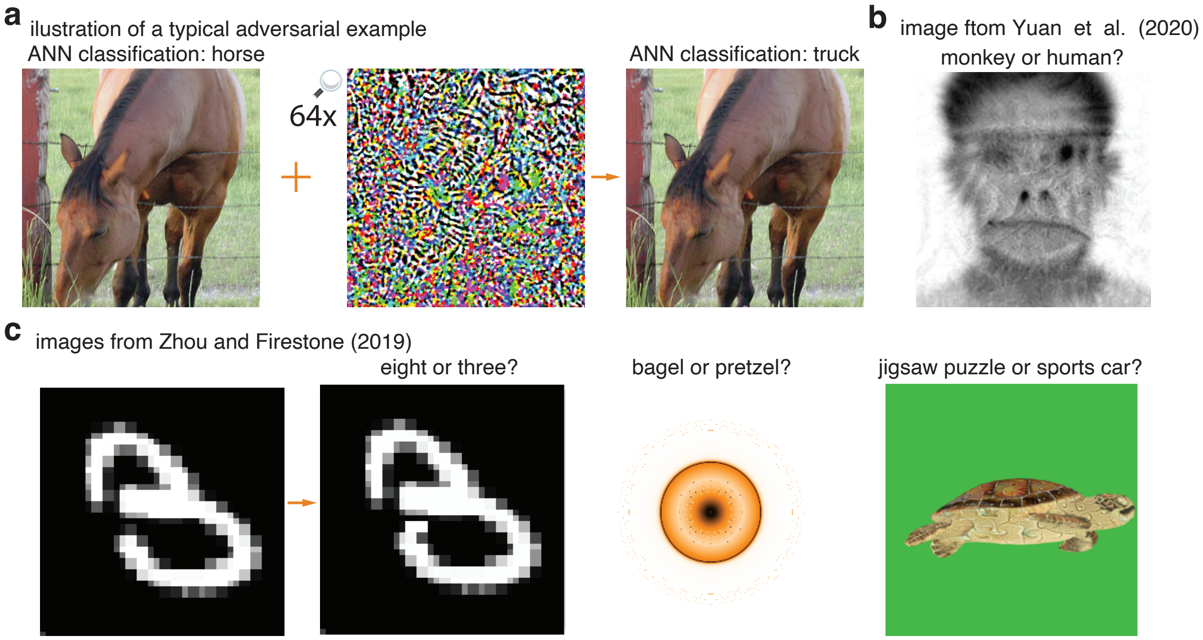

a
ilustration of a typical adversarial example
ANN classification: horse
64x
+
ANN classification: truck
b
image ftom Yuan et al. (2020)
monkey or human?
c
images from Zhou and Firestone (2019)
eight or three?
bagel or pretzel?
jigsaw puzzle or sports car?

Figure 5.1. Examples of adversarial images used as stimuli in past research. **(a)** A subtle perturbation added to a horse image that causes an ANN to switch its classification decision from horse to truck, similar to those first demonstrated in Figure 5 of Szegedy et al (Szegedy et al., 2013). (Original image shown here was obtained from MS-COCO dataset (Lin et al., 2014). In this and subsequent figures, perturbations are scaled up for better visualization. **(b)** An adversarial attack that causes face-selective neurons in macaque inferotemporal cortex to predict a perturbed human face image as monkey not human, obtained with permission from Yuan et al (Yuan et al., 2020). **(c)** Various adversarial images used in human behavioral studies by Zhou and Firestone (Zhou and Firestone, 2019). This paper present studies that go beyond the work in (b) and (c) by using perturbations more closely resembling those illustrated in Figure 5 of Szegedy et al (Szegedy et al., 2013), which seem relatively subtle and innocuous, the key properties of adversarial examples that made them 'intriguing' in earlier work (Szegedy et al., 2013). Illustration images in this Figure were obtained with permission from Yuan et al. (2020); Zhou and Firestone (2019); Papernot et al. (2016b); Athalye et al. (2018); Nguyen et al. (2015)

The standard procedure for generating adversarial perturbations starts with a pretrained ANN classifier that maps RGB images to a probability distribution over a fixed set of classes (Szegedy et al., 2013). When presented with an uncorrupted image, the ANN will typically assign a high probability to the correct class. Any change to the image, such as increasing the red intensity of a particular pixel, will yield a slight change to the output probability distribution. Adversarial images are obtained by searching—via gradient descent—for a perturbation of the original image that causes the ANN to reduce the probability assigned to the correct class (an untargeted attack) or to assign high probability to some specified alternative class (a targeted attack). To ensure that the perturbations do not wander too far from the original image, an L_∞-norm constraint is often applied in the adversarial machine-learning literature; this constraint specifies that no pixel can deviate from its original value by more than $\pm\varepsilon$, with ε usually much smaller than the $[0,255]$ pixel intensity range (Goodfellow et al., 2015). The constraint applies to pixels in each of the RGB color planes. Although this restriction does not prevent individuals from detecting changes to the image, with appropriate choice of ε the predominant signal indicating the original image class in the perturbed images is mostly intact. Yet ANNs largely change their predictions in response to adversarial perturbations.

These errors seem to point to a troubling fragility of ANNs, which makes them behave in a manner that is counter intuitive and ostensibly different than human perception, suggesting fundamental flaws in their design. Ilyas et al (Ilyas et al., 2019) propose that the existence of adversarial examples is due to ANNs exploiting features which are predictive but not causal, and perhaps ANNs are far more sensitive to these features than are humans. Kim et al (Kim et al., 2020) further argued that that neural mechanisms in the human visual pathway may filter out the signal contained in adversarial perturbations. However, Ford et al (Ford et al., 2019) make a theoretical claim that any classifier, human or machine, will be susceptible to adversarial examples in high dimensional input spaces if the classifier achieves less than perfect generalization to more standard types of corruption (e.g., Gaussian noise) or to naturally occurring variation in the input. As humans sometimes make classification mistakes, it may be inevitable that they also suffer from adversarial examples. However, it is not inevitable that ANNs and BNNs are susceptible to the same adversarial examples.

Although the fascination with adversarial perturbations stems from the assumption that ANNs are fooled by a manipulation to which humans are believed to be impervious, some evidence has arisen contradicting this assumption. Han et al (Han et al., 2019) used fMRI to probe neural representations of adversarial perturbations and found that early visual cortex represents adversarial perturbations differently than Gaussian noise. This difference is a necessary condition for human behavioral sensitivity to adversarial perturbations. Several research teams have shown that primate forced-choice classification decisions of modulated or synthetic images can be predicted by the responses of ANNs. First, Zhou and Firestone (Zhou and Firestone, 2019) performed a series of experiments on a variety of fooling images (Figure 5.1c) that ranged from synthetic shapes and textures to images having perceptually salient modulations. Although human-ANN agreement is found, Dujmović et al (Dujmović et al., 2020) argue that the agreement is weak and dependent on the choice of adversarial images and the design of the experiment. Second, Yuan et al (Yuan et al., 2020) trained an ANN to match the responses of face-selective neurons in macaque inferotemporal cortex and then used this model to modulate images toward a target category (Figure 5.1b). Both human participants and monkey subjects showed the predicted sensitivity to the modulations. In each of these two experiments, the image modulations were not subtle and a human observer could make an educated guess about how another observer would respond in a forced-choice judgment. As such, it remains an open question whether people are influenced by images that possess the intriguing property (Szegedy et al., 2013) that first drew machine-learning researchers to adversarial examples—that they are corruptions to natural images that are relatively subtle and might easily be perceived as innocuous noise.

In this work, we investigate whether adversarial perturbations—a subordinate signal in the image—influence individuals in the same way as they influence ANNs. We are not exploring here whether predominant and subordinate signals may have separation in human cognition, but rather that both signals may influence human perception. The challenge in making this assessment is that under ordinary viewing conditions, individuals are so strongly driven by the predominant signal that categorization responses ignore the subordinate signal. In contrast, ANNs clearly have a different balance of influence from the predominant and subordinate signals such that the subordinate signal dominates the decision of ANNs. Setting aside this notable and well-appreciated difference, which is a key reason for the interest in adversarial attacks, the question remains whether the subordinate signal reflects some high-order statistical regularity to which both ANNs and BNNs are sensitive. If so, we obtain even more compelling evidence for ANNs as a model of human perception; and if not, we can point to a brittleness of ANNs that should be rectified before trust is placed in them to make perceptual decisions. We conduct behavioral experiments in which participants performed forced-choice classification of a briefly presented perturbed image, or participants inspected pairs of perturbed images with no time constraint and selected the one that better represented an object class. We find converging evidence from these experiments suggesting that subordinate adversarial signals that heavily influence ANNs can also influence humans in the same direction. We further find the ANN properties that underlie this perceptual influence and identify reliance on shape cues as an important characteristic enhancing alignment with human perception.

5.2 Methods

5.2.1 Stimuli

Adversarial images used as stimuli in all experiments were generated using an ensemble of ANN models which are listed along with their ImageNet classification performance in Tables 5.7 and 5.8. Our stimuli were created to alter the ANN ensemble prediction score for sets of ImageNet classes that we refer to as *coarse classes*. Section 5.2.5 describes the coarsification procedure and adversarial classes used in our experiments. Images are separately generated for each level (ε) of perturbation. For Experiment 1, we generated adversarial perturbations with $\varepsilon = 32$ (out of 256 pixel intensity levels). For experiments 2–5, we generated adversarial stimuli at four perturbation magnitudes, $\varepsilon \in \{2, 4, 8, 16\}$. For experiment 1, we added adversarial perturbations to images from the ImageNet dataset (Deng et al., 2009). For Experiments 2–5, adversarial perturbations were added to a collection of images obtained from the Microsoft

COCO dataset (Lin et al., 2014) and OpenImages dataset (Kuznetsova et al., 2020). Our use of the above-mentioned image datasets to create our stimuli was in line with their terms of use. The image resolution used for Experiments 1–4 was 256 × 256 and 384 × 384 for Experiment 5.

5.2.2 Participants

Experiment 1 included 38 participants with normal or corrected vision. Participants gave informed consent and were awarded a reasonable compensation for their time and effort. Participants were recruited from our institute but were not involved in any projects with the research team. Experiment 1 control (i.e., Experiment SI-5) included 50 participants recruited from an online rating platform (Mason and Suri, 2012). For Experiments 2–5, we performed psychophysics experiments using an online rating platform. In each experimental condition, approximately 100 participants were recruited to participate in the task (see Table **??** for exact number). No statistical method was used to predetermine number of participants, but sample size was decided to be comparable to that used in previous similar studies Zhou and Firestone (2019); Kar et al. (2019). Participants received a compensation in the range of $8 - $15 per hour based on the expected difficulty of the task. No sex or age information was gathered from the participants for all our studies. Our participants were all located in North America and were financially compensated for their participation. We excluded participants if they were not engaged in the task, as assessed using randomly placed catch trials with an unambiguous answer (e.g., pairing an unperturbed dog image with a cat image and asking which image is more cat-like). If a participant failed one catch trial for Experiments 2, 3 and 5, or two catch trials for Experiment 4, the task automatically terminated and their data was not analyzed.

5.2.3 Experiment structure

For Experiment 1, participants sat on a fixed chair 61 cm away from a high refresh-rate computer screen (ViewSonic XG2530) in a room with dimmed light to classify brief, masked presentations of adversarial images. Participants classified images that appeared on the screen into one of two classes by pressing buttons on a response time box (LOBES v5/6:USTC) using two fingers on their right hand. The assignment of classes to buttons was randomized for each experiment session. Each trial started with a fixation cross displayed in the middle of the screen for $500 - 1000$ ms. After the fixation period, an image of fixed size 15.24 cm $\times$ 15.24 cm (14.2° visual angle) was presented briefly at the center of the screen for a period of 63 ms (71 ms for some sessions). The image was followed by a sequence of ten high contrast binary random masks, each displayed for 20 ms (see example in Figure 5.2a). Participants pressed one of two buttons to classify the object present in the image. The waiting period to start the next trial was of the same duration whether participants responded quickly or slowly. Each participant's response time was recorded by the response time box relative to the image presentation time (monitored by a photodiode). In the case where a participant pressed more than one button in a trial, only the class corresponding to their first choice was considered. Each participant completed between 140 and 950 trials.

For Experiments 2 – 5, Participants performed an extended viewing two-alternative forced-choice task where they saw a pair of images that appeared on the screen and chose the one that looked more like it belonged to a queried target class. On each experiment trial, participants were shown a question above the stimuli which read, "Which image looks more target-like?", where the target was cat, bottle, etc. (The complete set of categories is presented in Figure 5.3a). Two buttons labeled 'left' and 'right' were placed below the stimulus pair. Participants used the keyboard keys F and J to select the left and right responses, respectively. (Participants were required to use a computer and not a mobile device to participate.) After a response was made, the images and buttons disappeared followed by an inter-trial-interval (ITI) screen. This ITI screen contains a 'next trial' button with the following text: "Press SPACEBAR to move to the next trial." Participants advanced in the task at their own pace. Individuals were asked to use their own perceptual judgement to make this decision and were informed that their responses will be compared to a machine doing the same task. Each participant completed 104 experiment trials. 8 out of these 104 trials were catch trials with a clear solution placed randomly to measure user engagement.

5.2.4 ANN models used to create adversarial perturbations

We created adversarial attacks on ANN models trained on large collection of natural images to classify images to objects. We describe here the specific models used along with training strategies and datasets.

For experiment 1, we used an ensemble of 10 ANN models trained on the ImageNet dataset (Deng et al., 2009) to create adversarial perturbations; these models are listed in Table 5.7. For experiments 2–4, we used an ensemble of 6 highly accurate ANNs trained on the ImageNet dataset to create adversarial perturbations; these models are listed in Table 5.8. In order to match the initial visual processing of ANN models with human vision, we added an artificial 'retinal blurring' layer (described in Veerabadran et al. (2023a)) that mimics processing done by primates fovea. However, that choice was not essential and we could not detect a reliably strong increase in human perceptual bias from adding the Fovea layer. For experiment 5, we constructed two alternative ANN models (one based on convolution and one based on the self-attention operation) with which adversarial attacks were created. These ANNs were trained on the same data from in-house JFT-300MDosovitskiy et al. (2020) dataset and finetuned on ImageNet. For the former we used an ensemble of the ResNet-101 and ResNet-200 networks (BiT models described in Dosovitskiy et al Dosovitskiy et al. (2020)) and for the latter, we used an ensemble the ViT-L16 and ViT-B32 networks (Dosovitskiy et al., 2020) respectively. We show in Table 5.9 the accuracy of these models on the ImageNet validation set.

Ensembling predictions from a set of ANN models refers to the process of mathematically combining predictions of an input image from more than one ANN model. This is a common practice used in adversarial machine learning to generate adversarial attacks that transfers across ANN models (Liu et al., 2016). We perform a simple aggregation of prediction scores from the ensemble's individual models by taking an average of the unnormalized predictions (aka logits) across the ANN models. This aggregation is related to the geometric mean of the ANN prediction probability.

5.2.5 Coarsening of object categories

The ImageNet dataset consists of 1000 fine object categories such as breeds of dogs that a typical human participant may not be familiar with. For this reason, we compute predictions corresponding to groups of the fine object categories that may be more familiar to the experiment participants. We group the fine classes into one of nine common object categories (sheep, dog, cat, elephant, bird, chair, bottle, truck, and clock) we refer to as coarse categories (coarse categories used in our experiments along with ImageNet support can be found in Table 5.1). For example, we aggregate predictions from all 120 ImageNet classes that correspond to various dog breeds into a single ‘dog’ coarse category. These coarse categories were chosen arbitrarily from Geirhos et al (Geirhos et al., 2018a) in order to roughly balance natural and human-made categories. Let S_i be the score assigned by our ensemble to a fine ImageNet category i (i.e., the value of the i^{th} unnormalized prediction score; aka logit) and c be a coarse category. We compute the unnormalized prediction score to a coarse category c as:

$$S_c = (\log \sum_{i \in c} \exp S_i) - (\log \sum_{j \notin c} \exp S_j) \tag{5.1}$$

This score reflects the logit of the binary classification model that defines the probability of the existence of the coarse category c. Refer to Table 5.1 where we report the ImageNet support of each coarse category we have used in Experiments 2–5.

Table 5.1. Number of ImageNet fine classes (ImageNet support) corresponding to each of our experiment's coarse categories

Coarse category	Number of ImageNet fine categories
Cat	6
Dog	120
Bird	52
Bottle	7
Sheep	6
Chair	4
Elephant	2
Clock	3
Truck	8

5.2.6 Perturbation generation algorithm

We use the iterative Fast Gradient Sign Method (iFGSM) technique (Kurakin et al., 2016), an iterative adversarial attack method, to create targeted or untargeted adversarial attack on the ANN ensemble. iFGSM optimizes a small perturbation to the input image by iteratively perturbing the image using information from the image gradient corresponding to minimizing an adversarial objective function.

The adversarial objective function corresponding to a targeted attack towards target class y on input X is the binary cross entropy loss with label y:

$$J_{targeted}(\mathbf{X}, y) = -\log(P_{ens}(y|\mathbf{X})) \tag{5.2}$$

Similarly, the adversarial objective function corresponding to an untargeted adversarial attack reducing the prediction confidence of class y for input X is the binary cross entropy loss with label $\bar{y}$:

$$J_{untargeted}(\mathbf{X}, y) = -\log(P_{ens}(\bar{y}|\mathbf{X})) = -log(1 - P_{ens}(y|\mathbf{X})) \tag{5.3}$$

The following equation then outlines the iterative procedure used in combination with $J_{targeted}$ or $J_{untargeted}$ to create an adversarial attack that increases or decreases prediction confidence on a target class y respectively.

$$\tilde{\mathbf{X}}_i = \tilde{\mathbf{X}}_{i-1} + \alpha \times \text{sgn}(\nabla_{\mathbf{X}}(-J(\mathbf{X}, y))) \tag{5.4}$$

We constrained adversarial perturbations created with iFGSM using the L_∞ norm of the perturbation ($||\mathbf{X_{adv}} - \mathbf{X}||_\infty \leq \varepsilon$); we restrict the adversarial perturbations by clipping all perturbed image pixels with intensity less than $\mathbf{X} - \varepsilon$, and greater than $\mathbf{X} + \varepsilon$ as follows:

$$\mathbf{X}_i = \text{clip}(\tilde{\mathbf{X}}_i, \mathbf{X} - \varepsilon, \mathbf{X} + \varepsilon) \tag{5.5}$$

The above procedure is performed iteratively from $i \in \{1, \ldots, n\}$ until the final adversarial image $\mathbf{X}_n$ is computed.

Viewing conditions for human raters could vary significantly (e.g., raters could be viewing images from different angles, on monitors that have different sizes, or while sitting at various distance away from the monitor), which may interfere with the experiment. To address this problem, we created adversarial images that are largely invariant to the change in these viewing conditions. To achieve this invariance, we modeled the change in viewing condition as geometric transformation of the image (e.g., different rotation, scale, and translation). We compute adversarial examples that are robust to image transformations by sampling random geometric transformations applied to the original image at each step of the perturbation algorithm (rotation $\theta \sim$ U(0, $\pi/6$), scale $s_x, s_y \sim$ U(0.5*L, L), and translation $t_x, t_y \sim$ U(-L/4, L/4)) where L represents the image width in pixels (same as height as we use square images) of our stimuli. Let $t \in T$ be an identity-preserving geometric image transformation that is differentiable and $\mathbf{X}$

be the input image to be perturbed to be classified as y_{target}. We have the following optimization problem to solve to compute an adversarial image with invariance to geometric transformations.

$$\underset{\mathbf{X}'}{\arg\max} \underset{t \in T}{\mathbb{E}} [\log(P(y_{target}|t(\mathbf{X}')))] \quad \text{s.t. } ||\mathbf{X}' - \mathbf{X}||_\infty \leq \varepsilon \tag{5.6}$$

5.2.7 Class subsampling for Experiment 4 and 5 stimuli.

Experiments 4 and 5 are different in nature than the rest of the experiments in that we explicitly ask humans to choose an image from a pair of adversarial perturbations towards two different target classes. In this case, there exists a significant difference in ImageNet support corresponding to each individual coarse class in the pair, e.g. there are 120 ‘dog’ classes vs 7 ‘bottle’ classes in ImageNet. In order to prevent such difference from causing a hidden bias in human response, we randomly sampled a subset of the classes to compute coarse target class score for the class that had larger ImageNet support. Let n_A and $n_{A'}$ be the number of ImageNet classes corresponding to target classes A and A' that are paired in an experiment. For each image, we randomly subsample $\min(n_A, n_{A'})$ fine ImageNet classes corresponding to target class A (and third class A'); these fine classes are then used to for computing the coarse score for target class A (or class A') using Eqn 5.1, which is used in the adversarial objective function that is optimized by the algorithm.

5.3 Results

5.3.1 Adversarial perturbations increase human classification errors with brief presentations

In an initial experiment, we examined human classification responses to brief, masked presentations of adversarial images. By restricting exposure time to increase classification errors, the experiment aimed to increase individuals’ sensitivity to aspects of the stimuli that might otherwise not have influenced a classification decision. We created adversarial perturbations to images of a true class T by optimizing the perturbation such that an ensemble of ANNs produces a classification preference for an adversarial class, A. We refer to the perturbed image as $A\uparrow$. Participants were asked to make a forced choice between T and A (Figure 5.2a). We also tested participants on control images formed by top-down flipping adversarial perturbations obtained in the $A\uparrow$ condition. This simple transformation breaks the pixel-to-pixel correspondence between the adversarial perturbation and the image and largely obliterates the adversarial effect on ANNs while preserving norms and other statistics of the perturbation. Our results show that participants are more likely to choose the adversarial class A with $A\uparrow$ images than with control images (Figure 5.2b).

The increase in error rate appears to demonstrate a consistent influence of adversarial perturbations on both ANNs and BNNs. However, one can raise three concerns with this experiment. First, the specificity of the attack observed in ANNs—where targeting class A results in higher probability specifically for class A—has not been established by the current experimental paradigm. Forced choice precludes the possibility that participants perceive images to be of a third class. Participants may be sensitive only to the fact that images are less clean examples of class T in the $A\uparrow$ condition. Second, the perturbation magnitude we used ($\varepsilon = 32$), is larger than is typical in generating adversarial examples for ANNs. With lower magnitudes, measuring reliable effects on human classification may be challenging. Third, with large perturbations, it is possible that the increased error rate is due to the perturbations obscuring image regions critical to discriminating classes T and A in the $A\uparrow$ condition, but not in the control condition, which lacks the pixel-to-pixel correspondence between perturbation and source image. Indeed, in a time-unlimited variant of the cat versus dog discrimination of Experiment 1 (Experiment SI-5), error rates are slightly higher in the $A\uparrow$ condition (0.135 vs. 0.089; $t(50) = 3.19$, $p < 0.001$, Cohen's $d = 0.45$, 95% CI of difference in means is greater than 0.02, right-tailed), admitting the possibility that some of the effect observed in Experiment 1 is due to obfuscation.

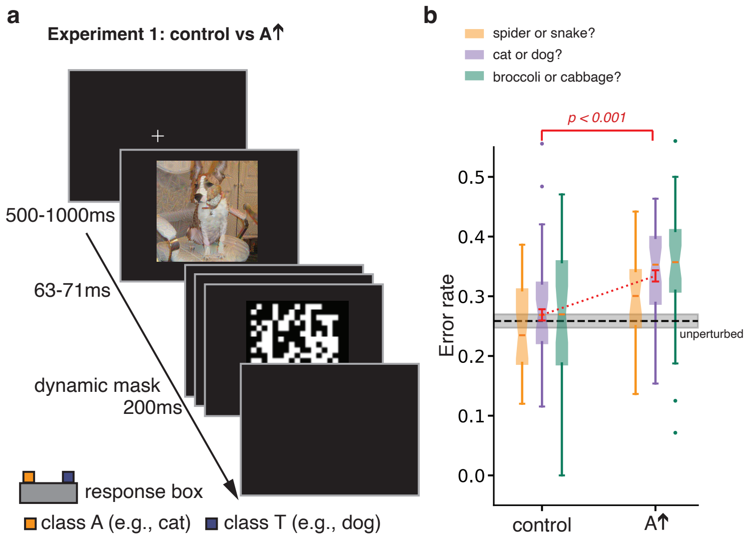


Figure 5.2. Experiment 1: Adversarial perturbations increase errors when participants are asked to classify briefly presented images. (a) Following fixation, a stimulus image is presented briefly followed by a dynamic, high contrast mask. Participants chose between two classes, the target class T (dog in the example) and an adversarial class A (cat in the example). Images used in the experiment are obtained from ImageNet dataset (Deng et al., 2009), but image used here as an illustration is obtained from MS-COCO dataset (Lin et al., 2014). **(b)** Box-and-whisker plots show the human error-rate distribution obtained from a pool of n=38 independent participants who performed a max of three discrimination conditions (spider vs. snack n=24, cat vs. dog n=35, broccoli vs. cabbage n=32). We use Tukey conventions: box lower border, middle line and upper border show 25th percentile, median, and 75th percentile, respectively, and whiskers show lowest and highest points within 1.5 × the interquartile range). The mean error rate across conditions is reliably higher for adversarial versus control stimuli ($t(91) = 4.463$, $p < 0.001$, Cohen's d=0.66, 95% CI of difference between means=[0.04, 0.09], one-tailed t-test). Red error bars indicate ±1 standard error of the mean (SE). The black dashed line is the baseline error rate (±1 SE) for unperturbed images.

5.3.2 Adversarial perturbations bias human choice in extended viewing

Experiment 1 used brief, masked presentations to limit the influence of the original-image class (the predominant signal) on responses, and thereby to reveal sensitivity to adversarial perturbations (the subordinate signal). We designed three additional experiments that had the same goal but avoided the need for large-magnitude perturbations and limited-exposure viewing. In these experiments, the predominant signal in an image could not systematically guide response choice, allowing the influence of the subordinate signal to be revealed. In each of these experiments, a pair of unmasked nearly-identical stimuli are presented and remain visible until a response is chosen (Figure 5.3a). The pair of stimuli share the same predominant signal, i.e., they are both modulations of the same underlying image, but they have different subordinate signals (Figure 5.3b). Participants are asked to select the image that is more like an instance of a target class (Figure 5.3c). In Experiment 2, the two stimuli are modulations of an image of class T, one perturbed such that ANNs predict it to be more T-like and one perturbed to be less T-like ($T\uparrow$ and $T\downarrow$, respectively). In Experiment 3, the stimuli are modulations of an image belonging to a true class T, one perturbed to alter ANN classification toward a target adversarial class A ($A\uparrow$) and the other using the same perturbation except flipped right-left as a control condition (control); this control serves to preserve norms and other statistics of the perturbation, but is more conservative than the control in Experiment 1 because left and right sides of an image may have more similar statistics than the upper and lower parts of an image. In Experiment 4, the pair are also modulations of an image of a true class T, one perturbed to be more A-like and one to be more like a third class, A' ($A\uparrow$ and $A'\uparrow$, respectively). Trial blocks alternated between participants being asked to choose the more A-like image and the more A'-like .

In each experiment, the ANN has higher confidence in the target class for one stimulus of the pair over the other because of the differential effect of the subordinate signal (Figure 5.3b); neither choice conflicts with the true class of the original image. And in each experiment, human perception is consistently biased by the adversarial perturbation in the direction predicted by the ANN (Figure 5.3d; E2: $F(1,385) = 156.7, p < .001, \eta_p^2 = 0.29$, 95% CI of perceptual bias=[0.03, 0.05], E3: $F(1,392) = 140.6$, $p < .001, \eta_p^2 = 0.26$, 95% CI of perceptual bias=[0.05, 0.07], E4: $F(1,385) = 126.7, p < .001, \eta_p^2 = 0.25$, 95% CI of perceptual bias=[0.05, 0.07]; see also Table 5.3 for a nonparametric Wilcoxin test of significance). Human perceptual bias is robust across target classes and grows with ε. Conducting a two factor ANOVA, we observe a main effect for target class, indicating that participants are more sensitive to some classes than others (E2: $F(3,385) = 12.94, p < .001$, $\eta_p^2 = 0.09$, E3: $F(3,392) = 19.86, p < .001$, $\eta_p^2 = 0.13$, E4: $F(3,385) = 16.84, p < .001$, $\eta_p^2 = 0.12$). We also observe a main effect for perturbation magnitude, with the perceptual bias growing with ε (E2: $F(3,1155) = 10.73, p < .001$, $\eta_p^2 = 0.03$, E3: $F(3,1176) = 22.08, p < .001$, $\eta_p^2 = 0.05$, E4: $F(3,1155) = 17.48, p < .001$, $\eta_p^2 = 0.04$). A reliable interaction occurs between target class and perturbation magnitude for E3 and E4, reflecting a larger slope with ε for some classes than others (E2: $F(9,1155) = 1.78, p = .067$, $\eta_p^2 = 0.01$; E3: $F(9,1176) = 2.55, p < 0.01$, $\eta_p^2 = 0.02$, E4: $F(9,1155) = 4.25, p < .001$, $\eta_p^2 = 0.03$). See Tables 5.2 for a summary of statistics from Experiments 1 – 5. Nonetheless, performing separate ANOVAs for each adversarial target class, every one of the four conditions in each of the three experiments yields a reliable above-chance bias.

Across Experiments 2-4, the per-image human perceptual bias is significantly positively correlated with the bias of a black box ANN (i.e., a model that was not used in generating perturbations. (E2: Spearman's $\rho(1534) = 0.13, p < 0.001$, E3: $\rho(1534) = 0.29, p < 0.001$, E4: $\rho(1534) = 0.18, p < 0.001$). Perturbation magnitudes varied from 2 to 16, smaller than has previously been studied with human participants, and similar in magnitude to perturbations used in adversarial machine learning research. Surprisingly, even a perturbation of 2 pixel intensity levels (on a 0-255 scale) is sufficient to reliably bias human perception (With Bonferoni corrected p values, E2: two-tailed $t(388) = 3.54, p = 0.002$, Cohen's d=0.18,two-tailed; E3: $t(395) = 3.95$, $p < 0.001$, Cohen's d=0.2,two-tailed; E4: $t(388) = 3.45$, $p = 0.002$, Cohen's d=0.18, two-tailed).

We further tested whether there exist a simpler explanation for observing the effect in our main experiment. First, we investigate whether the effect in experiment 4 is driven by a few outlier stimuli; we analyzed the distribution of per-stimulus perceptual bias (averaged across subject responses to a given stimulus) for all conditions in this experiment by performing the Shapiro-Wilk test for normality and found no credible evidence to reject the hypothesis that the distribution of perceptual bias across stimuli is normal for almost all the conditions (see Table 5.6). Second, we find no credible evidence that participants are more sensitive to perturbations that are highly salient (e.g., textures painted into a uniform background such as the sky) than to ones that are less salient, as measured by the structural similarity index, MS-SSIM (Wang et al., 2004) (See related Supplementary Note in Veerabadran et al. (2023a)). Third, participants make relatively few errors in a direct classification task involving single adversarial images, even with $\varepsilon = 16$, indicating that the perturbations are not altering the ostensible image class (see relevant Supplementary Note 3 in Veerabadran et al. (2023a)). Fourth, we conducted shuffling analyses, where the shuffling procedure eliminated all the effects, suggesting that the effects that we observed are robust and highly unlikely to occur by chance (see relevant Supplementary Note in Veerabadran et al. (2023a)) .

Each of Experiments 2-4 has a particular strength, but on its own, each has a potential confound. The strength of Experiment 2 is that participants are asked to make an intuitive judgment (e.g., which of two perturbed cat images are more cat-like); however, Experiment 2 allows the possibility that adversarial perturbations cause an image to be more or less cat-like simply by sharpening or blurring the image. The strength of Experiment 3 is that we match all statistics of the perturbations being compared, not just the maximum magnitude of the perturbations. However, matching perturbation statistics does not ensure that the perturbations are equally perceptible when added to the image; consequently, participants might have chosen on the basis of image distortion. However, we showed from the results of a control experiment we conducted and presented in Figure 5.4 that the judgments in Experiment 3 are not based on perceived distortion). The strength of Experiment 4 is that it proves that participants are sensitive to the question being asked because the same image pair ($A \uparrow$ and $A' \uparrow$) yields systematically different responses depending on the question asked ('more A-like' or 'more A'-like') but requires participants to answer a seemingly nonsensical question, leading to potential variability in the question's interpretation.

Taken together, Experiments 2-4 provide converging evidence that the subordinate adversarial signals that strongly influence ANNs also bias time-unlimited humans in the same direction, even under a subtle perturbation magnitude.

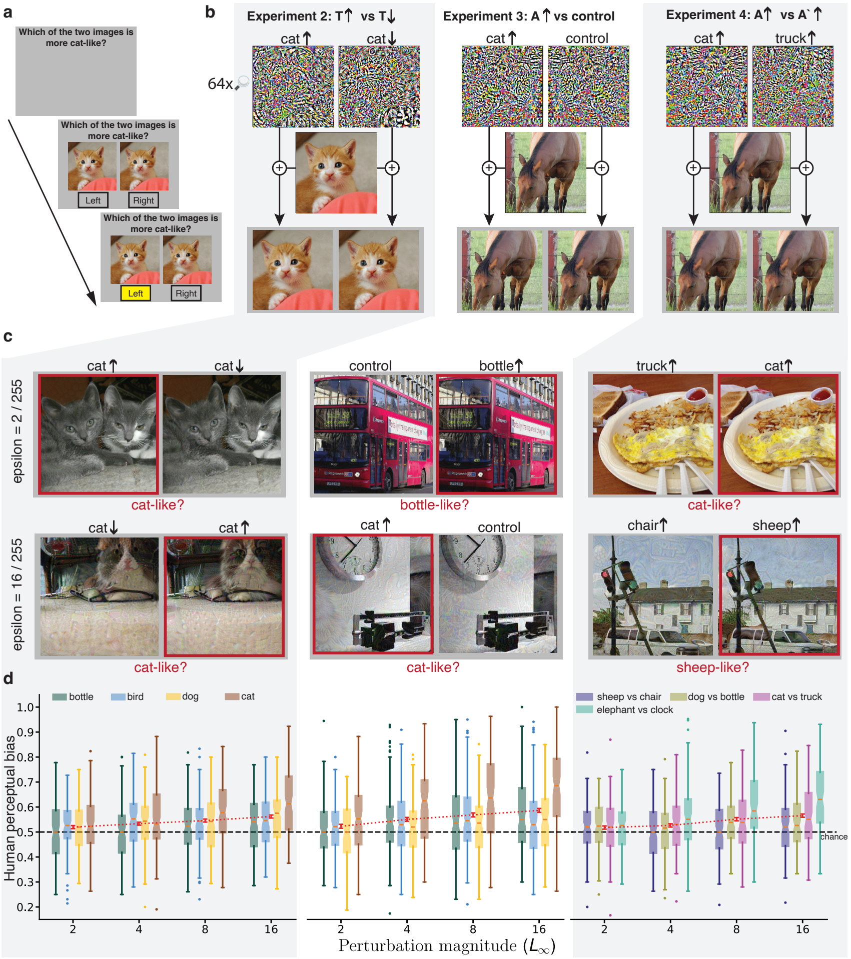


Figure 5.3. Experiments 2–4: Adversarial examples systematically bias choice. (a) Participants are shown two perturbations of the same image, of true class T, and are asked to select the image which is more like an instance of some adversarial class A. The image pair remains visible until a choice is made. **(b)** One of the two choices is an adversarial perturbation that increases the probability of classifying the image as A, denoted $A\uparrow$. Experiment 2: $T = A$; the second image is perturbed to be less A-like, denoted $A\downarrow$. Experiment 3: $T \neq A$; the second image is formed by adding a right-left flipped version of the adversarial perturbation, which controls for the magnitude of the perturbation while removing the image-to-perturbation correspondence. Experiment 4: $T \neq A$; the second image is an adversarial perturbation toward a third class A', denoted $A'\uparrow$. **(c)** We show adversarial images which empirically yielded human responses consistent with those of the ANN (indicated by the red box) for $\varepsilon = 2$ and 16, corresponding to the lowest and largest perturbation magnitudes used in these experiments. Example images in a–c are obtained from the Microsoft COCO dataset (Lin et al., 2014) and OpenImages dataset (Kuznetsova et al., 2020). **(d)** Box plots quantifying participant bias toward $A\uparrow$ (where $A = T$ for Experiment 2 and $A \neq T$ for Experiments 3 and 4), as a function of ε for four different conditions (each a different adversarial class A) collected from n=389 participants for Experiment 2, n=396 participants for Experiment 3 and n=389 independent participants for Experiment 4. The red points (with ±1 SE bars) indicate the mean across conditions. The black dashed line indicates performance of random strategy that is insensitive to the adversarial perturbations.

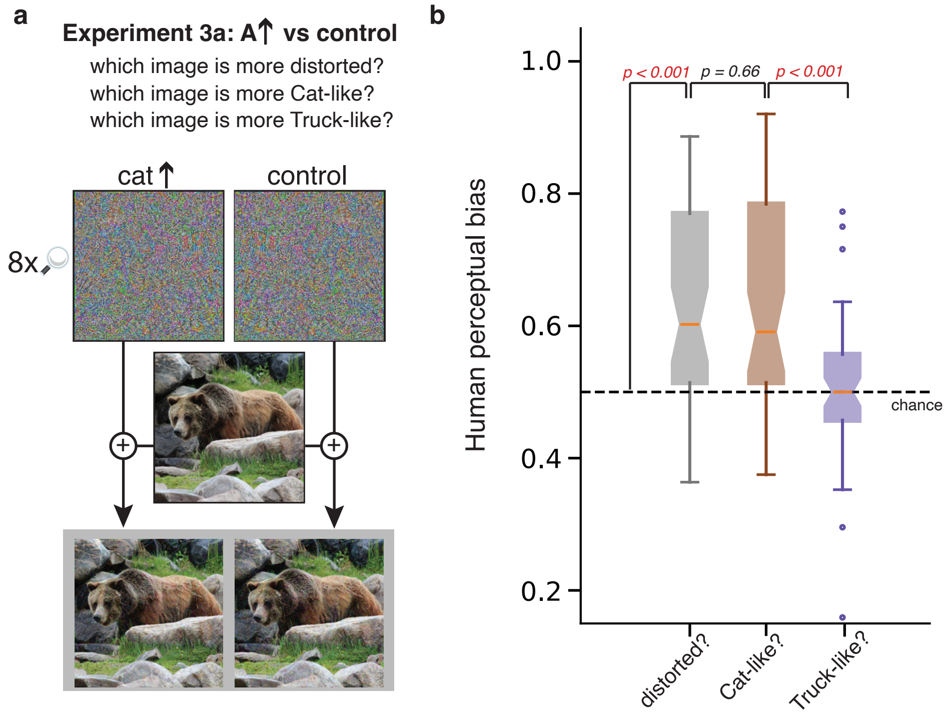


Figure 5.4. Experiment 3-control (a) This panel describes a control experiment in which participants are shown the same stimuli as the $\varepsilon = 16$ 'cat' condition of Experiment 3. Each trial consists of a pair of images, one adversarially perturbed toward the cat class and one with the same perturbation only flipped left-right. In a replication condition, we ask participants which image is more cat-like. In a second condition, we ask which image is more distorted. And in a third condition we ask, which image is more like a truck (i.e., a class which is different from the adversarial class). The three conditions were between participant, with n=50 independent participants per condition. Example image is drawn from a collection of images from the Microsoft COCO dataset (Lin et al., 2014) and OpenImages dataset (Kuznetsova et al., 2020). (b) Box plots (same convention as Figure 5.2c) showing that both the 'distorted' condition and the 'cat' condition obtained a reliable bias (from n=50 independent participants) in selecting the adversarial image (distorted: $t(49) = 6.241, p < 0.001$, Cohen's d=0.88, 95% CI of bias=[0.09, 0.17], two-tailed ; cat: $t(49) = 6.33, p < 0.001$, Cohen's d=0.9, 95% CI of bias=[0.1, 0.18], two-tailed ttest). However, there was no reliable difference between the two conditions ($t(49) = 0.439, p = 0.66$, Cohen's d=0.09, 95% CI of bias=[-0.05, 0.07], two-tailed). However, we find no credible evidence that the 'truck' condition produced any deviation from chance ($t(49) = 0.545, p = 0.588$, Cohen's d=0.077, 95% CI of bias=[-0.02, 0.04], two-tailed), and the 'cat' bias was larger than the 'truck' bias ($t(49) = 4.984, p < 0.001$, Cohen's d=0.996, 95% CI of the difference between their means is computed as greater than 0.09, one-tailed ttest). Thus, the experiment shows that choice based on image distortion could produce a bias as large as we observed for the $\varepsilon = 16$ 'cat' condition of Experiment 3, in actuality participants were not choosing based on distortion. Were participants choosing based on distortion, the adversarial class would be irrelevant, and one would expect no difference between the 'cat' and 'truck' conditions, which we observe here.

5.3.3 What properties of the ANN are critical to perturbation effectiveness?

Having shown human susceptibility to adversarial examples, we turn to investigating the particular ANN properties that influence this susceptibility. We utilize two model classes, convolutional and self-attention architectures. Convolutional networks (Fukushima, 1980b; LeCun et al., 1989) are the dominant architecture used in computer vision and in modeling the human visual system (Lindsay, 2020); they incorporate strong inductive biases such as local receptive fields and approximate translation equivariance. Convolutional models apply static local filters across the visual field and build a hierarchy of representations by repeating this operation multiple times, mimicking the hierarchy in the ventral pathway of visual cortex (Felleman and Van Essen, 1991). Convolutional networks are inspired by the primate visual system (Fukushima, 1980b; Fukushima and Miyake, 1982) with convolution and pooling operations connecting to the simple and complex cells in the visual system (Hubel and Wiesel, 1962). Recently, a new class of architectures has arisen for computer vision that incorporate mechanisms of self attention Ramachandran et al. (2019); Parmar et al. (2018); Chen et al. (2020); Dosovitskiy et al. (2020). Originally, these mechanisms were designed to tackle problems in natural-language processing e.g., transformers (Vaswani et al., 2017); and thus received no explicit architectural inspiration from the visual system. The self-attention operation determines a weighting for embeddings of different tokens or words to obtain the next level of representation in the network hierarchy. To adapt these models to image processing, the image is typically divided into non-overlapping patches and then these patches are processed as if they are a sequence of words in a sentence (Dosovitskiy et al., 2020; Chen et al., 2020). The main operation in self-attention models is nonlocal, allowing for global communication across the entire image space. Self-attention models achieve state-of-the-art performance and have some intriguing differences from convolutional models, including the fact that self-attention models have relatively greater bias toward shape features than texture features as compared to convolutional models

(Tuli et al., 2021; Naseer et al., 2021), consistent with human vision. The errors produced by self-attention models better match human error patterns than errors produced by convolutional models (Tuli et al., 2021), possibly due to their ability to extract shape features.

We constructed two alternative models, one convolutional and one based on self attention, trained on the same data. Both models achieve comparable top-1 and near state-of-the-art classification accuracy on ImageNet (86.3% and 86.6%, respectively). We conducted a version of Experiment 4 using perturbations generated by either convolutional or self-attention models. Human perception is biased in the predicted direction in both conditions (convolutional: $t(380) = 3.91, p < 0.001$, Cohen's d = 0.2, 95% CI of difference between means=[0.01, 0.02], two-tailed; self attention: $t(380) = 5.98, p < 0.001$, Cohen's d=0.31, 95% CI of difference between means=[0.02, 0.03], two-tailed), indicating that both models are aligned with human perception. Because structural differences between convolutional and self-attention models lead to somewhat different image representations (Raghu et al., 2021), we ask whether one or the other is better aligned.

As an assessment of the relative effectiveness of the two representations, we conducted an experiment requiring participants to select between adversarial images generated by the two models. Using each of the models, we generated adversarial examples from an image of a true class T and perturbed toward adversarial class A, denoted $A\uparrow_{conv}$ and $A\uparrow_{attn}$. We presented matched pairs of adversarial images—$A\uparrow_{conv}$ and $A\uparrow_{attn}$—to participants and asked which of the two images is more A-class like (Figure 5.5a). Participants are more likely to choose adversarial images from the self-attention model than the convolution model as being A-class like ($t(396) = 18.25, p < 0.001$, Cohen's d=0.91, 95% CI of difference between means=[0.1, 0.13], two-tailed); see Figure 5.5c).

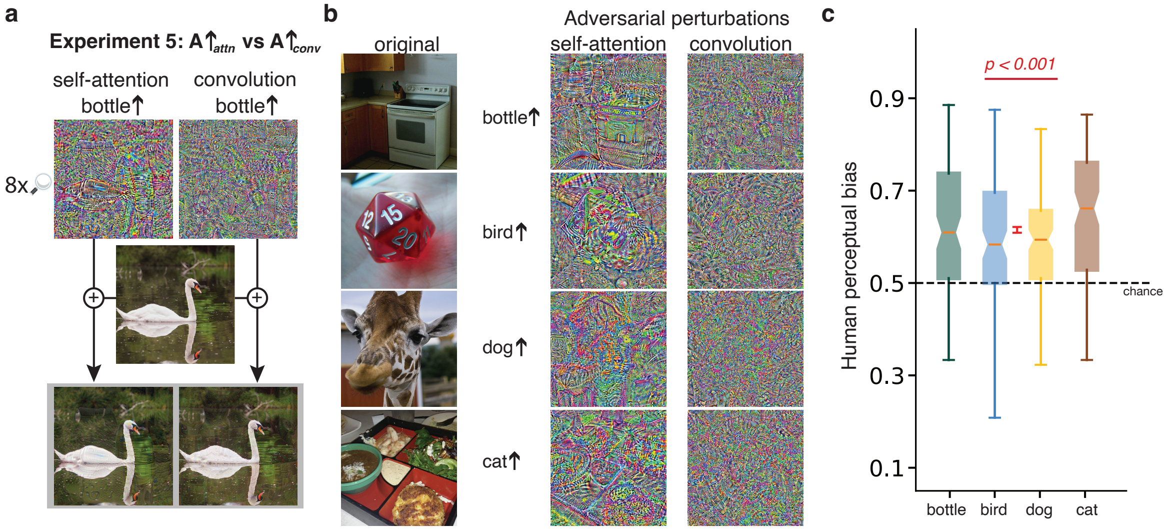


Figure 5.5. Experiment 5: Participants are more sensitive to adversarial images produced by self-attention ANNs than convolutional ANNs. (a) Adversarial perturbations ($\varepsilon = 16$) of an image of true class T are obtained from either self-attention or convolutional ANNs that increase the model's confidence in adversarial class A, $A \neq T$. Participants are asked to judge which of the two adversarial images are more like an instance of class A. **(b)** Examples of four original images and corresponding perturbations produced by the two models toward specific adversarial classes. The perturbations produced by the self-attention model have more apparent structure. Example images in a and b are obtained from the Microsoft COCO dataset (Lin et al., 2014) and OpenImages dataset (Kuznetsova et al., 2020) **(c)** Box plots (same convention as Figure 5.2c) indicating probability that n=396 independent participants (cat n=100, dog n=98, bird n=100, bottle n=98) prefer adversarial images produced by self-attention over convolutional ANNs for each of four classes. Participants reliably prefer adversarial images produced by self-attention over convolutional ANNs ($t(396) = 18.25, p < 0.001$, Cohen's d=0.91, 95% CI of difference between means=[0.1, 0.13], two-tailed ttest). Mean across conditions is shown as red point with ±1 SE errorbar.

The more interesting hypothesis for the selection preference, suggested by the literature (Tuli et al., 2021; Naseer et al., 2021), is that image representations in human perception better match representations obtained by self-attention models. We conducted further analyses to understand the nature of the representational differences between self-attention and convolutional models. We quantified the shape bias of the two models we used in this experiment using the Stylized-ImageNet dataset (Geirhos et al., 2018a) and found that indeed the self-attention model shows more shape bias (46.9%) than the convolution model (41.7%) (Tables 5.10 and 5.11). Further, the self-attention model is more robust to image noise corruptions (Tables 5.12 and 5.13) presumably as a result of its greater reliance on shape features.

Inspecting the perturbations produced by self-attention models, they appear to have more structure in that the perturbations are aligned with edges in the original image (e.g., see Figure 5.5b). To formalize a measure of edge strength, we use an automatic procedure to extract edges from the adversarial perturbations and sum the evidence for edges across image space. Perturbations generated by the self-attention model contained significantly more evidence for edges than those generated by the convolutional model ($t(415) = 9.8$, $p < 0.001$, Cohen's d=0.68, 95% CI of difference between means=[0.01, 0.02], two-tailed). Further, on an individual trial basis, human preference for one image of a pair is correlated with the difference in edge strengths of the perturbations forming the pair (Spearman's $\rho(413) = 0.16$, $p = 0.001$) This correlation indicates that participants are in fact sensitive to structural changes made to images, even when those structural changes are subtle (see, for example, Figure 5.5a).

5.4 Discussion

In this work, we showed that subtle adversarial perturbations, designed to alter classification decisions of artificial neural networks, also influence human perception. This influence is revealed when experimental participants perform forced-choice classification tasks, whether image exposures are brief or extended (Figures 5.2 and 5.3, respectively). Reliable effects of adversarial perturbations are observed even when the perturbation magnitudes are so tiny that there is little possibility of overtly transforming the image to be a true instance of the adversarial class. Adversarial perturbations have intrigued the academic community—as well as the the broader population—because the perturbations appear perceptually indistinguishable from noise, in that one might expect them to degrade human perceptual accuracy but not bias perception systematically in the same direction as neural networks are biased.

Even though our adversarial manipulations induced a reliable change in human perception, this change is not nearly as dramatic as what is found in artificial neural nets, which completely switch their classification decisions in response to adversarial perturbations. For example, in Experiment 4, whereas our participants chose the response consistent with the adversarial perturbation on 52% of trials (for epsilon=2), a black-box attack on a neural net showed a two-alternative choice preference consistent with the adversarial perturbation on 85.3% of trials. (A black-box attack refers to the fact that the neural net used for testing is different than the model used to generate the adversarial perturbation in the first place.)

One minor factor for the weak human response is that on any trial where participants are inattentive to the stimulus, choice probability will regress to the chance rate of 50%. The more substantive factors reflect fundamental differences between humans and the type of neural networks that are used to obtain adversarial images. While we cannot claim to enumerate all such differences, four points stand out in the literature: 1) Even with millions of training examples, the data that neural network classifiers are exposed to does not reflect the richness of naturalistic learning environments. Consequently, image statistics learned by neural nets are likely deficient. Bhojanapalli et al (Bhojanapalli et al., 2021) and Sun et al (Sun et al., 2019) found that as training corpus size increases, neural networks do show improved robustness to adversarial attacks; this robustness is observed for both convolutional and self-attention models and for a variety of attacks. 2) The models used for generating adversarial perturbations are trained only to classify, whereas human vision is used in the service of a variety of goals. Mao et al (Mao et al., 2020) indeed found that when models are trained on multiple tasks simultaneously, they become more robust to adversarial attacks on the individual tasks. 3) Typical neural networks trained to classify images have at best an implicit representation of three-dimensional structure and of the objects composing a visual scene. Recent vision ANN models have considered explicit figure-ground segmentation, depth representation, and separate encoding of individual objects in a scene. Evidence points to these factors increasing adversarial and other forms of robustness (Xiang et al., 2019; Akumalla et al., 2020; Dittadi et al., 2021). 4) Common ANN architectures for vision are feedforward, whereas a striking feature of visual cortex is the omnipresence of recurrent connectivity. Several recent investigations have found improvements in adversarial robustness for models with lateral recurrent connectivity (Paiton et al., 2020) and reciprocal connectivity between layers (Huang et al., 2020).

What is the explanation for the alignment found between human and machine perception? Because both convolutional and self-attention ANNs—models with quite different architectural details—are able to predict human choice , the alignment cannot primarily be due to ANNs having coarse structural similarities to the neuroanatomy of visual cortex. Indeed, if anything, self-attention models—whose global spatial operations are unlike those in the visual system—are better predictors, though this trend was not statistically reliable. Nonetheless, in a direct comparison between adversarial stimuli designed for the two models, experimental participants strongly choose images that fool self-attention models over images that fool convolutional models (Figure 5.5). Self-attention models have a greater tendency to be fooled by images that are perturbed along contours or edges, and we found a reliable correlation between the presence of contour or edge perturbations and the preference for that adversarial image. An interesting topic for future research would be to explore other techniques that better align human and machine representations (Roads and Love, 2021; Attarian et al., 2020) and to utilize human susceptibility to adversarial perturbations as a diagnostic of that alignment. This study did not include data on sex and age, which may be contributing factors to the susceptibility of humans to adversarial perturbations. Future studies may address this limitation and explore the impact of sex and age.

Taken together, our results suggest that the alignment between human and machine perception is due to the fact that both are exquisitely sensitive to subtle, higher-order statistics of natural images. This further supports the importance of higher order image statistics to neural representations (Simoncelli and Olshausen, 2001). Progress in ANN research has resulted in powerful statistical models that capture correlation structures inherent in an image data set. Our work demonstrates that these models can not only be exploited to reveal previously unnoticed statistical structure in images beyond low-order statistics, but that human perception is influenced by this structure.

5.5 Results statistics and analysis tables

Experiment-wise statistics table

For each of our experiments 1-5 in this Chapter, we conducted either a two-tailed t-test or a 2-way ANOVA (depending on experiment design). The statistics corresponding to these tests along with number of degrees of freedom, the test-statistic value, p-value and effect size of the result are reported here.

Table 5.2. Statistics table for all experiments

Analysis identifier	Test type	Exact n	Degrees of freedom	Test statistic	p value	effect size
E1: $A\uparrow$ vs control	Two-tailed t test	91	90	t=4.46	$p < 0.001$	Cohen's d=0.66
E2: $T\uparrow$ vs $T\downarrow$– ε	2-way ANOVA	1556	(3, 1155)	F=10.73	$p < 0.001$	η_p^2=0.03
E2: $T\uparrow$ vs $T\downarrow$– target class	2-way ANOVA	1556	(3, 385)	F=12.94	$p < 0.001$	η_p^2=0.09
E2: $T\uparrow$ vs $T\downarrow$– target class x ε	2-way ANOVA	1556	(9,1155)	F=1.78	$p = 0.067$	η_p^2=0.01
E3: $A\uparrow$ vs control – ε	2-way ANOVA	1584	(3, 1176)	F=22.08	$p < 0.001$	η_p^2=0.05
E3: $A\uparrow$ vs control – target class	2-way ANOVA	1584	(3, 392)	F=19.86	$p < 0.001$	η_p^2=0.13
E3: $A\uparrow$ vs control – target class x ε	2-way ANOVA	1584	(9,1176)	F=2.55	$p < 0.01$	η_p^2=0.02
E4: $A\uparrow$ vs $A'\uparrow$– ε	2-way ANOVA	1556	(3, 1155)	F=17.48	$p < 0.001$	η_p^2=0.04
E4: $A\uparrow$ vs $A'\uparrow$– target class	2-way ANOVA	1556	(3, 385)	F=16.84	$p < 0.001$	η_p^2=0.12
E4: $A\uparrow$ vs $A'\uparrow$– target class x ε	2-way ANOVA	1556	(9,1155)	F=4.25	$p < 0.001$	η_p^2=0.03
E4: $A\uparrow$ vs $A'\uparrow$– ε =2, Bonferonni correction for 4 comparisons	Two-tailed t test	389	388	t=3.45	$p < 0.001$	Cohen's d=0.18
E5: self-attention vs convolution	Two-tailed t test	396	395	t=18.25	$p < 0.001$	Cohen's d=0.92
E5: Edge strength(conv) >Edge strength(self-attention)	Two-tailed t test	415	414	t=9.8	$p < 0.001$	Cohen's d = 0.68

5.5.1 Alternate analysis of perceptual bias without assuming normal distribution of perceptual bias

In this section, we analyze compute the significane of perceptual bias observed in Experiments 1–5 using the non-parametric Wilcoxon test (one-sided). Unlike the t-test, this alternate test does not assume normality of the input samples and is hence a more rigorous evaluation of the observed significance.

Table 5.3. Wilcoxon test (one-sided) of significance of the main effect for Experiments 1-5

Experiment name	Wilcoxon statistic	p-value	RBC (effect size)	95% CI of bias
Experiment 1	647	$p < 0.001$	0.746	[0.052, ∞]
Experiment 2	60302	$p < 0.001$	0.598	[0.031, ∞]
Experiment 3	60754	$p < 0.001$	0.546	[0.045, ∞]
Experiment 4	58526	$p < 0.001$	0.543	[0.026, ∞]
Experiment 5	65221	$p < 0.001$	0.821	[0.103, ∞]

5.5.2 95% confidence interval of perceptual bias grouped by perturbation magnitude

In this Table below, we computed the 95% confidence interval of the human perceptual bias we observed in Experiments 2–4. In each of these experiments, we grouped trials with the same perturbation magnitude and report the 95% CI of perceptual bias at each magnitude. One can see here too, that the range of perceptual bias increases with increasing perturbation magnitude in all experiments.

Table 5.4. Experiments 2–4 - 95% CI of perceptual bias by ε

Experiment	Perturbation magnitude	95% CI of perceptual bias
Experiment 2	$\varepsilon = 2$	[0.51, 0.53]
	$\varepsilon = 4$	[0.52, 0.54]
	$\varepsilon = 8$	[0.53, 0.56]
	$\varepsilon = 16$	[0.55, 0.57]
Experiment 3	$\varepsilon = 2$	[0.51, 0.53]
	$\varepsilon = 4$	[0.54, 0.56]
	$\varepsilon = 8$	[0.55, 0.58]
	$\varepsilon = 16$	[0.57, 0.60]
Experiment 4	$\varepsilon = 2$	[0.51, 0.53]
	$\varepsilon = 4$	[0.51, 0.54]
	$\varepsilon = 8$	[0.54, 0.56]
	$\varepsilon = 16$	[0.55, 0.58]

5.5.3 Above-chance preference for adversarial image

In this Table, we tested the likelihood of above-chance preference of participants to the adversarial image in Experiments 2–4. In each of these experiments, we grouped participants' perceptual bias by object category (averaging across perturbation magnitude) and ran a one-sided t-test at each object category testing whether the distribution of participant preference to the adversarial image was greater than chance. These results are shown in the below table.

Table 5.5. Above-chance preference for adversarial image, by experiment and class

Experiment	Target class(es)	Perceptual bias	test-statistic	p value	effect size (η_p^2)	95% CI of bias
2	dog	0.54	$F(1,99) = 38.9$	$< .001$	0.21	[0.03, 0.05]
	cat	0.57	$F(1,99) = 88.4$	$< .001$	0.39	[0.06, 0.09]
	bottle	0.52	$F(1,98) = 7.61$	.007	0.06	[0.0, 0.03]
	bird	0.53	$F(1,89) = 33.2$	$< .001$	0.14	[0.02, 0.05]
3	dog	0.52	$F(1,99) = 10.6$	.001	0.08	[0.01, 0.04]
	cat	0.62	$F(1,95) = 98.3$	$< .001$	0.50	[0.1, 0.15]
	bottle	0.55	$F(1,98) = 23.9$	$< .001$	0.19	[0.03, 0.07]
	bird	0.54	$F(1,100) = 18.6$	$< .001$	0.09	[0.02, 0.06]
4	cat-truck	0.54	$F(1,97) = 28.5$	$< .001$	0.17	[0.03, 0.06]
	sheep-chair	0.51	$F(1,97) = 6.85$	.010	0.07	[0.0, 0.02]
	dog-bottle	0.53	$F(1,98) = 24.6$	$< .001$	0.39	[0.02, 0.04]
	elephant-clock	0.58	$F(1,93) = 70.6$	$< .001$	0.38	[0.06, 0.1]

5.5.4 Shapiro-Wilk test for normality of per-image responses

In this section, we test the normality of per-image perceptual bias for trials in Experiment 4. From the result of the Shapiro-Wilk test for normality, we cannot discard the null-hypothesis that the per-image responses are normally distributed. These results allow us to analyse our data using 2-way ANOVA and t-test, both of which assume that the input samples are normally distributed.

Table 5.6. Shapiro-Wilk test for normality of per-image responses - Experiment 4

Target class	Perturbation magnitude	W	p value
cat-truck	$\varepsilon = 2$	0.993	0.484
	$\varepsilon = 4$	0.986	0.050
	$\varepsilon = 8$	0.992	0.395
	$\varepsilon = 16$	0.989	0.165
sheep-chair	$\varepsilon = 2$	0.993	0.435
	$\varepsilon = 4$	0.988	0.096
	$\varepsilon = 8$	0.993	0.533
	$\varepsilon = 16$	0.990	0.172
dog-bottle	$\varepsilon = 2$	0.992	0.396
	$\varepsilon = 4$	0.989	0.138
	$\varepsilon = 8$	0.993	0.480
	$\varepsilon = 16$	0.983	0.020
elephant-clock	$\varepsilon = 2$	0.991	0.280
	$\varepsilon = 4$	0.989	0.127
	$\varepsilon = 8$	0.991	0.283
	$\varepsilon = 16$	0.984	0.027

5.5.5 List of ANN models used to create adversarial manipulations in Experiments 1–5

In this section, we list all the models that were used to create adversarial attacks for Experiments 1–5. We have also included the top-1 classification accuracy of these models on the ImageNet validation set. Models in Experiments 2–4 are grouped into one Table as the adversarial image stimuli for these experiments were created using the same ANN models.

Table 5.7. Accuracy of models used in Experiment 1 on ImageNet validation set. * models trained on ImageNet with retina layer pre-pended and with train data augmented with rescaled images in the range of $[40, 255-40]$; ** model trained with adversarial examples augmented data. First ten models are models used in the adversarial training ensemble. Last two models are models used to test the transferability of adversarial examples.

Model	**Top-1 accuracy**
Resnet V2 101	77.0
Resnet V2 101*	72.1
Inception V4	80.2
Inception V4*	75.2
Inception Resnet V2	80.4
Inception Resnet V2*	76.6
Inception V3	78.0
Inception V3*	74.5
Resnet V2 152	77.8
Resnet V2 50*	70.8
Resnet V2 50 (test)	75.6
Inception V3** (test)	77.6

Table 5.8. Top-1 classification accuracy (%) of models used in Experiments 2 – 4 on ImageNet validation set Accuracy of the models ensembled to create adversarial image manipulations in Experiments 2–4. All models were trained on ImageNet-1k, top-1 accuracy here is computed on the ImageNet-1k validation set.

Model name	Top-1 accuracy
EfficientNet-B4	83
EfficientNet-B5	83.7
ResNet-101	79.8
InceptionV4	80
PNASNet-5	82.9
HaloNet	80.6

Table 5.9. Top-1 classification accuracy (%) of models used in Experiment 5 on ImageNet validation set Accuracy of the models ensembled to create adversarial image manipulations in Experiments 5. Top-1 accuracy here is computed on the ImageNet-1k validation set.

Model name	Top-1 accuracy
ViT-B/32	80.7
ResNet-101x1	80.7
ViT-L/16	87.1
ResNet-200x3	87.2

5.5.6 Shape-bias of convolution- and self-attention networks.

In this section, we report the accuracy of convolution and self-attention networks used in Experiment 5 on the Shape-vs-Texture dataset introduced in Geirhos et al. (2019b). We report the fraction of decisions made by these networks based on shape and texture respectively. This analysis shows that ANNs with self-attention are relatively more shape-biased than convolutional ANNs.

Table 5.10. Shape bias of convolution networks

Category	shape decisions	texture decisions	other decisions	shape bias
boat	6	49	29	10.9
knife	7	41	33	14.6
airplane	5	46	33	9.80
bear	19	44	22	30.2
dog	15	40	29	27.3
keyboard	28	42	15	40.0
cat	27	33	25	45.0
elephant	21	38	26	35.6
oven	18	35	32	34.0
bird	10	44	31	18.5
car	39	33	13	54.2
truck	39	34	12	53.4
bottle	54	26	5	67.5
chair	29	29	27	50.0
clock	62	14	9	81.6
bicycle	35	32	18	52.2
Total	414	580	359	41.7

Table 5.11. Shape bias of self-attention networks

Category	shape decisions	texture decisions	other decisions	shape bias
boat	10	46	29	17.9
knife	15	39	29	27.8
airplane	17	46	22	27.0
bear	25	43	17	36.8
dog	15	45	24	25.0
keyboard	23	43	19	34.9
cat	25	43	17	36.8
elephant	38	29	18	56.7
oven	34	33	18	50.8
bird	12	57	16	17.4
car	41	30	14	57.8
truck	45	36	4	55.6
bottle	56	27	2	67.5
chair	41	31	13	56.9
clock	71	10	4	87.7
bicycle	46	25	14	64.8
Total	514	583	260	46.9

5.5.7 ImageNet-C top-1 accuracy of convolution and attention networks

In this section, we report the accuracy of convolution and self-attention networks used in Experiment 5 on the ImageNet-C dataset introduced in Hendrycks and Dietterich (2019). We report the accuracy of these networks on ImageNet validation images subject to various kinds of image distortions at 5 levels of severity. This analysis also shows that for many perceptual distortions, ANNs with self-attention are relatively more robust to perceptual distortions than convolutional ANNs.

Table 5.12. ImageNet-C Top 1 accuracy for convolution model. Model accuracy on clean ImageNet validation set is 86.3%

Noise Type	Scale 1	Scale 2	Scale 3	Scale 4	Scale 5
Gaussian noise	81.9	79.3	73.6	63.5	44.6
Shot noise	81.5	78.5	73.1	60.3	46.9
Impulse noise	80.0	77.0	73.8	64.2	48.6
Defocus blur	78.9	75.1	65.6	54.9	43.4
Glass blur	76.7	69.2	46.9	38.3	25.4
Motion blur	83.1	80.1	73.6	63.0	54.1
Zoom blur	77.0	71.9	67.1	61.3	53.8
Snow	80.1	72.0	73.5	67.2	60.8
Frost	80.0	72.6	65.7	64.2	58.9
Fog	83.2	82.5	81.2	80.0	75.5
Brightness	84.6	83.9	82.8	80.8	78.0
Contrast	83.4	82.6	80.9	74.7	60.0
Elastic transform	81.0	63.8	75.6	67.2	42.8
Pixelate	82.9	81.8	79.7	75.6	72.1
JPEG compression	80.4	78.9	77.8	73.5	67.2
Gaussian blur	82.4	76.5	67.9	58.3	38.3
Saturate	81.9	79.6	83.9	79.8	74.4
Spatter	84.0	81.5	79.1	76.7	70.6
Speckle noise	82.1	80.6	74.6	69.7	61.8
AVG	81.3	77.2	73.5	67.0	56.7
SEM	0.50	1.21	1.96	2.40	3.30

Table 5.13. ImageNet-C Top 1 accuracy for attention model. Model accuracy on clean ImageNet validation set is 86.6%

Noise Type	Scale 1	Scale 2	Scale 3	Scale 4	Scale 5
Gaussian noise	83.6	81.1	75.6	64.3	42.2
Shot noise	83.3	80.4	75.1	60.4	44.9
Impulse noise	82.6	79.5	76.1	65.0	46.4
Defocus blur	81.7	79.1	71.6	62.9	53.2
Glass blur	80.9	76.5	60.1	53.5	42.0
Motion blur	84.3	82.5	78.4	71.2	64.8
Zoom blur	79.3	74.9	70.8	66.2	59.3
Snow	81.9	75.7	77.1	72.4	67.6
Frost	81.7	75.7	69.4	68.0	62.9
Fog	83.7	82.8	81.6	80.6	76.5
Brightness	85.5	84.9	83.9	82.5	80.1
Contrast	84.3	83.4	81.4	74.0	52.4
Elastic transform	82.7	67.5	80.3	74.7	53.5
Pixelate	84.4	83.8	82.7	80.3	78.1
JPEG compression	82.3	81.2	80.2	76.8	70.4
Gaussian blur	84.1	80.7	75.1	67.8	51.1
Saturate	83.5	81.3	84.9	81.5	77.2
Spatter	85.0	83.2	81.4	81.1	77.5
Speckle noise	83.8	82.6	77.3	72.5	64.3
AVG	83.1	79.8	77.0	71.4	61.3
SEM	4.12	4.00	3.92	3.80	4.00

5.6 Acknowledgements

Chapter 5, in part, is obtained from the following publication. I, the dissertation author, was the primary investigator and author of this material.

Veerabadran, V., Goldman, J., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., Shlens, J., Sohl-Dickstein, J., Mozer, M. C., et al. (2023a). Subtle adversarial image manipulations influence both human and machine perception. *Nature Communications*, 14(1):4933

We are grateful to Bruno Olshausen, David Sussillo, John Cunningham, Zoubin Ghahramani, Vidhya Navalpakkam, Simon Kornblith, Been Kim, Ross Goroshin, Virginia de Sa, Ari Morcos, Hanlin Tang, Santani Teng, Daniel Yamins, Pradeep Shenoy, Alexey Dosovitskiy, Katherine Lee, Niru Maheswaranathan, Catherine Olsson, Jing Yang Zhou, Kohitij Kar, William P. McCarthy, and Dan Abolafia for useful discussions. We also thank Deanna Chen, Leslie Philips, Sally Jesmonth, Phing Turner, Melissa Strader, Lily Peng, and Ricardo Prada for assistance with IRB and experiment setup. We thank Tom Small for the help with visualizations.

Chapter 6

Conclusions and future directions

6.1 Conclusions

We advance two key research directions at the intersection of neuroscience with AI with the goal of contributing to human-aligned machine vision. In this Section, we summarize our key conclusions from Chapters 2 - 5. These findings are discussed in more detail at the end of each Chapter respectively.

(1) **Bio-Inspired Modeling of Computer Vision Algorithms:** Drawing inspiration from established computational neuroscience models of human visual perception, we crafted sophisticated models of lateral connections aimed to enhance spatial processing in DNNs. As discussed in Chapters 2 and 3, we developed feedforward and recurrent models of lateral connections which produced biologically realistic orientation-tuned receptive fields, robustness to out-of-distribution distortions and interpretable contour integration. We described in Chapter 4 how we introduced the joint training of these recurrent neural network models of lateral connections with Adaptive Computation Time (ACT), resulting in adaptive RNNs (AdRNNs) that learn (from unlabelled data) to scale their computation as a function of instance-level difficulty. These AdRNN models demonstrated superior performance against various challenging computer vision & visual reasoning benchmarks, surpassing state-of-the-art (non bio-inspired) deep learning architectures and showed excellent generalization to novel difficulty levels of the training task; and further the model and human response time trends correlated strongly on two

challenging visual reasoning problems. These above Chapters together highlighted the potency of careful bio-inspired architecture design and their integration with gradient-based training in producing human-like generalization characteristics to out-of-distribution settings in DNN computer vision models.

(2) **Assessment of Human-Machine Perceptual Gap Using Psychophysics:** Complementary to brain-inspired deep learning modeling, we also designed and conducted a comprehensive human-machine comparison study to assess the perceptual gap between human and machine vision. In Chapter 5, we described the design, analysis and results from this time-unlimited visual psychophysics study where we recorded human behavioral response to adversarial image manipulations in a set of 2AFC tasks. The analysis of the correspondence between human and AI responses to these manipulations in these tasks quantified the degree of human vision's alignment with DNN-based computer vision models, highlighting significant parallels between them. Our analysis identified (1) that learning of shared feature sensitivities is an integral element to enhancing human-machine alignment, and (2) factors of divergence between ANNs and biological brains which can be addressed by careful design of future deep learning architectures and training methods to further minimize the perceptual gap between them.

6.2 Future directions

Although we showed that our models of lateral connections—DivNormEI and LocRNN—are capable of generalizing to out-of-distribution perceptual distortions and novel test-time difficulty levels, we lack a sound interpretation of the model's components and their working. In future work, we will perform a deep-dive into mechanistically understanding them. We will study the role of various ablations of our networks, especially an ablation of the split populations of LocRNN. A principled understanding of the role of interneurons in our models can potentially better inform us about the role of interneurons in biological brains during spatial processing. From initial experimental results on the Mazes challenge, we found the division of

LocRNN's neurons into a population of principal neurons and a population of interneurons to be important for maintaining generalization ability. We will conduct in-silico electrophysiology and state-space visualization to interpret this ablation model's functioning. We shall further study whether such a division of LocRNN's neurons helps its generalization performance on PathFinder and on natural image computer vision benchmarks such as ImageNet-1k and ImageNet-C. Ultimately, observations from these new experiments will help us form a better picture of the how interneurons support robust spatial information processing in biological vision.

Next, through our human-machine comparison studies in Chapter 5 we concluded that a shared feature sensitivity between human and machine vision is essential to maximizing their alignment and we showed that this alignment in the space of current ANNs has significant room for growth. Our predictions elicit ANN design factors that can influence the learning of a shared feature sensitivity with human vision. Future work must follow-up on whether using ANNs designed with recurrent lateral connections and top-down feedback connections are better aligned with human vision; this line of research unifies our two directions of bio-inspired modeling of vision and human-machine comparison studies. Additionally, the availability of high resolution neural recordings also allows us to regularize ANNs to be predictive of primate visual neural responses. We hypothesize that such regularized ANNs optimized for representational similarity with primate vision will also produce emergent behavioral alignment. We are working with collaborators on collecting high resolution primate cortical recordings in response to novel visual stimuli which will will be used for training the next generation of state-of-the-art encoding models of primate ventral visual responses. With a combination of bio-inspired design, training, and human-machine perceptual comparison, we believe that future work expanding on this dissertation research will contribute towards the development of highly performant biologically inspired and aligned models for use in computer vision tasks—especially in critical applications where we seek maximal alignment between human and machine perception.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th $\{$USENIX$\}$ Symposium on Operating Systems Design and Implementation ($\{$OSDI$\}$ 16)*, pages 265–283.

Akumalla, A., Haney, S., and Bazhenov, M. (2020). Contextual fusion for adversarial robustness.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2010). Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916.

Arnab, A., Zheng, S., Jayasumana, S., Romera-Paredes, B., Larsson, M., Kirillov, A., Savchynskyy, B., Rother, C., Kahl, F., and Torr, P. H. (2018). Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation: Combining Probabilistic Graphical Models with Deep Learning for Structured Prediction. *IEEE Signal Processing Magazine*, 35(1):37–52.

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. (2018). Synthesizing robust adversarial examples. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 284–293. PMLR.

Attarian, I. M., Roads, B. D., and Mozer, M. C. (2020). Transforming neural network visual representations to predict human judgments of similarity. In *NeurIPS Workshop on Shared Visual Representations Between Humans and Machines*.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Baker, N., Lu, H., Erlikhman, G., and Kellman, P. J. (2018). Deep convolutional networks do not classify based on global object shape. *PLoS Computational Biology*, 14(12).

Ballas, N., Yao, L., Pal, C., and Courville, A. (2015). Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*.

Ballé, J., Laparra, V., and Simoncelli, E. P. (2015). Density modeling of images using a generalized normalization transformation. *arXiv preprint arXiv:1511.06281*.

Banino, A., Balaguer, J., and Blundell, C. (2021). Pondernet: Learning to ponder. *arXiv preprint arXiv:2107.05407*.

Bansal, A., Schwarzschild, A., Borgnia, E., Emam, Z., Huang, F., Goldblum, M., and Goldstein, T. (2022). End-to-end algorithm synthesis with recurrent networks: Logical extrapolation without overthinking. *arXiv preprint arXiv:2202.05826*.

Berardino, A., Laparra, V., Ballé, J., and Simoncelli, E. (2017). Eigen-distortions of hierarchical representations. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T., and Veit, A. (2021). Understanding robustness of transformers for image classification. *arXiv preprint arXiv:2103.14586*.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Srndic, N., Laskov, P., Giacinto, G., and Roli, F. (2013). Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III*, pages 387–402.

Blakeslee, B. and McCourt, M. E. (1999). A multiscale spatial filtering account of the white effect, simultaneous brightness contrast and grating induction. *Vision research*, 39(26):4361–4377.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Brendel, W. and Bethge, M. (2019a). Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations*.

Brendel, W. and Bethge, M. (2019b). Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet. *7th International Conference on Learning Representations, ICLR 2019*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Burg, M. F., Cadena, S. A., Denfield, G. H., Walker, E. Y., Tolias, A. S., Bethge, M., and Ecker, A. S. (2021). Learning divisive normalization in primary visual cortex. *PLOS Computational*

Biology, 17(6):e1009028.

Busse, L., Wade, A. R., and Carandini, M. (2009). Representation of concurrent stimuli by population activity in visual cortex. *Neuron*, 64(6):931–942.

Cadieu, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., Majaj, N. J., and DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS computational biology*, 10(12):e1003963.

Carandini, M. and Heeger, D. J. (2012). Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13(1):51–62.

Carter, B., Jain, S., Mueller, J. W., and Gifford, D. (2021). Overinterpretation reveals image classification model pathologies. *Advances in Neural Information Processing Systems*, 34.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018a). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848.

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018b). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11211 LNCS, pages 833–851. Springer Verlag.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. (2020). Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-January, pages 1800–1807. Institute of Electrical and Electronics Engineers Inc.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2023). Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee. IEEE. https://doi.org/10.1109/CVPR.2009.5206848.

Dittadi, A., Papa, S., De Vita, M., Schölkopf, B., Winther, O., and Locatello, F. (2021). Generalization and robustness implications in object-centric learning. *arXiv preprint arXiv:2107.00637*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dujmović, M., Malhotra, G., and Bowers, J. S. (2020). What do adversarial images tell us about human vision? *Elife*, 9:e55978.

Eigen, D., Rolfe, J., Fergus, R., and LeCun, Y. (2013). Understanding deep architectures using a recursive convolutional network. *arXiv preprint arXiv:1312.1847*.

Elder, J. H. (2018). Shape from Contour: Computation and Representation. *Annual Review of Vision Science*, 4(1):423–450.

Elder, J. H., Oleskiw, T. D., and Fruend, I. (2018). The role of global cues in the perceptual grouping of natural shapes. *Journal of Vision*, 18(12):1–21.

Eyzaguirre, C. and Soto, A. (2020). Differentiable adaptive computation time for visual reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12817–12825.

Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., and Feichtenhofer, C. (2021). Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835.

Feather, J., Leclerc, G., Mądry, A., and McDermott, J. H. (2023). Model metamers reveal divergent invariances between biological and artificial neural networks. *Nature Neuroscience*, 26(11):2017–2034.

Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211.

Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47.

Field, D. J., Hayes, A., and Hess, R. F. (1993). Contour integration by the human visual system:

Evidence for a local "association field". *Vision Research*, 33(2):173–193.

Firestone, C. (2020). Performance vs. competence in human–machine comparisons. *Proceedings of the National Academy of Sciences*, 117(43):26562–26571.

Ford, N., Gilmer, J., Carlini, N., and Cubuk, E. D. (2019). Adversarial examples are a natural consequence of test error in noise. In *International Conference on Machine Learning*, pages 2280–2289. PMLR.

Fukushima, K. (1980a). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.

Fukushima, K. (1980b). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

Fukushima, K. and Miyake, S. (1982). Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern recognition*, 15(6):455–469.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Geirhos, R., Medina Temme, C., Rauber, J., Schütt, H., Bethge, M., and Wichmann, F. (2019a). Generalisation in humans and deep neural networks. In *Thirty-second Annual Conference on Neural Information Processing Systems 2018 (NeurIPS 2018)*, pages 7549–7561. Curran.

Geirhos, R., Narayanappa, K., Mitzkus, B., Thieringer, T., Bethge, M., Wichmann, F. A., and Brendel, W. (2021). Partial success in closing the gap between human and machine vision. *arXiv preprint arXiv:2106.07411*.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2018a). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2019b). Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*.

Geirhos, R., Temme, C. R. M., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. (2018b). Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems 31*.

Gilbert, C. D. and Wiesel, T. N. (1989). Columnar specificity of intrinsic horizontal and corticocortical connections in cat visual cortex. *Journal of Neuroscience*, 9(7):2432–2422.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Graves, A. (2016). Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., and Wierstra, D. (2015). DRAW: A Recurrent Neural Network For Image Generation. *International Conference in Machine Learning*, 4(7540):14.

Grossberg, S. and Raizada, R. D. (2000). Contrast-sensitive perceptual grouping and object-based attention in the laminar circuits of primary visual cortex. *Vision research*, 40(10-12):1413–1432.

Han, C., Yoon, W., Kwon, G., Kim, D., and Nam, S. (2019). Representation of white-and black-box adversarial examples in deep neural networks and humans: A functional magnetic resonance imaging study. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Hasani, H., Baghshah, M. S., and Aghajan, H. (2019). Surround Modulation: A Bio-inspired Connectivity Structure for Convolutional Neural Networks. Technical report.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hendrycks, D. and Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

Huang, Y., Gornet, J., Dai, S., Yu, Z., Nguyen, T., Tsao, D. Y., and Anandkumar, A. (2020). Neural networks with recurrent generative feedback. *arXiv preprint arXiv:2007.09200*.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154.

Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial examples are not bugs, they are features. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

Iuzzolino, M., Mozer, M. C., and Bengio, S. (2021). Improving anytime prediction with parallel cascaded networks and a temporal-difference loss. *Advances in Neural Information Processing Systems*, 34:27631–27644.

Jacobs, R. A. and Bates, C. J. (2019). Comparing the visual representations and performance of humans and deep neural networks. *Current Directions in Psychological Science*, 28(1):34–39.

Jin, J., Dundar, A., and Culurciello, E. (2014). Flattened Convolutional Neural Networks for Feedforward Acceleration. *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*.

Jolicoeur, P. and Ingleton, M. (1991). Size invariance in curve tracing. *Memory & Cognition*, 19(1):21–36.

Jolicoeur, P., Ullman, S., and Mackay, M. (1986). Curve tracing: A possible basic operation in the perception of spatial relations. *Memory & Cognition*, 14(2):129–140.

Kar, K., Jonas, K., Schmidt, K., Issa, E. B., and DiCarlo, J. (2019). Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nature Neuroscience*.

Khaligh-Razavi, S.-M. and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, 10(11):e1003915.

Kim, B., Reif, E., Wattenberg, M., Bengio, S., and Mozer, M. C. (2021). Neural networks trained on natural scenes exhibit gestalt closure. *Computational Brain & Behavior*, 4(3):251–263.

Kim, E., Rego, J., Watkins, Y., and Kenyon, G. T. (2020). Modeling biological immunity to adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*

Pattern Recognition, pages 4666–4675.

Kim, J., Ricci, M., and Serre, T. (2018). Not-so-clevr: learning same–different relations strains feedforward neural networks. *Interface focus*, 8(4):20180011.

Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. (2021). Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR.

Kreiman, G. and Serre, T. (2020). Beyond the feedforward sweep: feedback computations in the visual cortex. *Annals of the New York Academy of Sciences*, 1464(1):222–241.

Kriegeskorte, N. (2015). Deep neural networks: A new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science*, 1(1):417–446.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Kubilius, J., Bracci, S., and Op de Beeck, H. P. (2016). Deep neural networks as a computational model for human shape sensitivity. *PLoS Comput Biol*, 12(4).

Kubilius, J., Schrimpf, M., Nayebi, A., Bear, D., Yamins, D. L., and DiCarlo, J. J. (2018a). Cornet: Modeling the neural mechanisms of core object recognition. *BioRxiv*, page 408385.

Kubilius, J., Schrimpf, M., Nayebi, A., Bear, D., Yamins, D. L. K., and DiCarlo, J. J. (2018b). CORnet: Modeling the Neural Mechanisms of Core Object Recognition. *bioRxiv*, page 408385.

Kurakin, A., Goodfellow, I., and Bengio, S. (2016). Adversarial examples in the physical world. In *ICLR'2017 Workshop*.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., and Ferrari, V. (2020). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*. Springer Nature. https://doi.org/10.1007/s11263-020-01316-z.

Lamme, V. A., Supèr, H., and Spekreijse, H. (1998). Feedforward, horizontal, and feedback processing in the visual cortex. *Current Opinion in Neurobiology*, 8(4):529–535.

Laskar, M. N. U., Giraldo, L. G. S., and Schwartz, O. (2018). Correspondence of Deep Neural Networks and the Brain for Visual Textures.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

LeCun, Y. et al. (1989). Generalization and network design strategies. connectionism in perspective. *Zurich, Switzerland, Elsiever*.

Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., and Hutter, M. (2020). Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47).

Li, W., Piëch, V., and Gilbert, C. D. (2006). Contour saliency in primary visual cortex. *Neuron*, 50(6):951–962.

Li, Z. (1998). A neural model of contour integration in the primary visual cortex. *Neural computation*, 10(4):903–940.

Liao, Q. and Poggio, T. (2016a). Bridging the Gaps Between Residual Learning, Recurrent Neural Networks and Visual Cortex.

Liao, Q. and Poggio, T. (2016b). Bridging the gaps between residual learning, recurrent neural networks and visual cortex. *arXiv preprint arXiv:1604.03640*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48.

Lindsay, G. W. (2020). Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of cognitive neuroscience*, pages 1–15.

Linsley, D., Karkada Ashok, A., Govindarajan, L. N., Liu, R., and Serre, T. (2020). Stable and expressive recurrent vision models. *Advances in Neural Information Processing Systems*, 33:10456–10467.

Linsley, D., Kim, J., Veerabadran, V., Windolf, C., and Serre, T. (2018). Learning long-range spatial dependencies with horizontal gated recurrent units. In *Advances in Neural Information Processing Systems*, pages 152–164.

Liu, Y., Chen, X., Liu, C., and Song, D. (2016). Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986.

Loffler, G. (2008). Perception of contours and shapes: Low and intermediate stage mechanisms. *Vision Research*, 48(20):2106–2127.

Long, B. and Konkle, T. (2018). The role of textural statistics vs. outer contours in deep cnn and neural responses to objects. In *Conference on Computational Cognitive Neuroscience*, volume 4.

Mao, C., Gupta, A., Nitin, V., Ray, B., Song, S., Yang, J., and Vondrick, C. (2020). Multitask learning strengthens adversarial robustness. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 158–174. Springer.

Mason, W. and Suri, S. (2012). Conducting behavioral research on amazon's mechanical turk. *Behavior research methods*, 44(1):1–23.

Miller, J. and Hardt, M. (2018). Stable recurrent models. *arXiv preprint arXiv:1805.10369*.

Mozer, M. C. (1991). *The perception of multiple objects: A connectionist approach*. MIT Press, Cambridge, MA.

Naseer, M., Ranasinghe, K., Khan, S., Hayat, M., Khan, F. S., and Yang, M.-H. (2021). Intriguing properties of vision transformers. *arXiv preprint arXiv:2105.10497*.

Nayebi, A., Bear, D., Kubilius, J., Kar, K., Ganguli, S., Sussillo, D., DiCarlo, J. J., and Yamins, D. L. (2018). Task-driven convolutional recurrent models of the visual system. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 5290–5301. Curran Associates, Inc.

Nayebi, A., Sagastuy-Brena, J., Bear, D. M., Kar, K., Kubilius, J., Ganguli, S., Sussillo, D., DiCarlo, J. J., and Yamins, D. L. (2022). Recurrent connections in the primate ventral visual stream mediate a tradeoff between task performance and network size during core object recognition. *bioRxiv*, pages 2021–02.

Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Nowak, L. G. and Barone, P. (2009). Contrast adaptation contributes to contrast-invariance of orientation tuning of primate v1 cells. *PLoS one*, 4(3):e4781.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016a). WaveNet: A Generative Model for Raw Audio.

Oord, A. v. d., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., and Kavukcuoglu, K. (2016b). Conditional Image Generation with PixelCNN Decoders. *Advances in Neural Information Processing Systems*, pages 4797–4805.

Paiton, D. M., Frye, C. G., Lundquist, S. Y., Bowen, J. D., Zarcone, R., and Olshausen, B. A. (2020). Selectivity and robustness of sparse coding networks. *Journal of Vision*, 20(12):10–10.

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

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016b). The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. (2018). Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639.

Pinheiro, P. and Collobert, R. (2014). Recurrent convolutional neural networks for scene labeling. In *International conference on machine learning*, pages 82–90. PMLR.

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., and Dosovitskiy, A. (2021). Do vision transformers see like convolutional neural networks? *arXiv preprint arXiv:2108.08810*.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. (2019). Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*.

Roads, B. D. and Love, B. C. (2021). Enriching imagenet with human similarity judgments and psychological embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3547–3557.

Robinson, A. E., Hammon, P. S., and de Sa, V. R. (2007). A filtering model of brightness perception using frequency-specific locally-normalized oriented difference-of-gaussians (flodog). *Journal of Vision*, 7(9):237–237.

Roelfsema, P. R. (2006). CORTICAL ALGORITHMS FOR PERCEPTUAL GROUPING. *Annual Review of Neuroscience*, 29(1):203–227.

Roelfsema, P. R., Lamme, V. A., and Spekreijse, H. (2000). The implementation of visual routines. *Vision research*, 40(10-12):1385–1411.

Rumelhart, D. E., McClelland, J. L., and Group, P. R., editors (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 2: Psychological and Biological Models*. MIT Press, Cambridge, MA, USA.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Schwartz, O. and Simoncelli, E. P. (2001). Natural signal statistics and sensory gain control. *Nature neuroscience*, 4(8):819–825.

Schwarzschild, A., Borgnia, E., Gupta, A., Huang, F., Vishkin, U., Goldblum, M., and Goldstein, T. (2021). Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. *Advances in Neural Information Processing Systems*, 34:6695–6706.

Schwarzschild, A., Gupta, A., Goldblum, M., and Goldstein, T. (2022). The uncanny similarity of recurrence and depth. *10th International Conference on Learning Representations, ICLR 2022*.

Serre, T. (2019). Deep learning: The good, the bad, and the ugly. *Annual Review of Vision Science*, 5(1):399–426.

Simoncelli, E. P. and Olshausen, B. A. (2001). Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24(1):1193–1216.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Spoerer, C. J., Kietzmann, T. C., Mehrer, J., Charest, I., and Kriegeskorte, N. (2020). Recurrent neural networks can explain flexible trading of speed and accuracy in biological vision. *PLoS computational biology*, 16(10):e1008215.

Subramanian, A., Sizikova, E., Majaj, N. J., and Pelli, D. G. (2023). Spatial-frequency channels, shape bias, and adversarial robustness. *arXiv preprint arXiv:2309.13190*.

Sun, K., Zhu, Z., and Lin, Z. (2019). Towards understanding adversarial examples systematically: Exploring data size, task and model factors. *arXiv preprint arXiv:1902.11019*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., and Metzler, D. (2020). Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*.

Tian, Y., Krishnan, D., and Isola, P. (2020). Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Triggs, B. and Verbeek, J. J. (2008). Scene segmentation with crfs learned from partially labeled images. In *Advances in neural information processing systems*, pages 1553–1560.

Troyer, T. W., Krukowski, A. E., Priebe, N. J., and Miller, K. D. (1998). Contrast-invariant orientation tuning in cat visual cortex: thalamocortical input tuning and correlation-based intracortical connectivity. *Journal of Neuroscience*, 18(15):5908–5927.

Tuli, S., Dasgupta, I., Grant, E., and Griffiths, T. L. (2021). Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*.

Ullman, S. (1984). Visual routines. *Cognition*, 18(1-3):97–159.

Van Den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. In *33rd International Conference on Machine Learning, ICML 2016*, volume 4, pages 2611–2620. International Machine Learning Society (IMLS).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Veerabadran, V. and de Sa, V. R. (2020). Learning compact generalizable neural representations supporting perceptual grouping. *arXiv preprint arXiv:2006.11716*.

Veerabadran, V., Goldman, J., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., Shlens, J., Sohl-Dickstein, J., Mozer, M. C., et al. (2023a). Subtle adversarial image manipulations influence both human and machine perception. *Nature Communications*, 14(1):4933.

Veerabadran, V., Raina, R., and de Sa, V. R. (2021). Bio-inspired learnable divisive normalization for anns. In *SVRHM 2021 Workshop@ NeurIPS*.

Veerabadran, V., Ravishankar, S., Tang, Y., Raina, R., and de Sa, V. R. (2023b). Adaptive recurrent vision performs zero-shot computation scaling to unseen difficulty levels. In *Thirty-seventh Conference on Neural Information Processing Systems*.

von Neumann, J. (1958). *The Computer and the Brain*. The Silliman Memorial Lectures Series. Yale University Press.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

Wei, C., Fan, H., Xie, S., Wu, C.-Y., Yuille, A., and Feichtenhofer, C. (2022). Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678.

Wenliang, L. K. and Seitz, A. R. (2018). Deep neural networks for modeling visual perceptual learning. *Journal of Neuroscience*, 38(27):6028–6044.

Xiang, C., Qi, C. R., and Li, B. (2019). Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9136–9144.

Xu, X., Xiong, T., Ding, Z., and Tu, Z. (2023). Masqclip for open-vocabulary universal image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 887–898.

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624.

Yamins, D. L. K. and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19:356–365.

Yu, F. and Koltun, V. (2015). Multi-Scale Context Aggregation by Dilated Convolutions. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*.

Yuan, L., Xiao, W., Kreiman, G., Tay, F. E., Feng, J., and Livingstone, M. S. (2020). Adversarial images for the primate brain. *arXiv preprint arXiv:2011.05623*.

Zamir, A. R., Wu, T.-L., Sun, L., Shen, W. B., Shi, B. E., Malik, J., and Savarese, S. (2017). Feedback networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1308–1317.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Zhou, Z. and Firestone, C. (2019). Humans can decipher adversarial images. *Nature communications*, 10(1):1–9.