# Deep learning and neuroscience : a match made in heaven?

Bhavin Yogesh Choksi


# THÈSE

**En vue de l'obtention du**

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

**Délivré par :** *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

---

**Présentée et soutenue le** *Defense date (20/10/2022)* **par :**

Bhavin CHOKSI

---

## Deep Learning and Neuroscience : a match made in heaven ?

---

**JURY**

| | |
|---|---|
| Laurent Perrinet | Président du Jury |
| Umut Güçlü | Rapporteur |
| Gemma Roig | Examiner |
| Jean-Remi King | Examiner |
| Leila Reddy | Directeur de Thèse |
| Rufin VanRullen | Co-directeur de Thèse |

---

**École doctorale et spécialité :**
*CLESCO : Neurosciences*
**Unité de Recherche :**
*CerCo-CNRS (UMR 5549)*
**Directeur(s) de Thèse :**
*Leila Reddy* et *Rufin VanRullen*
**Rapporteurs :**
*Laurent Perrinet* et *Umut Güçlü*

# Deep Learning and Neuroscience : a match made in heaven ?


A DISSERTATION PRESENTED
BY
BHAVIN CHOKSI
TO
CERCO, CNRS UMR 5549


IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF NEUROSCIENCE


UNIVERSITÉ DE TOULOUSE, PAUL SABATIER
TOULOUSE, FRANCE

SEPTEMBER 2022

Thesis advisor: Leila Reddy & Rufin VanRullen Bhavin Choksi

# Deep Learning and Neuroscience: a match made in heaven?

## Abstract

Deep Learning, as a field has tried to build networks that can perform intelligent tasks that previously only humans could perform. While doing this, the field sets an ambitious goal of creating a conscious machine, often symbolized by it's other name – Artificial Intelligence (AI). Up until now, it has been quite successful in making networks good at a few tasks such as classification, captioning, translation, etc. But still, there remains a lot more desired in these networks. Their sensitivity to very tiny perturbations in the input examples—called adversarial perturbations—has baffled the field since almost a decade. Similarly, the generalization ability of the networks to other tasks and categories is another problem that is actively under investigation.

At the same time, Neuroscience has aimed to understand the most profound network known to us—the human brain. For this quest, it has often relied on using substitute models, mathematical or biological, which are easier to experiment with and understand. But so far, Neuroscience has lacked an apt model to ask questions that go beyond neuronal cytoarchitectures and synapses; especially those regarding the structure of abstract representations in the brain. Investigating a question like *How a human learns or even represents a concept such as Death and links it to Fear and Sadness?* has remained challenging even using other primates such as macaque monkey.

The current thesis argues that these two fields, which have historically always been relevant for each other, can even now help each other in these regards. To illustrate this, the thesis first proposes recurrent dynamics for machine learning models using concepts from neuroscience, specifically a popular neurocomputational theory called predictive coding, and implements them into deep neural networks. It demonstrates that the resulting recurrent networks are more robust to various types of noise, natural and adversarial, when compared to their feedforward counterparts. Importantly, it reports that this robustness is achieved by the ability of the predictive coding dynamics to help the networks *project* the noisy

representations towards their clean versions that are learned during training—a property labeled as *projection towards the manifold.*

Second, going in the reverse direction, it uses neural networks for the benefit of neuroscience. First it compares various networks trained with different objectives—uni- or multimodality, robustness, or dataset sizes—in their ability to explain brain activity measured using functional Magnetic Resonance Imaging, or fMRI. The thesis then reports the uncanny ability of multimodal networks, i.e., networks trained with datasets spanning various modalities, to explain the activity of hippocampus—a region known to possess modality invariant concept cells. Later, it uses a region agnostic approach of systematically looking at smaller portions of voxels in the brain. Using such a searchlight-based approach, it reports that compared to other models, multimodal networks explained the fMRI activity better throughout the visual cortex, while also explaining the regions surrounding superior temporal sulcus.

Thus, overall in this overarching ambition of bridging the gap between the two fields, an aspiration also harboured by an emerging community of NeuroAI, the current thesis attempts to provide additional reasons as to why the two could be a match made in heaven.

# Abstract en Français

Deep Learning, en tant que domaine, a essayé de construire des réseaux capables d'effectuer des tâches intelligentes que seuls les humains pouvaient auparavant effectuer. Ce faisant, l'objectif fixé est ambitieux puisqu'il s'agit de créer une machine consciente, communément appelée intelligence artificielle (IA). Jusqu'à présent, le Deep Learning est efficace pour une série de tâches telles que la classification, le sous-titrage, la traduction, etc. Mais il reste encore beaucoup plus à souhaiter de ces réseaux. Leur sensibilité à de très petites perturbations dans les inputs — appelées perturbations adversaires — a déconcerté le domaine depuis près d'une décennie. De même, la capacité de généralisation des réseaux à d'autres tâches et catégories est une limite activement étudiée.

En parallèle, les neurosciences ont cherché à comprendre le réseau le plus complexe que nous connaissions : le cerveau humain. Dans cette quête, les neurosciences se sont souvent appuyées sur l'utilisation de modèles de substitution, mathématiques ou biologiques, plus faciles à expérimenter et à comprendre. Mais jusqu'ici, les neurosciences manquaient d'un modèle apte à poser des questions allant au-delà des cytoarchitectures neuronales et des synapses ; en particulier ceux concernant la structure des représentations abstraites dans le cerveau. Enquêter sur une question comme - Comment un humain apprend ou même représente un concept tel que la mort et le relie à la peur et à la tristesse? est resté difficile même en utilisant d'autres primates tels que le singe macaque.

Ce travail de thèse soutient que ces deux domaines– le Deep Learning et les neurosciences– qui ont historiquement toujours été pertinents l'un pour l'autre, peuvent même maintenant s'entraider. Pour illustrer cela, cette thèse propose d'abord d'utiliser les dynamiques récurrentes, un concept issu des neurosciences, au profit des modèles de machine learning. En particulier, le predictive coding, une théorie neuro-computationnelle populaire, est implémentée dans des deep neural networks. Il démontre que les réseaux récurrents résultants sont plus robustes à divers types de bruit, naturels et adversaires, par rapport à leurs homologues à anticipation. Il est important de noter que cette robustesse est obtenue grâce à la capacité de la dynamique de predictive coding à aider les réseaux à projeter les représentations bruitées vers leurs versions propres qui sont apprises pendant l'entraînement - une propriété appelée "projection vers la courbe" .

Deuxièmement, en sens inverse, nous utilisons les réseaux de neurones au profit des neurosciences. Tout d'abord, nous comparons divers réseaux formés avec différents objectifs — uni ou multimodalité, robustesse ou tailles d'ensembles de données — dans leur capacité à expliquer l'activité cérébrale mesurée à l'aide de l'imagerie par résonance magnétique fonctionnelle, ou IRMf. La thèse rapporte

ensuite l'étrange capacité des réseaux multimodaux, c'est-à-dire des réseaux entraînés avec des ensembles de données couvrant diverses modalités, à expliquer l'activité de l'hippocampe — une région connue pour posséder des concept cells invariantes de modalité. Plus tard, nous utilisons une approche indépendante de la région consistant à examiner systématiquement de plus petites portions de voxels dans le cerveau. En utilisant une telle approche basée sur un "searchlight", nous rapportons que par rapport à d'autres modèles, les réseaux multimodaux expliquaient mieux l'activité IRMf dans tout le cortex visuel, tout en expliquant également les régions entourant le sillon temporal supérieur.

Ainsi, dans l'ensemble, dans cette ambition globale de combler le fossé entre les deux domaines, une aspiration également nourrie par une communauté émergente de l'équipe NeuroAI, cette thèse tente de fournir des raisons supplémentaires pour lesquelles les deux domaines pourraient être un couple complémentaire parfait.

# Contents


1 Introduction **9**
- 1.1 Neuroscience and Deep Learning: a tale of two fields . . . . . 14
- 1.2 Using Neuroscience for Deep Learning . . . . . 18
- 1.3 Using Deep Learning for Neuroscience . . . . . 20
- 1.4 Predictive Coding . . . . . 27
- 1.5 Concept Cells . . . . . 40
- 1.6 Outline of the Thesis: . . . . . 42

2 Predify: Augmenting deep neural networks with brain-inspired predictive coding dynamics **44**
- 2.1 Prologue to the main article : . . . . . 45
- 2.2 Main article : . . . . . 46
- 2.3 Epilogue to the main article: . . . . . 69

3 Multimodal neural networks better explain multivoxel patterns in the hippocampus **70**
- 3.1 Prologue to the main article : . . . . . 71
- 3.2 Main article : . . . . . 71
- 3.3 Epilogue to the main article: . . . . . 83

4 Do multimodal neural networks better explain human visual representations than vision-only networks? **84**
- 4.1 Prologue : . . . . . 84
- 4.2 Abstract . . . . . 85
- 4.3 Introduction . . . . . 86
- 4.4 Materials and Methods . . . . . 87
- 4.5 Results and Discussion . . . . . 88

4.6 Acknowledgments . . . 90
4.7 Epilogue to the article: . . . 90

5 Conclusions **92**
5.1 Extended discussion on Chapter 2 . . . 92
5.2 Extended discussion on Chapters 3 and 4 . . . 98
5.3 Closing thoughts . . . 100
5.4 Final Summary . . . 104

Appendix A Appendix for Chapter 2 **106**
A.1 Getting Started with Predify . . . 106
A.2 Network Architectures . . . 109
A.3 Execution Time . . . 111
A.4 Gradient Scaling . . . 112
A.5 Prior work: PCNs . . . 114
A.6 Comparing with Rao and Ballard . . . 118
A.7 Tuning hyperparameters . . . 123
A.8 mCE scores of the optimized networks using AlexNet as a baseline 127
A.9 mCE scores of a predified robust network . . . 127
A.10 Original data for Adversarial Attacks . . . 129
A.11 Absolute values of the plots shown in the main text . . . 131

Appendix B Appendix for Chapter 3 **133**
B.1 The complete schematic . . . 134
B.2 Additional details on the models used in the analysis . . . 135
B.3 Licenses of the assets used . . . 136
B.4 Raw data shown in Figure 2 . . . 137
B.5 Voxel-selection based on a fixed beta-value threshold. . . . 138
B.6 RSA computed using different metrics . . . 140
B.7 Broader Impacts . . . 141

References **165**

# List of Publications

During the course of my Ph.D. I have been involved in various other works with varying degrees of capacity. I list all the publications, along with the links to the code to reproduce all the results, below:

**Predify: Augmenting deep neural networks with brain-inspired predictive coding dynamics**
*Bhavin Choksi*, Milad Mozafari*, Callum Biggs O'May, Benjamin Ador, Andrea Alamia, Rufin VanRullen*
Link to Code : `https://github.com/bhavinc/predify2021`

**Multimodal neural networks better explain multivoxel patterns in the hippocampus**
*Bhavin Choksi, Milad Mozafari, Rufin VanRullen, Leila Reddy*
Link to Code : `https://github.com/bhavinc/multimodal-concepts`

**Does language help generalization in vision models?**
*Benjamin Devillers, Bhavin Choksi, Romain Bielawski, Rufin VanRullen*
Link to Code : `https://github.com/bdvllrs/generalization-vision`

**Reconstruction of Perceived Images from fMRI Patterns and Semantic Brain Exploration using Instance-Conditioned GANs**
*Furkan Ozcelik, Bhavin Choksi, Milad Mozafari, Leila Reddy, Rufin VanRullen*
Link to Code : `https://github.com/bhavinc/IC-GAN_fMRI_Reconstruction`

**On the role of feedback in visual processing: a predictive coding perspective**
*Andrea Alamia, Milad Mozafari, Bhavin Choksi, Rufin VanRullen*
Link to Code : `https://github.com/artipago/Role_of_Feedback_in_Predictive_Coding`

**Predictive coding feedback results in perceived illusory contours in a recurrent neural network**
*Zhaoyang Pang, Callum Biggs O'May, Bhavin Choksi, Rufin VanRullen*
Link to Code : `https://github.com/rufinv/Illusory-Contour-Predictive-Networks`

# Listing of figures

1.1 **Papers published in NeurIPS conference until 1990 and until 2021:** The figure visualizes the fields of papers published in the NeurIPS, a very popular conference in machine learning. Each dot represents an individual paper and the color represents the year in which it was published. The rectangles depict the field of the papers. The papers published in Neuroscience are highlighted by a red rectangle and those with classical neural networks with a purple rectangle. The divergence in the interests of the machine learning community is apparent from the contributions made in past few decades. (Adapted from `neuripsav.vizhub.ai` and best viewed in a digital format) . . . . . 17

1.2 **General methodology used to build encoding and decoding models** : The figure illustrates the methodology used to build encoding and decoding models. (1) Generally, after collecting the data, it is split into training and validation sets. Later, an encoding model is learned on the training split such that it can predict brain activity (voxel activity in case of fMRI data). (2 and 3) This encoding model typically learns a feature space that can linearly map onto the voxel activity space. (4) The linear mapping learned can also be inversed to obtain a decoding model that can predict the stimulus features (and sometimes the stimulus itself) from a given brain activity pattern (Figure from Naselaris et al. 2010). . . . . 23

1.3 **Representational Similarity Analysis:** To perform Representational Similarity Analysis, one first calculates a Representational Dissimilarity Matrix (RDM) using pairwise distances between the different model features (or brain activity patterns). A similar RDM is constructed for another model of interest after which a similarity measure, generally correlation, is calculated between the two RDMs (figure adapted from Devillers et al.[51]). . . . . 26

1.4 **Feedforward receptive fields of neurons in a predictive coding network** after training (representative subset) on natural images. Basis vectors in the model, which can be considered as classical receptive fields of higher-level units, exhibit tuning to components of optic flow such as translation and expansion (Figure from Jehee et al.[106]) . . . . . 30

1.5 **The neural responses elicited due to an expectation of a stimulus are specific to the features of the stimulus** : (A) Experimental setup used by Kok et al.[121]. The trials started with an auditory cue that predicted the orientation of the subsequent stimulus (grating of 45 degrees or 135 degrees). On 75% of the trials, subjects were shown two gratings, first with the expected orientation based on the sound cue, followed by the second grating that was tilted clockwise or anticlockwise by a few degrees (with respect to the first). The subjects later performed a discrimination task where they judged this direction of rotation. (B) In 25% of trials, after the sound cue, no gratings was presented. The participants were asked to just fixate in the center (C) Throughout the experiment, two sound cues were used that hinted at the orientation of the grating (with 100% accuracy). (D) BOLD signals in V1. The time courses are locked to the (expected) onset of visual stimuli. (E) The plot shows the BOLD signal amplitude evoked by 45 degree and 135 degree gratings (in stimulus and omission condition), separately for voxels that preferred the particular grating orientation. The activity observed when the stimulus was omitted represents the prior expectations of the grating orientation. (Figure adapted from Kok et al.[121]) . . . . . 35

2.1 **General overview of our predictive coding strategy** as implemented in a feedforward hierarchical network with generative feedback connections. The architecture (roughly similar to stacked auto-encoders) consists of $N$ encoding layers $e_n$ and $N$ decoding layers $d_n$. $W_{m,n}$ denotes the connection weights from layer $m$ to layer $n$, with $W^f$ and $W^b$ for feedforward and feedback connections, respectively. The reconstruction errors at each layer are denoted by $\varepsilon_n$. The feedforward connections (green arrows) are trained for image classification (in a supervised fashion), while the feedback weights (red arrows) are optimized for a prediction (i.e. reconstruction) objective (unsupervised). Predictive coding minimizes the reconstruction errors in each layer by updating activations in the next layer accordingly (black arrows). Self-connections (memory) are represented by blue arrows. . . . . 49

2.2 **Performance under Gaussian noise and projection towards the learned manifold.** (a) Improvement in recognition accuracy with reference to the feedforward baseline under various levels of Gaussian noise. Both networks demonstrate significant accuracy improvement across timesteps under noisy conditions, while maintaining a performance close to the feedforward level for clean images. (b) Normalized MSE distance between the image reconstruction ($d_0$) and the clean image ($e_0$). Irrespective of the noise level, image reconstruction consistently gets closer to the clean image across timesteps in both models. (c) Examples of clean and noisy input images together with their final reconstruction by the model (the row order from top to bottom is: original image, PVGG16 reconstruction, PEfficientNetB0 reconstruction; noisy image, PVGG16 reconstruction, PEfficientNetB0 reconstruction). For best viewing, we recommend zooming in on the electronic version. (d) Normalized correlation distance between representation of clean and noisy images for each encoder ($e_i$) across timesteps. The values are normalized with respect to the feedforward baseline (timestep 0). In both models and all encoders, the noisy representations tend to move toward the clean copies. . . . . 57

2.3 **Benchmarking robustness to ImageNet-C.** (a) Normalized corruption errors (CE) of PVGG16 and PEfficientNetB0 under four types of additive noise corruptions. The values are normalized with respect to the feedforward baseline. Both networks show consistent reductions in the errors across timesteps. (b) Normalized mean Corruption Error (mCE) scores for PVGG16 and PEfficientNetB0 on all the 19 corruptions available in the ImageNet-C dataset, when optimized hyperparameters are used (as described in the Appendix A.7). The values are normalized with respect to the feedforward baseline. In both the panels, error bars represent the standard deviation of the bootstrapped estimate of the mean value. 60

2.4 **Benchmarking robustness to adversarial attacks.** Plots show the success rate of targeted adversarial attacks against DPCNs across timesteps. The values are baseline-corrected, relative to the success rate at timestep 0 (feedforward baseline). Both networks demonstrate improved robustness to different types and/or levels of perturbations. . . . . 63

3.1 **Noise ceilings after selecting subsets of voxels from each region** The panels show the noise ceilings (i.e., inter-subject correlation) calculated after selecting different numbers of voxels from each region of interest. The noise ceilings were computed using either voxels with the highest beta values (blue) or via a random sampling of voxels (orange). The gray regions denote the standard error of mean. For certain ROIs (visual region, fusiform), most voxels are informative about the visual stimulus, and the two selection methods yield similar results. For other ROIs (hippocampus, parahippocampus), the noise ceiling depends on the selection method, implying that some voxels (with the highest betas) are more informative than others (randomly selected). The hippocampus shows an improved noise ceiling when 30 voxels with the highest beta values are selected, with additional voxels degrading the signal. 78

3.2 **Multimodal models better explain fMRI response patterns in the hippocampus**: Panel A shows the correlation values obtained with different models across selected regions of interest (ROI). Only 30 voxels were selected from each ROI. The values are normalized with the noise-ceilings to facilitate comparisons across regions. Panel B shows the correlation values after aggregating them over mutlimodal (green), visual (red) and language (blue) models. Statistical significance is calculated by using Welch's t-test and is denoted by an asterisk. . . . . 80

4.1 **Partial correlations between model RDMs and brain RDMs (with the ResNet RDM as a control variable) :** Each column depicts four slices, the first two columns for uni- and the last four for multi-modal networks. The color scale represents partial correlation values. Multimodal networks show higher similarity to brain representations in the LOC and fusiform regions compared to their unimodal counterparts, thus explaining more unique variance in the brain data. They also explain variance in regions around STS, an effect unseen in the visual models. . . . . 89

A.1 **PCN**: Panel (a) shows the reconstruction errors of the model over timesteps. It does not decrease over timesteps, as would be expected in a predictive coding system. Panel (b) depicts the accuracy of the model on the CIFAR100 test dataset. The model performs at chance level at early timesteps and then becomes better in the last few timesteps. . . . . 115

A.2 **PVGG16 (optimised) Corruption Error (CE) scores for all distortions:** The panel shows the CE scores calculated on the distorted images provided in the ImageNet-C dataset. The values are normalized with the CE score obtained for the feedforward VGG. The error bars denote the standard deviation of the means obtained from bootstrapping (resampling multiple binary populations across all severities.) . . . . . 125

A.3 **PEfficientNetB0 (optimised) Corruption Error (CE) scores for all distortions:** The panel shows the CE scores calculated on the distorted images provided in the ImageNet-C dataset. The values are normalized with the CE score obtained for the feedforward EfficientNetB0. The error bars denote the standard deviation of the means obtained from bootstrapping (resampling multiple binary populations across all severities.) . . . . . 126
A.4 The mCE scores of the optimized networks (as shown in Figure 3) normalized using the score of the AlexNet network. Instead of normalizing using the score for the feedforward version of our recurrent network, to facilitate comparison with other works, we here normalize the scores using the score obtained for AlexNet network. . . . . 127
A.5 The Relative mCE scores of the optimized networks (as shown in Figure 3) normalized using the score of the AlexNet network. As suggested by[92], we use Relative mCE score which accounts for the changing baseline accuracy on the clean images over timesteps. . . 127
A.6 mCE scores of a predified version of an already robust PEfficientNetB0 . . . . . 128
A.7 $L_\infty$BIM attacks on PVGG16 network . . . . . 129
A.8 $L_\infty$BIM attacks on PEfficientNetB0 network . . . . . 129
A.9 $L_2$RPGD attacks on PEfficientNetB0 network . . . . . 129
A.10 $L_\infty$ HopSkipJump attacks on PEfficientNetB0 . . . . . 130
A.11 Adversarial Attacks with respect to epsilons. Here we show the number of successful attacks on 1000 (100 for HopSkipJump) images. Increasing the size of the epsilon leads to increase in the success rate of the attack as expected. As predictive coding timesteps increase, the curves shift slightly to the right, meaning that a slightly larger perturbation is required to fool the network. This robustness is more easily seen on Figure 2.4, where $\varepsilon$ values are sampled near each curve's inflection point. . . . . 130
A.12 **Correlation distances for representations obtained on noisy images:** Here we show the absolute correlation distances obtained between clean and noisy representations as shown in Figure 2d in the main text. . . . . 132

B.1 This schematic shows the methodology used for the analysis. In a first step, the fMRI data corresponding to the test images shown to the participants are processed. Next, from all the voxels in a selected region of interest, a subset of voxels with the highest reliability in the fMRI signal (as determined by an independent analysis on the noise ceiling) is chosen. Using pairwise distances on these brain features, an RDM (representational dissimilarity matrix) is constructed. In parallel, the same test images are passed through different models, and their features are obtained. These features are used to construct RDMs for each model. Finally, the similarity between each model RDM and the brain RDM is computed using different correlation measures. . . . . 134
B.2 **Non-normalized RSA values between model and brain RDMs.** The brain RDMs are calculated based on selecting 30 voxels from each ROI, as in the main analysis. The gray bands show the upper and lower bounds of the noise-ceilings calculated by adding and subtracting the s.e.m. values respectively. . . . . 137
B.3 Non-normalized RSA values after using the beta value of the 30th voxel from hippocampus as a threshold for other ROIs for each participant. . . . . 138
B.4 The RDMs were calculated using the Pearson correlation distance, and the Spearman rank correlation was used to compute the RSA. 140
B.5 The RDMs were calculated using the Cosine distance, and the Spearman rank correlation was used to compute the RSA. . . . . 140
B.6 The RDMs were calculated using the Cosine distance, and the Pearson correlation was used to compute the RSA. . . . . 141

To my Mom, Dad, and Rishabh.....

# Chapter 1

# Introduction

Humans have always been curious about our intelligence and consciousness. Since ancient times we have wondered about their origins, often even resorting to answers in physical or metaphysical elements. In terms of the biological actuality, ancient thinkers debated whether it was the brain or the heart that was the primary source of our intelligence. Most famous (or infamous) of all, Aristotle argued that "the seat and source of sensation is the region of the heart"[8,79] while the brain acted as a "coolant" that counterbalanced the hot heart. Interestingly, he made these arguments to contrast the "fallacious" idea of his predecessors, such as Alcmaeon and Hippocrates, who believed that the brain was the primary organ responsible for our intelligence[79].

Indeed, any modern day researcher would at best frown at this ludicrous idea of the brain being a glorified refrigerator. The advent of better technology, along with clinical and anatomical findings have since long settled that debate. But still, how the brain achieves its functionality remains an open question, a holy

grail for the fields of neuroscience, psychology, philosophy, and the new field of Artificial Intelligence (AI). Advances until now have established that the brain is an intricate network of neurons—units which communicate among each other using various charged ions and/or molecules. And numerous researchers from all the varying fields have dedicated significant amounts of their careers trying to understand this network at various levels—from the level of an individual neuron to their assemblies.

And partly due to the complexity of the human brain, many of these efforts have been driven by a very simple approach—of using a simpler system that, at least to some extent, can faithfully represent the human brain. Depending on the questions asked, these so-called "model organisms" can range from either a 1mm nematode, such as *Caenorhabditis elegans*, with only 302 neurons to organisms such as monkeys and gorillas that have billions of neurons and portray behaviours and social structures as complex as humans. Apart from the complexity, this has also helped us avoid, at least partially, the ethical and logistical issues often encountered in clinical studies with humans. This simple approach of analogical reasoning has been quite successful in advancing our understanding of the brain, especially when the model organism is a very good proxy for the particular aspect under investigation.

But still, directly investigating *human* intelligence and its various aspects has always eluded us, mostly because we never had a suitable model organism for that. Indeed, no one would argue against the inutility of *C. elegans* to meaningfully study the human intelligence. But even if one settles on a more intel-

ligent, *obviously* conscious animal, say a monkey, can we be sure of its utility? A very stereotypical example often used in natural language processing is that of "QUEEN - WOMAN + MAN = KING". When asked about the validity of this operation, a human would almost instinctively jump and say *"of course yes"*. Here, are we sure if a monkey agrees with this statement which involves a linear operation on complex abstract concepts encompassing sexuality, social hierarchy, etc.? Does it even understand what a KING or a QUEEN is? And say given the complex social structures in primates, it indeed possesses an understanding of a "QUEEN", can one be sure that it understands it the same way a human would? And going beyond, what about more abstract concepts such as SONDER or SCHADENFREUDE? And if we agree that no reasonable person can assume this to be trivially true, then one must concede the limited utility of even our evolutionary ancestors to study our own intelligence and consciousness.

And this is where the current advances in the field of artificial intelligence, especially Artificial Neural Networks (or ANNs), come in. These artificial networks were initially inspired by the brain; a fact that is apparent in the names of the units of these networks (neurons) and their *feedforward* connections as observed in the brain. Over the past few decades, these ANNs have become exceptionally good at performing tasks that previously only a human could do: classifying various images in different categories, generating long verbose essays or even an artistic image using just a simple text prompt, driving cars while identifying road signs and pedestrians, etc.[87,180,179,182,192,241] Performing these tasks implies learning of representations of these complex and abstract concepts. More importantly,

when tested with "QUEEN - WOMAN + MAN = ?", these networks gave the answer "KING"[77,232] demonstrating their ability to perform linear operations akin to humans! Thus, for neuroscience, they make a strong case for being a perfect candidate for fulfilling its lack of an apt model to study higher level cognition.

At the same time, though quite good, these ANNs themselves are not completely free of problems. For example, they are known to generalize poorly outside their training data, even on datapoints that are minutely perturbed. For computer vision applications, these perturbations can be simple natural noises like snow or fog or expensively hard-mined to be so small that they are undetectable to the human eye—called adversarial perturbations[214,92]. After the early characterization of this problem almost a decade ago, the Machine Learning (or ML) community is still trying to understand this sensitivity of ANNs and improve their robustness.

But of course, the brain is many times more robust to such perturbations. Any pet owner can easily identify their pet dog behind the backyard tree, covered in mud on a heavy rainy day. Thus, like early engineers who looked at birds to get the idea for wings for planes, what if current ML engineers looked at the brain to see what made it is so robust to input stimuli? And, theoretical and computational neuroscience have various candidate mechanisms that could be feasible in the brain. What if these are employed in the modern day ANNs? Will that help in improving their robustness? Maybe the current field of Deep Learning (or DL), which benefited from some inspiration a few decades ago, can learn a little more from the brain.

The current thesis lies in this context. It recognises that there is a gap between

these two very relevant fields; Neuroscience requires a more apt model to meaningfully progress on its quest to understand the brain, while Machine learning warrants directions to improve itself further and could heavily profit by looking at the best machine that is at its disposal—the human brain. The thesis argues that the two can benefit from a stronger communication between them.

The thesis tries to advocate for this by illustrating two approaches. In the first approach, it uses ideas from neuroscience to make brain-inspired neural networks and then study their properties. Such an approach helps both the fields. It helps neuroscience by corroborating ideas among their other alternatives, while elucidating the emerging properties resulting due to the theory. Simultaneously, it helps machine learning by providing mechanistic principles that it could use to build better networks. Specifically, in Chapter 2 we propose and implement recurrent dynamics inspired from "predictive coding", a popular theory in neuroscience, in ANNs and show that they render further robustness to networks via a mechanism known as *projection towards the manifold*.

On the other hand, the second approach exploits the aforementioned observation that current ANNs can learn representations of complex concepts and thus uses them to gain insights into the representational structure of the brain. More specifically, in Chapters 3 and 4, we compare the representational spaces of various ANNs to that of the brain obtained using functional Magnetic Resonance Imaging (or fMRI) data, and then try to distinguish factors that stand out in these comparisons. Insights from such experiments can be useful for not only inferring the role of specific regions in the brain, but also for building networks that

are more functionally similar to the brain in the future. Later in Chapter 5, the thesis concludes by discussing the merits and demerits of these approaches and comparisons in general.

But first, let us discuss the two approaches and the work relevant to the thesis in more detail.

## 1.1 Neuroscience and Deep Learning: a tale of two fields

Indeed, a claim such as bridging the gap between two fields implicitly assumes the existence of one in the first place. Though its existence can be hardly refuted today, a brief look at older works tells us that historically this gap has not been so trivially apparent. The two fields worked hand-in-hand, often taking ideas from one for the other.

### 1.1.1 Neuroscience and Networks

By 1900, the findings from Cajal and subsequent researchers had helped satisfactorily disband the reticular theory, which considered the brain as one continuous network, and instead establish "neurons" as one of the fundamental constituent of the brain. This discovery led the research into understanding these neurons in the brain—their density, types, firing patterns, etc. The cytoarchitectural findings, their localization in the brain, along with behavioral (and lesion) studies on patients, hinted at the modular nature of the brain. On the other hand, electrophysiological data, for example from patch clamp experiments, paved the way for mathematical models of the firing patterns of neurons[94,65].

Indeed, an understanding of the network is incomplete without investigating the connections between its units. Investigations into these connections between neurons, so-called synapses, led to our understanding of their formation, potentiation and depression[20]. From a mathematical perspective, these findings helped the formation of "learning rules" such as Hebb's rule, BCM thoery, Oja's rule, etc. that modeled the formation and adaptation of synapses between neurons[88,19].

The localization of the different neuron types in the brain had already hinted at its modular nature. Subsequent findings only helped solidify this idea. As a modular network, the brain also has a lot of connections interconnecting different modules. Neuroanatomical studies revealed that the different modules in the brain are connected in a hierarchical fashion with two* broad types of connections—feedforward connections that connected lower regions (in the hierarchy) to the higher regions, and feedback connections that started from higher regions to the lower regions[63,145,154].

As could be expected, this was followed by proposals of hierarchical networks comprising of both feedforward and feedback connections with different roles and functions. Theories such as Adaptive Resonance Theory (ART)[28], HyperBF[174], re-entrant signaling theory[56] etc. used feedforward and feedback connections to pass different information between the lower and the higher areas in the hierarchy fulfilling different objectives. In Section 1.4 we will look at one such particular use of feedforward and feedback connections.

Meanwhile, taking inspiration from these architectural principles, the field of

*There are also lateral connections that connect neurons within the same module. But they are omitted here for simplicity given the scope of the work.

Machine Learning was building more mathematically abstract (and better!) networks for various complex (human-like) tasks such as computer vision. Networks like the Hopfield network[96], Helmholtz machine[47] are some classic examples of early ANNs that highlight this inspiration. This is very elegantly highlighted in the very first few lines of the abstract of John Hopfield's 1982 paper that proposed the Hopfield network[96], considered as one of the early recurrent ANN:

> *Computational properties of use to biological organisms or to the construction of computers can emerge as collective properties of systems having a large number of simple equivalent components (or neurons)....A model of such a system is given, based on the aspects of neurobiology but readily adapted to integrated circuits.*

### 1.1.2 The divergence between the two fields

To the surprise of everyone, the mathematical principles—propagation of the gradients in the backward direction (or backpropagation[191]) in particular—that were developed in this proto-Deep learning era turned out be really effective.

Thus when the era of advanced computers brought with them the ability to perform faster and faster computations, it led to a sudden increase in the performances of these networks. More importantly, this exponential boom in the available computational resources allowed for two key things – (i) creation of huge datasets that can be stored and worked with, and (ii) the ability to test various architectures, biologically plausible or not for learning on these datasets. These factors, especially the latter, gradually transformed the goal of building networks

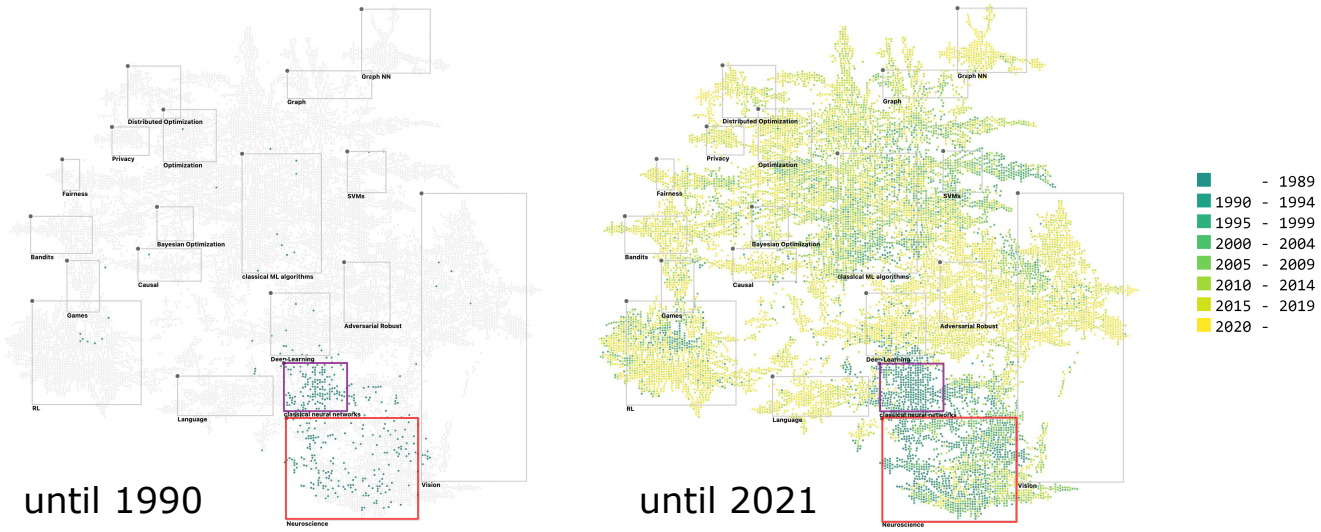



**Figure 1.1: Papers published in NeurIPS conference until 1990 and until 2021:** The figure visualizes the fields of papers published in the NeurIPS, a very popular conference in machine learning. Each dot represents an individual paper and the color represents the year in which it was published. The rectangles depict the field of the papers. The papers published in Neuroscience are highlighted by a red rectangle and those with classical neural networks with a purple rectangle. The divergence in the interests of the machine learning community is apparent from the contributions made in past few decades. (Adapted from `neuripsav.vizhub.ai` and best viewed in a digital format)

into an engineering problem. *How can we get a network learn things better? with larger and broader layers? with more layers? with different connections? with more data? etc.*

And with success, the field grew but pivoted away from neuroscience in due course of time. This rather unfortunate turn can be very well visualized on `https://neuripsav.vizhub.ai`. The site shows the contributions of papers from each field in NeurIPS – a popular conference in the field of modern day Deep Learning. Figure 1.1 contrasts the papers accepted at the conference based on its field. The decrease in the relative contribution of the papers, and thus ideas, from neuroscience elucidates the divergence in the ideas pursued by the machine learning

community.

## 1.2 Using Neuroscience for Deep Learning

But I would like to submit that present day Machine Learning can still learn a lot from neuroscience; after all the brain is still the best network in business!

And we have a precedence for this. Besides the architectural inspiration of feedforward connections, the deep learning community has used other ideas such as attention and hallucinations, at least functionally, to improve the performance of neural networks[246,130]. Transformers are some of the state-of-the-art networks that rely on attention mechanism inspired from the brain[229].

Similarly, another popular concept from neuroscience—reinforcement learning—initially inspired various reward based learning objectives in Machine Learning. These learning paradigms are quite successful in building well-performing networks, which often beat humans in games such as Chess, Go, Dota, Starcraft[200,17,230]. Indeed, as can also be seen in the Figure 1.1, Reinforcement Learning has evolved into its own field right now, often gathering most of the crowd at NeurIPS poster sessions.

And though one can argue that deep learning has cracked the code for performing really well on vision or semantic tasks individually, it still has a long way to go. For example, one front where it still struggles is making networks that can combine and/or generalize to tasks across modalities and datasets. There do exist methods for multimodal integration, that often combine audio, semantic and video modalities, but the results are still not as would be expected and there still

remains a lot to be improved[51,222].

Thus, efforts are being made to take further inspiration from neuroscience. Global workspace theory, is a neuroscience theory of perception that posits that information from all the modalities is combined at one common location in the brain[9,10]. Thus, architectures implementing versions of this theory using modern-day ANNs have been proposed[227]. Even attention, as it implemented in transformers, heavily inspires from so-called bottom-up attention. But, neuroscience posits/recognizes the existence of other type of attention—top-down attention—which can be implemented at a global level. Indeed, VanRullen & Alamia[226] have tried implementing such a global attention system into current deep neural networks.

As mentioned earlier, ANNs are easily fooled if one adds a tiny perturbation to the input examples. This sensitivity to small perturbations, called adversarial perturbations, is especially troublesome given their increasing use for tasks in the wider population. Given that the brain is instead a robust network, ideas from neuroscience have been used to counter this sensitivity of the ANNs and improve their robustness[142,158,45]. For example, taking a page from neuroscience, Nayebi & Ganguli[158] used neurons with saturating values to build robust networks. Similarly based on the visual cortex, Dapello et al.[45] added gabor filters in their ANNs and observed improved robustness to a variety of corruptions. As a more interesting approach, studies have directly trained the ANNs on brain data and have observed that they were more robust[132]. Investigations into these networks led to the discovery of guiding principles that could help in building more robust

networks in the future[133].

Out of these ideas from neuroscience, one idea in particular is recently gaining a lot more traction—that of *recurrence*. As mentioned earlier, biological brains possess a high amount of feedback connections but are not so commonly seen in the modern day typical ANNs[63,154]. Various studies have tried reconciling this architectural difference and investigate the role of feedback connections in information processing. A seeming consensus is emerging that the recurrent connections help the neural networks when the inputs are either degraded with noise[237,157], with occlusion[62,206], or for efficiently learning long range dependencies[138]. Indeed, the work discussed in Chapter 2 also implements recurrent dynamics in ANNs; that prefer the feedback connections in the presence of noise[4].

## 1.3 Using Deep Learning for Neuroscience

Just as the evolution of networks has benefited from neuroscience, neuroscience also has benefited from the networks. Until now, the early networks were built to be used as a testbed for various hypotheses, predominantly in theoretical neuroscience. But as pointed out earlier, the current ANNs, with their ability to perform complex tasks at human-level, have opened up the possibility of directly studying the representations in the human brain.

### 1.3.1 Establishing the model organism

But before that, there lies an important step of establishing sufficient equivalence between an ANN and the brain; after all it's always smarter to ensure that a

given map is accurate before using it to cross the world. Such an equivalence can be built at various levels. Given the impressive performance of ANNs on various tasks, it is easy to start establishing an equivalence at a behavioral level. Indeed, studies have analyzed the similarities between the performances of ANNs and humans under various conditions[74,72,206,118,175,58]. Most of these typically focus on ways in which ANNs learn their tasks; more importantly under which situations do they fail. Thus, though very important for neuroscience, their appeal generally lies more for the Machine Learning community.

For the Neuroscience community, a more appealing aspect lies in establishing an equivalence at a representational level—after all it is interested in understanding how representations of complex concepts are encoded in the brain, not the ANNs. To accomplish this, current methods rely heavily on two broad techniques: Neural encoding and decoding, and Representational Similarity Analysis.

#### 1.3.1.1 Neural Encoding and Decoding

One way to directly analyze the neural code in the brain is to learn a mapping from this code to a more interpretable, understandable space which can be used for directly performing tasks, such as classification.

Though not the first, one of the prominent studies that performed such an analysis was from Cox & Savoy[41]. Using linear and polynomial support vector machines, they were able to classify the fMRI data measured on subjects viewing images from ten different categories. Later, Kay et al.[111] learned a linear regression from features extracted from images (in the training dataset) to accurately

predict activities of voxels in an fMRI experiment. These regression weights were later able to accurately identify the underlying image from a test dataset by just using its voxel activity.

Encoding methods allow explaining the variance in the activity of the voxels, and as a result can also help in elucidating information about the stimuli encoded in specific voxels. For example, to analyze the same data from Kay et al.[111], Naselaris et al.[155] built two different models—one based on phase-invariant Gabor wavelets (also called as Gabor wavelet model), and other based on category label for each natural scene (semantic model). They found that though both the models predicted the voxel activities well, the underlying population of voxels was different; while the Gabor wavelet model worked best on voxels from the early visual areas, the semantic model worked best for higher visual areas.

Encoding models typically learn a linear mapping between the image features and the brain space (see Figure 1.2). This assumption of linearity between these two spaces is generally used as it allows for greater interpretability. For example, even though the underlying neural code is nonlinear, the intuitiveness of operations like QUEEN - WOMAN + MAN = KING or FACE = EYES + NOSE + EARS illustrates the linear way in which humans think about these concepts. Thus, one reasonably expects that any mapping learned (between these two nonlinear spaces), should at least allow us to manipulate these concepts in a linear fashion. Indeed, this assumption is further strengthened by the empirical success of various methods using linear mappings[41,111,187,109,228]. Here, one should be careful to note that this is just an engineering expectation, and previous studies have

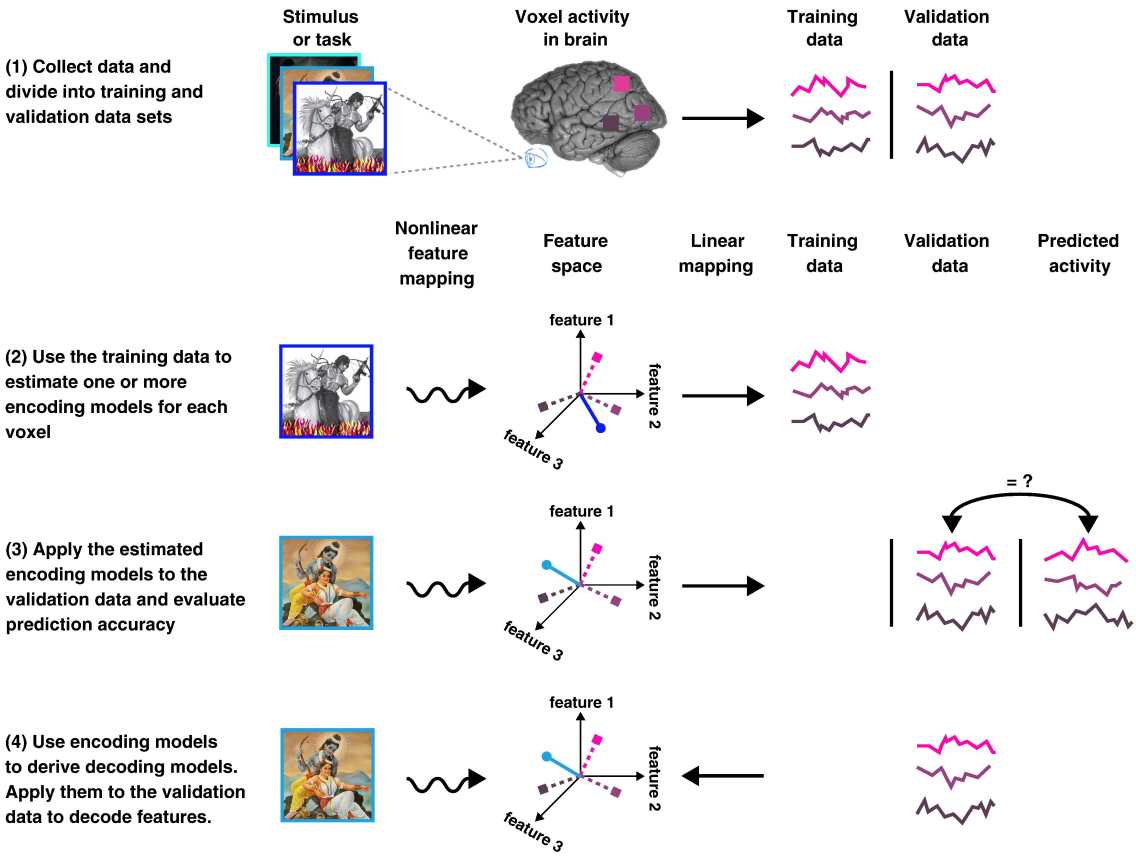



**Figure 1.2: General methodology used to build encoding and decoding models** : The figure illustrates the methodology used to build encoding and decoding models. (1) Generally, after collecting the data, it is split into training and validation sets. Later, an encoding model is learned on the training split such that it can predict brain activity (voxel activity in case of fMRI data). (2 and 3) This encoding model typically learns a feature space that can linearly map onto the voxel activity space. (4) The linear mapping learned can also be inversed to obtain a decoding model that can predict the stimulus features (and sometimes the stimulus itself) from a given brain activity pattern (Figure from Naselaris et al. 2010).

used nonlinear mappings to learn mappings onto the brain activity space[41,46,85].

As one can expect, the ability to learn a linear mapping between the brain and model space will thus depend on the structure of the model space—a place where current mathematically pliable ANNs can shine. This has prompted various researchers to use the latent spaces of different ANNs to encode the brain data.

For example, in 2015, using various ANNs Horikawa & Kamitani[97] were able to identify images from over 50 categories from the popular ImageNet dataset[48].

Neural decoding instead, as an approach aims to do the opposite : to directly use brain activity and predict what stimulus could possibly have caused such a pattern (see Figure 1.2.4). A first example of such a decoding was attempted by Thirion et al.[220] where they tried to reconstruct the stimuli viewed by subjects in an fMRI machine. While their stimuli were simple gabor patches, the complexity of the stimuli was very soon scaled up by Miyawaki et al.[151] where they instead used contrast patches with various patterns.

Even ANNs, especially those trained with generative objectives, are used for this task. For example, VanRullen & Reddy[228] used a combination of state-of-the-art networks at that time – variational autoencoders (VAE) and Generative Adversarial Networks (GAN) – and showed their ability to reconstruct faces viewed during the fMRI experiment. Moreover, they were able to perform simple linear operations in the VAE latent space and visualize the effects in the reconstructions. Various other studies later improved on the methods, often faithfully reconstructing lower level details in the reconstructions[70,15]. Mozafari et al.[152] on the other hand were able to reconstruct images that captured the semantic attributes of the stimuli. These methods were later improved by Ozcelik et al.[169] who were successful in capturing both lower and higher level details by using Instance-Conditioned GANs[30].

While encoding and decoding models help infer the information encoded in various representational spaces, they still involve an indirect comparison between two

latent spaces by learning a linear regression between them. These methods are also hampered by the dimensionalities of the two spaces, which is often countered using some form of regularization. Thus, an alternative method to directly compare the representational spaces by measuring some form of similarity between them would be desirable. This is where Representational Similarity Analysis (RSA) comes in.

#### 1.3.1.2 Representational Similarity Analysis

Representational Similarity Analysis, allows one to directly compare different representation spaces[125]. Intuitively, it supposes that if in two different representational spaces, the relative distances between the inter-category representations are similar, then the representational spaces can be considered similar to each other.

Mathematically, RSA starts by calculating the pair-wise distances (correlation, cosine or euclidean) between representations of different categories and constructing a Representational Distance Matrix (or an RDM) for a particular representation space (see Figure 1.3). It then compares this RDM with the RDM of another representational space by measuring the correlation (generally rank correlation) between them. Since the entries in an RDM only represent a measure of distance between the representations with a single scalar quantity, RSA can be used to compare representational spaces of different dimensionalities. Thus, one can compare not only a brain and a latent space of a neural network, but also two different neural networks, at varying depths. One can also compare representations from a human and a primate, as was done for the first inception of RSA[126].

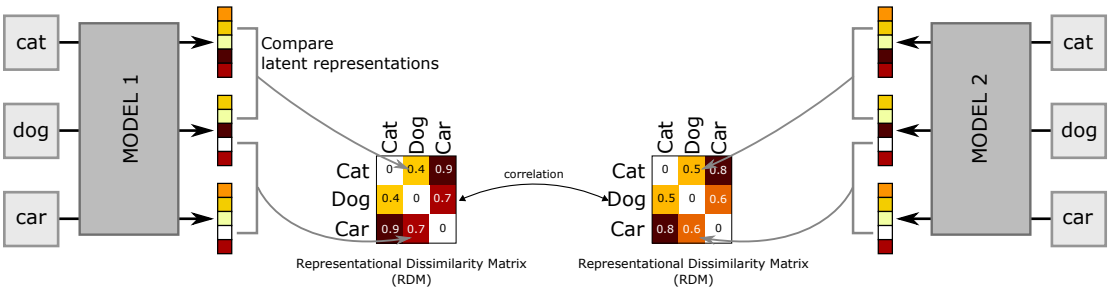



**Figure 1.3: Representational Similarity Analysis:** To perform Representational Similarity Analysis, one first calculates a Representational Dissimilarity Matrix (RDM) using pairwise distances between the different model features (or brain activity patterns). A similar RDM is constructed for another model of interest after which a similarity measure, generally correlation, is calculated between the two RDMs (figure adapted from Devillers et al.[51]).

Often studies combine these methods to assess the abilities of an ANN to explain the brain data. One such early study was from Yamins et al.[243] who showed that ANNs that were trained to be good at the final task such as object recognition were able to explain the brain data exceptionally well. Since then, other studies have only corroborated this findings across other animals such as primates and rodents.[27,112,25,26,242,156,31]

The different layers in the neural networks learn different levels of representations, often getting more and more abstract with depth. Studies have tried successfully testing if this hierarchy in the representations is also mapped and matched onto the brain data.[81,82,97]

To further improve the utility of the model, various researchers have argued for building networks that act as good models for the brain[189]. This can be achieved by either investigating the impacts of (i) architectural changes such as convolutions versus transformers[223] or recurrence vs feedforward[181], (ii) objective function based changes such as supervised versus unsupervised[114],uni- versus multi-

modality[167] etc. To systematize the search for such networks, proposals have also emerged for benchmarks such as Brain Score[198], the Algonauts Challenge[38], Brain Hierarchy Score[160]. Already, such benchmarks in turn, have inspired building of better ANNs for future research[129].

Chapters 3 and 4 present similar attempts to uncover factors that affect the ability of neural networks to explain the brain data. But before that, let us look at two particular topics in more detail—the theory of predictive coding in neuroscience and concept cells as they are found in neuroscience (and recently even in ANNs).

## 1.4 Predictive Coding

As discussed earlier, the modular and hierarchical nature of the network allowed classification of the connections into three categories—feedforward, feedback and lateral. Of these, the feedforward connections received a major portion of the attention, particularly in the ML community after the success story of the ANNs. But given their widespread nature and abundance in the brain, theories have long tried understanding the role of the feedback connections[50,28,174,63].

Around the 1980s, one idea in particular had started gaining some traction – of feedback connections relaying *predictions* about the representations to the lower layers[174,28]. Stemming from his work on template and pattern matching, Mumford[154] hypothesized that a feedback connection can be used to relay a representative template of the activity at the lower layer (that is, a prediction) as expected by the corresponding higher layer. Such a communication will also allow

the higher layers to communicate with the lower layers in the latter's representation space.

In 1999, Rao and Ballard[183] used this idea to propose *Predictive Coding* to explain changes in the response pattern of neurons in the visual cortex when a stimulus is presented outside of their classical receptive fields, also known as extra-classical receptive field effects. The contemporary approach was to use feedforward connections to relay information regarding the representations. In contrast, the Rao and Ballard architecture instead used feedback connections from the higher layers to relay this information about the representations to the lower level, while using the feedforward connections to relay the errors made in the predictions (or *residual errors*) to the higher layers. The whole network, along with its weights (during learning) and activations (during inference) aimed at optimizing the overall residuals error across its layers (and time).

The Rao and Ballard predictive coding model had a very strong appeal from an information theoretic perspective; it allowed the feedforward connections to only relay information that was not already explained by the higher layers, thus leading to an efficient coding strategy with reduced redundancy in the transmitted signal. Indeed, a similar strategy of delta compression already existed as a solution for signal transmission in engineering[164,86], and is still used to store programming codes on websites such as Github. Funnily enough, in 1982, Srinivasan et al.[210] and colleagues had already proposed a theory based on a similar principle (and name) to explain the role of inhibitory connections in the retina.

The predictive coding principle has found its utility in explaining various as-

pects of the brain. For example, in the retina, the retinal ganglion cells exhibit a phenomenon known as surround suppression, whereby the firing rate of a cell decreases when one enlarges the stimulus beyond its classical receptive fields[101]. From a predictive coding perspective, this can be accounted for by postulating that during natural vision the cells always encounter large stimuli encompassing regions beyond their receptive fields. Hence in natural conditions, the predictions from the higher layers are quite good and thereby result in ample inhibition. But presenting a stimulus only in the center of a receptive field is an unnatural scenario, and results in inaccurate predictions, increasing the residual error and thus response of the cell. More interestingly, when Rao & Ballard[183] trained a predictive coding network on natural images, it exhibited several interesting neurons. The neurons in the early layers showed orientation tuning, while neurons in the higher layers learned complex features as observed in visual cortex. The neurons in the network also demonstrated surround suppression.

The removal of correct predictions received from higher layers helps in explaining away the variance in the response, and thus allows a layer to just transmit the unexplained variance further via the feedforward connections. This results in a decorrelation of the transmitted signal, and in the case of retina is in the spatial domain[100]. Similar arguments have been used to extend the principle to the temporal domain to explain the response and tuning properties of cells in the lateral geniculate nucleus (LGN), an immediate higher-order region of the retina[44,54,105].

Predictive coding has been also used to explain other aspects of visual processing. To explain motion processing, Jehee et al.[106] showed that a modified

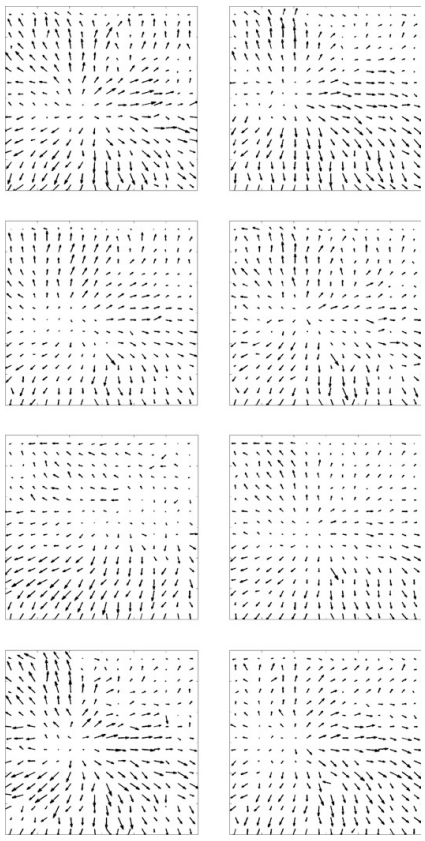


**Figure 1.4: Feedforward receptive fields of neurons in a predictive coding network** after training (representative subset) on natural images. Basis vectors in the model, which can be considered as classical receptive fields of higher-level units, exhibit tuning to components of optic flow such as translation and expansion (Figure from Jehee et al. [106])

predictive coding model can learn neurons with tuning properties (to optic flow) similar to that of the neurons in the medial superior temporal (MST) regions (see Figure 1.4). Hohwy et al. [95] also used predictive coding to explain binocular rivalry, a phenomenon where when each eye is presented with different stimuli, the subjective perception alternates between the two stimuli.

Predictive coding has also been useful beyond the visual system. It has been used to explain the auditory system [202], as it also deals with a lot of temporally

correlated input. Similarly, it is also used to model the hippocampus[146,35], ventral midbrain and striatum[161]. For example, recently Chen et al.[35] proposed a temporal predictive coding inspired network, called PredRAE, for the hippocampus. Their network possessed neurons acting similar to place cells in the hippocampus[163], while also showing other hippocampus-like behaviors such as memory relay and prediction.

Going beyond specific regions, predictive coding has also been used to explain complex cognitive phenomenon. For example, Spratling[207] demonstrated that tweaking the predictive coding dynamics can reconcile them with those of biased competition (another theory that used feedback connections to modulate responses) and can also explain attention[209]. The latter task was earlier attempted by Rao & Ballard[184] while modeling human eye movements. More significantly, predictive coding was used to account for complex cognitive phenomena such as perception, decision-making, etc. when Friston[66] combined it with his free-energy principle.

Alamia & VanRullen[5] implemented a predictive coding network with neural communication delays and observed oscillatory behaviour in the network. More interestingly, when biologically plausible values were used for the communication delays, the oscillations observed occurred at biologically plausible frequencies of around 10 Hertz, characteristic of alpha oscillations.

The fact that higher layers send information to, and thus modulate, lower layers has prompted the use of predictive coding for explaining another cognitive phenomenon – perceptual illusions (and bi- and multi-stable perceptions)[234]. Even

later ANN implementations of predictive coding, including the one presented in Chapter 2, demonstrated an ability to perceive perceptual illusions[140,171]. The information from the higher layers could also be valuable in case of noisy environments. Thus, predictive coding networks have been used to explain robustness in recognition of occluded objects, or as Chapter 2 will illustrate in noisy conditions.

### 1.4.1 Biological evidence for predictive coding

Though the appeal of a unifying theory that explains various phenomena is quite high, the biggest hurdle, the Achilles' heel for predictive coding has been its empirical foundation. As noted earlier, one observes a remarkable consistency in feedback and feedforward connections in the cortex, an observation that has prompted many to wonder whether there is a universal computation that is followed throughout the cortex. Such expectations have found an easy amalgamation with the predictive coding principle. For example, Bastos et al.[14] proposed a "canonical microcircuit" that is repeated throughout the cortex and that implements predictive coding computations. Based on experimental evidence, they also hypothesized the various neurobiological analogues in the laminar structure of the cortex that can be involved in such computations, a task that was further refined by others[199].

But, despite various efforts, finding satisfactory evidence in support of the theory has remained quite difficult. The difficulties in finding satisfactory evidence arise due to multiple reasons. For example, a predictive coding network makes certain assumptions about the brain. First, it assumes an inverted signal trans-

mission where it is the feedback connections that carry information regarding the representations while the feedforward connections carry the corrections for those predictions. It also assumes that these two distinct types of neuronal populations with antagonistic behaviour—one encoding the predictions and the other the errors in them—inhabit the same region of the cortical hierarchy. This makes attributing the increase in activity of a certain region in the brain challenging, especially when the techniques used span a large patch in the brain (See Walsh et al.[233] for a detailed review).

For example, Summerfield et al.[212] repeatedly showed images of human faces to subjects while performing fMRI and measured their blood oxygen level dependent (BOLD) responses in the fusiform face area (FFA). They observed a consistent decrease in the response of the FFA when an expected stimulus was repeatedly shown; a trend that was reversed when the participants were instead shown an unexpected stimulus. This modulation of activity in presence of a repeated stimulus, known as repetition suppression, can be easily inferred from a predictive coding perspective – a repeated stimuli is predictable, thus the prediction error, and consequently the response in the FFA is reduced. As soon as an unexpected face is observed, the prediction error (thus the response) goes up. But the study encountered the same questions as discussed earlier. Attributing a response in a big region such as FFA to just prediction errors was quite controversial. Most important of all, the same results can be explained by accounting for the ability of the neurons to adapt in the presence of a repeated stimulus, called *neural adaptation*. Such a hypothesis doesn't require any reference to predictive coding

and is observed in other places in the brain[231,213,203]. Though the above example of one of the popular studies in neuroscience is quite old by now, subsequent efforts to devise better paradigms have still failed to find conclusive support for the existence of predictive coding[231,203].

Here, it should be noted that not all is lost for the proponents of the theory. Little to no doubt exists in the literature that perception is affected by cognitive and attentional state, effects which are very likely to be of a top-down nature[29,40]. Kok et al.[121] showed that when an expected stimuli is omitted, it elicits an increased response in the visual cortex. This response is specific to the features of the expected stimuli, hinting that they could potentially be predictions from the higher layers (see Figure 1.5). Ekman et al.[57] used 7T fMRI (ultra high field) to measure the BOLD activity in participants observing moving dots on the screen. They reported that only flashing the starting sequence triggered an activity wave in the V1 that resembled the full stimulus sequence, indicating the possibility of (temporal) predictions of the moving dot. Or to explain bistable perception, Weilnhammer et al.[234] used a simple predictive coding based model to simulate and fit fMRI data collected while the participants were observing rotating Lissajous figures. They found that their modelled prediction errors showed a good match in higher regions such the inferior frontal gyrus and in the insula.

Preliminary evidence also shows some support for the existence of the canonical microcircuit. Muckli et al.[153] presented occluded images to participants in a similar 7T fMRI machine. They observed that only the superficial layers in V1 were activated by the occluded part of the image. These superficial cortical layers

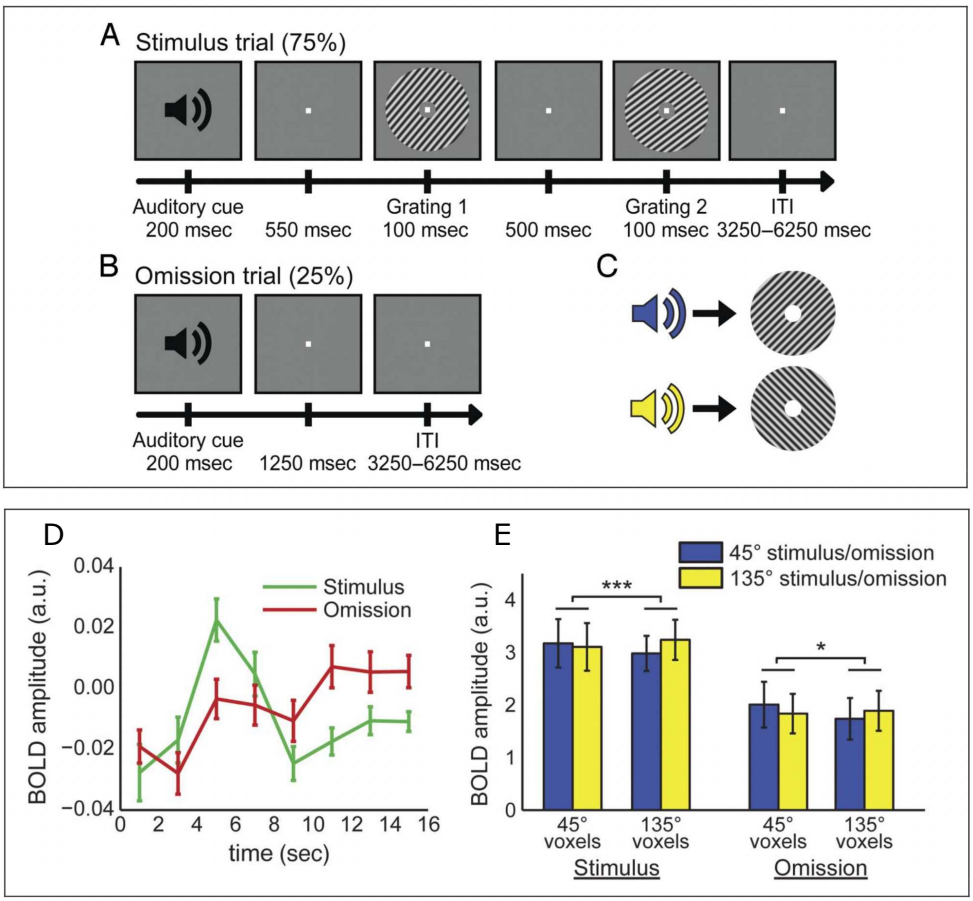



**Figure 1.5: The neural responses elicited due to an expectation of a stimulus are specific to the features of the stimulus** : (A) Experimental setup used by Kok et al.[121]. The trials started with an auditory cue that predicted the orientation of the subsequent stimulus (grating of 45 degrees or 135 degrees). On 75% of the trials, subjects were shown two gratings, first with the expected orientation based on the sound cue, followed by the second grating that was tilted clockwise or anticlockwise by a few degrees (with respect to the first). The subjects later performed a discrimination task where they judged this direction of rotation. (B) In 25% of trials, after the sound cue, no gratings was presented. The participants were asked to just fixate in the center (C) Throughout the experiment, two sound cues were used that hinted at the orientation of the grating (with 100% accuracy). (D) BOLD signals in V1. The time courses are locked to the (expected) onset of visual stimuli. (E) The plot shows the BOLD signal amplitude evoked by 45 degree and 135 degree gratings (in stimulus and omission condition), separately for voxels that preferred the particular grating orientation. The activity observed when the stimulus was omitted represents the prior expectations of the grating orientation. (Figure adapted from Kok et al.[121])

in V1 are known to possess many feedback connections, possibly from higher regions such as V2-V7, and thus support the possibility of this activity being a signature of predictions inline with the assumptions made earlier by Mumford and Bastos et al.

Overall, for now the verdict on the biological plausibility of predictive coding is still out and the question remains one of the biggest hurdles for the proponents of the theory.

### 1.4.2 Predictive coding in the era of deep learning

Despite its shaky empirical support, predictive coding as a principle has also found a place in the modern era of deep learning. Many studies have tried to implement predictive coding inspired principles into ANNs—at times even diverging from the traditional Rao and Ballard architecture—to leverage the efficiency of ML tools such as convolutions, backpropagation, etc.

In the Deep Learning era, one of the first attempts to implement predictive coding dynamics was made by Chalasani & Principe[32] where their simple model dynamically changed the priors using predictive coding based updates. Later, Lotter et al.[139] et al proposed an LSTM based model that performed predictive coding updates while learning video sequences. Their network, called PredNet, not only possessed neurons with response properties very similar to the neurons of macaque visual cortex but also showed gestalt behaviours such as illusory contours[140].

Both Chalasani & Principe[32] and Lotter et al.[139] trained their networks on generative objective functions. Such unsupervised training can lead to represen-

tations that are not ideal for a discrimination task such as object recognition. Thus Spratling[208] and Wen et al.[235] proposed a predictive coding based network that was more attuned for object recognition. Spratling[208] worked with a small, shallow network, and incorporated multiplicative error inline with his previously proposed PC/BC (predictive coding/biased competition) model, whereas Wen et al.[235] worked with large-scale deep networks. A detailed discussion on them is made in Section A.5.

Boutin et al.[21] combined the principles of predictive coding and sparse coding. Their network even possessed neurons with receptive fields akin to those in the primate visual cortex. Their model was able to reconstruct robustly in the presence of small perturbations in the input images. Dora et al.[55] and Brucklacher et al.[23] also designed deep models that were inspired from predictive coding and adhered to simple Hebbian learning rule. Their networks also showed properties like invariant object representations.

Another way in which predictive coding has inspired ideas in ML is that of contrastive predictive coding[166]—an unsupervised objective that tries to reduce prediction errors in the latent space.

Overall, this is a small field within Machine Learning that aims to combine and scale predictive coding with more powerful Machine learning tools.

#### 1.4.2.1 Predictive coding, variational inference and normalizing flows

Starting from its first iteration in 1999, predictive coding was always discussed as a dynamical model in terms of information theory. Friston instead reformulated the predictive coding theory as a variational Bayesian approach while combining it with his free energy principle[66,67]. He demonstrated that the energy function minimized in the Rao and Ballard model can be interpreted as a variational free energy and can be minimized through variational inference.

Variational techniques are a broad set of techniques that aim to learn an often intractable problem by optimizing an approximate tractable alternative. Originally stemming from statistical mechanics in physics[64], they are extensively developed and used in Machine learning[119,108], under the commonly known name of variational autoencoders (or VAEs). Thus, Boutin et al.[22] tried implementing predictive coding iterations more explicitly in VAEs and reported that their network, called iterative-VAE (or iVAE), was more robust against distributional shifts in the data.

As mentioned earlier in the context of retina, the decorrelation (or normalization) observed in the neural response across hierarchies can be explained using the predictive coding theory[100,210]. Thus recently, Marino[144] extended the principle of predictive coding to normalizing flows—a machine learning approach where successive invertible functions are learned that, in the end, approximate a normal distribution[120,188].

#### 1.4.2.2 Predictive coding as an alternative to backpropagation

Predictive coding has found another important utility in the deep learning era—as a potential biological alternative for backpropagation. Backpropagation, in machine learning is the learning rule used to train ANNs. It assumes a differentiable function around the parameter values and applies the (reverse) chain rule to calculate the parameter updates. Thus, as a weight update rule, it assumes that the information about the global loss function is present at a particular level in the system. But such transfer of non-local information throughout the network adds additional constraints to the network architecture and seem to be a strong assumption for the brain[42]. Thus, various biologically plausible learning rules have been proposed to address this problem of backpropagation[16,1,134,131].

In 2017, Whittington & Bogacz[236] demonstrated that a predictive coding network with simple Hebbian learning rule, and thus which relies solely on local information, can approximate the weight updates from backpropagation. Initially tested on simple multilayer perceptrons (MLPs), studies have now extended this ability of (modified) predictive coding networks to efficiently learn complex modern-day networks[150,194,149,205]. Though quite new, this is one active and interesting avenue that predictive coding enthusiasts should keep an eye on.

Now, let's have a brief look at concept cells, as they are found in neuroscience, and machine learning. This will be relevant for the work done in Chapter 2.

## 1.5 Concept Cells

The establishment of the brain as an intricate network of interlinked neurons, raised other important questions related to the nature of the representations. *How does it process the stimulus it observes? If that information is stored somewhere, what is its nature? Is it sparse, where each individual neuron encodes for specific concepts? Or is it distributed, where a representation is spread across multiple neurons?*

One particular hypothesis, prominent in the mid 1900s[11,123] was that it is perhaps an individual neuron that stores such information. This hypothesis was hotly debated in the field for a couple of decades. Jerome Lettvin, a professor at MIT, famously mocked this idea during one of his lectures while mentioning a fictional story in which a doctor treated a character with a troubled relationship with his mother. The doctor removed the cells from the patient's brain that represented the mother; an operation whose success led him on a search for *grandmother cells*[80,13,12]—a term that later became popularly used to dismiss the possibility of such cells.

Further insights into this debate came when Quiroga and colleagues were performing some intracranial recordings. Generally, patients with epileptic seizures are implanted with intracranial electrodes to locate the source of the epileptiform activity; the typical approach being to identify and remove that particular troublesome region. While performing such recordings in the medial temporal region, they found neurons in the brain that selectively fired for stimuli belonging to a spe-

cific concepts—i.e. concept cells![178] These cells showed selective activity whenever the subject was shown any stimulus related to a specific concept. For example, they discovered a neuron that fired specifically for the actress Halle Berry, including her image, her sketch, her written text, and even an image of catwoman, a role she played in the movie. Later efforts also discovered concept cells that responded while performing visual imagery and internal recall of the concepts[124,75].

The discovery of these so-called concept cells once again fueled the age-old debate. But, various researchers argued that the grandmother cell interpretation is an extreme one, and an intermediate interpretation of a sparse network, without the "a neuron per concept" interpretation is quite possible[177].

More importantly, questions emerged on the functional purpose of these concept cells in the medial temporal lobe (MTL). After all, object related information which can lead to rapid recognition, is already detectable in the upstream region of IT cortex[113,186]. While the hippocampus, one of the medial temporal regions, has been linked to function of memory formation[43]. The current leading hypothesis is that these modality invariant concept cells are used in the MTL to encode the concepts into new memories via associative learning[186]. Such a mechanism can explain their delayed responses, their abstract encoding and is inline with the studies where MTL damage showed difficulties in memory formation[93].

Concept cells in ANNs : Multimodal neural networks are studied in Machine Learning for a variety of purposes. Apart from initial attempts to make networks good at one-shot or few-shot tasks, the current methods allow training on a huge corpus of dataset from the internet[49,179,180]. The use of another modal-

ity, language for example, often allows to counterbalance the effect of the noise in the dataset. And these models are known to learn good representations[179,51,18].

Studying concept cells using ANNs is generally tough since, at least as of now, ANNs don't explain the activity in regions apart from the ventral visual stream that well[240]. But when Goh et al.[77] discovered the presence of concept cells in CLIP—a multimodal network trained on data from more than one modality (visual and semantic)—it suddenly raised an important question. Is this ANN model now a better, more apt model for the hippocampus? It is this question that is addressed in the Chapter 3.

## 1.6 Outline of the Thesis:

The current thesis argues that the two fields of Neuroscience and Deep Learning are still very relevant for each other, and neuroscientists should actively look at the developments Deep Learning, and Machine Learning scientists should keep an eye out at what Neuroscience is doing. To support its argument, in the following Chapters it provides two illustrations of how one can benefit from the other.

In Chapter 2, it proposes recurrent dynamics that are inspired from the Predictive coding theory. These dynamics, which are well-suited for Machine Learning models, also induce interesting properties in the networks, in particular robustness against a variety of corruptions. The Chapter then goes beyond and looks under the hood to uncover the mechanisms by which predictive coding renders such robustness.

In Chapter 3, we will use Machine learning for the benefit of neuroscience. Cells

similar to concept cells were discovered in a multimodal network called CLIP[77]. The Chapter reports that CLIP, and multimodal networks in general, are much better models of hippocampus representations—a region known to possess concept cells in humans. Chapter 4 extends this approach to the whole brain by instead performing a searchlight analysis across the brain. It looks at the ability of various uni- and multi-modal networks to explain the activity throughout the brain. Apart from correlation-based analysis, as is typically performed in RSA, it also looks at partial correlations, removing the correlation explained by a control visual model (ResNet50).

Chapter 5 will start by first discussing the works reported in Chapters 2, 3, and 4 in a broader setting. Then it will discuss the implications of this coupling between these two fields, its untapped potential, and the limitations that researchers should be aware of.

Chapter 2

# Predify: Augmenting deep neural networks with brain-inspired predictive coding dynamics

In this Chapter we will propose and implement recurrent dynamics inspired from predictive coding for modern day ANNs. We will then test the performance of the resulting bio-inspired networks against various types of natural and adversarial noises. Insights obtained will help ML to incorporate better biases in future networks.

To further increase their utility to the ML community, an easy-to-use open-source package (along with the weights of all the networks) are made available.

## 2.1 Prologue to the main article :

Every project has a story. The scientific story is itself published in the form of an article and, rightfully so, gets the most attention. Instead, I will use this section to discuss the other, human aspect of the project here. This will also help clarify the contributions made in this essentially collaborative work. The aim of implementing predictive coding into modern day CNNs was Rufin's. Rufin and I started working on finalizing the equation, testing it on a small autoencoder model, characterizing and testing properties such as projection towards the manifold, robustness to natural and adversarial noise, etc. Benjamin Ador tried scaling the implementation to VGG16, a work which later Milad Mozafari brought to fruition. Milad also later generalized the implementation as a python package – later came to be known as *predify* – which helped us test various other state-of-the-art networks. Callum Biggs O'May, instead, took a different approach; he started testing a previously published implementation of predictive coding and highlighted the differences between that and ours. His suggestions not only helped us improve the final equation but were also valuable for our understanding. Andrea Alamia, since the start of the project asked all the relevant questions. His questions and experiments, apart from being compiled as a separate article on its own, played a crucial role in our understanding of the recurrent dynamics. A previous version of this work was presented at Shared Visual Representations in Human and Machine Intelligence (SVRHM), a NeurIPS 2020 workshop. Currently, the work is published in the proceedings of NeurIPS 2021 conference.

## 2.2 Main article :

### 2.2.1 Abstract

Deep neural networks excel at image classification, but their performance is far less robust to input perturbations than human perception. In this work we explore whether this shortcoming may be partly addressed by incorporating brain-inspired recurrent dynamics in deep convolutional networks. We take inspiration from a popular framework in neuroscience: "predictive coding". At each layer of the hierarchical model, generative feedback "predicts" (i.e., reconstructs) the pattern of activity in the previous layer. The reconstruction errors are used to iteratively update the network's representations across timesteps, and to optimize the network's feedback weights over the natural image dataset–a form of unsupervised training. We show that implementing this strategy into two popular networks, VGG16 and EfficientNetB0, improves their robustness against various corruptions and adversarial attacks. We hypothesize that other feedforward networks could similarly benefit from the proposed framework. To promote research in this direction, we provide an open-sourced PyTorch-based package called *Predify*, which can be used to implement and investigate the impacts of the predictive coding dynamics in any convolutional neural network.

### 2.2.2 Introduction

Deep convolutional neural networks (DCNNs), initially inspired by the primate visual cortex architecture, have taken big strides in solving computer vision

tasks in the last decade. State-of-the-art networks can learn to classify images with high accuracy from huge labeled datasets[127,201,87,98,3,215]. This rapid progress and the resulting interest in these techniques have also highlighted their various shortcomings. Most widely studied is the sensitivity of neural networks, not only to perturbations specifically designed to fool them (so-called "adversarial examples") but also to regular noises typically observed in natural scenes[214,92,159]. These shortcomings indicate that there is still room for improvement in current techniques.

One possible way to improve the robustness of artificial neural networks could be to take further inspiration from the brain. In particular, one major aspect of the cerebral cortex that is missing from standard feedforward DCNNs is the presence of feedback connections. Recent studies have stressed the importance of feedback connections in the brain[116,110], and have shown how artificial neural networks can take advantage of such feedback for various tasks such as object recognition with occlusion[62], or panoptic segmentation[137]. Feedback connections convey contextual information about the state of the higher layers down to the lower layers of the hierarchy; in this way, they can constrain lower layers to represent inputs in meaningful ways. In theory, this could make neural representations more robust to image degradation[237]. Merely including feedback in the pattern of connections, however, may not always be sufficient; rather, it should be combined with proper mechanistic principles.

To that end, we explore the potential of recurrent dynamics for augmenting deep neural networks with brain-inspired predictive coding (supported by am-

ple neuroscience evidence[14,100,90,2,233]). We build large-scale hierarchical networks with both feedforward and feedback connections that can be trained using error backpropagation. Several prior studies have explored this interesting avenue of research[32,139,235,21], but with important differences with our approach (see Section 2.2.4). We demonstrate that our proposed method adds desirable properties to feedforward DCNNs, especially when viewed from the perspective of robustness.

Our contributions can be summarized as follows:

- We propose a novel strategy for effectively incorporating recurrent feedback connections based on the neuroscientific principle of predictive coding.
- We implement this strategy in two pre-trained feedforward architectures with unsupervised training of the feedback weights, and show that this improves their robustness against different types of natural and adversarial noise.
- We suggest and verify that an emergent property of the network is to iteratively shift noisy representations towards the corresponding clean representations—a form of "projection towards the learned manifold" as implemented in certain adversarial defense methods.
- To facilitate research aimed at using such neuroscientific principles in machine learning, we provide a Python package called *Predify* that can easily implement the proposed predictive coding dynamics in any convolutional neural network with a few lines of code.

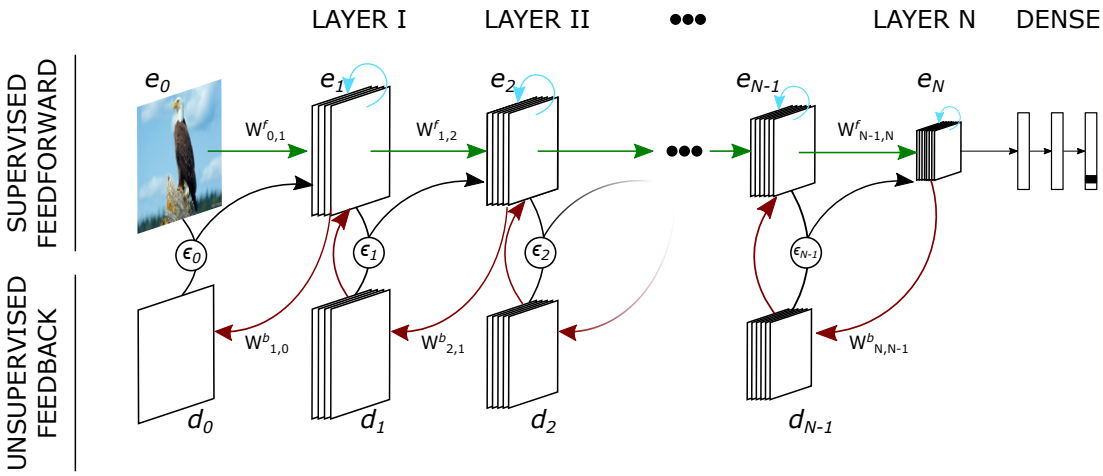



**Figure 2.1: General overview of our predictive coding strategy** as implemented in a feedforward hierarchical network with generative feedback connections. The architecture (roughly similar to stacked auto-encoders) consists of $N$ encoding layers $e_n$ and $N$ decoding layers $d_n$. $W_{m,n}$ denotes the connection weights from layer $m$ to layer $n$, with $W^f$ and $W^b$ for feedforward and feedback connections, respectively. The reconstruction errors at each layer are denoted by $\varepsilon_n$. The feedforward connections (green arrows) are trained for image classification (in a supervised fashion), while the feedback weights (red arrows) are optimized for a prediction (i.e. reconstruction) objective (unsupervised). Predictive coding minimizes the reconstruction errors in each layer by updating activations in the next layer accordingly (black arrows). Self-connections (memory) are represented by blue arrows.

### 2.2.3 Our Approach

#### 2.2.3.1 The proposed predictive coding dynamics

Predictive coding, as introduced by Rao & Ballard [183], is a neurocomputational theory positing that the brain maintains an internal model of the world, which it uses to actively predict the observed stimulus. Within a hierarchical architecture, each higher layer attempts to predict the activity of the layer immediately below, and the errors made in this prediction are then utilized to correct the higher-layer activity.

To establish our notation, let us consider a hierarchical feedforward network

equipped with generative feedback connections, as represented in Figure 2.1. The network contains $N$ encoding layers $e_n$ ($n \in \mathbb{N}$) and $N$ corresponding decoding layers $d_{n-1}$. The feedforward weights connecting layer $n-1$ to layer $n$ are denoted by $W^f_{n-1,n}$, and the feedback weights from layer $n+1$ to $n$ by $W^b_{n+1,n}$. For a given input image, we first initiate the activations of all encoding layers with a feedforward pass. Then, over successive recurrent iterations (referred to as timesteps $t$), both the decoding and encoding layer representations are updated using the following equations (also refer to Pseudocode 1):

$$d_n(t) = W^b_{n+1,n} e_{n+1}(t) \tag{2.1}$$

$$e_n(t+1) = \beta_n W^f_{n-1,n} e_{n-1}(t+1) + \lambda_n d_n(t) + (1 - \beta_n - \lambda_n) e_n(t) - \alpha_n \nabla \varepsilon_{n-1}(t), \tag{2.2}$$

where $\beta_n$, $\lambda_n$ ($0 \leq \beta_n + \lambda_n \leq 1$), and $\alpha_n$ act as layer-dependent balancing coefficients for the feedforward, feedback, and error-correction terms, respectively. $\varepsilon_{n-1}(t)$ denotes the reconstruction error at layer $n-1$ and is defined as the mean squared error (MSE) between the representation $e_{n-1}(t)$ and the predicted reconstruction $d_{n-1}(t)$ at that particular timestep. Layer $e_0$ is defined as the input image and remains constant over timesteps. All the weights $W^f_{n-1,n}$ and $W^b_{n+1,n}$ are fixed during these iterations.

Each of the four terms in Equation 2.2 contributes different signals, reflected by different arrow colors in Figure 2.1: (i) the feedforward term (green arrows;

controlled by parameter $\beta$) provides information about the (constant) input and changing representations in the lower layers, (ii) the feedback correction term (red arrows; parameter $\lambda$), as proposed in[183,89], guides activations towards their representations from the higher levels, thereby reducing the reconstruction errors over time, (iii) the memory term (blue arrows) acts as a time constant to retain the current representation over successive timesteps, and (iv) the feedforward error correction term (black arrows; controlled by parameter $\alpha$) corrects representations in each layer such that their generative feedback can better match the preceding layer. For this error correction term, we directly use the error gradient $\nabla\varepsilon_{n-1} = [\frac{\partial\varepsilon_{n-1}}{\partial e_n^0}, ..., \frac{\partial\varepsilon_{n-1}}{\partial e_n^k}]$ to take full advantage of modern machine learning capabilities (where $k$ is the number of elements in $e_n$). While the direct computation of this error gradient is biologically implausible, it has been noted before that it is mathematically equivalent to propagating error residuals up through the (transposed) feedback connection weights $(W^b)^T$, as often done in other predictive coding imple-

**Pseudocode 1** Predictive Coding Iterations

Input image: $e_0$
**for** $n = 1$ to $N$ **do**
    $e_n \leftarrow Conv(e_{n-1})$
    $d_{n-1} \leftarrow deConv(e_n)$
    $\varepsilon_{n-1} \leftarrow ||d_{n-1} - e_{n-1}||_2^2$
**end for**
**for** $t = 1$ to $T$ **do**
    **for** $n = 1$ to $N$ **do**
        $ff \leftarrow \beta_n \cdot Conv(e_{n-1})$
        $fb \leftarrow 0$
        **if** $n < N$ **then**
            $fb \leftarrow \lambda_n \cdot d_n$
        **end if**
        $e_n \leftarrow ff + fb + (1 - \beta_n - \lambda_n) \cdot e_n - \alpha_n \cdot \nabla\varepsilon_{n-1}$
        $d_{n-1} \leftarrow deConv(e_n)$
        $\varepsilon_{n-1} \leftarrow ||d_{n-1} - e_{n-1}||_2^2$
    **end for**
**end for**

mentations[235,183]. Together, the feedforward and feedback error correction terms fulfill the objective of predictive coding as laid out by Rao and Ballard[183]. We discuss the similarities and differences between our equations and those proposed in the original Rao and Ballard implementation in the Appendix A.6.

While it is certainly possible to train such an architecture in an end-to-end fashion, by combining a classification objective for the feedforward weights $W^f$ with an unsupervised predictive coding objective (see Section 2.2.3.2) for the feedback weights $W^b$, we believe that the benefits of our proposed scheme are best demonstrated by focusing on the added value of the feedback pathway onto a pre-existing state-of-the-art feedforward network. Consequently, we implement the proposed strategy with two existing feedforward DCNN architectures as backbones: VGG16 and EfficientNetB0, both trained on ImageNet. We show that predictive coding confers higher robustness to these networks.

#### 2.2.3.2 Model architectures and training

We select VGG16 and EfficientNetB0, two different pre-trained feedforward networks on ImageNet, and augment them with the proposed predictive coding dynamics. The resulting models are called PVGG16 and PEfficientNetB0, respectively. The networks' "bodies" (without the classification head) are split into a cascade of $N$ sub-modules, where each plays the role of an $e_n$ in equation (2.2). We then add deconvolutions as feedback layers $d_{n-1}$ connecting each $e_n$ to $e_{n-1}$, with kernel sizes accounting for the increased receptive fields of the neurons in $e_n$ or upsampling layers to match the size of the predictions and their targets (see Ap-

pendix A.2). We then train the parameters of the feedback deconvolution layers with an unsupervised reconstruction objective (with all feedforward parameters frozen). We minimize the reconstruction errors just after the first forward pass, and after a single deconvolution step (i.e. no error correction or predictive coding recurrent dynamics are involved at this stage):

$$\mathcal{L} = \sum_{n=0}^{N-1} \| e_n - d_n \|_2^2, \tag{2.3}$$

where $e_n$ is the output of the $n^{th}$ encoder after the first forward pass and $d_n$ is the estimated reconstruction of $e_n$ via feedback/deconvolution (from $e_{n+1}$).

For both the networks, after training the feedback deconvolution layers, we freeze all of the weights, and set the values of hyperparameters to $\beta_n = 0.8$, $\lambda_n = 0.1$, and $\alpha_n = 0.01$ for all the encoders/decoders in Equation (2.2). We also explore various strategies for further tuning hyperparameters to improve the results (see Appendix A.7 for the chosen hyperparameter values).

#### 2.2.3.3 *Predify*

To facilitate and automate the process of adding the proposed predictive coding dynamics to existing deep neural networks, we have developed an open-source Python package called *Predify*. The package is developed based on PyTorch[173] and provides a flexible object oriented framework to convert any PyTorch-compatible network into a predictive network. While an advanced user may find it easy to integrate *Predify* in their project manually, a simple text-based user interface (in

TOML* format) is also provided to automate the steps. For the sake of improved performance and flexibility, *Predify* generates the code of the predictive network rather than the Python object. Given the original network and a configuration file (e.g. `'config.toml'`) that indicates the intended source and target layers for the predictive feedback, three lines of code are enough to construct the corresponding predictive network:

```
from predify import predify

net = # load PyTorch network
predify(net,'./config.toml') # config file indicates the layers that
                             # will act as outputs of encoders.
```

The Appendix A.1 provides further details on the package, along with a sample config file and certain default behaviours. *Predify* is an ongoing project available on GitHub† under GNU General Public License v3.0. Scripts for creating PVGG16 and PEfficientNetB0 from their feedforward instances and reproducing the results presented in this paper, as well as the pre-trained weights are also available on another GitHub repository‡.

### 2.2.4 Related work

There is a long tradition of drawing inspiration from neuroscience knowledge to improve machine learning performance. Some studies suggest using sparse coding, a concept closely related to predictive coding[100,165,162,33,170], for image denois-

*https://toml.io/en/
†https://github.com/miladmozafari/predify
‡https://github.com/bhavinc/predify2021

ing[141] and robust deep learning[211,117], while other studies focus on implementing feedback and horizontal recurrent pathways to tackle challenges beyond the core object recognition[137,237,89,206,138,69,128,129,181].

Here, we focus specifically on those studies that tried implementing predictive coding mechanisms in machine learning models[32,139,235,21]. Out of these, our implementation is most similar to the Predictive Coding Networks (PCNs) of Wen et al.[235]. These hierarchical networks were designed with a similar goal in mind: improving object recognition with predictive coding dynamics. However, their network (including the feedback connection weights) is solely optimized with a classification objective. As a result, their network does not learn to uniformly reduce reconstruction errors over timesteps, as the predictive coding theory would mandate. We also found that their network performs relatively poorly until the final timestep (see corresponding Figure A.1 in the Appendix A.5), which does not seem biologically plausible: biological systems typically cannot afford to wait until the last iteration before detecting a prey or a predator. In the proposed method, we incorporate the feedforward drive into a similar PC dynamics and train the feedback weights in an unsupervised way using a reconstruction loss. We then show that these modifications help resolve PCNs' issues. We discuss these PCNs[235] further in the Appendix A.5, together with our own detailed exploration of their network's behavior.

Other approaches to predictive coding for object recognition include Boutin et al.[21], who used a PCN with an additional sparsity constraint. The authors showed that their framework can give rise to receptive fields which resemble those

of neurons in areas V1 and V2 of the primate brain. They also demonstrated the robustness of the system to noisy inputs, but only in the context of reconstruction. Unlike ours, they did not show that their network can perform (robust) classification, and they did not extend their approach to deep neural networks.

Spratling[208] also described PCNs designed for object recognition, and demonstrated that their network could effectively recognise digits and faces, and locate cars within images. Their update equations differed from ours in a number of ways: they used divisive/multiplicative error correction (rather than additive), and a form of biased competition to make the neurons "compete" in their explanatory power. The weights of the network were not trained by error backpropagation, making it difficult to scale it to address modern machine learning problems. Conversely, our proposed network architecture and PC dynamics are fully compatible with error backpropagation, making them a suitable option for large-scale problems. Indeed, the tasks on which they tested their network are simpler than ours, and the datasets are much smaller.

Huang et al.[99] also aimed to extend the principle of predictive coding by incorporating feedback connections such that the network maximizes "self consistency" between the input image features, latent variables and label distribution. The iterative dynamics they proposed, though different from ours, improved the robustness of neural networks against gradient-based adversarial attacks on datasets such as Fashion-MNIST and CIFAR10.

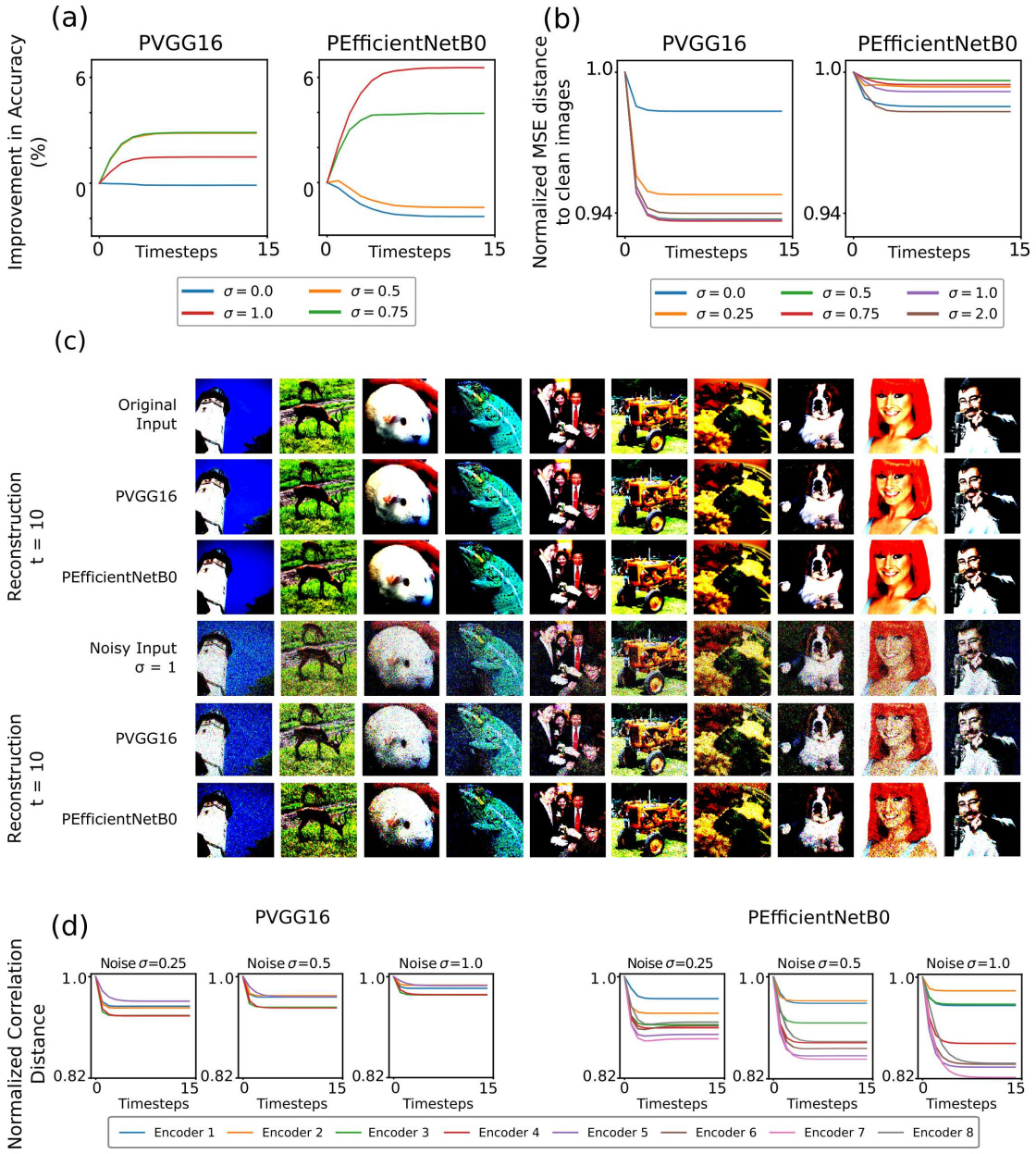



**Figure 2.2: Performance under Gaussian noise and projection towards the learned manifold.** (a) Improvement in recognition accuracy with reference to the feedforward baseline under various levels of Gaussian noise. Both networks demonstrate significant accuracy improvement across timesteps under noisy conditions, while maintaining a performance close to the feedforward level for clean images. (b) Normalized MSE distance between the image reconstruction ($d_0$) and the clean image ($e_0$). Irrespective of the noise level, image reconstruction consistently gets closer to the clean image across timesteps in both models. (c) Examples of clean and noisy input images together with their final reconstruction by the model (the row order from top to bottom is: original image, PVGG16 reconstruction, PEfficientNetB0 reconstruction; noisy image, PVGG16 reconstruction, PEfficientNetB0 reconstruction). For best viewing, we recommend zooming in on the electronic version. (d) Normalized correlation distance between representation of clean and noisy images for each encoder ($e_i$) across timesteps. The values are normalized with respect to the feedforward baseline (timestep 0). In both models and all encoders, the noisy representations tend to move toward the clean copies.

### 2.2.5 Results

Here we contrast the behavior of feedforward networks with their predictive coding augmentations. When considered at timestep 0 (i.e., after a single feedforward and feedback pass through the model), the deep predictive coding networks (DPCNs) and their accuracy are—by construction—exactly identical to their standard pretrained feedforward versions. Over successive timesteps, however, the influence of feedback and predictive coding iterations becomes visible. Here, we investigate for both DPCNs (PVGG16 and PEfficientNetB0): (i) how the PC dynamics update the networks' representations across timesteps, and in which direction relative to the learned manifold; (ii) how the networks benefit from PC under noisy conditions, or against adversarial attacks.

#### 2.2.5.1 Performance under Gaussian noise

To understand the evolution of representations and the behavior of the proposed DPCNs, we first investigate their performance under the influence of different levels of Gaussian noise. To this end, we inject additive Gaussian noise to the ImageNet validation set, and monitor the models' performance across timesteps.

In Figure 2.2a we provide the classification accuracy on these noisy images and absolute values in the Table A.4. We observed that both models progressively improve their recognition accuracy relative to their feedforward baseline (timestep 0) over successive iterations while imposing only a minor performance reduction on clean images. In other words, the networks are able to discard some of the noise by leveraging the predictive coding dynamics over timesteps.

#### 2.2.5.2 Projection towards the learned manifold

In order to quantify DPCNs' denoising ability, we evaluate the quality of image reconstructions generated by each network using the mean squared error (MSE) between the clean image and its reconstruction generated by the first decoder. For each DPCN, we normalize these distances, by dividing them by the value obtained for the corresponding feedforward network (at t=0). We provide the absolute values in the Table A.5. As Figures 2.2b-c illustrate, the reconstructions become progressively cleaner over timesteps. It should be noted that the feedback connections were trained only to reconstruct clean images; therefore, this denoising property is an emerging feature of the PC dynamics.

Next, we test whether the higher layers of the proposed DPCNs also manifest this denoising property. Hence, we pass clean and noisy versions of all images from the ImageNet validation set through the networks, and measure the average correlation distance between the clean and noisy representations of each encoder at each timestep. As done above, these correlation distances are then normalized with the distance measured at timestep 0 (i.e., relative to the standard feedforward network). For both the networks, the correlation distances decrease consistently over timesteps across all layers (see Figure 2.2d). This implies that predictive coding iterations help the networks to steer the noisy representations closer to the representations elicited by the corresponding (unseen) clean image.

This is an important property for robustness. When compared to clean images, noisy images can result in different representations at higher layers[239] and consequently, produce significant classification errors. Various defenses have aimed

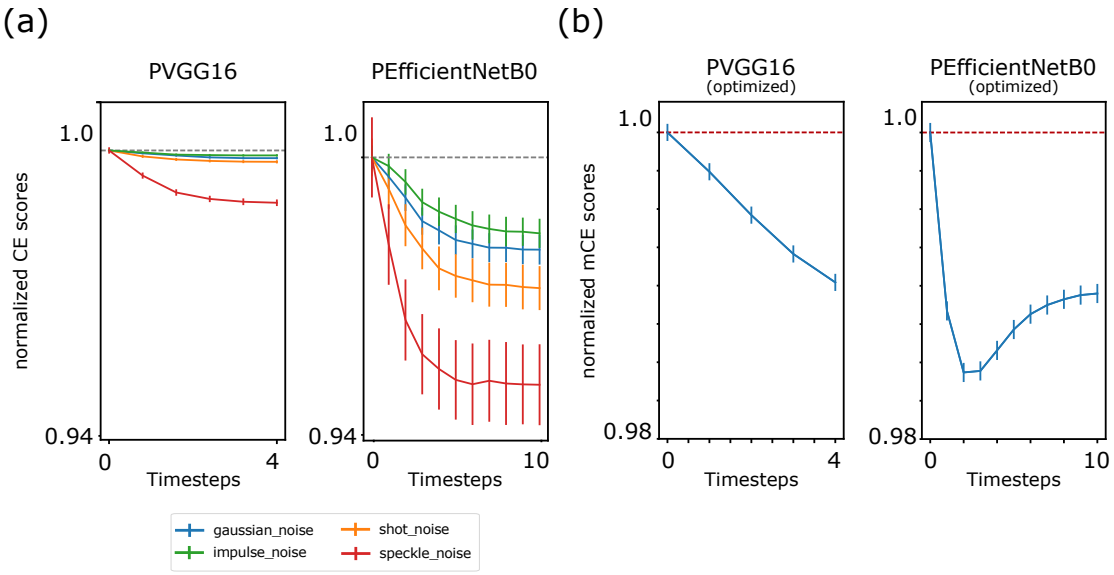



**Figure 2.3: Benchmarking robustness to ImageNet-C.** (a) Normalized corruption errors (CE) of PVGG16 and PEfficientNetB0 under four types of additive noise corruptions. The values are normalized with respect to the feedforward baseline. Both networks show consistent reductions in the errors across timesteps. (b) Normalized mean Corruption Error (mCE) scores for PVGG16 and PEfficientNetB0 on all the 19 corruptions available in the ImageNet-C dataset, when optimized hyperparameters are used (as described in the Appendix A.7). The values are normalized with respect to the feedforward baseline. In both the panels, error bars represent the standard deviation of the bootstrapped estimate of the mean value.

to protect neural networks from perturbations and adversarial attacks by constraining the images to the "original data manifold". Accordingly, studies have used generative models such as GANs[195,107,147,104] or PixelCNNs [204] to constrain the input to the data manifold. Similarly, multiple efforts have been made to clean the representations in higher layers and keep them closer to the learned latent space[239,218,168,190]. Here, we demonstrate that feedback predictive coding iterations can achieve a similar goal by iteratively projecting noisy representations towards the manifolds learned during training, both in pixel (Figure 2.2b-c) and representation spaces (Figure 2.2d).

#### 2.2.5.3 Benchmarking robustness to ImageNet-C

Given the promising results with additive Gaussian noise (Figure 2.2), we extend the noise variety and quantify the classification accuracy of the networks under different types of perturbations. We use ImageNet-C, a benchmarking dataset for noise robustness provided by Hendrycks & Dietterich[92], including 19 types of image corruptions across 5 severity levels each. To begin with, we evaluate DPCNs with pre-defined hyperparameter values (as provided in subsection 2.2.3.2). We observe that they improve the Corruption Error (CE) scores over timesteps for several of the additive-noise corruptions: Gaussian noise, shot noise, impulse noise or speckle noise (see Figure 2.3), but fail to improve the overall mean Corruption Error, or mCE score (the recommended score for this benchmark[92]).

Thus, instead of using pre-defined hyperparameter values, we fine-tune them using two different methods (see Appendix A.7), and repeat the above experiment. As shown in Figure 2.3b, when the hyperparameters are more appropriately tuned for the task, the PC dynamics can increase noise robustness more generally across noise types, resulting in improvements of the mean Corruption Error (mCE) score. The CE plots for individual perturbations along with other recommended metrics (values normalized with AlexNet scores, Relative mCE scores) are provided in the Appendix A.8.

Furthermore, in the Appendix A.9, we demonstrate that we can replicate these observations with a version of PEfficientNetB0 provided by Xie et al.[238] that is robust to corruptions in the ImageNet-C dataset. We show that the recurrent dynamics we propose still help in further improving the mCE score of this already

robust network.

#### 2.2.5.4 Benchmarking robustness to adversarial attacks

Finally, we evaluate the robustness of the networks across timesteps against adversarial attacks. The proposed DPCNs are recurrent models, meaning that their layer representations change on every timestep, and consequently, so do the classification boundaries in the last layer, leading to different accuracy and generalization errors across time (as seen above). To mitigate this effect and properly assess the changes in robustness due to the PC dynamics, for each network we start by selecting 1000 images from the ImageNet validation dataset such that they are correctly classified across all timesteps. Also, we only perform *targeted* attacks so that for each image, the same attack target is given for all timesteps. Using the *Foolbox* library[185], we conduct targeted Basic_Iterative_Method attacks (BIM, with $L_\infty$ norm)[78] for both networks; although it would prove computationally prohibitive to systematically explore all standard types of adversarial attacks, we also evaluated random Projected Gradient Descent attacks (RPGD, with $L_2$ norm)[143], and non-gradient-based HopSkipJump attacks[34] on a subset of 100 images, specifically for PEfficientNetB0. Across various levels of allowed image perturbations (denoted as $\varepsilon$s), the predictive coding iterations tend to decrease the success rate of the attacks across timesteps, for both networks and attacks (see Figure 2.4). That is, DPCNs are more robust against these adversarial attacks than their feedforward counterparts.

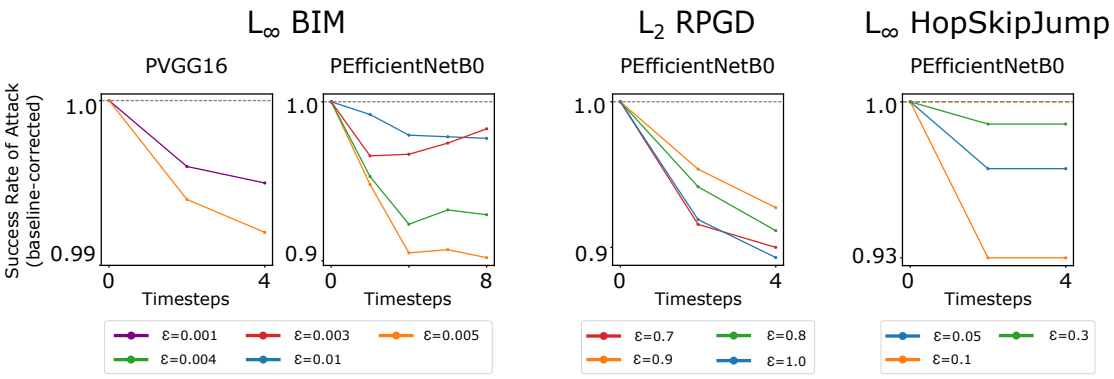



**Figure 2.4: Benchmarking robustness to adversarial attacks.** Plots show the success rate of targeted adversarial attacks against DPCNs across timesteps. The values are baseline-corrected, relative to the success rate at timestep 0 (feedforward baseline). Both networks demonstrate improved robustness to different types and/or levels of perturbations.

### 2.2.6 Discussion and Conclusion

In this work, we explore the use of unsupervised recurrent predictive coding (PC) dynamics, based on neuroscientific principles, to augment modern deep neural networks. The resulting models have an initial feedforward sweep, compatible with visual processing in human and macaque brains[110,221,103,224]. Following this feedforward sweep, consecutive layers iteratively exchange information regarding predictions and prediction errors, aiming to converge towards a stable explanation of the input. This dynamic system is inspired by, and reminiscent of, the "canonical microcircuit" (a central component of cortical structure[14]) that relies on feedback signaling between hierarchically adjacent layers to update its activity. Overall, the augmented networks are closer to the architecture of biological visual systems, while gaining some desirable functional properties. For example, in[172], we also demonstrated that the proposed dynamics help the networks perceive illusory contours in a similar way to humans.

Here, we implemented these PC dynamics in two state-of-the-art DCNs, VGG16 and EfficientNetB0, and showed that they helped to improve the robustness of the networks against various corruptions (e.g. ImageNet-C). We demonstrated that this behavior, at least partly, stems from PC's ability to project both the corrupted image reconstructions and neural representations towards their clean counterparts.

We also tested the impact of our network augmentations against adversarial attacks; here again, we showed that PC helps to improve the robustness of the networks. So far, the most promising strategy for achieving robustness has been adversarial training, whereby adversarial datapoints are added to the training dataset. While efficient, this strategy was also shown to be strongly limited[197,245]. Apart from factors like the choice of the norms used for training, or the high computation requirements, it is ultimately performed with a supervised loss function that can alter the decision boundaries in undesirable ways[245,217]. Most importantly, adversarial training shares very little, if any, resemblance to the way the brain achieves robustness. Instead, here we start from biological principles and show that they can lead to improved adversarial robustness. It is worth mentioning that both our networks achieved robustness totally via unsupervised training of the feedback connections (while of course, the backbone feed-forward networks that we used were pretrained in a supervised manner). We avoided using costly adversarial training, or tuning our hyperparameters specifically for classification under each attack. This likely explains why the models, while improving in robustness compared to their feedforward versions, remain far from state-of-the-art

adversarial defenses. On the other hand, we believe that addition of these methods (adversarial training, hyperparameter tuning) to the training paradigm, in future work, could further improve the networks' adversarial robustness.

For the present experiments, we made a choice of using different objectives for training the feedforward and feedback weights: pre-trained feedforward weights optimized for classification, feedback weights trained with a reconstruction objective (computed after a single time-step). On the one hand, we note that it is perfectly feasible to train a similar predictive coding architecture with a single objective (classification, reconstruction, or otherwise) for both feedforward and feedback weights[172,4]. On the other hand, our choice has several advantages. First, using a feedforward backbone pretrained for classification allowed us to demonstrate the effect of our dynamics on pre-existing state-of-the-art neural networks. Some authors have tried training both feedforward and feedback connections together for classification[235] at the final timestep for relatively smaller networks, but as we discussed in our explorations in the Appendix A.5, we found that the resulting network ended up classifying correctly at the last timestep, with very poor performance during early timesteps. This problem could be addressed by training over time-averaged metrics, such as the average cross-entropy loss for $N$ timesteps. Nonetheless, training the feedback weights for reconstruction instead of classification has the additional advantage that it can be done entirely without supervision. We chose to train the feedback weights for a single time-step, because training with recurrence over multiple timesteps would have required unrolling the network over time. Hence, training a large network like PVGG16 for say 5 or 10

timesteps would incur significant computational challenges. Furthermore, our use of a one-step reconstruction objective allowed us to train the feedback weights independently of the various hyperparameters of our predictive coding dynamics ($\beta$, $\lambda$, and $\alpha$), which only influence the model behavior after the second timestep. Training these weights using recurrence would have required to (i) either fix the values of these hyperparameters beforehand, leading to constraints of expensive hyperparameter explorations; (ii) or directly train these hyperparameters as parameters of the model, probably with additional constraints to prevent the network from reaching trivial values (e.g., if all hyperparameters but the feedforward term $\beta$ converge to zero, the network performs identically to a feedforward one). Finally, from a neuroscience perspective, whether and how the brain combines discriminative and generative representations has been an open question addressed by many researchers, e.g. Al-Tahan & Mohsenzadeh[3], DiCarlo et al.[53], and Huffman & Stark[102]. Our approach of a discriminative (classification-trained) feedforward coupled with generative (reconstruction-trained) feedback could be considered another attempt in this direction.

We speculate that the proposed PC dynamics could help improve robustness in most feedforward neural architectures. To facilitate further explorations in this direction, we provided a Python package, called *Predify*, which allows users to implement recurrent PC dynamics in any feedforward DCN, with only a few lines of code. *Predify* automates the network building, and thus simplifies experiments. On the other hand, there is as yet no established method or criteria to automate the process of identifying the appropriate number of encoding layers,

their source and target layers in the DCN hierarchy, and the corresponding hyperparameter values. This remains an open research question, and a requirement for manual explorations and tuning from *Predify* users. For instance, our own explorations with augmenting ResNets through *Predify* proved difficult, and failed in some situations but succeeded in others. More specifically, as developed in Alamia et al.[4] using *Predify*, ResNet augmentations always achieved noise robustness when the hyperparameter values (controlling the feedforward, feedback, and memory terms) could be tuned separately for each noise type; but we found it challenging to identify a single set of hyperparameters that could generalize to all noise types. Nonetheless, we are hopeful that the package will prove useful to the community. The code is structured such that users can readily adapt it to test their hypotheses. In particular, it should allow both proponents and opponents of the predictive coding theory to investigate its effects on any DCN.

Overall, this work contributes to the general case for continuing to draw inspiration from biological visual systems in computer vision, both at the level of model architecture and dynamics. We believe that our user-friendly Python package *Predify* can open new opportunities, even for neuroscience researchers with little background in machine learning, to investigate bio-inspired hypotheses in deep computational models, and thus bridge the gap between the two communities.

### 2.2.7 Broader Impacts

The research discussed above proposes novel ways of using brain-inspired dynamics in current machine learning models. Specifically, it demonstrates a neuro-inspired method for improving the robustness of machine learning models. Given that such models are employed by the general public, and are simultaneously shown to be heavily vulnerable, research efforts to increase (even marginally) or to understand their robustness against mal-intentioned adversaries has high societal relevance.

Importantly, the research also aims to bridge techniques between two different fields–neuroscience and machine learning, which can potentially open new avenues for studying the human brain. For example, it could help better understand the unexplained neural activities in patients, to improve their living conditions, and in the best case, in the treatment of their conditions. While this may also be associated with inherent risks (related to privacy or otherwise), there are clear potential benefits to society.

The likelihood of sentient AI arising from this line of research is estimated to be rather low.

### 2.2.8 Acknowledgements

This work was funded by an ANITI (Artificial and Natural Intelligence Toulouse Institute) Research Chair to RV (ANR grant ANR-19-PI3A-0004), as well as ANR grants AI-REPS (ANR-18-CE37-0007-01) and OSCI-DEEP (ANR-19-NEUC-0004).

## 2.3 Epilogue to the main article:

In this Chapter, we first implemented bio-inspired recurrent dynamics into Artificial Neural Networks. The resulting networks were then demonstrated to be robust against a variety of perturbations – natural and adversarial. Thus, in-line with the aim of first approach, the Chapter provides one illustration as to how inspiration from biological systems can help modern networks, demonstrating the utility of an overlap in one direction (from Neuroscience to AI).

The subsequent Chapters take the reverse approach of using Machine Learning for Neuroscience.

# Chapter 3

# Multimodal neural networks better explain multivoxel patterns in the hippocampus

In-line with the second approach of using Machine Learning models to understand Neuroscience, in this Chapter, we will aim to use ANNs to understand the representations in the human brain activity. The current Chapter focuses on one particular anatomically define region—the hippocampus—mostly due to the nature of the question it asks. It then tries to uncover how various constraints during training affect the ability of the representations (of ANNs) to explain the activity in the human hippocampus. Such insights will help neuroscience to build better networks in the future that are more apt for investigating the brain.

## 3.1 Prologue to the main article :

Given our interest in multimodal networks, the CLIP model from OpenAI was already discussed quite heavily in the lab. I had also tried using it in my searchlight analysis (discussed in Chapter 4) to understand which parts of the brain were explained better/differently than purely visual models. Thus when OpenAI released their paper reporting concept cells in the CLIP, it raised a very simple question – *Can CLIP now explain the activity of the hippocampus better?* I worked on this question with the help of Leila and Rufin. Milad provided all the beta values (voxel activities) of the subjects in the publicly available fMRI dataset from Kamitani's lab, and helped in solving my innumerous fMRI questions. This paper was first accepted at Shared Visual Representations in Human and Machine Intelligence (SVRHM, a NeurIPS workshop) as an oral presentation, where it was also awarded the "Creative Directions in AI" award. Currently, the work is published in the journal *Neural Networks*.

## 3.2 Main article :

### 3.2.1 Abstract

The human hippocampus possesses "concept cells", neurons that fire when presented with stimuli belonging to a specific concept, regardless of the modality. Recently, similar concept cells were discovered in a multimodal network called CLIP[179]. Here, we ask whether CLIP can explain the fMRI activity of the human hippocampus better than a purely visual (or linguistic) model. We extend

our analysis to a range of publicly available uni- and multi-modal models. We demonstrate that "multimodality" stands out as a key component when assessing the ability of a network to explain the multivoxel activity in the hippocampus.

### 3.2.2 Introduction

Deep neural networks, or DNNs—a hallmark of the recent breakthroughs in machine learning—can solve complex tasks going beyond computer vision to tasks requiring semantic knowledge and understanding, features characteristic of human intelligence (e.g., story completions, context-based question answering, code generation etc.,). This feat has been made possible by both the ability of DNNs to learn expressive representational spaces that enable them to carry out these complex tasks, as well as by the development of improved optimization algorithms required to train them.

Importantly for the neuroscience community, DNNs also provide a potential model for understanding the human brain. Their mathematical pliability combined with their unprecedented expressivity has opened up novel avenues to investigate the human brain. Efforts are being made to understand the similarities and differences between these two systems due to their architectures, dynamics, behavioral patterns, and representational structures[243,81,39,114].

At the same time, DNNs themselves are getting better and better on more human-like tasks. Recently, Radford et al.[179] proposed a model that could simultaneously learn visual and linguistic information from a huge dataset using a constrastive loss function. Importantly, this multimodal model, known as CLIP

(Contrastive Language-Image Pre-training), was found to possess neurons in its last layer that encoded specific concepts[77]. These artificial neurons are reminiscent of 'concept cells' in the human medial temporal lobe (MTL)[178,186], biological neurons that appear to represent the meaning of a given stimulus or concept in a manner that is invariant to how that stimulus is actually experienced by the observer. For example, a single neuron in the human hippocampus showed incredible specificity in its response to the actress Halle Berry. This neuron responded to different images of the actress, including to photographs in which she was disguised as Catwoman (her starring role in a movie by the same name). The same neuron also responded to a semantic representation of the concept, i.e. to the letter string "HALLE BERRY". Other studies have since shown that "concept cells'' are also activated when stimulus information is provided in other sensory modalities, for example when the name of the person is spoken out loud[176].

The discovery of concept cells in artificial networks raises a natural question — *Can a multimodal model like CLIP explain the activity of brain regions known to possess concept cells better than a purely visual model?* In this work we investigate this question by using publicly available fMRI data[97], and asking if CLIP can explain the activity of the hippocampus region better than a comparable feedforward visual model, i.e., ResNet. Because fMRI data does not provide us with the spatial resolution to identify individual concept cells, we address this question at the level of multi-dimensional representation spaces rather than at the level of individual neurons. We also extend our analysis to a variety of models from the literature, trained with unimodal or multimodal objectives. Using Representational

Similarity Analysis (RSA)[125], we report that multimodal networks consistently rank higher than their unimodal counterparts in their ability to explain fMRI activity in the hippocampus*.

### 3.2.3 Methods

#### 3.2.3.1 RSA

Representational Similarity Analysis (RSA) [125] compares representations across different high-dimensional spaces (e.g., brain multi-voxel spaces, model latent spaces, etc.). A first step in RSA consists of constructing Representational Dissimilarity Matrices (RDMs) in each space. RDMs are two-dimensional matrices, in which each element measures the pairwise distance between two stimulus conditions. In this work, we use the Pearson correlation distance (defined as 1 - correlation) to construct the RDMs, and subsequently compare them with the Pearson's $r$ correlation coefficient. Results with other choices of metrics are shown in the Appendix B.

#### 3.2.3.2 fMRI data

For our investigations, we use publicly available data from Horikawa & Kamitani[97]. This dataset consists of fMRI data collected on five healthy participants viewing images from a subset of categories available in ImageNet. Participants performed a one-back test in the scanner in which they had to press a button when the same image was repeated on two consecutive presentations. The data were

*Code to reproduce our results is available at: `https://github.com/bhavinc/mutlimodal-concepts`

collected on 1200 training images that were presented once, and 50 test images presented 35 times each. For our experiments, we restrict ourselves to the subset of test images since the higher number of repetitions provides a more robust estimate of the multi-voxel representation of each image.

We preprocessed the raw data with a standard pipeline using SPM12[68]: slice-time correction, realignment, and coregistration to the T1W anatomical images. We performed a GLM using regressors for each image (the onset and duration), along with regressors for 'fixation' and 'one-back'. For each subject, the beta coefficients obtained from the GLM were transformed into a common MNI305 space using FreeSurfer[†] to allow analysis across subjects. We defined four regions of interest (ROIs) using the Desikan-Killiany atlas for both the left and right hemispheres: a visual ROI comprising the lateraloccipital and pericalcarine regions, a fusiform ROI, a hippocampal ROI and a parahippocampal ROI. fMRI RDMs were built using the beta values in each ROI for each subject. Since 50 image conditions were compared, each RDM was 50x50 in squareform.

#### 3.2.3.3 Models

We include a variety of models in our analysis to facilitate interpretation and discovery of underlying trends in different classes of models. All the included models are publicly available, and possess a ResNet50 backbone to minimize architectural differences.

For CLIP, we used the visual CLIP-RN50 backbone (called CLIP hereafter),

[†] `http://surfer.nmr.mgh.harvard.edu`

which was jointly trained along with a linguistic head (called CLIP-L hereafter) on a contrastive learning task on 400M image-caption pairs[179]. Additionally we also considered visual features from TSM[6], another multimodal model that is trained with a contrastive objective on the HowTo100M dataset[148], in a task that comprises three modalities (video, text, and audio). The impact of contrastive learning objectives on the features of these models can be compared to VirTex and ICMLM, multimodal networks trained with different objectives. For Virtex, the visual backbone is trained on an image captioning task[49], while for ICMLM, the visual features are trained on a text-unmasking task[196]. Both VirTex and ICMLM are trained on MS-COCO[135], a much smaller dataset compared to those used for CLIP or TSM.

To tease apart the effect of multimodal training, we also included visual-only models in our comparisons. Since dataset size has been suggested to affect the quality of representations learned by a network, we considered two visual-only models trained on different datasets. We used the standard ResNet50 model (the control visual model) trained on ImageNet-1K, as well as BiT-M, a ResNet50 backbone trained on the significantly larger ImageNet-21K dataset[122]. We also included adversarially robust models (AR-L2, AR-L4, AR-L8) from[59] in our comparisons. These models are trained to be robust to minute perturbations to the input images by explicitly incorporating such perturbed (adversarial) images[214] in the training dataset. These models have been observed to possess more human-like features[193], making them particularly relevant to our analysis.

Unlike human observers who rely on shapes, standard ImageNet models are

strongly biased by the texture of images[73]. Therefore, Geirhos et al.[73] designed a stylized version of ImageNet to train models that have a stronger bias towards shape than texture. To assess whether representations optimized for human-like biases are better at explaining brain activity in MTL regions, we included three StylizedImageNet models in our comparisons: (i) a model pretrained on only StylizedImageNet (SIN) images, (ii) a model trained on SIN images and ImageNet combined (SIN-IN), and (iii) the SIN-IN model further fine-tuned on ImageNet (SIN-IN+FIN).

Finally, apart from visual and multimodal models, we also included language models: GPT-2[180], BERT[52], as well as CLIP-L. Although these models are not trained to process visual data, they provide an important basis for comparison along with visual and multimodal networks.

For multimodal and visual backbones, we used the test images shown to the human participants and obtained their feature representations from the final average pooling layer. For language models, for each image, we encoded the text 'a photo of {ImageNet label of the image}' to obtain the latent representations. These latent representations were then used to obtain the RDMs of shape 50x50.

#### 3.2.3.4 Voxel Selection from anatomical ROIs based on Noise ceilings

We start by evaluating the signal of the selected beta coefficients. In each ROI, we calculated the noise-ceiling, defined as the average inter-subject correlation between RDMs. The noise ceiling provides an estimate of the reliability of the fMRI

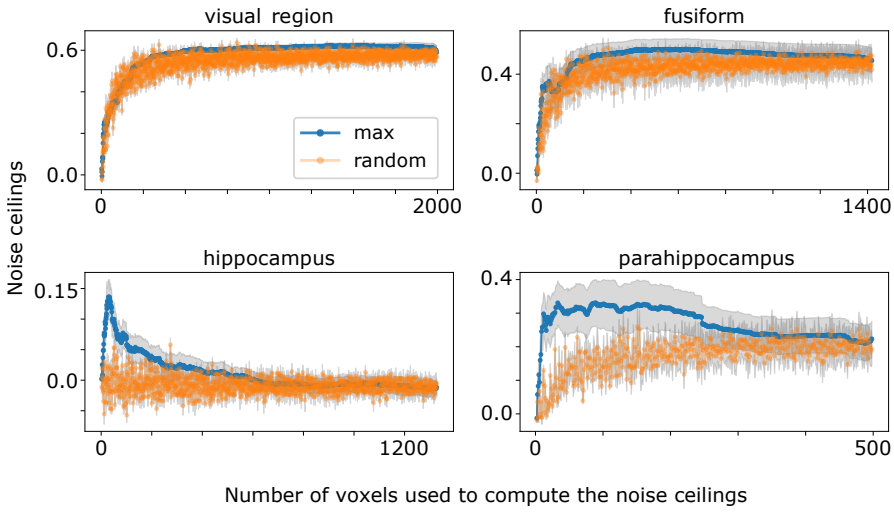



**Figure 3.1: Noise ceilings after selecting subsets of voxels from each region** The panels show the noise ceilings (i.e., inter-subject correlation) calculated after selecting different numbers of voxels from each region of interest. The noise ceilings were computed using either voxels with the highest beta values (blue) or via a random sampling of voxels (orange). The gray regions denote the standard error of mean. For certain ROIs (visual region, fusiform), most voxels are informative about the visual stimulus, and the two selection methods yield similar results. For other ROIs (hippocampus, parahippocampus), the noise ceiling depends on the selection method, implying that some voxels (with the highest betas) are more informative than others (randomly selected). The hippocampus shows an improved noise ceiling when 30 voxels with the highest beta values are selected, with additional voxels degrading the signal.

signal in a given ROI across subjects. Due to the visual nature of the task, the more visual regions (visual ROI, fusiform and parahippocampus) unsurprisingly showed higher values for the noise-ceiling (between 0.2 and 0.6). In contrast, the noise ceiling in the hippocampus was relatively low, and not significantly different from zero ($-0.012 \pm 0.012$). This could be due, in part, to the fact that the fMRI signal in the hippocampus is generally less reliable. However, single neuron recordings in the hippocampus have revealed that only a small proportion of cells ($\approx$15% of recorded cells) is responsive to visual stimuli, and even fewer ($\approx$5%) qualify as "concept cells". In fact, the hippocampus is well-known for its

implication in non-visual tasks, e.g. spatial navigation or memory retrieval and consolidation. If only a small subset of voxels respond to visual stimuli, it stands to reason that a noise ceiling computed across all voxels would not capture any meaningful visual information.

To circumvent this issue, we defined a quantifiable criterion to select a limited number of voxels from each ROI. Specifically, we selected the $N$ voxels with the highest beta value (for any of the 50 stimuli), and calculated the noise ceiling based on this voxel selection. We varied $N$ systematically. As a control, we used random selections of $N$ voxels. As Figure 3.1 shows, the noise ceiling in the more visual regions (visual region, fusiform, parahippocampus) increased rapidly and then stabilized after the inclusion of $\approx$20% of the total voxels. This was true, even when the voxels were randomly selected, indicating that most voxels in these regions carry information about visual stimuli. In the hippocampus however, the noise ceiling was virtually zero when based on random voxel selections: most hippocampal voxels do not appear to encode visual information. Nonetheless, when selecting the $N$ most-activated voxels, the noise ceiling peaked at $\approx$30 voxels, before sharply dropping down to random levels. This is consistent with our hypothesis that although a relatively small number of hippocampal voxels are reliably activated by visual inputs, the signal in these voxels (as measured by the noise ceiling) is reliably above chance. For the main RSA analysis, we thus considered only these top-30 hippocampus voxels. Note that this selection criterion only ensures that the considered brain responses are meaningful, but does not bias the outcome of the RSA with neural network models (i.e., there is no danger of

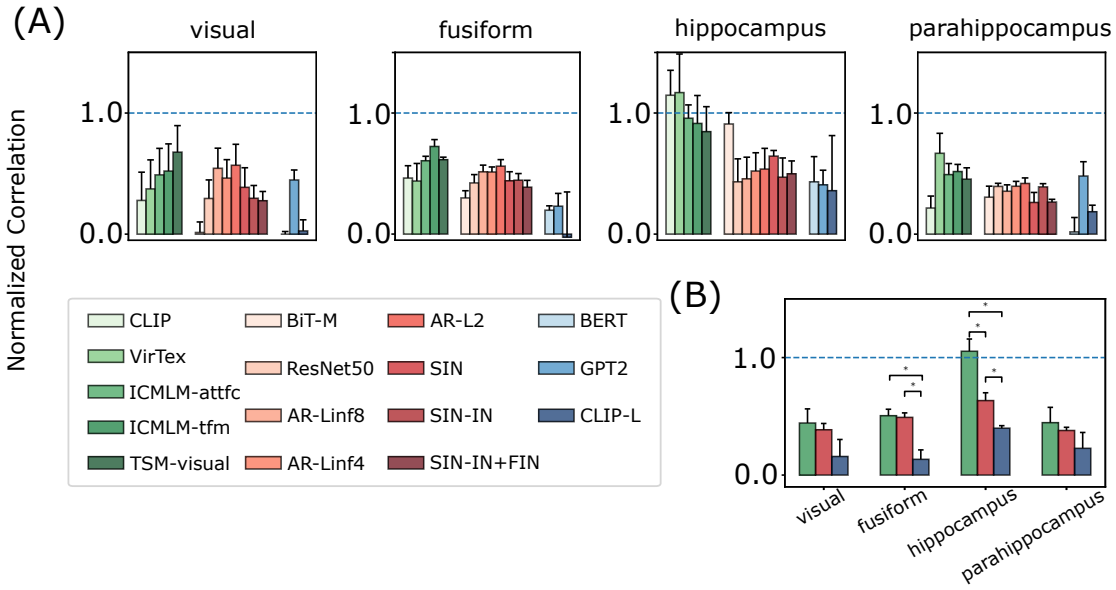



**Figure 3.2: Multimodal models better explain fMRI response patterns in the hippocampus**: Panel A shows the correlation values obtained with different models across selected regions of interest (ROI). Only 30 voxels were selected from each ROI. The values are normalized with the noise-ceilings to facilitate comparisons across regions. Panel B shows the correlation values after aggregating them over mutlimodal (green), visual (red) and language (blue) models. Statistical significance is calculated by using Welch's t-test and is denoted by an asterisk.

circular reasoning). For the other ROIs, we also considered the top-30 voxels for a fair comparison; yet we also report a different selection procedure (based on a fixed beta threshold) in the Appendix B.

### 3.2.4 Results

To investigate whether CLIP explains multivoxel activity patterns in MTL regions better, we computed RSA between the brain RDMs and each model RDM. The noise ceiling places an upper limit on brain-model comparisons because it is an estimate of inter-subject variability. Thus, we normalized the RSA values by the noise-ceiling to allow for comparisons across models and across regions. The

normalized RSA values for each model in each ROI are shown in Figure 3.2A, and averaged across groups of models in Figure 3.2B. (The corresponding non-normalized values are shown in the Appendix B.)

RSA values for the majority of models and brain regions were positive. However, comparisons between individual models (e.g., CLIP vs. ResNet in hippocampus) were not significant (Wilcoxon signed-rank test, $p<0.05$), possibly because of the small number of fMRI participants. Thus, we grouped the models according to their modalities (for example, BERT, GPT2, and CLIP-L as the *language models*) to ask whether one class outperformed the others in explaining brain activity in each ROI. In the hippocampus, in line with our main hypothesis, multimodal models significantly better explained activity patterns compared to both visual and linguistic models (Welch's t-test, $p < 0.05$. Figure 3.2B). In fact, the multimodal networks reached the noise ceiling in the hippocampus, meaning that they could explain all of the explainable variance in brain responses–this result did not happen for any other model group in any other ROI. A similar trend was observed in other regions (even reaching statistical significance in the fusiform ROI), but the RSA values were lower and more variable compared to the hippocampus. Finally, the visual and vision-language models performed systematically better than the linguistic models–as expected since all stimuli were visual.

Above we performed RSA using a subset of 30 voxels that showed the highest beta values in each ROI. While this threshold is reasonable in the hippocampus based on our noise-ceiling calculations (Fig. 3.1), visual regions did have a larger number of voxels with reliable beta values. Thus, in a control analysis, in each ROI

we selected the $N$ voxels that had beta values greater than a common threshold (determined so as to yield 30 hippocampal voxels). Figure B.3 in the appendix shows that including a larger number of voxels had little impact on the main results shown in Figure 3.2. Finally, in the Appendix B, we confirmed that the trends observed in Figure 3.2 are robust to the choice of distance metrics by using other metrics commonly used for fMRI data.

### 3.2.5 Discussion

We applied RSA to study the ability of different neural network models – multi- or uni-modal – to explain the fMRI activity patterns in various brain regions. Based on recent findings[77], our hypothesis was that CLIP (and similar multimodal networks) would be specifically adept at explaining brain activity in the hippocampus–where 'concept cells' are found. This hypothesis was supported by the data: the *multimodal* nature of a model was a key component in explaining the activity in the human hippocampus—a trend that proved robust to different methods of voxel selection and distance metrics.

Recently, Xu & Vaziri-Pashkam[240] casted doubt on the utility of DNNs for explaining representations in higher brain regions—questioning their use for building more brain-like models. Our findings provide a potential way forward to address this limitation: building models that explain higher regions in the brain might require using datasets spanning different modalities. This can be further combined with bio-plausible architectural changes to the DNNs. For example, it would be interesting to investigate the effects of training a bio-inspired recurrent

neural network[36] using multimodal objectives. Combining these architecture- and objective-based approaches could potentially have synergistic effects in learning human-like representations.

### 3.2.6 Acknowledgements

RV is funded by ANR grant OSCI-DEEP (ANR-19-NEUC-0004). LR and RV are both funded by an ANITI (Artificial and Natural Intelligence Toulouse Institute) Research Chair (ANR grant ANR-19-PI3A-0004), as well as ANR grant AI-REPS (ANR-18-CE37-0007-01).

## 3.3 Epilogue to the main article:

In this work, we looked at the ability of different networks to explain the hippocampus activity. The work highlights the effect of different training paradigms on the representations of the ANNs and provides a potential way to improve them in the future. It also aligns well with the known understanding of the modality invariant responses of the hippocampus.

The work hints at inductive biases that can lead to better models of the brain activity in the future. Indeed, they might also aid Machine Learning to build better networks.

# Chapter 4

# Do multimodal neural networks better explain human visual representations than vision-only networks?

## 4.1 Prologue :

The work in this Chapter can be considered a superset of that in Chapter 3, and lies in the same spirit of using ANN models to better explain brain activity. We go from a single region (such as hippocampus) to the whole brain and perform a searchlight analysis. Similarly, in addition to the correlations, we also look at the partial correlations to isolate the effect (on the explained variance) due to each individual factor.

This work started before the previously mentioned project on hippocampus. The aim of this approach was to be region agnostic and analyze the factors that affect the representations. My initial ambition was to use CCA-based analysis on

the fMRI data, which partially stemmed from other projects that I was working on. After discussion with Rufin, we realized that partial correlations was a much more easier approach which Milad already possessed an implementation of. I started the project with the code and the preprocessed fMRI data provided by Milad (on the same publicly available dataset used in Chapter 3). Throughout the process, Leila and Rufin helped in analysing and making sense of the humongous number of partial correlation and correlation plots. The work documented below was submitted and presented at Cognitive Computational Neuroscience (CCN) conference in 2022.

## 4.2 Abstract

Multiple studies have used the representations learned by modern Artificial Neural Networks (ANNs) to explain the activity of the human brain. Efforts up until now have looked at the differences in explainability due to different architectures (such as recurrence vs feedforward or convolution-based vs transformer-based) or different objective functions (supervised vs unsupervised). Here, using multiple uni- and multimodal networks from the literature, we look at another key factor – the modality of the training inputs. Moreover, instead of looking at specific regions of interest or restricting our analysis to the visual ventral stream, we perform correlation- and partial correlation-based searchlight analyses to look at the whole brain. We report that multimodal networks are more similar than their visual counterparts to human fMRI activity in visual regions, and also stand out in their unique ability to explain higher order regions around the superior

temporal sulcus (STS).

## 4.3 Introduction

The remarkable ability of ANNs to learn good representations of complex concepts has led various researchers to hypothesize that they can be used to explore representational spaces of the human brain. Studies have tried finding design principles, based on either architecture or objective functions, that can aid in making more brain-like networks – further improving their utility for neuroscience.

Extending on these approaches, in this work we ask if the uni- or multimodality of the training data is an important factor for explaining human brain activity. Indeed, previous studies have shown that multimodal networks are better at explaining specific regions using representational similarity analysis (RSA) or voxel-based encodings[37,167]. But many of these efforts were limited to specific regions of interest, potentially missing other relevant regions. Here, we overcome this limitation by using a searchlight based analysis on whole brain fMRI data. Additionally, instead of simply assessing whether *multimodality* better explains brain activity, we use partial correlations to tease apart the contribution of other baseline factors such as network architecture.

Specifically, using various publicly available multimodal and unimodal networks, we perform correlation- and partial correlation-based searchlight analyses on fMRI activity over the whole brain. We observe that visual features trained in conjunction with matched language inputs, i.e., multimodal features, are better at explaining the visual cortex compared to features trained only with images.

**Table 4.1:** Models used in this work

| | Model | Training Objective | Dataset |
|---|---|---|---|
| Unimodal | ResNet50[87] | classification | ImageNet[48] |
| | Adversarially Robust (AR) models[60] | adversarial training | ImageNet + adversarial images |
| | SIN models[73] | biased towards shape | StylizedImageNet + ImageNet |
| Multimodal | CLIP[179] | contrastive loss | approx. 400M from internet |
| | ICMLM (-attfc and -tfm)[196] | masked captioning | MS-COCO[135] |
| | VirTex[49] | bicaptioning | MS-COCO[135] |

More surprsingly, they also explain areas around the superior temporal sulcus – regions known to be involved in higher order tasks such as audio-visual integration, motion, face perception and theory of mind[91].

## 4.4 Materials and Methods

**fMRI dataset :** We used a publicly available fMRI dataset[97] to perform our analysis. We preprocessed the raw fMRI data of five subjects and obtained voxel activitation values. We then transformed these values into a common MNI305 space for inter-subject analysis.

For the current work, we restricted ourselves to the test data which contains 50 images shown 35 times each (see[97] for more details).

**Models :** We used various publicly available networks from the literature (see Table 4.1 for a summary). All models had a ResNet50 backbone, thereby minimizing any architectural differences. The 50 test images were passed through each model to obtain their feature vectors (output of the last residual block). Then, using pairwise correlation distance on these feature vectors, a 50x50 representational

dissimilarity matrix (RDM) was constructed.

**Searchlight analysis :** For each participant, we selected a sphere with a radius of five voxels and used their activation values to calculate an RDM. To perform Representation-Similarity Analysis (RSA), a rank correlation coefficient was then calculated between the brain RDM within the searchlight and the model RDM. Additionally, we also calculated partial correlations between these two RDMs, using the standard ResNet50 RDM as the control variable. The searchlight was moved along the brain volume, and RSA was performed within each searchlight.

## 4.5 Results and Discussion

Using a searchlight approach and RSA, we observed that both multimodal and visual networks explained large parts of the visual cortex (data not shown). Though there were some differences in their localization, almost all networks showed high correlation values in lateraloccipital, lingual, and fusiform regions, with an apparent continuum: ResNet50 $<$ SIN $<$ AR $<$ multimodal.

The partial correlation based searchlight allowed us to isolate the effect of each factor – multimodality, adversarial robustness (AR models), or shape-bias (SIN models) – from the common effects captured in the ResNet50 baseline model. We found a few interesting observations. First, compared to the ResNet baseline, both regularization methods used for the visual models (adversarial robustness for all AR models, shape-bias for SININ) made representations more similar to human visual cortex – an effect that was more noticeable for AR models (Figure 4.1, two leftmost columns).

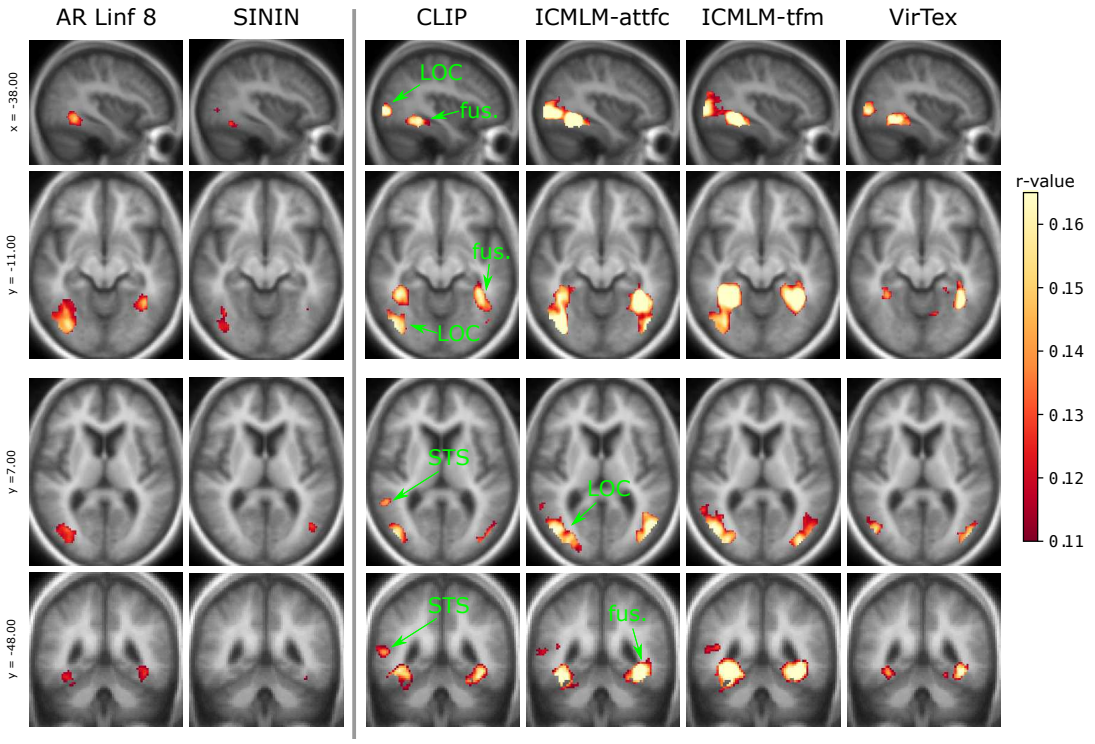



**Figure 4.1: Partial correlations between model RDMs and brain RDMs (with the ResNet RDM as a control variable) :** Each column depicts four slices, the first two columns for uni- and the last four for multi-modal networks. The color scale represents partial correlation values. Multimodal networks show higher similarity to brain representations in the LOC and fusiform regions compared to their unimodal counterparts, thus explaining more unique variance in the brain data. They also explain variance in regions around STS, an effect unseen in the visual models.

This observation was even more striking for all the multimodal networks (CLIP, ICMLMs, and to some extent VirTex) which stood out in these partial correlation brain maps. They explained unique variance in larger parts of the visual cortex, hinting that language-supervised training might be a better form of regularization for learning visual features. More interestingly, the multimodal networks uniquely explained the regions surrounding the right STS (bankssts and inferiorparietal, according to the Deskian-Killiany Atlas; see Figure 4.1, four rightmost columns)–

something unseen for the visual networks.

Overall, in this work, we extend the set of techniques used to study the similarity between the brain and ANN representations. Using partial correlations, we also isolate the impact of multimodal training for explaining visual cortex and higher order regions such as STS. We believe future work that looks at other factors using such region-agnostic methods would provide further insights into the nature of brain representations, and the requirements for building brain-like networks.

## 4.6 Acknowledgments

The authors would like to thank Milad Mozafari for providing his code for the searchlight analysis. RV is funded by ANR grant OSCI-DEEP (ANR-19-NEUC-0004). LR and RV are both funded by an ANITI (Artificial and Natural Intelligence Toulouse Institute) Research Chair (ANR grant ANR-19-PI3A-0004), as well as ANR grant AI-REPS (ANR-18-CE37-0007-01).

## 4.7 Epilogue to the article:

In this Chapter, we looked at the effect of various factors such as robustness, shape bias, unimodality, multimodality, etc. while explaining the different regions in the brain. Two aspects highlight the novelty of the work — (i) the extension of the RSA technique to include partial correlations and region-agnostic searchlight, and (ii) the empirical findings on the unique ability of the multimodal networks to explain the brain activity. The data further corroborates the insights obtained

from Chapter 3, and hints that building multimodal networks will help us explain the activity in various regions in the brain. Efforts like these will help us go farther in our aim to build better models for neuroscience.

*...I believe that the success of deep learning at emulating biological perception is a game-changer that our field cannot ignore. It would be like lighting a fire by hitting stones, with a flamethrower lying on our side...*

– *Rufin VanRullen*[225]

# Chapter 5

# Conclusions

The current thesis attempts at first highlighting, and then advocating for bridging the gap between two seemingly distinct fields that study networks—Neuroscience and Machine Learning. It does this by first demonstrating the utility of ideas from Neuroscience for Machine Learning, and then using and finding suitable ML models for making progress to understand brain activity.

Indeed, an increased overlap between two fields with inherently different objectives possesses numerous limitations of both technological and theoretical nature. But before discussing those, let's look at the work done in the Chapters a little more closely.

## 5.1 Extended discussion on Chapter 2

The work showcased in Chapter 2 aimed to implement predictive coding dynamics into Machine learning networks and test if that makes the networks robust. One of the big advantage of the work was that it allowed us to build relatively more bio-inspired networks where we were able to investigate the implications of

a neurocomputational theory more closely.

As mentioned earlier, the advent of Deep Learning has put a lot of emphasis on building networks that are trained in an 'end-to-end' fashion, most of which are typical feedforward networks. Such a pursuit has also come at a cost of the biological plausibility of such networks. And biological plausibility is an important factor when it comes to building networks suitable for understanding the brain. Since, in the end, how biologically plausible a network is will directly affect how applicable it is to study a real-world biological brain. In this regard, Chapter 2 addressed this concern at two levels. First at an architectural level, it augmented the feedforward ANNs with feedback connections to make them architecturally more similar to the brains. Second, the work also adapted the networks to incorporate bio-inspired dynamics into the neural networks. Based on the principles of a neuroscience theory known as predictive coding, the dynamics iteratively changed the activations in the hierarchical network with an aim to reduce the overall prediction error in the networks.

Generally, implementing predictive coding in large scale hierarchical architectures has remained challenging, especially when paired with (as is typically done) Hebbian based learning rules. In this regards, the work in Chapter 2 makes some important modifications. First, it leverages powerful ML libraries performing out-of-the-box automatic differentiation. This explicit design-choice is also reflected in the error-correction term of the dynamics where the operation is not resolved into a transposed convolution of the feedforward weights but rather performed using the libraries. While this choice of auto-differentiation reduces the strict

biological veridicality of the networks, it allows for learning (or retaining) expressive representations that are useful for tasks such as classification. The proposed dynamics also cast the different streams of inputs to a layer into a simple linear operation modulated by tunable coefficients. These coefficients can be either treated as hyperparameters as they are done in this work, or as parameters that can be trained for different tasks. For example, Alamia et al.[4] took the latter approach and trained these coefficients for different types of noises and investigated the final values obtained by the networks. They found that, with increasing amount of noise in the dataset, the networks (increasingly) relied on higher feedback information. Thus, the reformulation of the dynamics allowed for an explicit exploration of the impacts of top-down and bottom-up information. Also, to make the dynamics easily available for the wider community, Chapter 2 provides an open-source package which is simple enough to be used by engineers and neuroscientists alike. Already a few labs have implemented the dynamics and found the package useful. They have not only successfully reproduced the results shown in Chapter 2, further increasing the confidence on them, but also further explored the impact of predictive coding as a theory.

Most importantly, apart from just implementing the dynamics into a neural network and testing its effect on the network's classification performance, Chapter 2 goes beyond and digs deeper with an aim to uncover the underlying phenomenon. It hints at the interplay between feedforward and feedback information which allows the networks to project the representations obtained on noisy images towards their clean counterparts. This property, dubbed as "projection

towards the learned data manifold", is proposed to be very important for robust systems[195,104]. For example, currently, to guard the networks against adversarial examples, a prominent method that is employed is that of adversarial training, wherein one generates these adversarial images and adds them directly into the training dataset[214,78]. This technique, though quite successful, has a few limitations. First, it is very computationally expensive, as it involves the generation and training of adversarial examples within a particular training loop. Second, it is quite sensitive to the parameters chosen to generate these examples. For example, attributes like the norm and the size of the perturbations become important hyperparameters for performing the adversarial training. Studies have also argued that the current implementations of adversarial training, which typically involve an optimization of a supervised loss function leads to some undesired effects. Hence, various unsupervised alternatives have been proposed in the past few years[245,217]. But more importantly, especially as neuroscientists, the whole principle seems quite unsatisfactory; because we know that the brain does not perform adversarial training! Rather, adversarial training seems like a hack wherein we put the perturbations in the training dataset. This almost sounds like putting the exam questions in class notes so that the students don't fail.

Thus, one can argue that adversarial training doesn't solve the underlying problem. Hence, various groups have advocated for an alternative mechanism where we use some understanding of the training manifold to project the input points. For example, Samangouei et al.[195], one of the early proponents of this approach, used a GAN to select only those points that are closer to the ones seen by the

discriminator during training. This approach has been further echoed by various other works[104,107,195,204]. What is more interesting is that the predictive coding dynamics proposed in Chapter 2 seem to do something very similar. They projected both—the noisy representations and reconstructions—towards their clean counterparts. This could also hint at one potential explanation as to how the brain might be achieving such robustness to noisy stimuli.

This principle of projection towards the learned data manifold can also be extended to explain the perception of illusions in humans. As mentioned earlier, Pang et al.[171] used the dynamics proposed in Chapter 2 and demonstrated that upon incorporation into ANNs, the dynamics rendered the ability to perceive illusory contours in Kanizsa squares to the artificial neural networks.

Efforts like these that aim at understanding the impact of feedback connections and predictive coding are indeed very important. While Chapter 2 makes some progress, a lot remains to be understood as to how top-down information affects the representations. Another interesting approach in this direction was taken by Lindsay et al.[136] where they augmented their ANNs with recurrent connections of three different types — (i) those trained directly for denoising, (ii) those that performed surround suppression, and (iii) those that implemented predictive coding dynamics proposed in Chapter 2. They investigated the effects of these different types of recurrent information on the representations of the networks, and found some interesting differences between them. Such efforts are crucial to further our understanding as to how these two streams of information affect each other.

Predictive coding in particular, seems to have a promising future. In the past

few years, it has found new connections to popular Machine Learning concepts such as backpropagation, normalizing flows, etc. These provide interesting topics for future research. As a purely personal opinion, its use as a biologically plausible alternative to backpropagation seems to hold the highest promise. As discussed earlier, critics of backpropagation have always raised concerns over its biological implausibility. The fact that now predictive coding weight updates can approximate the weights updates from backpropagation, at least under certain conditions, makes it a lot more appealing for both the communities.

One must also note that the Chapter 2 still possesses a multitude of limitations. For example, from a neuroscience perspective, it still fails to provide a conclusive evidence for the existence of predictive coding in the brain. Similarly, while it makes progress in deploying bio-plausible mechanisms into ANNs, it still relies on the use of biologically unlikely automatic differentiation and falls short of building a completely biological network. Even for Deep Learning, when considered solely from the lens of robustness, the proposed dynamics don't help in building fancy state-of-the-art robust networks. It is still much easier to build a more robust network by directly training it on the exact or similar noise types. The work also makes a unique choice of starting with discriminative supervised features and learning unsupervised generative features on top. While this choice was mostly motivated by empirical success, other potentially better choices for optimizing the weights, or the coefficients, need to be explored. Future work should focus on addressing many of such limitations.

## 5.2 Extended discussion on Chapters 3 and 4

In Chapters 3 and 4, the thesis uses ML models to understand the representations in the brain. Here, the motivation was to figure out how the explanation of the brain representations is affected by the different training paradigms. Chapter 3 looks at a specific anatomical region to ask a specific question that was raised after the discovery of concept cells in a multimodal ANN. Chapter 4 instead generalizes this approach and instead looks at a selected group of voxels throughout the brain.

The chapters provide ways to improve the models for studying human representations. For example, one common concern that is raised while using ANNs to study brain representations is that they still can't provide good representations useful for studying the higher order regions in the brain[240], implying that the models that we have are still not suitable for the whole brain and need a little more refining. Thus, pursuits like these that help in identifying and constructing better ANN representations will be very worthwhile for neuroscience. Of course, by elucidating ways in which the factors affect the representational spaces, they will also help in pruning training paradigms for Deep Learning, especially for applications where it aims to build human-interactable ANNs.

Among various factors, Chapters 3 and 4 particularly focus on multimodality. Multimodal objectives are typically used in AI to learn networks that can generalize to other categories, which the networks have either never seen before (called zero-shot learning) or seen only a couple of times (called few-shot learn-

ing), by leveraging the semantic knowledge. This generally also allows to build huge datasets by scraping internet websites for images and their captions, and are known to improve the performance of ANNs[179]. Generally, language supervision is expected to constrain the representations learned in a more natural or human-like way[77,18,179] (but see Devillers et al.[51], Thrush et al.[222] for contradictory evidence and limitations of current techniques). This makes them a strong candidate for explaining the representations in the brain—another network that relies heavily on language supervision and is also known to be modality invariant. Indeed, the current efforts only focus on two modalities, but future efforts should look at networks with multiple modalities, such as audio, video and textual data. Such efforts, if successful, can provide better representations, useful for both, AI and Neuroscience.

The Chapters 3 and 4 are also limited in various ways. The work relies on a low amount of fMRI data. Also, the networks used are just single instances trained with one random initialization. Thus, making any strong claims with high statistical power becomes particularly challenging. Future work should indeed focus on addressing these limitations by using better/larger datasets (such as Natural Scenes Dataset (NSD)[7]) and more instances of networks trained with similar strategies.

## 5.3 Closing thoughts

### 5.3.1 Why should the two meet each other?

A strengthened communication between these two fields will help each in its own quest. Neuroscience can heavily benefit from an easily pliable mathematical model that is expressive enough to understand the brain representations. Such a model will allow it to study the human brains at a representational level, filling an important gap in its understanding.

The ability to understand the brain at a representational level will already open a variety of avenues where interesting questions can be investigated. Answering questions such as: *Do Alzheimer's patients lose memory representations? And if yes, are these losses related to specific memory types or concepts? In which regions? Or what about subjects with synesthesia? Are their representations for numbers or colors structured differently? Again, if yes, in which regions of the brain? On a more clinical side, do the brain representations (say in motor cortex) in patients using prosthetic devices alter over time? If yes, then how?* suddenly become in the realm of technological possibility. Already the empirical success of using ANNs for studying the brain representations in fMRI data, which is limited in its spatial and temporal resolution, has hinted us about the smooth nature of neural code (at least until the ventral stream) in the brain[83].

Indeed, up until now Neuroscience is still establishing the equivalence between the ANNs and the brains at a functional level. A complete understanding of the brain will require its understanding even at a mechanistic level (the last level

of the three Marr's levels). For this, we need more accurate implementations of brain-like networks and ANNs, which can be easily tinkered to add appropriate amount of detail, can be of use here as well (for one such use of ANNs see Tanaka et al.[216]).

For Machine Learning, these insights from neuroscience, functional or mechanistic, will be very useful and can act as guiding principles. Functionally, even now, Machine Learning relies on humans as oracles for various definitions. From what constitutes a conscious AI, or what constitutes an adversarial example or even a shortcut learning rule[71], all definitions that rely on humans in some way. Thus any progress in understanding the human cognition will be directly relevant for improving the ANNs.

Mechanistically, machine learning can make use of the vast amount of ideas from theoretical neuroscience. The field has a number of possible ideas ranging from predictive coding, as Chapter 2 illustrated, to others such as sparse coding and spiking neurons which it can take inspiration from. For example, spiking neural networks can be very energy-efficient compared to their floating point counterparts[219]. Thus, various groups are aiming to build spiking neural networks that can be used on energy efficient neuromorphic hardware[115,24,244]. Even the insights from Chapter 3 and 4, where multimodal networks were able to explain brain representations, can be useful for building future Machine Learning networks that learn better representations. This understanding of human representations will be particularly relevant for ANNs that will find their applications directly dealing with humans either in an augmented and/or virtual reality setting. People

have also trained the ANNs directly on brain data and found that it improves the robustness of the networks. Investigations into these ANNs revealed biases in the feature spaces learned by the networks[132,133]. Machine Learning networks can now actively implement these principles to build future robust networks.

Thus, the two fields have a great deal to learn from each other and can come further together in a potentially synergistic fashion.

### A word of caution for Neuroscience :

Though the advantages of using the ANNs as a proxy for the brain are innumerous, Neuroscience should tread carefully while drawing inferences from them, at least for now.

Since long, Neuroscience has used different metaphors to understand the brain—sometimes equating it to a combination of various aqueducts, or a complex machine such as a tractor, to recently even a computer. Indeed, the brain is none of those. Thus, though the utility of such metaphors might seem to be quite high, various people have warned caution for the use of a new metaphor of the brain—as an ANN or even a mere information processing unit[61,76].

More importantly, on a practical note, there are still a lot of differences between the way brains and ANNs function. Firstly, their architectures are completely different. Brains possess neurons and synapses that are richly complex, show spiking behavior, and are temporal in nature. Moreover, the neurons are affected by a plethora of chemical (neuromodulators) signals apart from their synaptic signaling. These complexities are completely missing in their artificial

cousins which are instantaneous and have floating point activation values. Second, their also seems very little evidence for the possibility of backpropagation in the brain. Of course, how many of these details are important for modeling the brain, and how many are unnecessary is an open question (which probably will be answered only empirically), but one should be careful before equating these two networks. Thus, even if one succeeds in establishing the ANNs as a functional model for the brain at a behavioral or neural activity level, one must be careful in realizing that a more implementational level understanding still eludes us. And an implementational level understanding remains the hole grail for Neuroscience. Guest & Martin[84] provide a comprehensive analysis of the pitfalls that the proponents of ANNs (for studying the brain) often fall into and caution against the prevalent logical fallacies in the literature.

5.3.1.0.1 The ethical limitations No discussion of using AI for the brain and vice versa can be complete without acknowledging the potential harm. While the use of AI and neuroscience will help in building better prosthetic devices that can be no little than a boon for clinical patients, they also have a potential of being misused by nefarious players. Similarly, given that the current ANNs can possess various biases often carried from the datasets, the fields should be careful of designing and drawing inferences from future clinical tests that combine the two. More importantly, given the high rate of deployment of these ML systems, and their possible far reaching effects, the margin for errors, even unintentional, can be quite low. Thus, the two fields should carefully move forward, and while doing so actively aid public awareness and political regulations to deal with the

upcoming challenges.

## 5.4 Final Summary

To summarize, the current thesis argues that the two fields of Neuroscience and Machine Learning are very relevant to each other. Both want to understand and use some form of networks—biological or artificial—to either build better models of the brain or artificial sentient agents. The two fields can join their forces, potentially synergistically, to tackle the questions in their pursuits. Neuroscience can use the high expressive power and mathematical pliability of the ANNs to study the complex representations in the brain, and Machine Learning can utilize the principles from Neuroscience to make better neural networks.

The thesis supports this by providing demonstrations of two approaches. In Chapter 2, it implements recurrent dynamics inspired from neuroscience to make robust neural networks, and in Chapters 3 and 4 it uses ANN representations to explain the fMRI activity of the brains. Both the approaches provide novel insights that have implications for the two fields. Predictive coding networks are able to project noisy representations towards clean ones using feedback connections—a principle that Machine Learning can use to build better ANNs in the future. The insight also provides potential ways the brain might be achieving robustness, thus opening avenues for future experiments in neuroscience. Similarly, the representations learned by multimodal networks were better suited for explaining the representations in the brain (obtained using fMRI data). For ML, this provides principles to learn better representations in the future, whereas for Neuroscience,

the work provides ways to make or choose better models to investigate the brain data.

While arguing for it, the thesis also warns against the potential limitations—technological, theoretical, and ethical—of their intersection. Recently, a new field of NeuroAI is emerging with a similar goal of promoting further discussion between Neuroscience and Artificial Intelligence. The thesis aligns itself with this broad goal, and partakes in it by providing two illustrations as to how the two fields can benefit each other, with an aim to answer and hopefully remove the question mark at the end of its title.

# Appendix A

# Appendix for Chapter 2

## A.1 Getting Started with Predify

Both VGG16 and EfficientNetB0 are converted to predictive coding networks PVGG16 and PEfficientNetB0, using the Predify package. The fastest and easiest way to convert a feedforward network into its predictive coding version is to use Predify's text-based interface which supports configuration files in TOML format.

The current version of Predify assumes that there is no gap between the encoders. Therefore, in the minimal case, one only needs to provide a list of submodule names in the target feedforward network. Then, Predify takes care of the rest by converting each of them into an encoder and assigning default decoders. More precisely, let $x$ and $y$ denote the input and output of a layer (or complex submodule, potentially including multiple layers) that is selected to be an encoder ($e_n$). If $x$ and $y$ respectively have the size $(c_{in}, h_{in}, w_{in})$ and $(c_{out}, h_{out}, w_{out})$; then, the default decoder's structure that predicts this encoder ($d_{n+1}$) is a 2D upscaling operation by the factor of $(h_{in}/h_{out}, w_{in}/w_{out})$ followed by a transposed convolutional

layer with $c_{out}$ channels and $3 \times 3$ window size. The values of hyperparameters will be set to $\beta_n = 0.3$, $\lambda_n = 0.3$, and $\alpha_n = 0.01$

In Predify, each encoder ($e_n$) and the decoder that uses its output to predict the activity of the encoder below ($d_{n-1}$) is called a *PCoder*. To verify the functionality of Predify's default settings, we applied it for PEfficientNetB0 used in this work. Here is the corresponding minimal configuration file:

```
name = "PEfficientNetB0"


input_size = [3,224,224]
gradient_scaling = true
shared_hyperparameters = false


[[pcoders]]
module = "act1"
[[pcoders]]
module = "blocks[0]"
[[pcoders]]
module = "blocks[1]"
[[pcoders]]
module = "blocks[2]"
[[pcoders]]
module = "blocks[3]"
[[pcoders]]
module = "blocks[4]"
[[pcoders]]
module = "blocks[5]"
[[pcoders]]
```

```
module = "blocks[6]"
```

One can easily override the default setting by providing all the details for a PCoder. Here is the configuration corresponding to the PVGG16 used in this work:

```
imports = [
"from torch.nn import Sequential, ReLU, ConvTranspose2d",
]


name = "PVGG16"


input_size = [3, 224, 224]
gradient_scaling = true
shared_hyperparameters = false

[[pcoders]]
module = "features[3]"
predictor = "ConvTranspose2d(64, 3, kernel_size=(5, 5), stride=(1, 1)
    , padding=(2, 2))"
hyperparameters = {feedforward=0.2, feedback=0.05, pc=0.02}

[[pcoders]]
module = "features[8]"
predictor = "Sequential(ConvTranspose2d(128, 64, kernel_size=(10, 10)
    , stride=(2, 2), padding=(4, 4)), ReLU(inplace=True))"
hyperparameters = {feedforward=0.4, feedback=0.1, pc=0.05}


[[pcoders]]
```

```
module = "features[15]"
predictor = "Sequential(ConvTranspose2d(256, 128, kernel_size=(14,
    14), stride=(2, 2), padding=(6, 6)), ReLU(inplace=True))"
hyperparameters = {feedforward=0.4, feedback=0.1, pc=0.008}

[[pcoders]]
module = "features[22]"
predictor = "Sequential(ConvTranspose2d(512, 256, kernel_size=(14,
    14), stride=(2, 2), padding=(6, 6)), ReLU(inplace=True))"
hyperparameters = {feedforward=0.5, feedback=0.1, pc=0.0024}

[[pcoders]]
module = "features[29]"
predictor = "Sequential(ConvTranspose2d(512, 512, kernel_size=(14,
    14), stride=(2, 2), padding=(6, 6)), ReLU(inplace=True))"
hyperparameters = {feedforward=0.6, feedback=0.0, pc=0.006}
```

The network configuration files (in TOML format) are available to download on GitHub*.

## A.2 Network Architectures

VGG16 consists of five convolution blocks and a classification head. Each convolution block contains two or three convolution+ReLU layers with a max-pooling layer on top. For each $e_n$ in PVGG16, we selected the max-pooling layer in block $n-1$ and all the convolution layers in block $n$ of VGG16 (for $n \in \{1, 2, 3, 4, 5\}$)

*https://github.com/bhavinc/predify2021

as the sub-module that provides the feedforward drive. Afterwards, to predict the activity of each $e_n$, a deconvolution layer $d_n$ is added which takes the $e_{n+1}$ as the input. Here, deconvolution kernel sizes are set by taking the increasing receptive field sizes into account.

In the case of PEfficientNetB0, we used PyTorch implementation of EfficientNetB0 provided in https://github.com/rwightman/pytorch-image-models. This implementation of EfficientNetB0 consists of eight blocks of layers (considering the first convolution and batch normalization layers as a separate block). Similar to PVGG16, we convert each of these blocks into an encoder ($e_n$) and add deconvolution layers accordingly. This time we set the kernel size of all deconvolution layers to 3x3 and use upsampling layers to compensate the shrinkage of layer size through the feedforward pathway (i.e. Predify's default setting).

Table A.1 summarizes PVGG16's architecture. Moreover, the hyperparameter values are provided in Tables A.2 and A.3.

**Table A.1:** Architectures of $e_n$s and $d_n$s for PVGG16 and PEfficientNetB0. Conv (channel, size, stride), MaxPool (size, stride), Deconv (channel, size, stride), Upsample (scale_factor), BN is BatchNorm, $[\ ]_+$ is ReLU, and $[\ ]_*$ is SiLU. EfficientBlock corresponds to each block in PyTorch implementation of EfficientNetB0.

| | PVGG16 Input Size: 3x224x224 | | PEfficientNetB0 Input Size: 3x224x224 | |
|---|---|---|---|---|
| | $e_n$ | $d_{n-1}$ | $e_n$ | $d_{n-1}$ |
| PCoder1 | $\left[\text{Conv (64, 3, 1)}\right]_+$<br>$\left[\text{Conv (64, 3, 1)}\right]_+$ | Deconv (3, 5, 1) | [BN (Conv (32, 3, 2))]$_*$ | Upsample (2)<br>Deconv (3, 3, 1) |
| PCoder2 | MaxPool (2, 2)<br>$\left[\text{Conv (128, 3, 1)}\right]_+$<br>$\left[\text{Conv (128, 3, 1)}\right]_+$ | $\left[\text{Deconv (64, 10, 2)}\right]_+$ | EfficientBlock0 | Deconv (32, 3, 1) |
| PCoder3 | MaxPool (2, 2)<br>$\left[\text{Conv (256, 3, 1)}\right]_+$<br>$\left[\text{Conv (256, 3, 1)}\right]_+$<br>$\left[\text{Conv (256, 3, 1)}\right]_+$ | $\left[\text{Deconv (128, 14, 2)}\right]_+$ | EfficientBlock1 | Upsample (2)<br>Deconv (16, 3, 1) |
| PCoder4 | MaxPool (2, 2)<br>$\left[\text{Conv (512, 3, 1)}\right]_+$<br>$\left[\text{Conv (512, 3, 1)}\right]_+$<br>$\left[\text{Conv (512, 3, 1)}\right]_+$ | $\left[\text{Deconv (256, 14, 2)}\right]_+$ | EfficientBlock2 | Upsample (2)<br>Deconv (24, 3, 1) |
| PCoder5 | MaxPool (2, 2)<br>$\left[\text{Conv (512, 3, 1)}\right]_+$<br>$\left[\text{Conv (512, 3, 1)}\right]_+$<br>$\left[\text{Conv (512, 3, 1)}\right]_+$ | $\left[\text{Deconv (512, 14, 2)}\right]_+$ | EfficientBlock3 | Upsample (2)<br>Deconv (40, 3, 1) |
| PCoder6 | - | - | EfficientBlock4 | Deconv (80, 3, 1) |
| PCoder7 | - | - | EfficientBlock5 | Upsample (2)<br>Deconv (112, 3, 1) |
| PCoder8 | - | - | EfficientBlock6 | Deconv (192, 3, 1) |

## A.3 Execution Time

Since we used a variable number of GPUs for the different experiments, an exact execution time is hard to pinpoint. Briefly, depending on the number of

timesteps, analysing mCE scores and adversarial attacks on PEfficientNetB0 took around 15-20 hours on an NVIDIA TitanV gpu. These numbers were about three to four times higher for experiments on PVGG16. For both the networks, training the feedback weights on the ImageNet dataset generally finished before 5 epochs, which took approximately 7-8 hours for a single GPU.

## A.4 Gradient Scaling

In our dynamics, the error ($\varepsilon_{n-1}$) is defined as a scalar quantity whose gradient is taken with respect to the activation of the higher layer ($e_n$). That is,

$$\nabla \varepsilon_{n-1} = \begin{bmatrix} \frac{\partial \varepsilon_{n-1}}{\partial e_n^1} \\ \vdots \\ \frac{\partial \varepsilon_{n-1}}{\partial e_n^L} \end{bmatrix} \tag{A.1}$$

where L denotes the number of elements in $e_n$. The partial derivative with respect to $e_n^j$ can then be written as,

$$\frac{\partial \varepsilon_{n-1}}{\partial e_n^j} = \frac{1}{K} \sum_i^K \frac{\partial (e_{n-1}^i - d_{n-1}^i)^2}{\partial e_n^j} \tag{A.2}$$

$$\tag{A.3}$$

where K is the number of elements in $e_{n-1}$ ( = channels x width x height). Equation A.2 highlights how the dimensionality of the prediction (equivalently

the error term) affects the gradients, scaling them down by a factor K.

This can be easily seen by supposing that the gradients with respect to $e_n^j$ are i.i.d normally distributed around 0 with standard deviation $\sigma$,

$$\frac{\partial(e_{n-1}^i - d_{n-1}^i)^2}{\partial e_n^j} \sim \mathcal{N}(0, \sigma^2) \tag{A.4}$$

$$\sum_i^K \frac{\partial(e_{n-1}^i - d_{n-1}^i)^2}{\partial e_n^j} \sim \mathcal{N}(0, K\sigma^2) \tag{A.5}$$

Thus,

$$\frac{\partial \varepsilon_{n-1}}{\partial e_n^j} = \frac{1}{K}\sum_i^K \frac{\partial(e_{n-1}^i - d_{n-1}^i)^2}{\partial e_n^j} \sim \mathcal{N}(0, \frac{\sigma^2}{K}) \tag{A.6}$$

This scaling is further troublesome in DCNs, where most gradients are zero since they are not part of the receptive field of the element $e_n^j$. Hence assuming that there are only C elements (kernel*channels) that are part of the receptive field of $e_n^j$,

$$\sum_i^K \frac{\partial(e_{n-1}^i - d_{n-1}^i)^2}{\partial e_n^j} = \sum_i^C \frac{\partial(e_{n-1}^i - d_{n-1}^i)^2}{\partial e_n^j} \sim \mathcal{N}(0, C\sigma^2) \tag{A.7}$$

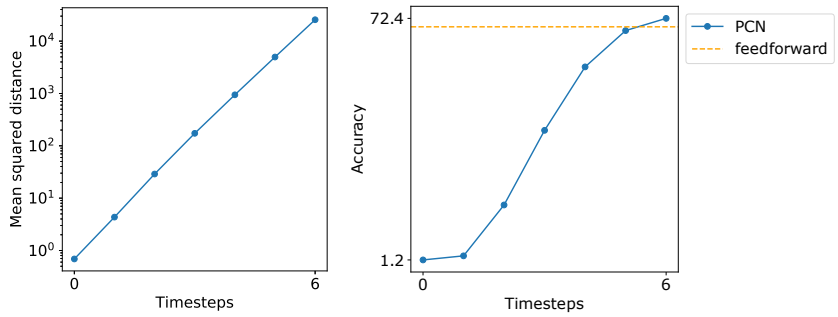
Hence,

$$\frac{\partial \varepsilon_{n-1}}{\partial e_n^j} = \frac{1}{K} \sum_i^C \frac{\partial (e_{n-1}^i - d_{n-1}^i)^2}{\partial e_n^j} \sim \mathcal{N}(0, \frac{C\sigma^2}{K^2}) \tag{A.8}$$

We use Equation A.8 to, at least partly, counteract this effect due to the dimensionality of the prediction errors. We multiply the gradient by a factor of $\sqrt{K^2/C}$ to scale them in a way that is more comparable across layers, and thus apply a more meaningful step size for correcting the errors.

## A.5 Prior work: PCNs

To better understand the model proposed by Wen et al.[235] and its differences to ours, we conducted several experiments using the code that they provided, and report here our most compelling observations. A first striking shortcoming was that the accuracy of their feedforward baseline was far from optimal. Using their code, with relatively minor tweaks to the learning rate schedule, we were able to bring it up from 60% to 70% – just a few percentage points below their recurrent network. We expect that this could be further improved with a more extensive and systematic hyperparameter search. In other words, their training hyperparameters appeared to have been optimised for their predictive coding network, but not – or not as much – for their feedforward baseline. We further found that a minor change to the architecture - using group normalisation layers after each ReLU – leads to a feedforward network which performs on par with the recurrent network, with a mean over 6 runs of 72% and best of 73%. Adding the same layers to the




**(a)** Reconstruction error over timesteps

**(b)** Accuracy over timesteps on CIFAR100 testset

**Figure A.1: PCN**: Panel (a) shows the reconstruction errors of the model over timesteps. It does not decrease over timesteps, as would be expected in a predictive coding system. Panel (b) depicts the accuracy of the model on the CIFAR100 test dataset. The model performs at chance level at early timesteps and then becomes better in the last few timesteps.

recurrent network did not lead to a corresponding improvement in accuracy.

We also found that the network had poor accuracy (underperforming the optimized feedforward baseline) until the final timestep, as can been seen in Figure A.1b. This can be clarified by a closer reading of Figure 3 of their paper: the reported improvements over cycles from 60% at timestep 0 to more than 70% at timestep 6 are for seven distinct networks, each evaluated only at the timestep they were trained for. So in fact, in their model the predictive coding updates do not gradually improve on an already reasonably guess. This is clearly not biologically plausible: visual processing would be virtually useless if the correct interpretation of a scene only crystallised after a number of "timesteps". By the time a person has identified an object that object is likely to have disappeared or, in a worst case scenario, eaten them. We also experimented with feeding the classification error at each timestep into an aggregate loss function, but this lead to a network which, while performing well, essentially did not improve over timesteps.

Figure A.1a shows that the network does not uniformly minimise reconstruction errors over time for all layers, and thus is not performing correct predictive coding updates. In fact the total reconstruction error (across all layers) increases exponentially over timesteps. There are a number of possible explanations for this. Firstly, in the case of the network with untied weights, the authors choose to make a strong assumption in the update equations (seen as the equivalence of their Equations 5 and 6): that the feedback weights can be assumed to be the transpose of the feedforward weights, i.e. $W^b = (W^f)^T$. They thus propagate the feedforward error through the feedforward weights. However, it might be that the network learns feedback weights which essentially invert the feedforward transformation as assumed, but this is not guaranteed, and nor is it explicitly motivated through the classification loss function. Indeed, because the network is not motivated to learn a representation at earlier timesteps which produces a good prediction, it does not necessarily need to learn the inverse transformation: it can instead learn some other transformation which, when applied with the update equations, leads the network to *end up* in the right place. That being said, this assumption is valid for the network with tied weights, and this network also does not uniformly reduce reconstruction error over timesteps. Possibly, the presence of ReLU non-linearities means that the forward convolution may still not be perfectly invertible by a transposed convolution. Finally, in line with this unexpected increase of reconstruction errors over time, we have also failed to extract good image reconstructions from the network as seen in Figure 5 of their paper, although in private communication the authors indicated that this was possible

with some other form of normalisation.

In short, while the ideas put forward in[235] share similarities with our own, their exact implementation did not support the claims of the authors, and the question of whether predictive coding can benefit deep neural networks remained an open one. We hope that our approach detailed in the present study can help resolve this question.

## A.6 Comparing with Rao and Ballard

This section aims to start from the equations initially provided in Rao and Ballard[183] and compare them to ours. The parallels drawn will help to highlight the similarities and the differences between both the approaches.

Rao and Ballard consider a two-layer system, and start with the assumption that the brain possesses a set of internal causes, denoted as $\mathbf{r}$ (in matrix notation), that it uses to predict the visual stimulus, for example an input image $\mathbf{I}$, such that

$$\mathbf{I} \approx f(U\mathbf{r}) \tag{A.9}$$

where $f(.)$ is some nonlinear activation function. This $\mathbf{r}$ can be equalled to encoding layer $e_1$ in our equations, with $\mathbf{I}$ being the input image $e_0$ or its reconstruction $d_0$. $U$ here, represents the top-down weight matrix (equivalent to $W^b_{1,0}$) that helps to make a prediction about the input image. That is,

$$\mathbf{I} \approx f(U\mathbf{r}) \equiv e_0 \approx d_0 = W^b_{1,0} e_1 \tag{A.10}$$

In this two-layer hierarchical architecture, $\mathbf{r}$ itself is predicted by the higher layer $\mathbf{r}^b$ using the weight matrix $U^b$, equivalent to how $e_1$ is predicted by $e_2$ using $W^b_{2,1}$ in our model. This prediction denoted as $\mathbf{r}^{td}$ in Rao and Ballard's original implementation can be equalled to $d_1$ in our equations.

$$\mathbf{r}^{td} = f(U^b \mathbf{r}^b) \equiv d_1 = W^b_{2,1} e_2 \tag{A.11}$$

The errors made in making the predictions are defined, like ours, as the mean squared distance,

$$\varepsilon_0 = (\mathbf{I} - f(U\mathbf{r}))^T(\mathbf{I} - f(U\mathbf{r})) \tag{A.12}$$

$$\varepsilon_1 = (\mathbf{r} - \mathbf{r}^{td})^T(\mathbf{r} - \mathbf{r}^{td}) \tag{A.13}$$

Please note that differentiating the prediction error $\varepsilon_0$ with respect to $\mathbf{r}$ (similar to taking the gradient of $\varepsilon_{n-1}$ with respect to $e_n$ as done in our error-correction term) gives us,

$$\nabla_0 = -2U^T \frac{\partial f}{\partial U\mathbf{r}}^T (\mathbf{I} - fU\mathbf{r}) \tag{A.14}$$

$$= -kU^T(\mathbf{I} - f(U\mathbf{r})) \tag{A.15}$$

which will be useful later.

As per the predictive coding theory, the brain tries both to learn parameters ($U$ and $U^h$) over a dataset of natural inputs, and tries to modify its neural activations ($\mathbf{r}$ and $\mathbf{r}^h$) over time given a particular input, in such a way as to minimize the total error $E$, defined as:

$$E = a \cdot \underbrace{(\mathbf{I} - f(U\mathbf{r}))^T(\mathbf{I} - f(U\mathbf{r}))}_{\varepsilon_0} + b \cdot \underbrace{(\mathbf{r} - \mathbf{r}^{td})^T(\mathbf{r} - \mathbf{r}^{td})}_{\varepsilon_1} \tag{A.16}$$

Here $a$ and $b$ act as constants that weigh the errors in this two-level hierarchichal network. Equation A.16 is reflected as Equation 4 on Page 86 of the original paper[183]. The original implementation also contains terms that account for the prior probability distributions of $\mathbf{r}$ and $U$; these terms can be equated to regularization terms, and thus we omit them for the sake of simplicity.

Equation A.16 represents the overall error, calculated as sum of the mean squared errors across the hierarchy of the network. It should be noted that we use this same objective function ($-E$) to train the feedback weights of our networks.

As stated above, the predictive coding dynamics aim to modify neural representations $\mathbf{r}$ so as to minimize the error $E$, i.e., differentiating the above equation:

$$\frac{d\mathbf{r}}{dt} = -\frac{\partial E}{\partial \mathbf{r}} = a \cdot U^T \frac{\partial f^T}{\partial U\mathbf{r}}(\mathbf{I} - f(U\mathbf{r})) + b \cdot (\mathbf{r}^{td} - \mathbf{r}) \tag{A.17}$$

Barring a regularization term, the above equation is equivalent to Equation 7 on page 86 of[183]. One can see that the first term in the RHS of equation A.17 can be substituted with our error-correction term $\nabla\varepsilon_0$ (see Eq. A.15). Hence, Equation A.17 after simultaneously expanding the LHS becomes,

$$\frac{\mathbf{r}(t+dt) - \mathbf{r}(t)}{dt} = -a_1 \cdot \nabla\varepsilon_r(t) + b \cdot (\mathbf{r}^{td}(t) - \mathbf{r}(t)) \tag{A.18}$$

We use subscript $r$ for $\varepsilon$ to emphasize that this error can be calculated at any level/stage $\mathbf{r}$ represents in a multi-layer hierarchical system, and is not restricted to just the first layer of the hierarchy. Similarly, the time resolution $dt$ can be equated to 1 timestep (of arbitrary duration) for simulations. Hence, rearranging

the equation further,

$$\mathbf{r}(t+1) = \underbrace{b \cdot \mathbf{r}^{td}(t)}_{feedback} + \underbrace{(1-b)\mathbf{r}(t)}_{memory} - \underbrace{a_1 \nabla \varepsilon_r(t)}_{error-correction} \tag{A.19}$$

In the above equation, the first term corresponds to our feedback term, the second term corresponds to our memory term and the last term corresponds to our feedforward error-correction term. That is, exchanging constants to match our notation:

$$\mathbf{r}(t+1) = \underbrace{\boxed{\phantom{xxxxxxxxx}}}_{feedforward} + \underbrace{\lambda \cdot \mathbf{r}^{td}(t)}_{feedback} + \underbrace{(1-\lambda)\mathbf{r}(t)}_{memory} - \underbrace{a_1 \nabla \varepsilon_r(t)}_{error-correction} \tag{A.20}$$

This can be directly compared to our main Equation 2.2.

Equation A.20 also highlights the fact that our approach has an extra feedforward term that is not present in the original Rao and Ballard proposal. We believe that such a modification allows for rethinking the role of error-correction in network dynamics; where error-correction constituted the predominant mode of feed-forward communication in the Rao and Ballard implementation, it plays a more supporting role in our implementation, iteratively correcting the errors made by the feedforward convolutional layers. We empirically found that the feedforward term helped to improve the stability of the training. Interestingly, a common criticism of predictive coding lies in its inability to explain the dominance of feedforward brain activity compared to prediction error signals[90,2]. We believe that our proposed implementation allows for a flexible modulation of these two

terms, and thus systematic investigation of these factors–as done in [4].

From a practical perspective, we expect that our framework can be readily used by both proponents and opponents of the predictive coding theory. Setting the feedforward term $\beta$ equal to zero produces a pure predictive coding network as proposed in Rao and Ballard[183]. Alternatively, one can set the error-correction term $\alpha$ equal to zero to study a bidirectional network with feedback and feedforward drives, in the style of Heeger[89]. The framework has been implemented such that the basic update rule (as class `Pcoder` in the package) is easily adaptable, allowing one to try other complex interactions between these terms; for example, one could easily include multiplicative interactions between feedback and feedforward terms to emulate forms of biased competition (see[208,207]).

## A.7 Tuning hyperparameters

In addition to the fixed set of hyperparameters used in our initial experiments (Figures 2.2, 2.3a and 2.4), we also experimented with optimizing our hyperparameters. To tune the hyperparameters for the models, we applied two different strategies for both the models–tuning hyperparameters for the whole network vs tuning hyperparameters for each pcoder separately. After a few initial explorations on clean images, we discovered that the hyperparameters dictate where the network dynamics converge, and consequently its performance for noisy situations. This effect is characterized and investigated thoroughly in[4]. Thus, in this study, we decide to use gaussian noise of standard deviation 0.5 to tune the hyperparameters and test it on all other types of noises from the ImageNet-C dataset.

For PVGG16, we start by fixing the value of alpha for each layer to zero and only search for $\beta_n$'s and $\lambda_n$'s. We calculate the average cross-entropy loss for 4 timesteps on 2000 images and use it as a metric for choosing the hyperparameters. The hyperparameters chosen are as follows :

**Table A.2:** Values of the Hyperparameters

| $n$ | $\beta_n$ | $\lambda_n$ | $\alpha_n$ |
|---|---|---|---|
| 1 | 0.2 | 0.05 | 0.01 |
| 2 | 0.4 | 0.10 | 0.01 |
| 3 | 0.4 | 0.10 | 0.01 |
| 4 | 0.5 | 0.10 | 0.01 |
| 5 | 0.6 | 0.00 | 0.01 |

For PEfficientNetB0, we take a different approach. Instead of the whole network, we start by finetuning each pcoder using the same metric (average crossentropy for 4 timesteps) on 4050 images. We then combine all hyperparameters found for each pcoder. The hyperparameters chosen are as follows :

**Table A.3:** Values of the Hyperparameters

| $n$ | $\beta_n$ | $\lambda_n$ | $\alpha_n$ |
|---|---|---|---|
| 1 | 0.77 | 0.08 | 0.01 |
| 2 | 0.76 | 0.11 | 0.01 |
| 3 | 0.83 | 0.03 | 0.01 |
| 4 | 0.94 | 0.01 | 0.01 |
| 5 | 0.73 | 0.25 | 0.01 |
| 6 | 0.81 | 0.01 | 0.01 |
| 7 | 0.85 | 0.10 | 0.01 |
| 8 | 1.0 | 0.00 | 0.01 |

We then, calculate the mCE scores using all the 19 noises for both the networks. The CE scores for each noise are shown below :

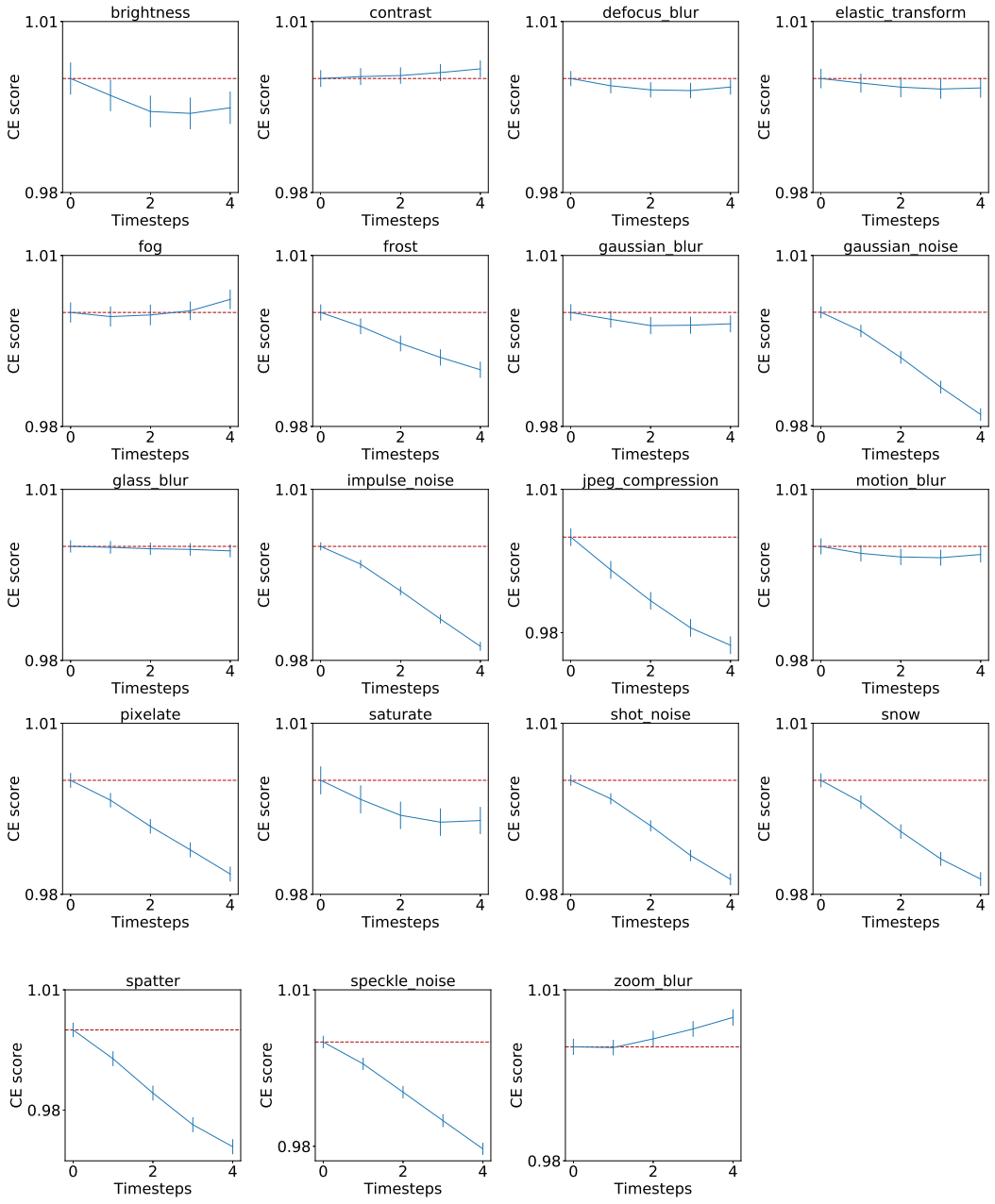



**Figure A.2: PVGG16 (optimised) Corruption Error (CE) scores for all distortions:** The panel shows the CE scores calculated on the distorted images provided in the ImageNet-C dataset. The values are normalized with the CE score obtained for the feedforward VGG. The error bars denote the standard deviation of the means obtained from bootstrapping (resampling multiple binary populations across all severities.)

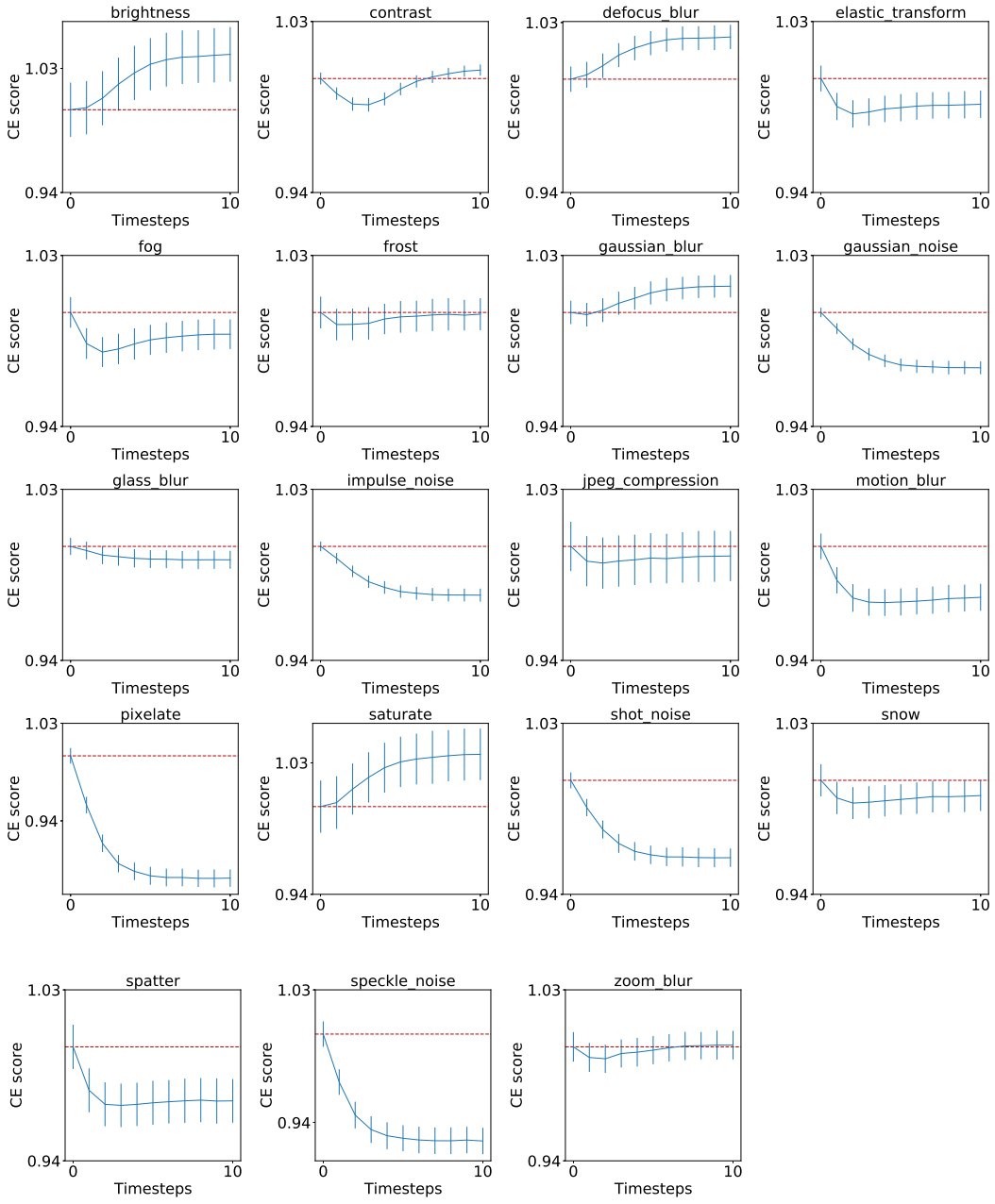



**Figure A.3: PEfficientNetB0 (optimised) Corruption Error (CE) scores for all distortions:** The panel shows the CE scores calculated on the distorted images provided in the ImageNet-C dataset. The values are normalized with the CE score obtained for the feedforward EfficientNetB0. The error bars denote the standard deviation of the means obtained from bootstrapping (resampling multiple binary populations across all severities.)

## A.8 mCE scores of the optimized networks using AlexNet as a baseline

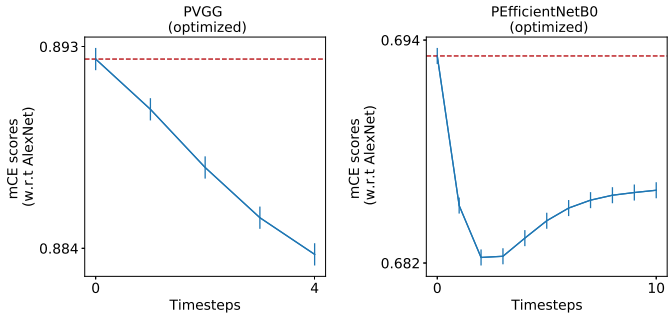



**Figure A.4:** The mCE scores of the optimized networks (as shown in Figure 3) normalized using the score of the AlexNet network. Instead of normalizing using the score for the feedforward version of our recurrent network, to facilitate comparison with other works, we here normalize the scores using the score obtained for AlexNet network.

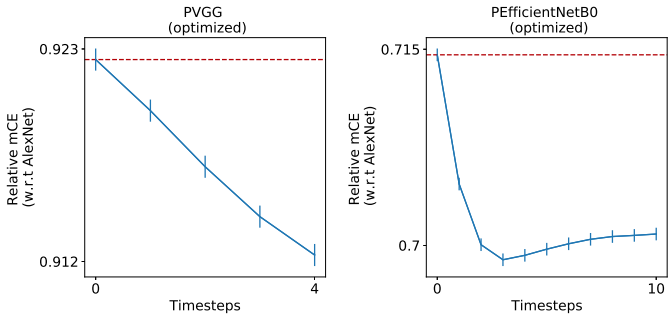



**Figure A.5:** The Relative mCE scores of the optimized networks (as shown in Figure 3) normalized using the score of the AlexNet network. As suggested by[92], we use Relative mCE score which accounts for the changing baseline accuracy on the clean images over timesteps.

## A.9 mCE scores of a predified robust network

We also incorporated our recurrent dynamics in an already robust PEfficientNet network. As a simple approach, we just used the hyperparameters ($\alpha$, $\beta$ and

$\lambda$) that were optimized for the non-robust version of PEfficientNEtB0 (on 0.25 gaussian noise) and measured its robustness against the corruptions in ImageNet-C dataset. We observed that the proposed predictive coding dynamics further helped in improving the robustness of this already robust network.

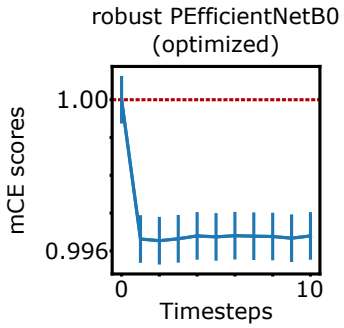



**Figure A.6:** mCE scores of a predified version of an already robust PEfficientNetB0

## A.10 Original data for Adversarial Attacks

We provide here the non-baseline corrected versions of the data presented for adversarial attacks in Figure 4. The panels below show the success rate of the targeted attacks across timesteps calculated on 1000 images. The perturbations allowed ($\varepsilon$) and the type of attack are denoted at the top.

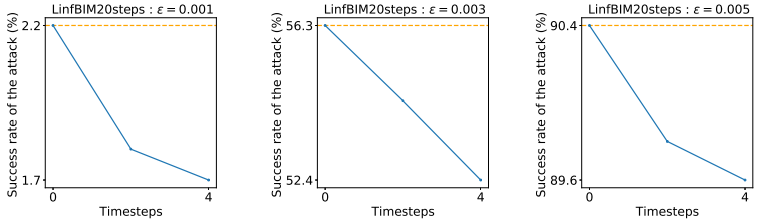



**Figure A.7:** $L_\infty$BIM attacks on PVGG16 network

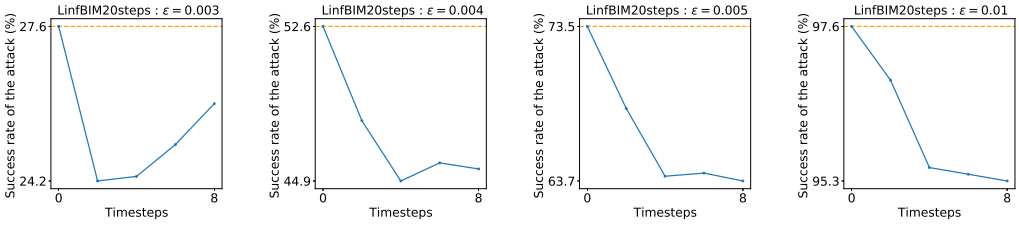



**Figure A.8:** $L_\infty$BIM attacks on PEfficientNetB0 network

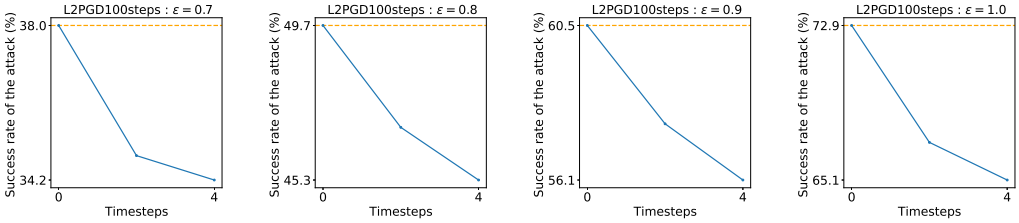



**Figure A.9:** $L_2$RPGD attacks on PEfficientNetB0 network

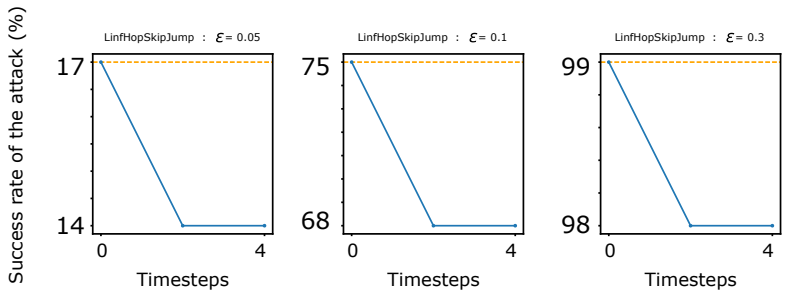



**Figure A.10:** $L_\infty$ HopSkipJump attacks on PEfficientNetB0

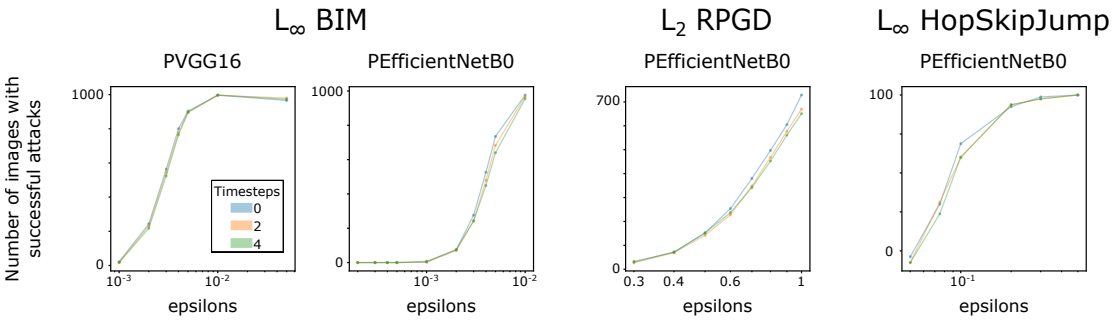



**Figure A.11:** Adversarial Attacks with respect to epsilons. Here we show the number of successful attacks on 1000 (100 for HopSkipJump) images. Increasing the size of the epsilon leads to increase in the success rate of the attack as expected. As predictive coding timesteps increase, the curves shift slightly to the right, meaning that a slightly larger perturbation is required to fool the network. This robustness is more easily seen on Figure 2.4, where $\varepsilon$ values are sampled near each curve's inflection point.

## A.11 Absolute values of the plots shown in the main text

| Noise Level | PVGG16 | | PEfficientNetB0 | |
|---|---|---|---|---|
| | Accuracy at t=0 | Accuracy at t=15 | Accuracy at t=0 | Accuracy at t=15 |
| $\sigma = 0.00$ | 71.63 | 71.47 | 77.29 | 75.35 |
| $\sigma = 0.50$ | 35.61 | 38.59 | 57.66 | 56.24 |
| $\sigma = 0.75$ | 16.69 | 18.46 | 37.11 | 41.05 |
| $\sigma = 1.00$ | 5.59 | 7.05 | 17.03 | 23.59 |

**Table A.4:** Accuracy on gaussian noise-corrupted images. Here we show the accuracy obtained on images corrupted using gaussian noise (at t=0) as shown in figure 2a. All the values are calculated on the corrupted versions of the ImageNet validation dataset.

| Noise Level | PVGG16 | | PEfficientNetB0 | |
|---|---|---|---|---|
| | MSE at t=0 | MSE at t=15 | MSE at t=0 | MSE at t=15 |
| $\sigma = 0.00$ | 0.224 | 0.220 | 0.186 | 0.184 |
| $\sigma = 0.25$ | 0.342 | 0.324 | 0.223 | 0.222 |
| $\sigma = 0.50$ | 0.518 | 0.485 | 0.303 | 0.302 |
| $\sigma = 0.75$ | 0.705 | 0.660 | 0.394 | 0.392 |
| $\sigma = 1.00$ | 0.898 | 0.842 | 0.486 | 0.482 |
| $\sigma = 2.00$ | 1.689 | 1.587 | 0.848 | 0.834 |

**Table A.5:** MSE distances for reconstructions on noisy images. Here we show the MSE distances obtained between the noisy images corrupted using gaussian noises and the reconstructions made by the models as shown in Figure 2b.

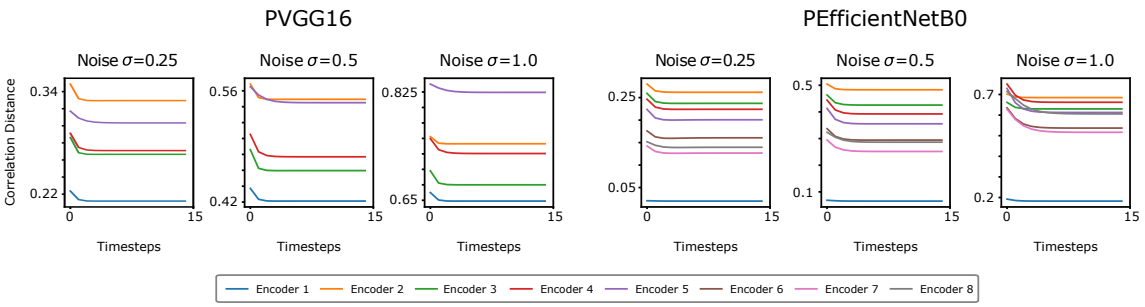



**Figure A.12: Correlation distances for representations obtained on noisy images:** Here we show the absolute correlation distances obtained between clean and noisy representations as shown in Figure 2d in the main text.

# Appendix B

# Appendix for Chapter 3

## B.1 The complete schematic

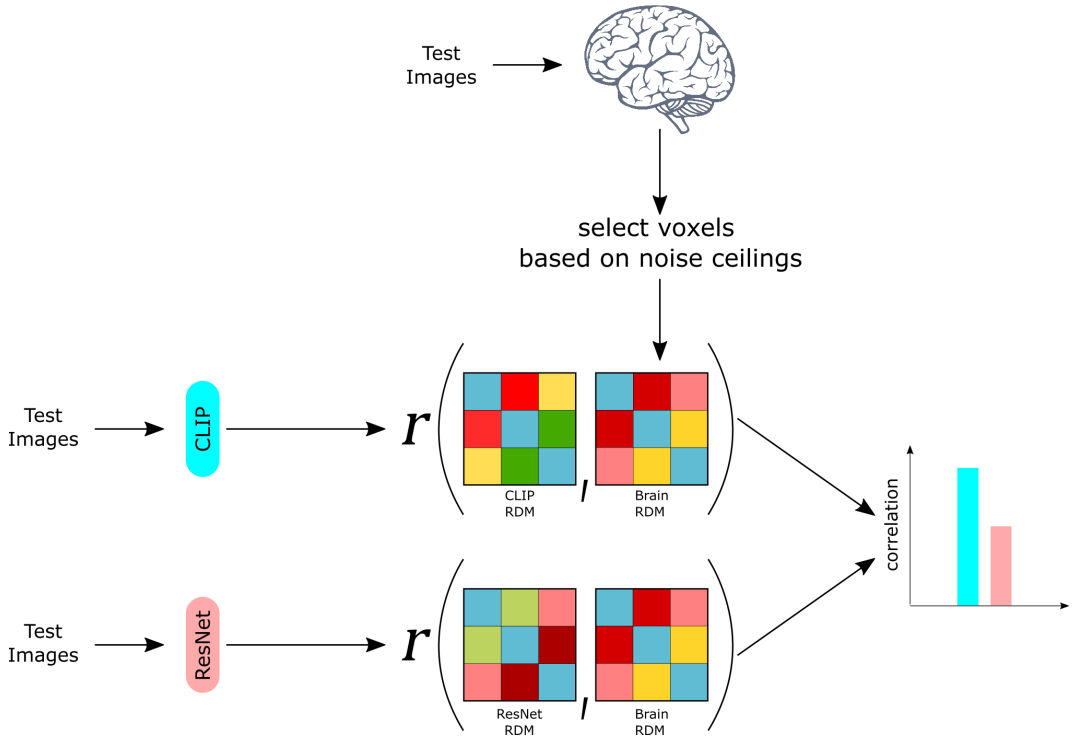



**Figure B.1:** This schematic shows the methodology used for the analysis. In a first step, the fMRI data corresponding to the test images shown to the participants are processed. Next, from all the voxels in a selected region of interest, a subset of voxels with the highest reliability in the fMRI signal (as determined by an independent analysis on the noise ceiling) is chosen. Using pairwise distances on these brain features, an RDM (representational dissimilarity matrix) is constructed. In parallel, the same test images are passed through different models, and their features are obtained. These features are used to construct RDMs for each model. Finally, the similarity between each model RDM and the brain RDM is computed using different correlation measures.

## B.2 Additional details on the models used in the analysis

| | Model | Objective | Dataset |
|---|---|---|---|
| **Mutlimodal** | CLIP | constrastive loss | approx. 400M images from Internet |
| | Virtex | bicaptioning | MS-COCO |
| | ICMLMs (-attfc and -tfm) | masked captioning | MS-COCO |
| | TSMResNet | trimodal contrastive objective | HowTo100M |
| **Visual** | ResNet | crossentropy | ImageNet |
| | BiT-M | crossentropy | ImageNet21K |
| | AR models | adversarial robustness | ImageNet + adversarial images |
| | SIN models | crossentropy on shape-biased images | ImageNet + StylizedImageNet |
| **Language** | GPT2 | unsupervised langugae modeling | WebText |
| | BERT | bidirectional masked language modeling | BookCorpus and English Wikipedia |
| | CLIP-L | constrastive loss | approx. 400 M images |

**Table B.1:** Further details of the networks used in this work.

## B.3 Licenses of the assets used

| Asset | License |
|---|---|
| FreeSurfer | https://surfer.nmr.mgh.harvard.edu/fswiki/FreeSurferWiki |
| SPM12 | GNU GPL |
| fMRI data | CC0 |
| CLIP | MIT |
| VirTex | MIT |
| TSM | Apache-2.0 |
| ICMLM | N/A |
| BiT-M | Apache-2.0 |
| ResNet | MIT |
| AR models | MIT |
| SIN models | https://github.com/rgeirhos/texture-vs-shape/blob/master/DATASET_LICENSE |
| GPT2 | MIT |
| BERT | Apache-2.0 |

**Table B.2:** Available Licences of the assets used in the study. Links to the appropriate webpages are provided for special licenses.

## B.4 Raw data shown in Figure 2

| Model | **visual region** | **fusiform** | **hippocampus** | **parahippocampus** |
|---|---|---|---|---|
| CLIP | 0.065 ± 0.049 | 0.172 ± 0.047 | 0.139 ± 0.024 | 0.091 ± 0.040 |
| VirTex | 0.086 ± 0.051 | 0.173 ± 0.048 | 0.129 ± 0.023 | 0.177 ± 0.030 |
| ICMLM-attfc | 0.108 ± 0.044 | 0.205 ± 0.035 | 0.125 ± 0.030 | 0.140 ± 0.030 |
| ICMLM-tfm | 0.117 ± 0.046 | 0.237 ± 0.034 | 0.140 ± 0.053 | 0.176 ± 0.046 |
| TSMResNet50-visual | 0.154 ± 0.044 | 0.211 ± 0.037 | 0.125 ± 0.042 | 0.160 ± 0.050 |
| BiT-M | 0.004 ± 0.018 | 0.109 ± 0.030 | 0.116 ± 0.021 | 0.076 ± 0.004 |
| ResNet50 | 0.068 ± 0.031 | 0.156 ± 0.039 | 0.078 ± 0.036 | 0.121 ± 0.022 |
| AR-Linf8 | 0.122 ± 0.032 | 0.173 ± 0.033 | 0.072 ± 0.031 | 0.102 ± 0.018 |
| AR-Linf4 | 0.106 ± 0.030 | 0.175 ± 0.033 | 0.084 ± 0.032 | 0.118 ± 0.021 |
| AR-L2 | 0.128 ± 0.033 | 0.190 ± 0.036 | 0.088 ± 0.035 | 0.123 ± 0.020 |
| SIN | 0.086 ± 0.032 | 0.147 ± 0.033 | 0.085 ± 0.017 | 0.072 ± 0.016 |
| SIN-IN | 0.070 ± 0.021 | 0.160 ± 0.037 | 0.079 ± 0.031 | 0.119 ± 0.021 |
| SIN-IN+FIN | 0.066 ± 0.016 | 0.135 ± 0.033 | 0.075 ± 0.025 | 0.080 ± 0.014 |
| BERT | 0.000 ± 0.006 | 0.074 ± 0.018 | 0.053 ± 0.026 | -0.009 ± 0.023 |
| GPT2 | 0.106 ± 0.022 | 0.094 ± 0.034 | 0.043 ± 0.011 | 0.123 ± 0.013 |
| CLIP-L | 0.006 ± 0.023 | 0.068 ± 0.075 | 0.069 ± 0.052 | 0.055 ± 0.016 |

**Table B.3:** Raw data shown in Figure 2

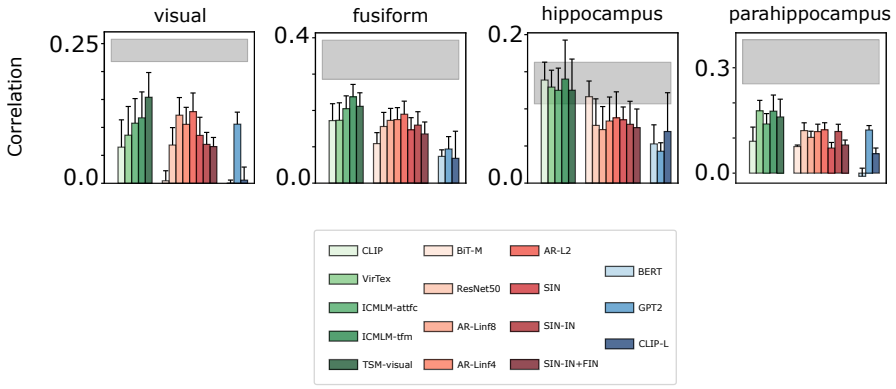



**Figure B.2: Non-normalized RSA values between model and brain RDMs.** The brain RDMs are calculated based on selecting 30 voxels from each ROI, as in the main analysis. The gray bands show the upper and lower bounds of the noise-ceilings calculated by adding and subtracting the s.e.m. values respectively.

## B.5 Voxel-selection based on a fixed beta-value threshold.

In the main analysis, we selected 30 voxels in each ROI based on the noise-ceiling analysis in the hippocampus. In other words, in each ROI we selected the 30 voxels with the highest beta values.

As a control method, instead of restricting the number of voxels to 30, we used the value of the 30th voxel from hippocampus as a threshold for other ROIs. The number of voxels found in each ROI for each participant is depicted in Table B.4 and the RSA values in Figure B.3. We observed that this alternate criterion did not affect the overall trend in our results.

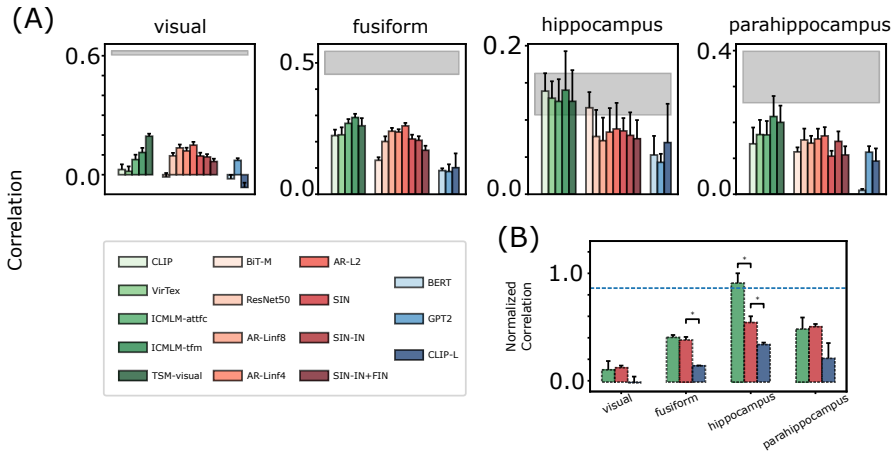



**Figure B.3:** Non-normalized RSA values after using the beta value of the 30th voxel from hippocampus as a threshold for other ROIs for each participant.

| | Subject 1 | Subject 2 | Subject 3 | Subject 4 | Subject 5 |
|---|---|---|---|---|---|
| visual | 530 | 831 | 343 | 707 | 592 |
| fusiform | 532 | 376 | 217 | 368 | 508 |
| hippocampus | 30 | 30 | 30 | 30 | 30 |
| parahippocampus | 122 | 67 | 85 | 111 | 167 |

**Table B.4:** Number of voxels found in each region after thresholding

| Model | **visual region** | **fusiform** | **hippocampus** | **parahippocampus** |
|---|---|---|---|---|
| CLIP | 0.026 ± 0.026 | 0.223 ± 0.022 | 0.139 ± 0.024 | 0.140 ± 0.045 |
| VirTex | 0.017 ± 0.025 | 0.226 ± 0.028 | 0.129 ± 0.023 | 0.166 ± 0.041 |
| ICMLM-attfc | 0.076 ± 0.025 | 0.270 ± 0.016 | 0.125 ± 0.030 | 0.165 ± 0.039 |
| ICMLM-tfm | 0.112 ± 0.023 | 0.292 ± 0.014 | 0.140 ± 0.053 | 0.216 ± 0.057 |
| TSMResNet50-visual | 0.194 ± 0.013 | 0.261 ± 0.029 | 0.125 ± 0.042 | 0.200 ± 0.047 |
| BiT-M | -0.011 ± 0.020 | 0.129 ± 0.012 | 0.116 ± 0.021 | 0.118 ± 0.013 |
| ResNet AvgPool | 0.095 ± 0.015 | 0.201 ± 0.020 | 0.078 ± 0.036 | 0.151 ± 0.032 |
| AR-Linf8 | 0.135 ± 0.017 | 0.240 ± 0.013 | 0.072 ± 0.031 | 0.142 ± 0.020 |
| AR-Linf4 | 0.120 ± 0.016 | 0.237 ± 0.011 | 0.084 ± 0.032 | 0.154 ± 0.029 |
| AR-L2 | 0.149 ± 0.017 | 0.260 ± 0.011 | 0.088 ± 0.035 | 0.162 ± 0.024 |
| SIN | 0.095 ± 0.016 | 0.210 ± 0.016 | 0.085 ± 0.017 | 0.106 ± 0.015 |
| SIN-IN | 0.090 ± 0.014 | 0.205 ± 0.016 | 0.079 ± 0.031 | 0.147 ± 0.028 |
| SIN-IN+FIN | 0.066 ± 0.014 | 0.167 ± 0.017 | 0.075 ± 0.025 | 0.109 ± 0.024 |
| BERT | -0.020 ± 0.020 | 0.090 ± 0.009 | 0.053 ± 0.026 | 0.011 ± 0.004 |
| GPT2 | 0.072 ± 0.011 | 0.086 ± 0.027 | 0.043 ± 0.011 | 0.117 ± 0.017 |
| CLIP-L | -0.064 ± 0.024 | 0.101 ± 0.055 | 0.069 ± 0.052 | 0.092 ± 0.035 |

**Table B.5:** Exact values corresponding to the data shown in Figure B.3

## B.6 RSA computed using different metrics

We verified the robustness of our results by using other metrics to compute the RDMs and RSA (Figures B.4–B.6).

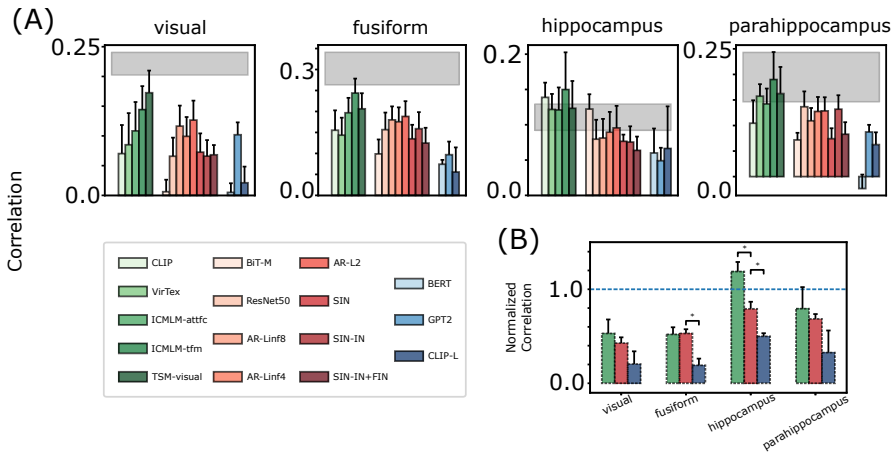



**Figure B.4:** The RDMs were calculated using the Pearson correlation distance, and the Spearman rank correlation was used to compute the RSA.

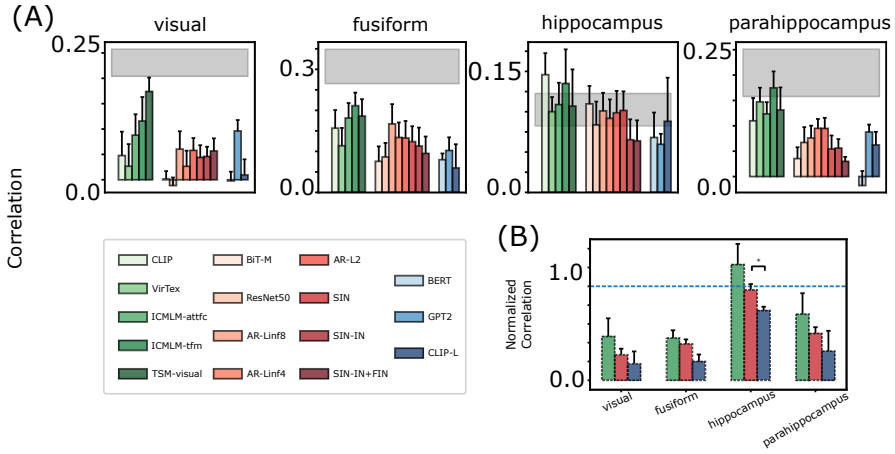



**Figure B.5:** The RDMs were calculated using the Cosine distance, and the Spearman rank correlation was used to compute the RSA.

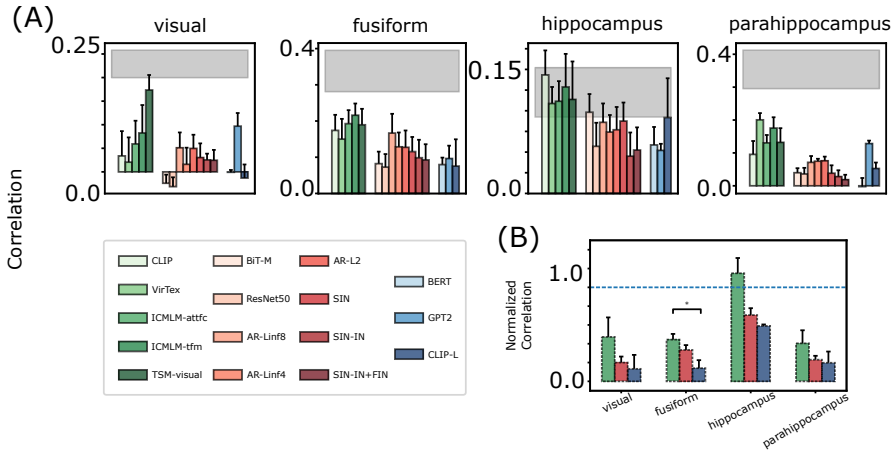



**Figure B.6:** The RDMs were calculated using the Cosine distance, and the Pearson correlation was used to compute the RSA.

## B.7 Broader Impacts

The research discussed above analyzes the ability of neural networks to explain human brain activity. Specifically, it demonstrates that multimodal neural networks are better than visual or linguistic models in explaining the activity in the hippocampus during visual tasks.

Importantly, this research provides potential insights for designing better bioplausible networks, which could elucidate underlying mechanisms in biological brains. At the same time, we are aware of the possibilities for the nefarious use of such systems, and urge all researchers to consider their implications.

# References


[1] Ahmad, N., van Gerven, M. A., & Ambrogioni, L. (2020). Gait-prop: A biologically plausible learning rule derived from backpropagation of error. *Advances in Neural Information Processing Systems*, 33, 10913–10923.

[2] Aitchison, L. & Lengyel, M. (2017). With or without you: predictive coding and bayesian inference in the brain. *Current opinion in neurobiology*, 46, 219–227.

[3] Al-Tahan, H. & Mohsenzadeh, Y. (2021). Reconstructing feedback representations in the ventral visual pathway with a generative adversarial autoencoder. *PLoS Computational Biology*, 17(3), e1008775.

[4] Alamia, A., Mozafari, M., Choksi, B., & VanRullen, R. (2021). On the role of feedback in visual processing: a predictive coding perspective. *arXiv preprint arXiv:2106.04225*.

[5] Alamia, A. & VanRullen, R. (2019). Alpha oscillations and traveling waves: Signatures of predictive coding? *PLoS Biology*, 17(10), e3000487.

[6] Alayrac, J.-B., Recasens, A., Schneider, R., Arandjelovic, R., Ramapuram, J., De Fauw, J., Smaira, L., Dieleman, S., & Zisserman, A. (2020). Self-supervised multimodal versatile networks. *NeurIPS*, 2(6), 7.

[7] Allen, E. J., St-Yves, G., Wu, Y., Breedlove, J. L., Prince, J. S., Dowdle, L. T., Nau, M., Caron, B., Pestilli, F., Charest, I., et al. (2022). A massive 7t fmri dataset to bridge cognitive neuroscience and artificial intelligence. *Nature neuroscience*, 25(1), 116–126.

[8] Aristotle, Peck, A. L., & Forster, E. S. (1956). *Parts of animals*.

[9] Baars, B. J. (1993). *A cognitive theory of consciousness*. Cambridge University Press.

[10] Baars, B. J. (2005). Global workspace theory of consciousness: toward a cognitive neuroscience of human experience. *Progress in brain research*, 150, 45–53.

[11] Barlow, H. (1995). *The neuron doctrine in perception.* The MIT Press.

[12] Barlow, H. (2009). Grandmother cells, symmetry, and invariance: how the term arose and what the facts suggest. *The cognitive neurosciences*, (pp. 309–320).

[13] Barwich, A.-S. (2019). The value of failure in science: The story of grandmother cells in neuroscience. *Frontiers in neuroscience*, 13, 1121.

[14] Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., & Friston, K. J. (2012). Canonical microcircuits for predictive coding. *Neuron*, 76(4), 695–711.

[15] Beliy, R., Gaziv, G., Hoogi, A., Strappini, F., Golan, T., & Irani, M. (2019). From voxels to pixels and back: Self-supervision in natural-image reconstruction from fmri. *Advances in Neural Information Processing Systems*, 32.

[16] Bengio, Y. & Fischer, A. (2015). Early inference in energy-based models approximates back-propagation. *arXiv preprint arXiv:1510.02777*.

[17] Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. (2019). Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.

[18] Bielawski, R., Devillers, B., Van de Cruys, T., & Vanrullen, R. (2022). When does clip generalize better than unimodal models? when judging human-centric concepts. In *Proceedings of the 7th Workshop on Representation Learning for NLP* (pp. 29–38).

[19] Bienenstock, E. L., Cooper, L. N., & Munro, P. W. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2(1), 32–48.

[20] Bliss, T. V. & Lømo, T. (1973). Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of physiology*, 232(2), 331–356.

[21] Boutin, V., Franciosini, A., Chavane, F., Ruffier, F., & Perrinet, L. (2019). Sparse deep predictive coding captures contour integration capabilities of the early visual system. *arXiv preprint arXiv:1902.07651*.

[22] Boutin, V., Zerroug, A., Jung, M., & Serre, T. (2020). Iterative vae as a predictive brain model for out-of-distribution generalization. *arXiv preprint arXiv:2012.00557*.

[23] Brucklacher, M., Bohte, S. M., Mejias, J. F., & Pennartz, C. M. (2022). Local minimization of prediction errors drives learning of invariant object representations in a generative network model of visual perception. *bioRxiv*.

[24] Büchel, J., Zendrikov, D., Solinas, S., Indiveri, G., & Muir, D. R. (2021). Supervised training of spiking neural networks for robust deployment on mixed-signal neuromorphic processors. *Scientific reports*, 11(1), 1–12.

[25] Cadena, S. A., Denfield, G. H., Walker, E. Y., Gatys, L. A., Tolias, A. S., Bethge, M., & Ecker, A. S. (2019a). Deep convolutional models improve predictions of macaque v1 responses to natural images. *PLoS computational biology*, 15(4), e1006897.

[26] Cadena, S. A., Sinz, F. H., Muhammad, T., Froudarakis, E., Cobos, E., Walker, E. Y., Reimer, J., Bethge, M., Tolias, A., & Ecker, A. S. (2019b). How well do deep neural networks trained on object recognition characterize the mouse visual system?

[27] Cadieu, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., Majaj, N. J., & 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.

[28] Carpenter, G. A. & Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer vision, graphics, and image processing*, 37(1), 54–115.

[29] Carrasco, M. (2011). Visual attention: The past 25 years. *Vision research*, 51(13), 1484–1525.

[30] Casanova, A., Careil, M., Verbeek, J., Drozdzal, M., & Romero Soriano, A. (2021). Instance-conditioned gan. *Advances in Neural Information Processing Systems*, 34, 27517–27529.

[31] Caucheteux, C. & King, J.-R. (2022). Brains and algorithms partially converge in natural language processing. *Communications biology*, 5(1), 1–10.

[32] Chalasani, R. & Principe, J. C. (2013). Deep predictive coding networks. *arXiv preprint arXiv:1301.3541*.

[33] Chalk, M., Marre, O., & Tkačik, G. (2018). Toward a unified theory of efficient, predictive, and sparse coding. *Proceedings of the National Academy of Sciences*, 115(1), 186–191.

[34] Chen, J., Jordan, M. I., & Wainwright, M. J. (2020). Hopskipjumpattack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy (SP)* (pp. 1277–1294).: IEEE Computer Society.

[35] Chen, Y., Zhang, H., & Sejnowski, T. J. (2022). Hippocampus as a generative circuit for predictive coding of future sequences. *bioRxiv*.

[36] Choksi, B., Mozafari, M., O'May, C. B., Ador, B., Alamia, A., & VanRullen, R. (2021a). Predify: Augmenting deep neural networks with brain-inspired predictive coding dynamics. *Neural Information Processing Systems (NeurIPS)*.

[37] Choksi, B., Mozafari, M., Vanrullen, R., & Reddy, L. (2021b). Multimodal neural networks better explain multivoxel patterns in the hippocampus. *arXiv preprint arXiv:2201.11517*.

[38] Cichy, R. M., Dwivedi, K., Lahner, B., Lascelles, A., Iamshchinina, P., Graumann, M., Andonian, A., Murty, N., Kay, K., Roig, G., et al. (2021). The algonauts project 2021 challenge: How the human brain makes sense of a world in motion. *arXiv preprint arXiv:2104.13714*.

[39] Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., & Oliva, A. (2016). Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific reports*, 6(1), 1–13.

[40] Cohen, M. R. & Maunsell, J. H. (2014). Neuronal mechanisms of spatial attention in visual cerebral cortex. *The Oxford handbook of attention*, (pp. 318–345).

[41] Cox, D. D. & Savoy, R. L. (2003). Functional magnetic resonance imaging (fmri)"brain reading": detecting and classifying distributed patterns of fmri activity in human visual cortex. *Neuroimage*, 19(2), 261–270.

[42] Crick, F. (1989). The recent excitement about neural networks. *Nature*, 337(6203), 129–132.

[43] Cutsuridis, V. & Wennekers, T. (2009). Hippocampus, microcircuits and associative memory. *Neural Networks*, 22(8), 1120–1128.

[44] Dan, Y., Atick, J. J., & Reid, R. C. (1996). Efficient coding of natural scenes in the lateral geniculate nucleus: experimental test of a computational theory. *Journal of neuroscience*, 16(10), 3351–3362.

[45] Dapello, J., Marques, T., Schrimpf, M., Geiger, F., Cox, D. D., & DiCarlo, J. J. (2020). Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations. In *NeurIPS*.

[46] Davatzikos, C., Ruparel, K., Fan, Y., Shen, D., Acharyya, M., Loughead, J. W., Gur, R. C., & Langleben, D. D. (2005). Classifying spatial patterns of brain activity with machine learning methods: application to lie detection. *Neuroimage*, 28(3), 663–668.

[47] Dayan, P., Hinton, G. E., Neal, R. M., & Zemel, R. S. (1995). The helmholtz machine. *Neural computation*, 7(5), 889–904.

[48] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255).: Ieee.

[49] Desai, K. & Johnson, J. (2021). VirTex: Learning Visual Representations from Textual Annotations. In *CVPR*.

[50] Desimone, R. (1996). Neural mechanisms for visual memory and their role in attention. *Proceedings of the National Academy of Sciences*, 93(24), 13494–13499.

[51] Devillers, B., Choksi, B., Bielawski, R., & VanRullen, R. (2021). Does language help generalization in vision models? *arXiv preprint arXiv:2104.08313*.

[52] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[53] DiCarlo, J. J., Haefner, R., Isik, L., Konkle, T., Kriegeskorte, N., Peters, B., Rust, N., Stachenfeld, K., Tenenbaum, J. B., Tsao, D., et al. (2021). How does the brain combine generative models and direct discriminative computations in high-level vision? *CCN 2021 Workshop GAC*. https://openreview.net/forum?id=zlTiwFtLlR4.

[54] Dong, D. W. & Atick, J. J. (1995). Temporal decorrelation: a theory of lagged and nonlagged responses in the lateral geniculate nucleus. *Network: Computation in neural systems*, 6(2), 159–178.

[55] Dora, S., Bohte, S. M., & Pennartz, C. M. (2021). Deep gated hebbian predictive coding accounts for emergence of complex neural response properties along the visual cortical hierarchy. *Frontiers in Computational Neuroscience*, 15, 666131.

[56] Edelman, G. M. (1993). Neural darwinism: selection and reentrant signaling in higher brain function. *Neuron*, 10(2), 115–125.

[57] Ekman, M., Kok, P., & de Lange, F. P. (2017). Time-compressed preplay of anticipated events in human primary visual cortex. *Nature Communications*, 8(1), 1–9.

[58] Elsayed, G., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., & Sohl-Dickstein, J. (2018). Adversarial examples that fool both computer vision and time-limited humans. *Advances in neural information processing systems*, 31.

[59] Engstrom, L., Ilyas, A., Salman, H., Santurkar, S., & Tsipras, D. (2019a). Robustness (python library).

[60] Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Tran, B., & Madry, A. (2019b). Adversarial robustness as a prior for learned representations.

[61] Epstein, R. (2016). The empty brain. *Aeon, May*, 18(2016), 3.

[62] Ernst, M. R., Triesch, J., & Burwick, T. (2019). Recurrent connections aid occluded object recognition by discounting occluders. In *International Conference on Artificial Neural Networks* (pp. 294–305).: Springer.

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

[64] Feynman, R. P. (2018). *Statistical mechanics: a set of lectures*. CRC press.

[65] FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6), 445–466.

[66] Friston, K. (2005). A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, 360(1456), 815–836.

[67] Friston, K. (2008). Hierarchical models in the brain. *PLoS computational biology*, 4(11), e1000211.

[68] Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J.-P., Frith, C. D., & Frackowiak, R. S. (1994). Statistical parametric maps in functional imaging: a general linear approach. *Human brain mapping*, 2(4), 189–210.

[69] Frosst, N., Sabour, S., & Hinton, G. E. (2018). Darccc: Detecting adversaries by reconstruction from class conditional capsules. *ArXiv*, abs/1811.06969.

[70] Gaziv, G., Beliy, R., Granot, N., Hoogi, A., Strappini, F., Golan, T., & Irani, M. (2020). Self-supervised natural image reconstruction and rich semantic classification from brain activity. *bioRxiv*.

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

[72] Geirhos, R., Janssen, D. H., Schütt, H. H., Rauber, J., Bethge, M., & Wichmann, F. A. (2017). Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv preprint arXiv:1706.06969*.

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

[74] Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., & Wichmann, F. A. (2018). Generalisation in humans and deep neural networks. *Advances in neural information processing systems*, 31.

[75] Gelbard-Sagiv, H., Mukamel, R., Harel, M., Malach, R., & Fried, I. (2008). Internally generated reactivation of single neurons in human hippocampus during free recall. *Science*, 322(5898), 96–101.

[76] Georgios, Z. (2016). In our own image: Savior or destroyer?: The history and future of artificial intelligence.

[77] Goh, G., Cammarata, N., Voss, C., Carter, S., Petrov, M., Schubert, L., Radford, A., & Olah, C. (2021). Multimodal neurons in artificial neural networks. *Distill*, 6(3), e30.

[78] Goodfellow, I., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

[79] Gross, C. G. (1995). Aristotle on the brain. *The Neuroscientist*, 1(4), 245–250.

[80] Gross, C. G. (2002). Genealogy of the "grandmother cell". *The Neuroscientist*, 8(5), 512–518.

[81] Güçlü, U. & van Gerven, M. A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27), 10005–10014.

[82] Güçlü, U. & van Gerven, M. A. (2017). Increasingly complex representations of natural movies across the dorsal stream are shared between subjects. *NeuroImage*, 145, 329–336.

[83] Guest, O. & Love, B. C. (2017). What the success of brain imaging implies about the neural code. *Elife*, 6, e21397.

[84] Guest, O. & Martin, A. E. (2021). On logical inference over brains, behaviour, and artificial neural networks.

[85] Hansen, L. K. (2007). Multivariate strategies in functional magnetic resonance imaging. *Brain and language*, 102(2), 186–191.

[86] Harrison, C. (1952). Experiments with linear prediction in television. *Bell System Technical Journal*, 31(4), 764–783.

[87] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. corr abs/1512.03385 (2015).

[88] Hebb, D. O. (2005). *The organization of behavior: A neuropsychological theory*. Psychology Press.

[89] Heeger, D. J. (2017). Theory of cortical function. *Proceedings of the National Academy of Sciences*, 114(8), 1773–1782.

[90] Heilbron, M. & Chait, M. (2018). Great expectations: is there evidence for predictive coding in auditory cortex? *Neuroscience*, 389, 54–73.

[91] Hein, G. & Knight, R. T. (2008). Superior temporal sulcus—it's my area: or is it? *Journal of cognitive neuroscience*, 20(12), 2125–2136.

[92] Hendrycks, D. & Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*.

[93] Higuchi, S.-I. & Miyashita, Y. (1996). Formation of mnemonic neuronal responses to visual paired associates in inferotemporal cortex is impaired by perirhinal and entorhinal lesions. *Proceedings of the National Academy of Sciences*, 93(2), 739–743.

[94] Hodgkin, A. L. & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4), 500.

[95] Hohwy, J., Roepstorff, A., & Friston, K. (2008). Predictive coding explains binocular rivalry: An epistemological review. *Cognition*, 108(3), 687–701.

[96] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8), 2554–2558.

[97] Horikawa, T. & Kamitani, Y. (2017). Generic decoding of seen and imagined objects using hierarchical visual features. *Nature communications*, 8(1), 1–15.

[98] Huang, G., Liu, Z., & Weinberger, K. Q. (2016). Densely connected convolutional networks. corr abs/1608.06993 (2016). *arXiv preprint arXiv:1608.06993*.

[99] Huang, Y., Gornet, J., Dai, S., Yu, Z., Nguyen, T., Tsao, D., & Anandkumar, A. (2020). Neural networks with recurrent generative feedback. *Advances in Neural Information Processing Systems*, 33.

[100] Huang, Y. & Rao, R. P. (2011). Predictive coding. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(5), 580–593.

[101] Hubel, D. H. & Wiesel, T. N. (1965). Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of neurophysiology*, 28(2), 229–289.

[102] Huffman, D. J. & Stark, C. E. (2014). Multivariate pattern analysis of the human medial temporal lobe revealed representationally categorical cortex and representationally agnostic hippocampus. *Hippocampus*, 24(11), 1394–1403.

[103] Hung, C. P., Kreiman, G., Poggio, T., & DiCarlo, J. J. (2005). Fast readout of object identity from macaque inferior temporal cortex. *Science*, 310(5749), 863–866.

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

[105] Jehee, J. F. & Ballard, D. H. (2009). Predictive feedback can account for biphasic responses in the lateral geniculate nucleus. *PLoS computational biology*, 5(5), e1000373.

[106] Jehee, J. F., Rothkopf, C., Beck, J. M., & Ballard, D. H. (2006). Learning receptive fields using predictive feedback. *Journal of Physiology-Paris*, 100(1-3), 125–132.

[107] Jin, G., Shen, S., Zhang, D., Dai, F., & Zhang, Y. (2019). Ape-gan: Adversarial perturbation elimination with gan. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 3842–3846).

[108] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2), 183–233.

[109] Kamitani, Y. & Tong, F. (2005). Decoding the visual and subjective contents of the human brain. *Nature neuroscience*, 8(5), 679–685.

[110] Kar, K., Kubilius, J., Schmidt, K., Issa, E. B., & DiCarlo, J. J. (2019). Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nature neuroscience*, 22(6), 974–983.

[111] Kay, K. N., Naselaris, T., Prenger, R. J., & Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature*, 452(7185), 352–355.

[112] Kell, A. J., Yamins, D. L., Shook, E. N., Norman-Haignere, S. V., & McDermott, J. H. (2018). A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3), 630–644.

[113] Keysers, C., Xiao, D.-K., Földiák, P., & Perrett, D. I. (2001). The speed of sight. *Journal of cognitive neuroscience*, 13(1), 90–101.

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

[115] Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., & Masquelier, T. (2018). Stdp-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99, 56–67.

[116] Kietzmann, T. C., Spoerer, C. J., Sörensen, L. K., Cichy, R. M., Hauk, O., & Kriegeskorte, N. (2019). Recurrence is required to capture the representational dynamics of the human visual system. *Proceedings of the National Academy of Sciences*, 116(43), 21854–21863.

[117] Kim, E., Rego, J., Watkins, Y., & Kenyon, G. T. (2020). Modeling biological immunity to adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4666–4675).

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

[119] Kingma, D. P. & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[120] Kobyzev, I., Prince, S. J., & Brubaker, M. A. (2020). Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11), 3964–3979.

[121] Kok, P., Failing, M. F., & de Lange, F. P. (2014). Prior expectations evoke stimulus templates in the primary visual cortex. *Journal of cognitive neuroscience*, 26(7), 1546–1554.

[122] Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., & Houlsby, N. (2019). Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 6(2), 8.

[123] Konorski, J. (1967). *Integrative activity of the brain; an interdisciplinary approach*. University of Chicago Press.

[124] Kreiman, G., Koch, C., & Fried, I. (2000). Imagery neurons in the human brain. *Nature*, 408(6810), 357–361.

[125] Kriegeskorte, N., Mur, M., & Bandettini, P. (2008a). Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2, 4.

[126] Kriegeskorte, N., Mur, M., Ruff, D. A., Kiani, R., Bodurka, J., Esteky, H., Tanaka, K., & Bandettini, P. A. (2008b). Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron*, 60(6), 1126–1141.

[127] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

[128] Krotov, D. & Hopfield, J. (2018). Dense associative memory is robust to adversarial inputs. *Neural computation*, 30(12), 3151–3167.

[129] Kubilius, J., Schrimpf, M., Kar, K., Rajalingham, R., Hong, H., Majaj, N., Issa, E., Bashivan, P., Prescott-Roy, J., Schmidt, K., Nayebi, A., Bear, D., Yamins, D. L., & DiCarlo, J. J. (2019). Brain-like object recognition with high-performing shallow recurrent anns. In H. Wallach, H. Larochelle, A.

Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, volume 32: Curran Associates, Inc.

[130] Lazarou, M., Avrithis, Y., & Stathaki, T. (2021). Few-shot learning via tensor hallucination. *arXiv preprint arXiv:2104.09467*.

[131] Lee, D.-H., Zhang, S., Fischer, A., & Bengio, Y. (2015). Difference target propagation. In *Joint european conference on machine learning and knowledge discovery in databases* (pp. 498–515).: Springer.

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

[133] Li, Z., Caro, J. O., Rusak, E., Brendel, W., Bethge, M., Anselmi, F., Patel, A. B., Tolias, A. S., & Pitkow, X. (2022). Robust deep learning object recognition models rely on low frequency information in natural images. *bioRxiv*.

[134] Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6), 335–346.

[135] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *Lecture Notes in Computer Science*, (pp. 740–755).

[136] Lindsay, G. W., Mrsic-Flogel, T. D., & Sahani, M. (2022). Bio-inspired neural networks implement different recurrent visual processing strategies than task-trained ones do. *bioRxiv*.

[137] Linsley, D., Ashok, A. K., Govindarajan, L. N., Liu, R., & Serre, T. (2020). Stable and expressive recurrent vision models. *arXiv preprint arXiv:2005.11362*.

[138] Linsley, D., Kim, J., Veerabadran, V., Windolf, C., & Serre, T. (2018). Learning long-range spatial dependencies with horizontal gated recurrent units. In *Advances in neural information processing systems* (pp. 152–164).

[139] Lotter, W., Kreiman, G., & Cox, D. (2016). Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*.

[140] Lotter, W., Kreiman, G., & Cox, D. (2020). A neural network trained for prediction mimics diverse features of biological neurons and perception. *Nature machine intelligence*, 2(4), 210–219.

[141] Lu, X., Yuan, Y., & Yan, P. (2013). Sparse coding for image denoising using spike and slab prior. *Neurocomputing*, 106, 12–20.

[142] Luo, Y., Boix, X., Roig, G., Poggio, T., & Zhao, Q. (2015). Foveation-based mechanisms alleviate adversarial examples. *arXiv preprint arXiv:1511.06292*.

[143] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

[144] Marino, J. (2022). Predictive coding, variational autoencoders, and biological connections. *Neural Computation*, 34(1), 1–44.

[145] Maunsell, J. & van Essen, D. C. (1983). The connections of the middle temporal visual area (mt) and their relationship to a cortical hierarchy in the macaque monkey. *Journal of Neuroscience*, 3(12), 2563–2586.

[146] Mehta, M. R. (2001). Neuronal dynamics of predictive coding. *The Neuroscientist*, 7(6), 490–495.

[147] Meng, D. & Chen, H. (2017). Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (pp. 135–147).

[148] Miech, A., Zhukov, D., Alayrac, J.-B., Tapaswi, M., Laptev, I., & Sivic, J. (2019). Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2630–2640).

[149] Millidge, B., Seth, A., & Buckley, C. L. (2021). Predictive coding: a theoretical and experimental review. *arXiv preprint arXiv:2107.12979*.

[150] Millidge, B., Tschantz, A., & Buckley, C. L. (2022). Predictive coding approximates backprop along arbitrary computation graphs. *Neural Computation*, 34(6), 1329–1368.

[151] Miyawaki, Y., Uchida, H., Yamashita, O., Sato, M.-a., Morito, Y., Tanabe, H. C., Sadato, N., & Kamitani, Y. (2008). Visual image reconstruction from human brain activity using a combination of multiscale local image decoders. *Neuron*, 60(5), 915–929.

[152] Mozafari, M., Reddy, L., & VanRullen, R. (2020). Reconstructing natural scenes from fmri patterns using bigbigan. In *2020 international joint conference on neural networks (IJCNN)* (pp. 1–8).: IEEE.

[153] Muckli, L., De Martino, F., Vizioli, L., Petro, L. S., Smith, F. W., Ugurbil, K., Goebel, R., & Yacoub, E. (2015). Contextual feedback to superficial layers of v1. *Current Biology*, 25(20), 2690–2695.

[154] Mumford, D. (1992). On the computational architecture of the neocortex. *Biological cybernetics*, 66(3), 241–251.

[155] Naselaris, T., Prenger, R. J., Kay, K. N., Oliver, M., & Gallant, J. L. (2009). Bayesian reconstruction of natural images from human brain activity. *Neuron*, 63(6), 902–915.

[156] Nayebi, A., Attinger, A., Campbell, M., Hardcastle, K., Low, I., Mallory, C. S., Mel, G., Sorscher, B., Williams, A. H., Ganguli, S., et al. (2021). Explaining heterogeneity in medial entorhinal cortex with task-driven neural networks. *Advances in Neural Information Processing Systems*, 34, 12167–12179.

[157] Nayebi, A., Bear, D., Kubilius, J., Kar, K., Ganguli, S., Sussillo, D., DiCarlo, J. J., & Yamins, D. L. (2018). Task-driven convolutional recurrent models of the visual system. *Advances in neural information processing systems*, 31.

[158] Nayebi, A. & Ganguli, S. (2017). Biologically inspired protection of deep networks from adversarial attacks. *arXiv preprint arXiv:1703.09202*.

[159] Nguyen, A., Yosinski, J., & Clune, J. (2014). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. arxiv e-prints, art. *arXiv preprint arXiv:1412.1897*.

[160] Nonaka, S., Majima, K., Aoki, S. C., & Kamitani, Y. (2021). Brain hierarchy score: Which deep neural networks are hierarchically brain-like? *IScience*, 24(9), 103013.

[161] O'Doherty, J. P., Buchanan, T. W., Seymour, B., & Dolan, R. J. (2006). Predictive neural coding of reward preference involves dissociable responses in human ventral midbrain and ventral striatum. *Neuron*, 49(1), 157–166.

[162] Oja, E., Hyvärinen, A., & Hoyer, P. (1999). Image feature extraction and denoising by sparse coding. *Pattern Analysis & Applications*, 2(2), 104–110.

[163] O'Keefe, J. & Dostrovsky, J. (1971). The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain research*.

[164] Oliver, B. (1952). Efficient coding. *The Bell System Technical Journal*, 31(4), 724–750.

[165] Olshausen, B. A. & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583), 607–609.

[166] Oord, A. v. d., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[167] Oota, S. R., Arora, J., Rowtula, V., Gupta, M., & Bapi, R. S. (2022). Visio-linguistic brain encoding. *arXiv preprint arXiv:2204.08261*.

[168] Orhan, A. E. & Lake, B. M. (2019). Improving the robustness of imagenet classifiers using elements of human visual cognition. *arXiv preprint arXiv:1906.08416*.

[169] Ozcelik, F., Choksi, B., Mozafari, M., Reddy, L., & VanRullen, R. (2022). Reconstruction of perceived images from fmri patterns and semantic brain exploration using instance-conditioned gans. *arXiv preprint arXiv:2202.12692*.

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

[171] Pang, Z., O'May, C. B., Choksi, B., & VanRullen, R. (2021a). Predictive coding feedback results in perceived illusory contours in a recurrent neural network. *Neural Networks*, 144, 164–175.

[172] Pang, Z., O'May, C. B., Choksi, B., & VanRullen, R. (2021b). Predictive coding feedback results in perceived illusory contours in a recurrent neural network. *Neural Networks*, 144, 164–175.

[173] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 8024–8035). Curran Associates, Inc.

[174] Poggio, T. & Girosi, F. (1989). *A theory of networks for approximation and learning*. Technical report, Massachusetts INST of TECH Cambridge Artificial Intelligence LAB.

[175] Pramod, R. & Arun, S. (2016). Do computational models differ systematically from human object perception? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1601–1609).

[176] Quiroga, R. Q., Kraskov, A., Koch, C., & Fried, I. (2009). Explicit encoding of multimodal percepts by single neurons in the human brain. *Current Biology*, 19(15), 1308–1313.

[177] Quiroga, R. Q., Kreiman, G., Koch, C., & Fried, I. (2008). Sparse but not 'grandmother-cell'coding in the medial temporal lobe. *Trends in cognitive sciences*, 12(3), 87–91.

[178] Quiroga, R. Q., Reddy, L., Kreiman, G., Koch, C., & Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature*, 435(7045), 1102–1107.

[179] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

[180] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.

[181] Rajaei, K., Mohsenzadeh, Y., Ebrahimpour, R., & Khaligh-Razavi, S.-M. (2019). Beyond core object recognition: Recurrent processes account for object recognition under occlusion. *PLoS computational biology*, 15(5), e1007001.

[182] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

[183] Rao, R. P. & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1), 79–87.

[184] Rao, R. P. & Ballard, D. H. (2005). Probabilistic models of attention based on iconic representations and predictive coding. In *Neurobiology of attention* (pp. 553–561). Elsevier.

[185] Rauber, J., Brendel, W., & Bethge, M. (2017). Foolbox: A python toolbox to benchmark the robustness of machine learning models. In *Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning*.

[186] Reddy, L. & Thorpe, S. J. (2014). Concept cells through associative learning of high-level representations. *Neuron*, 84(2), 248–251.

[187] Reddy, L., Tsuchiya, N., & Serre, T. (2010). Reading the mind's eye: decoding category information during mental imagery. *Neuroimage*, 50(2), 818–825.

[188] Rezende, D. & Mohamed, S. (2015). Variational inference with normalizing flows. In *International conference on machine learning* (pp. 1530–1538).: PMLR.

[189] Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R. P., de Berker, A., Ganguli, S., et al. (2019). A deep learning framework for neuroscience. *Nature neuroscience*, 22(11), 1761–1770.

[190] Roth, K., Kilcher, Y., & Hofmann, T. (2019). The odds are odd: A statistical test for detecting adversarial examples. In *International Conference on Machine Learning* (pp. 5498–5507).: PMLR.

[191] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533–536.

[192] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.

[193] Salman, H., Ilyas, A., Engstrom, L., Kapoor, A., & Madry, A. (2020). Do adversarially robust imagenet models transfer better? In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems*, volume 33 (pp. 3533–3545).: Curran Associates, Inc.

[194] Salvatori, T., Song, Y., Xu, Z., Lukasiewicz, T., Bogacz, R., Lin, H., Fan, Y., Zhang, J., Bai, B., Xu, Z., et al. (2022). Reverse differentiation via predictive coding. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence ,AAAI 2022 ,Vancouver, BC, Canada, February 22–March 1 ,2022*, volume 10177 (pp. 507–524).: AAAI Press.

[195] Samangouei, P., Kabkab, M., & Chellappa, R. (2018). Defense-gan: Protecting classifiers against adversarial attacks using generative models. *International Conference on Learning Representations*.

[196] Sariyildiz, M. B., Perez, J., & Larlus, D. (2020). Learning visual representations with caption annotations. In *European Conference on Computer Vision (ECCV)*.

[197] Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., & Madry, A. (2018). Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems* (pp. 5014–5026).

[198] Schrimpf, M., Kubilius, J., Hong, H., Majaj, N. J., Rajalingham, R., Issa, E. B., Kar, K., Bashivan, P., Prescott-Roy, J., Geiger, F., et al. (2020). Brain-score: Which artificial neural network for object recognition is most brain-like? *BioRxiv*, (pp. 407007).

[199] Shipp, S. (2016). Neural elements for predictive coding. *Frontiers in psychology*, 7, 1792.

[200] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419), 1140–1144.

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

[202] Smith, E. C. & Lewicki, M. S. (2006). Efficient auditory coding. *Nature*, 439(7079), 978–982.

[203] Solomon, S. S., Tang, H., Sussman, E., & Kohn, A. (2021). Limited evidence for sensory prediction error responses in visual cortex of macaques and humans. *Cerebral Cortex*, 31(6), 3136–3152.

[204] Song, Y., Kim, T., Nowozin, S., Ermon, S., & Kushman, N. (2018). Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*.

[205] Song, Y., Lukasiewicz, T., Xu, Z., & Bogacz, R. (2020). Can the brain do backpropagation?—exact implementation of backpropagation in predictive coding networks. *Advances in neural information processing systems*, 33, 22566–22579.

[206] Spoerer, C. J., McClure, P., & Kriegeskorte, N. (2017). Recurrent convolutional neural networks: a better model of biological object recognition. *Frontiers in psychology*, 8, 1551.

[207] Spratling, M. W. (2008). Predictive coding as a model of biased competition in visual attention. *Vision research*, 48(12), 1391–1408.

[208] Spratling, M. W. (2017). A hierarchical predictive coding model of object recognition in natural images. *Cognitive computation*, 9(2), 151–167.

[209] Spratling, M. W. & Johnson, M. H. (2004). A feedback model of visual attention. *Journal of cognitive neuroscience*, 16(2), 219–237.

[210] Srinivasan, M. V., Laughlin, S. B., & Dubs, A. (1982). Predictive coding: a fresh view of inhibition in the retina. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 216(1205), 427–459.

[211] Sulam, J., Muthukumar, R., & Arora, R. (2020). Adversarial robustness of supervised sparse coding. *Advances in neural information processing systems*.

[212] Summerfield, C., Trittschuh, E. H., Monti, J. M., Mesulam, M., Egner, T., et al. (2008). Neural repetition suppression reflects fulfilled perceptual expectations. *Nature neuroscience*, 11(9), 1004–1006.

[213] Symonds, R. M., Lee, W. W., Kohn, A., Schwartz, O., Witkowski, S., & Sussman, E. S. (2017). Distinguishing neural adaptation and predictive coding hypotheses in auditory change detection. *Brain topography*, 30(1), 136–148.

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

[215] Tan, M. & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning* (pp. 6105–6114).: PMLR.

[216] Tanaka, H., Nayebi, A., Maheswaranathan, N., McIntosh, L., Baccus, S., & Ganguli, S. (2019). From deep learning to mechanistic understanding in neuroscience: the structure of retinal prediction. *Advances in neural information processing systems*, 32.

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

[218] Tao, G., Ma, S., Liu, Y., & Zhang, X. (2018). Attacks meet interpretability: Attribute-steered detection of adversarial samples. In *Advances in Neural Information Processing Systems* (pp. 7717–7728).

[219] Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., & Maida, A. (2019). Deep learning in spiking neural networks. *Neural networks*, 111, 47–63.

[220] Thirion, B., Duchesnay, E., Hubbard, E., Dubois, J., Poline, J.-B., Lebihan, D., & Dehaene, S. (2006). Inverse retinotopy: inferring the visual content of images from brain activation patterns. *Neuroimage*, 33(4), 1104–1116.

[221] Thorpe, S., Fize, D., & Marlot, C. (1996). Speed of processing in the human visual system. *nature*, 381(6582), 520–522.

[222] Thrush, T., Jiang, R., Bartolo, M., Singh, A., Williams, A., Kiela, D., & Ross, C. (2022). Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5238–5248).

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

[224] VanRullen, R. (2007). The power of the feed-forward sweep. *Advances in Cognitive Psychology*, 3(1-2), 167.

[225] VanRullen, R. (2017). Perception science in the age of deep neural networks.

[226] VanRullen, R. & Alamia, A. (2021). Gattanet: Global attention agreement for convolutional neural networks. In *International Conference on Artificial Neural Networks* (pp. 281–293).: Springer.

[227] VanRullen, R. & Kanai, R. (2021). Deep learning and the global workspace theory. *Trends in Neurosciences*, 44(9), 692–704.

[228] VanRullen, R. & Reddy, L. (2019). Reconstructing faces from fmri patterns using deep generative neural networks. *Communications biology*, 2(1), 1–10.

[229] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[230] Vinyals, O., Babuschkin, I., Chung, J., Mathieu, M., Jaderberg, M., Czarnecki, W. M., Dudzik, A., Huang, A., Georgiev, P., Powell, R., et al. (2019). Alphastar: Mastering the real-time strategy game starcraft ii. *DeepMind blog*, 2.

[231] Vogels, R. (2016). Sources of adaptation of inferior temporal cortical responses. *Cortex*, 80, 185–195.

[232] Vylomova, E., Rimell, L., Cohn, T., & Baldwin, T. (2015). Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. *arXiv preprint arXiv:1509.01692*.

[233] Walsh, K. S., McGovern, D. P., Clark, A., & O'Connell, R. G. (2020). Evaluating the neurophysiological evidence for predictive processing as a model of perception. *Annals of the New York Academy of Sciences*, 1464(1), 242.

[234] Weilnhammer, V., Stuke, H., Hesselmann, G., Sterzer, P., & Schmack, K. (2017). A predictive coding account of bistable perception-a model-based fmri study. *PLoS computational biology*, 13(5), e1005536.

[235] Wen, H., Han, K., Shi, J., Zhang, Y., Culurciello, E., & Liu, Z. (2018). Deep predictive coding network for object recognition. *arXiv preprint arXiv:1802.04762*.

[236] Whittington, J. C. & Bogacz, R. (2017). An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation*, 29(5), 1229–1262.

[237] Wyatte, D., Curran, T., & O'Reilly, R. (2012). The limits of feedforward vision: Recurrent processing promotes robust object recognition when objects are degraded. *Journal of Cognitive Neuroscience*, 24(11), 2248–2261.

[238] Xie, C., Tan, M., Gong, B., Wang, J., Yuille, A. L., & Le, Q. V. (2020). Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 819–828).

[239] Xie, C., Wu, Y., Maaten, L. v. d., Yuille, A. L., & He, K. (2019). Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 501–509).

[240] Xu, Y. & Vaziri-Pashkam, M. (2021). Limits to visual representational correspondence between convolutional neural networks and the human brain. *Nature communications*, 12(1), 1–16.

[241] Xu, Z. & Duffy, V. G. (2021). A systematic review of autonomous driving in transportation. In *International Conference on Human-Computer Interaction* (pp. 389–402).: Springer.

[242] Yamins, D. L. & DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3), 356–365.

[243] Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & 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.

[244] Zarrin, P. S., Zimmer, R., Wenger, C., & Masquelier, T. (2020). Epileptic seizure detection using a neuromorphic-compatible deep spiking neural network. In *International Work-Conference on Bioinformatics and Biomedical Engineering* (pp. 389–394).: Springer.

[245] Zhang, H. & Wang, J. (2019). Defense against adversarial attacks using feature scattering-based adversarial training. In *Advances in Neural Information Processing Systems* (pp. 1831–1841).

[246] Zhang, H., Zhang, J., & Koniusz, P. (2019). Few-shot learning via saliency-guided hallucination of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2770–2779).