TOWARDS ROBUST VISION BY MULTI-TASK LEARNING ON MONKEY VISUAL CORTEX

Shahd Safarani,1,* Arne Nix,1-2 Konstantin Willeke,1-2 Santiago A. Cadena,2-3 Kelli Restivo,4-5 George Denfield,6 Andreas S. Tolias,4-5 Fabian H. Sinz1-5,**

1 Institute for Bioinformatics and Medical Informatics, University Tübingen, Germany
2 International Max Planck Research School for Intelligent Systems, Tübingen, Germany
3 Bernstein Center for Computational Neuroscience, University of Tübingen, Germany
4 Department for Neuroscience, Baylor College of Medicine, Houston, TX, USA
5 Center for Neuroscience and Artificial Intelligence, Baylor College of Medicine, Houston, TX, USA
6 Columbia University, Department of Psychiatry, New York, USA

*shahd.safarani@sinzlab.net, **fabian.sinz@uni-tuebingen.de

ABSTRACT

Deep neural networks set the state-of-the-art across many tasks in computer vision, but their generalization ability to simple image distortions is surprisingly fragile. In contrast, the mammalian visual system is robust to a wide range of distortions. Recent work suggests that this generalization power can be explained by useful inductive biases encoded in the representations of visual stimuli throughout the visual cortex. Here, we successfully leveraged these inductive biases with a multi-task learning approach: we jointly trained a deep network to perform image classification and to predict neural activity in macaque primary visual cortex (V1) in response to the same natural stimuli. We measured the out-of-distribution generalization abilities of our resulting network by testing its robustness to common image distortions. We found that co-training on monkey V1 data indeed leads to increased robustness despite the absence of those distortions during training. Additionally, we show that our network’s robustness is often very close to that of an oracle network where parts of the architecture are directly trained on the test corruptions. Finally, our results also demonstrate that, as the network’s representations become more brain-like, their robustness consistently improves. Overall, our work expands the promising research avenue of transferring inductive biases from biological to artificial neural networks.

1 INTRODUCTION

Although machine learning algorithms have witnessed enormous progress thanks to the deep learning revolution (LeCun et al., 2015), current state-of-the-art deep models (Hinton et al., 2012; Rawat & Wang, 2017; Krizhevsky et al., 2012) still fall behind the generalization abilities of biological brains. A growing body of literature (Szegedy et al., 2013; Geirhos et al., 2018) shows that these models are brittle when tested on out-of-distribution data samples, i.e. their ability to extrapolate is weak, unlike the mammalian visual system which is known to be very robust. For instance, Geirhos et al. (2018) show that humans can easily identify the objects in images that were exposed to common distortions, while the performance of state-of-the-art convolutional neural networks (CNNs) strongly deteriorates on those images. This gap in extrapolation has been attributed to differences in feature representations (Geirhos et al., 2019; Brendel & Bethge, 2019) and internal strategies for decision making (Geirhos et al., 2020) between humans and CNNs.

Historically, neuroscience has inspired many innovations in artificial intelligence (Hassabis et al., 2017; Fukushima, 1980). While most of the transfer between neuroscience and machine learning happens on the implementational level (Marr, 1982; Hassabis et al., 2017), too little is known about the structure of the brain at the level of detail needed to transfer functional generalization properties (Sinz et al., 2019). To transfer functional inductive biases from the brain to deep neural networks...
(DNNs), it may thus be better to consider the representational level by capturing biological feature representations in the responses of biological neurons to visual input – abstracting away from the implementational level. Prior work suggests that enforcing brain-like representations in CNNs via neural data from humans (Fong et al., 2018), mice (Li et al., 2019), or monkeys (Federer et al., 2020) can indeed have beneficial effects on the generalization abilities of these networks.

Our work expands on this line of research, by exploring the extrapolation capabilities of multi-task learning models (MTL; Caruana (1993)) trained on image classification and prediction of neural responses from monkey V1 – as proposed in the neural co-training hypothesis by Sinz et al. (2019).

We implement MTL via a shared representation between image classification and neural response prediction (Fig. 1). The motivation is that MTL with neural data regularizes the shared representation to inherit good functional inductive biases from neural data, and to help it extrapolate better to out-of-distribution images, thus yielding a more robust neural network.

We empirically investigate this idea using common corruptions on tiny ImageNet (TIN)1. We show that 1 MTL can transfer robustness properties even when trained on undistorted images only, and 2 MTL with monkey V1 data has a positive effect on robustness. We 3 analyze our findings using a robust oracle model quantifying what performance improvement can be expected given that only parts of the network are shared during MTL (Fig. 1). Our results provide evidence in favor of the neural co-training hypothesis and further expand the scope of prior results, by exploring the relationship between brain-like representations and robustness.

2 NEURAL MULTI-TASK LEARNING

Data Images for the classification task and the neurophysiological experiments are based on ImageNet (Deng et al., 2009). For the classification task, we use a grayscale version of TIN1. For neural prediction, we use neurophysiological recordings of 458 neurons from the primary visual cortices (area V1) of two fixating awake macaque monkeys, recorded with a 32-channel depth electrode during 15 (monkey 1) and 17 (monkey 2) sessions. In each session, approximately 1000 trials of 15 images are presented – each image for 120ms. We extract the spike count from 40ms to 160ms after image onset. The image set presented to the monkey consists of 24075 images from 964 categories – 25 images per category. Of those, 24000 were designated to model training and 75 to testing. For each training trial, a new subset of 15 images was randomly sampled from the training set. Test images are displayed in fixed order during 5 test trials, randomly interleaved among training trials, and repeated 40-50 times per session. All images are converted to gray-scale and presented at 420×420 px, covering 6.7° visual angle for the monkey, resulting in 63 pixels per degree (ppd). For model training, images are downscaled and cropped to 64×64 pixels, corresponding to 14.5 ppd. Similar to Li et al. (2019), we first train a model on recorded neural data and use it to predict neural responses for all input images of the TIN classification data. These predicted responses form the neural dataset we use in MTL. This allows us to balance the amount of data we have for each task and removes trial-to-trial noise in the neural data.

Models All our experiments are based on a variant of the VGG-19 architecture (Simonyan & Zisserman, 2015) with additional batch normalization layers (Ioffe & Szegedy, 2015) after every convolutional layer (Figure 1). To allow for arbitrary image sizes, we make the network fully convolutional

http://cs231n.stanford.edu/
by replacing the fully connected readout by three convolutional layers with dropout of 0.5 after the first two, and a final pooling operation and softmax \cite{bridle1990}. We predict neural responses by feeding the output of the convolutional layer \texttt{conv-3-1} \cite{cadena2019} into a Gaussian readout \cite{lurz2021} yielding a spike count prediction per neuron and image.

**Training** We use cross-entropy loss for single task image classification and Poisson loss for single task neural prediction. For multitask training, the challenge is finding the optimal balance between the two objectives to achieve reasonable performance on each task individually, and allow both tasks to benefit from each other by learning common representations. To put both objectives on the same scale, we use their corresponding negative log-likelihood and learn their balance through trainable observation noise parameters $\sigma$ \cite{kendall2018}. This yields a combined loss of $\frac{1}{2\sigma^2} \mathcal{L}_{CE}(\theta, \psi_c) + \frac{1}{2\sigma_n^2} \mathcal{L}_{MSE}(\theta, \psi_n) + \log \sigma_c + \log \sigma_n$, where $\theta$ are the shared parameters and $\psi_c$, $\psi_n$, $\sigma_c$, and $\sigma_n$ are the task-specific parameters for classification and neural prediction, respectively. The classification objective $\mathcal{L}_{CE}$ is the standard cross-entropy, analogous to the single-task case. For MTL on neural data, we use mean-squared error $\mathcal{L}_{MSE}$ because the targets are predictions from the network trained on neural data and not the original noisy neural responses. For optimization, we accumulate the gradients over the different losses to optimize the shared parameters $\theta$ in a single combined gradient step. By definition, the two loss components will contribute equally to the learning process. However, we can manually steer the focus towards either task via the proportion (batch-ratio) of neural to classification batches accumulated for each optimization step.

We standardize all pixel values with the mean and standard deviation of the training set, and augment the images by random cropping, horizontal flipping, and rotations in a range of $15^\circ$ for classification. We use stochastic gradient descent with momentum in all classification-related cases, and Adam for single task neural prediction \cite{kingma2015}. We use a batch-size of 128 and weight decay with a factor of $5 \cdot 10^{-4}$ throughout all our experiments, as well as a batch-ratio of 1:1 during MTL. The initial learning rate is determined for each task individually and reduced by a (task-specific) factor via an adaptive learning schedule. The schedule reduces the learning rate depending on the validation performance – classification performance in the case of MTL – when the rate of improvement is not above a $10^{-4}$ for 5 consecutive epochs. The training is stopped when we reach either five learning rate reduction steps or a maximum number of epochs, that we define for each task. This training setup was determined via prior hyper-parameter searches on the validation set. We repeat every experiment with five different random initializations. Error bars were obtained by bootstrapping (250 repetitions).

3 **RESULTS**

Our main goal is to find evidence for improved extrapolation abilities. To this end, we evaluate our model’s robustness on distorted copies of the TIN validation set – used as a test set in our experiments – following the corruption paradigm in \cite{hendrycks2019}. We reproduce the distortions with an on-the-fly implementation \cite{michaelis2019}, drop glass blur because it is computationally expensive, and refer to our resulting test set as \texttt{TIN-TC}. We quantify the robustness for

![Figure 2: Exemplary classification results on TIN-TC, showing 3 corruption types with the best (left), median (center) and worst (right) robustness score for MTL-monkey across 5 increasing levels of severity each.](image-url)

3
Figure 3: A Robustness scores for each model grouped by corruption category, as defined in Hendrycks & Dietterich (2019). B Overall robustness scores for our 5 different models. C Robustness and neural prediction correlate positively for MTL-monkey models across 12 different batch ratios and 5 random seeds per model (grey line: linear regression from robustness to neural performance). A darker color indicates higher accuracy on the clean TIN test set.

Since co-training only affects the shared representation up to layer conv-3-1, we cannot expect the network to be as robust as a network where all layers are trained on data augmented with the image distortions. To explore the limits on robustness resulting from sharing lower layers only, we train a classification model with a 1:1 mixture of clean and distorted images drawn from the pool of 14 IN-C corruptions, freeze all layers up to conv-3-1, and re-train the remaining network on clean data only. To push the robustness to the frozen part, we add a second loss that penalizes the Euclidean distance between the outputs of layer conv-3-1 for the same image augmented with different corruptions – similar to Chen et al. (2020). We refer to this model as the oracle since it has access to the image distortions during training – unlike our MTL models.

MTL can successfully transfer robustness. To demonstrate that MTL can in principle transfer robustness properties without showing distorted images in training, we generate neural responses from our oracle model for all images of the clean TIN dataset by freezing the oracle model and training a Gaussian readout on top of layer conv-3-1 for 10 epochs to predict V1 data. Then, we train a model on the resulting neural responses alongside clean image classification using MTL. We call this model MTL-oracle. Comparing the robustness of this model on TIN-TC to the robustness of the single-task baseline model trained on clean TIN only, we see clear signs of successful transfer (Fig. 2 and Fig. 3A,B) although the MTL network has never seen the image distortions of TIN-TC. In fact, the MTL-oracle performs close to the oracle model in most cases.

Co-training with monkey V1 increases robustness. The results on MTL-oracle show that MTL on neural responses predicted from a robust network on undistorted images successfully transfers robustness properties. For our main experiment, we use MTL on neural responses from the single task monkey V1 model (see section 2), and refer to it as MTL-monkey. This model has never seen distorted images at any point. We call the corresponding control model trained on the same neural data but shuffled across images MTL-shuffled. Similar to the MTL-oracle model, MTL-monkey generalizes better to the TIN-TC image distortions than the baseline model, although it has not seen distorted images at any stage during the training process. We find increased robustness for 9/14 image corruptions. This improvement is mainly observed across 3 groups of distortions: Noise, Blur and Digital (Fig. 3A), whereas MTL-monkey did not exceed the baseline performance for the Weather group. The shuffled control did not provide any benefits (Fig. 2 and Fig. 3A,B).

The more “brain-like” the neural network, the better it generalizes to image distortions. If features in the neural data affect the robustness, we would expect that the robustness correlates positively with the neural prediction performance in MTL-monkey. To test this hypothesis, we create a pool of MTL-monkey models with varying neural performance by altering the amount of neu-
ral data introduced during co-training through the batch-ratio hyperparameter. We find that both
the model accuracy on clean images and neural prediction improves the network’s robustness (Figure 3C; \( p < 10^{-5} \) (t-test) for both factors in a 2-factor linear regression). Analysis for MTL-shuffled shows no connection between robustness and neural performance \((p > 0.5\) for neural prediction and \( p < 10^{-13} \) for clean accuracy). Overall, our results are consistent with previous work finding a
positive correlation between model robustness and brain-likeness (Dapello et al., 2020).

Conclusion and Outlook

To the best of our knowledge, this work is the first that investigates the neural co-training hypothesis, adding further evidence to existing literature that useful representa-
tional inductive biases can be transferred from neural data. By carefully controlling the amount of
neural data and robustness in the neural data for MTL through the batch ratio parameter and the
MTL-oracle model respectively, we were able to show that robustness correlates with neural pre-
diction performance and that the MTL-monkey model is close to the expected ideal performance in
many cases. In the future, we hope to include higher brain areas for neural co-training to achieve
stronger effects and robustness against more complex distortions.

ACKNOWLEDGMENTS

We thank Konstantin Lurz, Mohammad Bashiri, Christoph Blessing and Pawel Pierzchlewicz for
helpful comments on the manuscript. The authors thank the International Max Planck Research
School for Intelligent Systems (IMPRS-IS) for supporting Arne Nix and Konstantin Willeke. This
work was partially supported by the Cyber Valley Research Fund (CyVy-RF-2019-01). FHS is sup-
ported by the Carl-zeiss-Stiftung and acknowledges the support of the DFG Cluster of Excellence
“Machine Learning – New Perspectives for Science”, EXC 2064/1, project number 390727645. This
work was supported by an AWS Machine Learning research award to FHS. Supported by the Intel-
ligence Advanced Research Projects Activity via Department of Interior/Interior Business Center
contract number D16PC00003, DP1 EY023176 Pioneer Grant (to A.S.T.) and grants from the US
Department of Health & Human Services, National Institutes of Health, National Eye Institute (nos.
R01 EY026927 to A.S.T. and T32 EY00252037 and T32 EY07001).

REFERENCES

W. Brendel and M. Bethge. Approximating CNNs with bag-of-local-features models works surpris-
ingly well on ImageNet. In International Conference on Learning Representations (ICLR), May
2019.

John S Bridle. Probabilistic interpretation of feedforward classification network outputs, with rela-

Deep convolutional models improve predictions of macaque V1 responses to natural images.


contrastive learning of visual representations. In International conference on machine learning,

Joel Dapello, Tiago Marques, Martin Schrimpf, Franziska Geiger, David D Cox, and James J Di-
Carlo. Simulating a primary visual cortex at the front of CNNs improves robustness to image

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical

Callie Federer, Haoyan Xu, Alona Fyshe, and Joel Zylberberg. Improved object recognition using
neural networks trained to mimic the brain’s statistical properties. Neural Networks, 131:103–114,
2020.


