Towards robust vision by multi-task learning on monkey visual cortex

Anonymous Author(s)
Affiliation
Address
email

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 perturbations. Recent work suggests that this generalization ability 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 showed that our network's robustness is often very close to that of an Oracle network where parts of the architecture are directly trained on noisy images. Our results also demonstrated that the network's representations become more brain-like as their robustness improves. Using a novel constrained reconstruction analysis, we investigated what makes our brain-regularized network more robust. We found that our monkey co-trained network is more sensitive to content than noise when compared to a Baseline network that we trained for image classification alone. Using DeepGaze-predicted saliency maps for ImageNet images, we found that the monkey co-trained network tends to be more sensitive to salient regions in a scene, reminiscent of existing theories on the role of V1 in the detection of object borders and bottom-up saliency. Overall, our work expands the promising research avenue of transferring inductive biases from biological to artificial neural networks on the representational level, and provides a novel analysis of the effects of our transfer.

1 Introduction

Although machine learning algorithms have witnessed enormous progress thanks to the recent success of deep learning methods [1], current state-of-the-art deep models [2–4] still fall behind the generalization abilities of biological brains [5, 6]. This includes lack of robustness to image corruptions as pointed out by Hendrycks and Dietterich [7], who measured a network's performance on 15 different image corruptions applied to the ImageNet [8] test-set. Studies with similar image corruptions have proven to severely decrease performance in classification networks while having a smaller impact on human perception [9] suggesting that the ability to *extrapolate* is weak in these networks compared to the mammalian visual system. This gap in extrapolation has previously been attributed to differences in feature representations [10, 11] and internal strategies for decision making [12] between humans and CNNs.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

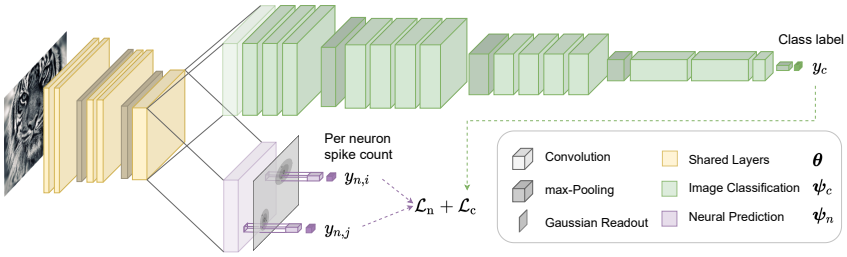

Figure 1: VGG-19 architecture for MTL on image classification and neural prediction.

Historically, neuroscience has inspired many innovations in artificial intelligence [13, 14], and most of this transfer between neuroscience and machine learning happens on the implementational level [15, 13]. However, we currently know too little about the structure of the brain at the level of detail needed to transfer functional generalization properties from biological to artificial systems [5]. 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. In fact, deep neural networks have set new standards in capturing brain activity across multiple areas in the visual cortex [16], becoming the state-of-the-art for neural response prediction of the primary visual cortex (area V1, [17]). Recent work has shown that CNNs, which were fitted to V1 neural data, can generalize well to other neurons, animals, and stimuli [18, 19]. Vice versa, prior work suggests that enforcing brain-like representations in CNNs via neural data from humans [20], mice [21], or monkeys [22] can have beneficial effects on the generalization abilities of these networks to stimuli outside their training distribution for object recognition.

Our work expands on this line of research by exploring the extrapolation capabilities of multi-task learning models (MTL; [23]) trained on image classification and prediction of neural responses from monkey V1. This approach was proposed as the *neural co-training hypothesis* by Sinz et al. [5], but untested to the best of our knowledge. We implement MTL via a shared representation between image classification and neural response prediction (Fig. 1). Our hypothesis is that MTL with neural data regularizes the shared representation to inherit good functional inductive biases from neural data, and improves the network's generalization abilities to out-of-distribution images, thus rendering it more robust. We empirically test this idea using common corruptions on tiny ImageNet (TIN)[1]. We show that MTL on monkey V1 has a positive effect on generalization as it increases the model's robustness to image distortions, even though it is trained on undistorted images only. We compare our model with a robust Oracle model to quantify what performance improvement can be expected given that only parts of the network are shared during MTL. Subsequently, we develop a constrained reconstruction based method to analyze learned sensitivities and invariances in the feature representation of the different models. We find that the robust models qualitatively exhibit different feature sensitivities than a standard classification model. Finally, we show that the feature sensitivity of the monkey V1 co-trained model is related to salient image features, consistent with existing theories about the role of V1 [24]. Overall, our results support the neural co-training hypothesis and further expand the scope of prior results, by exploring the relationship between brain-like representations and robustness. To the best of our knowledge, we are the first to analyze learned feature representations of neurally co-trained models and thus take a step towards a semantic understanding of what makes biological vision robust.

2 Neural multi-task learning

Data The images for the classification task and the neurophysiological experiments were selected from the ImageNet dataset [8]. For the classification task, we used a grayscale version of TIN[1]. For neural prediction, we used 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 were presented – each image for 120ms. We extracted the spike count from 40ms

[1] http://cs231n.stanford.edu/

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 were displayed in fixed order during 5 test trials, randomly interleaved among training trials, and repeated 40-50 times per session. All images were 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 were downscaled and cropped to 64×64 pixels, corresponding to 14.5 ppd. Similar to Li et al. [21], we first trained a model on the image-response pairs of our recorded neural data. We then used this to predict neural responses for all input images of the TIN classification data. These predicted responses served as the basis neural dataset we used in our MTL approach. This allowed us to balance the amount of data we have for each task and it removed trial-to-trial variability in the neural data.

Models All our experiments were based on a variant of the VGG-19 architecture [25] with batch normalization layers [26] after every convolutional layer (Figure 1). To allow for arbitrary image sizes, we made the network fully convolutional 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 [27]. We predicted neural responses by feeding the output of the convolutional layer `conv-3-1`, shown by Cadena et al. [17] to be optimal for predicting V1 responses, into a *Gaussian readout* [18] yielding a spike count prediction per neuron and image.

Training We used cross-entropy loss for single task *image classification* and Poisson loss for single task *neural prediction*. For *multitask training*, the challenge was to find the optimal balance between the two objectives that achieves a reasonable performance on each task individually, and allows both tasks to benefit from each other by learning common representations. To put both objectives on the same scale, we used their corresponding negative log-likelihood and learned their balance through trainable observation noise parameters σ [28]. This yields a combined loss of $\frac{1}{\sigma_c^2}\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}, \boldsymbol{\psi}_c) + \frac{1}{2\sigma_n^2}\mathcal{L}_{\text{MSE}}(\boldsymbol{\theta}, \boldsymbol{\psi}_n) + \log\sigma_c + \log\sigma_n$ where $\boldsymbol{\theta}$ are the shared parameters and $\boldsymbol{\psi}_c$, σ_c and $\boldsymbol{\psi}_n$, σ_n are the task-specific parameters for $c$lassification and $n$eural prediction, respectively. The classification objective $\mathcal{L}_{\text{CE}}$ was the standard cross-entropy, analogous to the single-task case. For MTL on neural data, we used mean-squared error $\mathcal{L}_{\text{MSE}}$ because the targets are predictions from the network trained on neural data and not the original noisy neural responses. For optimization, we accumulated the gradients over the different losses to optimize the shared parameters $\boldsymbol{\theta}$ in a single combined gradient step. By definition, the two loss components would contribute equally to the learning process. However, we can manually steer the focus towards either task via the *batch-ratio* hyperparameter, which defines the proportion of neural to classification batches accumulated for each optimization step.

We standardized all pixel values with the mean and standard deviation of the training set, and augmented the images by random cropping, horizontal flipping, and rotations in a range of $15°$ for classification. Optimization was performed using stochastic gradient descent with momentum in all classification-related cases, and Adam for single task neural prediction [29]. We used 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 for MTL. The initial learning rate was determined for each task individually and reduced by a (task-specific) factor via an adaptive learning rate 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 10^{-4} for 5 consecutive epochs. The training was stopped when we reach either five learning rate reduction steps or a maximum number of epochs, that we defined for each task. This training setup was determined via prior hyper-parameter searches on the validation-set. We repeated every experiment with five different random initializations. Error bars were obtained by bootstrapping (250 repetitions).

3 Results

Our goal is to test whether co-training on neural data can lead to improved extrapolation abilities. To this end, we evaluated our model's robustness on distorted copies of the TIN validation-set – used as a test-set in our experiments – following the corruption paradigm of Hendrycks and Dietterich [7]. We reproduced the distortions with an on-the-fly implementation [30], dropped *glass blur* because it is computationally expensive, and refer to our resulting test-set as *TIN-TC*. We quantified the robustness

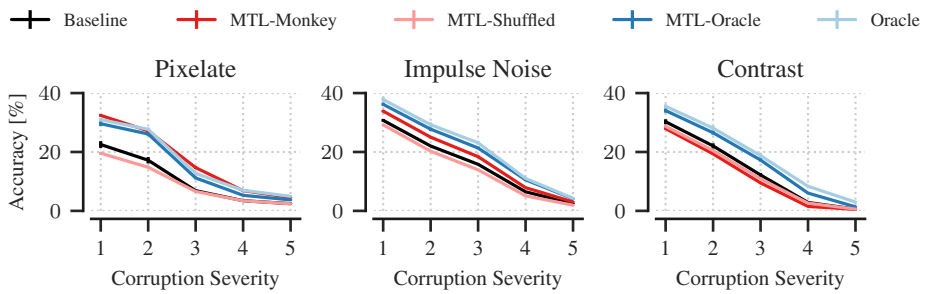

Figure 2: Exemplary classification results on TIN-TC, showing 3 corruption types with the best (left), median (center) and worst (right) robustness score for MTL-Monkey across 5 increasing levels of severity each.

for each of the remaining 14 noise types and five levels of corruption severity separately, and computed a summary robustness score adopted from Hendrycks and Dietterich [7]: $\frac{1}{14}\sum_{c=1}^{14} \overline{A}_c^{\text{robust}} / \overline{A}_c^{\text{Baseline}}$, where $\overline{A}_c = \frac{1}{5}\sum_{l,s=1}^{5} A_{l,c,s}$ denotes the mean accuracy on corruption c across levels l and seeds s.

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 trained a classification model with a 1:1 mixture of clean and distorted images drawn from the pool of 14 IN-C corruptions, froze all layers up to `conv-3-1`, and re-trained the remaining network on clean data only. To push the robustness to the frozen part, we added 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. [31]. 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 generated 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 trained 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 saw 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 performed close to the Oracle model in most cases.

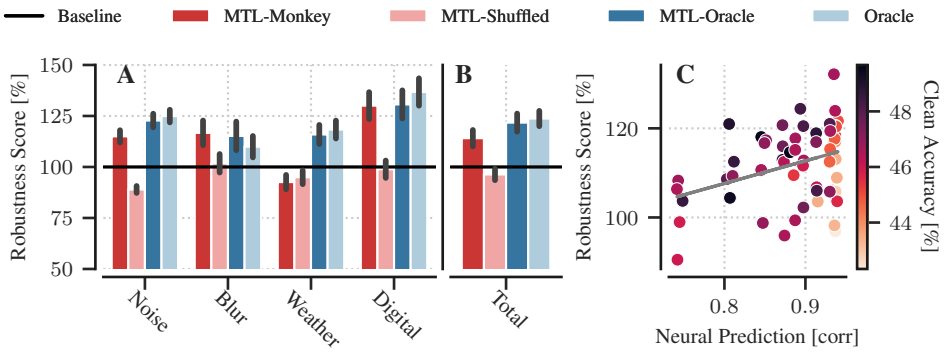

Figure 3: **A** Robustness scores for each model grouped by corruption category, as defined in Hendrycks and Dietterich [7]. **B** Overall robustness scores for our 5 different models. **C** Robustness and neural prediction correlate positively for MTL-Monkey models across 12 different batch ratios and 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.

Co-training with monkey V1 increases robustness. The results of MTL-Oracle show that MTL with neural responses from a robust network in response to undistorted images successfully transfers robustness properties. For our main experiment, we trained MTL with the neural responses generated from the single task predictive 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 with shuffled responses across images for all neurons *MTL-Shuffled*. Similar to the MTL-Oracle model, MTL-Monkey generalized 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 found 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), indicating that the improved robustness of our monkey network is due to learning on real neural data.

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 created a pool of MTL-Monkey models with varying neural performance by altering the amount of neural data introduced during co-training through the batch-ratio hyperparameter. We found that both the model accuracy on clean images and neural prediction improved the network's robustness (Figure 3C; $p < 10^{-5}$ (t-test) for both factors in a 2-factor linear regression). Analysis for MTL-Shuffled showed 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" [32].

4 Analysis

In the previous section, we showed that our MTL approach can transfer robustness properties and that improved robustness correlates with more brain-like representations learnt from monkey V1 data. The aim of this section is to understand the representational differences compared to other models that could be responsible for the increased robustness of our MTL-Monkey model. To this end, we visualize which image features the networks are sensitive to using a novel resource constrained image reconstruction from a given layer across all of the models used in this study. The rationale behind a constrained reconstruction is to put a resource limitation on the total power of an image, thereby force the reconstruction to put contrast in the image wherever it is necessary to recreate the activity of a given layer, and thus visualize the sensitivities and invariances of this layer (see Figure 4). Specifically, given a noisy target image $\mathbf{x}_0$, we computed the corresponding activations $f(\mathbf{x}_0)$ of a particular layer and reconstructed the original image by minimizing the squared loss between the target activations and the activations from the reconstructed image $\ell(\mathbf{x}_0, \mathbf{x}) = \|f(\mathbf{x}) - f(\mathbf{x}_0)\|^2$ subject to a norm constraint

$$\mathbf{x}^* = \text{argmin}_{\mathbf{x}} \|f(\mathbf{x}) - f(\mathbf{x}_0)\|^2 \text{ s.t.} \|\mathbf{x}\|^2 \leq r^2.$$

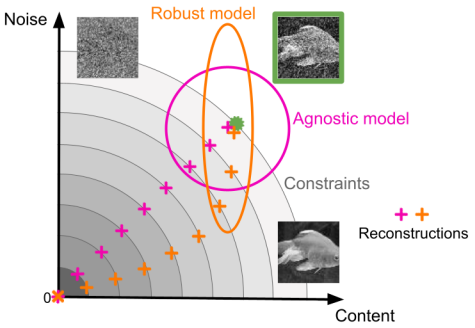

Figure 4: An illustration of the reconstruction process. We reconstruct a noise corrupted target image (green) from a given model under a resource constraint (gray circles). An agnostic model (dark pink) would be, by definition, equally sensitive to: The image content and noise. In contrast, a robust model (orange) would be more sensitive to content and less to noise. Resource constraints that do not allow for a full reconstruction force the optimization to put more image power towards directions to which the model is more sensitive.

Note that, if $\|\mathbf{x}_0\| \leq r$, one trivial optimal solution is $\mathbf{x} = \mathbf{x}_0$. However, if $\|\mathbf{x}_0\| > r$, the constraint becomes active and the reconstruction has to choose where to put power in the image (see Figure 4). This can be seen more formally if we approximate the loss function with a second order Taylor approximation $\ell(\mathbf{x}_0, \mathbf{x}) \approx \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^\top H(\mathbf{x} - \mathbf{x}_0)$ around the optimal solution $\mathbf{x}_0$. Using the local

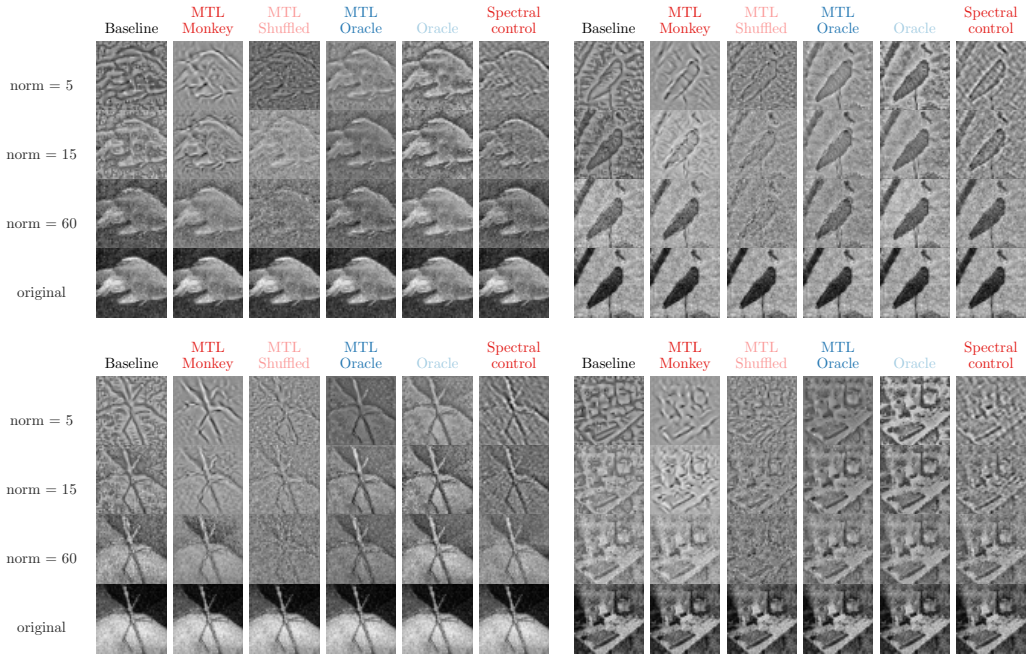

Figure 5: Reconstruction examples from all the 5 models of 4 noisy images (fish, bird, tomato and desk images) with Gaussian noise of severity level 2, under 3 norm constraints: 5, 15 and 60. See main text for details on the spectral control.

approximation in the optimization problem and solving for $\mathbf{x}$ using Lagrange multipliers

$$\max_\gamma \min_{\mathbf{x}} \frac{1}{2}\left(\mathbf{x}-\mathbf{x}_0\right)^\top H\left(\mathbf{x}-\mathbf{x}_0\right)+\gamma\cdot\frac{1}{2}\left(\|\mathbf{x}\|_2^2-r^2\right) \text{ s.t. } \gamma\geq 0$$

yields $\mathbf{x}=(H+\gamma I)^{-1}H\mathbf{x}_0$ where γ denotes the Lagrange multiplier chosen such that $\|\mathbf{x}^*\|=r$ if the constraint is active. To see that this preferentially reconstructs images along directions where the loss in activation space is more sensitive, consider $\mathbf{x}$ and $\mathbf{x}_0$ in the eigenbasis U of the Hessian $H=U\gamma U^\top$, which we denote by $\mathbf{v}=U^\top\mathbf{x}$ and $\mathbf{v}_0$, respectively. In that space, the solution is

$$\mathbf{v}=(\Lambda+\gamma I)^{-1}\Lambda\mathbf{v}_0 \quad \text{or} \quad v_i=\frac{\lambda_i}{\lambda_i+\gamma},$$

which means that directions with $\lambda_i \gg 0$, i.e. high curvature or strong sensitivity, stay as they are while directions with small λ_i, i.e. low curvature or "invariance", get diminished. How strongly they are diminished depends on the Lagrange multiplier γ, or, equivalently, the norm constraint.

In our analysis, we optimized the pixels based on the activations of layer `conv-3-1` – the co-trained layer – for all five models *MTL-Monkey*, *Baseline*, *Oracle*, *MTL-Oracle* and *MTL-Shuffled*. We found SGD with a learning rate of 5 to be most suitable to reconstruct from all the MTL models, and the Adam optimizer with a learning rate of 0.01 best for the Baseline and Oracle models. We always optimized for 8000 steps per image per model for each norm constraint.

Reconstructions from different models show qualitative differences Reconstructions from test images distorted with Gaussian noise under three different norm constraints qualitatively show that the models are sensitive to different features (Figure 5). When looking at a mid-range norm constraint (norm=15), where we expected the difference between the models to be the largest (see Figure 4), we found that the Baseline model seems to be sensitive to distortions present in the original image. This manifests itself in a stronger noise component in the background of the image reconstructed from the Baseline model compared to other models. The robust networks (MTL-Monkey, oracle and MTL-Oracle), on the other hand, were less sensitive to these perturbations and exhibited more content structure for that norm constraint (see appendix for more reconstructions). However, when comparing our MTL-Monkey model with the other robust networks, we noticed that the Oracle and MTL-Oracle

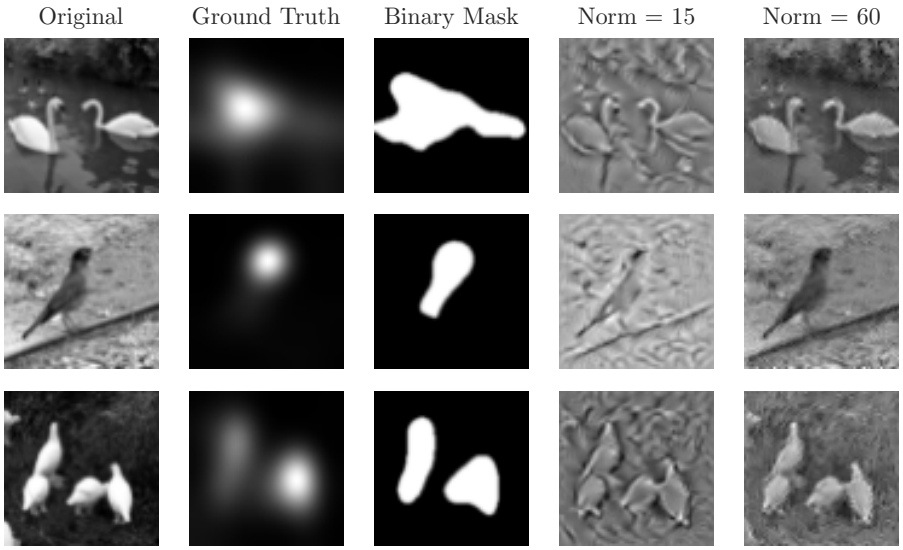

Figure 6: Examples from the ImageNet images used for testing our neural saliency hypothesis in the MTL-Monkey model. In addition to the original image, we show the DeepGaze predicted saliency map and the resulting binarized mask for computing the norm ratio as well as the MTL-Monkey reconstructions with norm constraints (15 and 60).

models generally preserved the original image content as much as possible, while reconstructions from the MTL-Monkey model seem to put slightly more emphasis on edges and object boundaries.

The MTL-Monkey model's sensitivities behavior cannot be fully explained by frequency filtering One simple mechanism, that would make a network more robust against high-frequency types of noise, would be to change the frequency sensitivity towards low-pass components. To assess whether this might be the case for the MTL-Monkey network, we used a simple "spectral control", where we transferred the Fourier amplitude spectrum of the reconstructed image to the original noisy image. This enforces the norm constraint on the original image, and exactly matches the frequency content. If the MTL-Monkey model were simply changing the frequency sensitivity, we would expect the reconstruction and the spectral control to match. However, this does not seem to be the case. The spectral control exhibited more noise in the background and showed less edge enhancing compared to the MTL-Monkey reconstruction. Thus, while changed frequency sensitivity might play a role, there might be additional effects that lead to more noise suppression and more edge enhancement.

The MTL-Monkey exhibits increased sensitivity to salient image regions Our constrained reconstruction analysis exposes image content that is relevant to recreate the responses of a particular layer. In this final section, we try to characterize this content in terms of known perceptual mechanisms of mammalian vision. Motivated by the potential role of V1 in bottom-up saliency [24], we specifically investigated the relation between salient regions of an image and the regions that get emphasized in the reconstructions from different models. It could be possible that focusing on more salient regions of an image, that often include boundaries and shapes of central objects [24, chapter 5], improves the robustness of a model to corruptions as a correlation between shape-like feature detection and robustness has been reported before [10, 30]. From our qualitative observation of the reconstruction examples (Figure 5), we noticed that the reconstructions from the MTL-Monkey model seem to often enhance the central object in a scene by emphasizing edges and boundaries compared to the reconstructions from other robust models. To investigate whether the MTL-Monkey model is more sensitive to salient regions compared to other models, we collected 70 undistorted images with structured background from ImageNet. We used grayscale versions of these images as targets for the reconstruction from our MTL-Monkey, Oracle, and Baseline models. Additionally, we used the DeepGaze II model [33] to predict the saliency maps of these images. To define a binary saliency mask, we took the predicted density map, sorted all resulting pixel values in a descending order, and

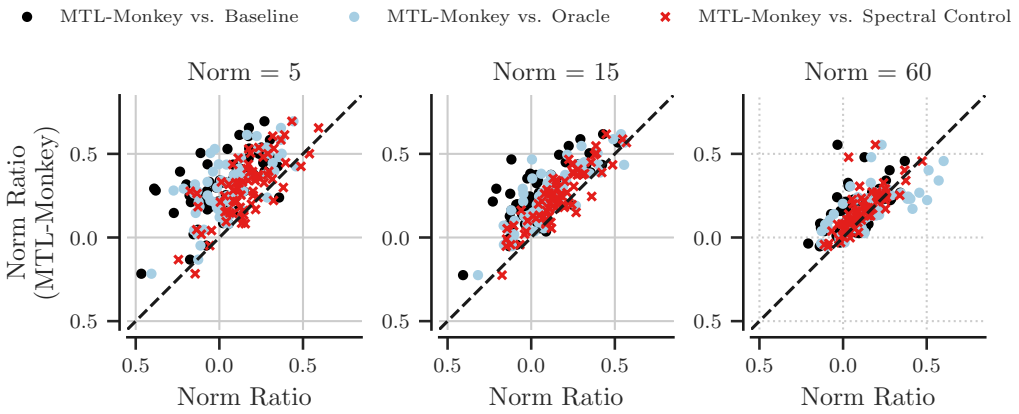

Figure 7: Each datapoint corresponds to one ImageNet image, for which we compute the normalized norm ratio (equation 1) of the reconstructed image from the MTL-Monkey model (y-axis) compared to the Oracle, Baseline or spectral control model in blue, black or red respectively (x-axis). The panels show varying norm constraints as the basis of the reconstructions. For all panels, if the datapoint is above the diagonal, it means that the salient regions in the corresponding image are more emphasized by our MTL-Monkey model than the model associated with the particular marker style.

selected all pixels up to a cumulative sum of 0.7 as "salient". Afterwards, we resized the target image and the binarized mask to 64x64 pixels, which is the standard size used for our models (see Figure 6). To quantify how much contrast is spent on the salient region salient(I) compared to the entire image I, we computed the ratio ϱ of the norm of the salient region against the full image's norm:

$$\varrho(I) = \frac{\|\text{salient}(I)\|_2^2}{\|I\|_2^2}.$$

We then computed the difference $\varrho\left(I_r\right) - \varrho\left(I_o\right)$ between the norm ratio of each reconstructed image I_r and the original image I_o. To put the values on a common scale, we normalized this difference by its maximally achievable value $1 - \varrho\left(I_o\right)$ [inspired by 12]:

$$\bar{\varrho}_r = \frac{\varrho\left(I_r\right) - \varrho\left(I_o\right)}{1 - \varrho\left(I_o\right)}. \tag{1}$$

Our results show that the MTL-Monkey is more sensitive to salient regions than the Oracle and Baseline models across images under low to mid-range norm constraints (Figure 7). And as the norm approaches the norm of the full image, the ratios for both models become more equal to those of the MTL-Monkey model, as expected (right). In comparison to the Baseline and Oracle models, the spectral control seems to be closer to the diagonal. However, in most cases the MTL-Monkey model emphasized the salient regions more strongly, supporting our previous observation that frequency filtering cannot fully explain the sensitivities of the co-trained model (see section 4). We want to stress that our analysis is purely correlational at this point: Robust MTL monkey models seem to be more sensitive to salient image content than other –robust and non-robust– models. However, if the focus on salient features turns out to be causal, it opens up the possibility that the Oracle models and the MTL-Monkey model are robust due to different reasons.

5 Related work

A number of previous studies also transferred useful inductive biases from biological to artificial neural networks. Fong et al. [20] used fMRI data of the human brain's activity to guide the training of neural networks on a classification task, obtaining general performance improvements for CNNs. Other works used neural data from animals to regularize the training of artificial networks on image classification, such as in Li et al. [21], through representational distance learning [34]. Li and colleagues used mouse V1 to regularize CNNs towards a more brain-like representation. They used the CIFAR-10 and CIFAR-100 datasets, and evaluated robustness on Gaussian noise corruptions.

Similar to our findings, they observed an increase in robustness when comparing their network to the Baseline while the model regularized by the shuffled neural data did not yield similar benefits. They also observed increased robustness to adversarial attacks. We extend their work by using a larger set of 14 distinct types of corruption to evaluate the robustness on tiny ImageNet. Federer et al. [22] used neural data from the primary visual cortex of macaque monkeys for regularization, and reported an improvement in test accuracy and an increased robustness to label corruption.

Finally, previous work, summarized by Yamins and DiCarlo [16], found a strong correlation between how well a neural network performs on a classification task and how well it predicts brain neural responses. Recent work by Dapello et al. [32] also found that robust networks tend to be better at predicting V1 responses in macaque monkeys than non-robust models. They also reported a correlation between the brain-likeness on the representational level of a network and robustness, without explicitly training on neural data. In this work, we show similar results on the representational level by directly co-training on neural data (see section 3).

6 Discussion and conclusion

In this work, we show a successful transfer of robustness properties via multi-task learning on neural data and object classification. Our findings are generally consistent with prior works on inductive bias transfer from the brain, and constitutes a first test of the neural co-training hypothesis [5]. Furthermore, we are the first, to the best of our knowledge, to go a step further by introducing a novel attribution method to understand what makes our neurally co-trained model robust. Through that analysis, we find that our MTL monkey model is more sensitive to salient regions of an image compared to other models. Since V1 has been implicated in bottom-up saliency before [24], it could be that our finding might be connected to this computational property of V1 neurons. In that respect, our results are consistent with the V1 saliency hypothesis, which has been already supported by several studies in the literature [35–38].

Although we achieve improved robustness through our MTL approach, our models are still far behind the human visual system in terms of generalization. Nevertheless, we believe that our work is an important step to bridge the gap between artificial and biological intelligence. Closing this gap could help the reliability and generalization capabilities of computer vision which becomes increasingly important as systems are used in more and more critical applications. In addition, our findings might help to get a better understanding of the computational role of V1, such as its role in bottom-up saliency. A promising future direction is to include higher brain areas for neural co-training that will presumably yield stronger robustness against more complex distortions. This exercise, combined with our analyses, could improve our understanding of the functional role of these areas as well.[2]

Our work represents fundamental research into the link between biological and artificial vision. Since this direction is at an early stage, the risk of misuse or unethical use of our results is present but not larger than in other fundamental investigations of the principles of vision. Our data is collected through animal experiments which are currently the only method to get high numbers of single unit responses across brain regions. All data coming from monkeys complied with the approved protocol of local authorities. A key advantage of the type of data we used is that it is suited for a wide range of analyses beyond this paper. Therefore, our paper simply improves the scientific yield of animal recordings.

References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. doi: 10.1038/nature14539. URL `https://doi.org/10.1038/nature14539`.

[2] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[3] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[2] Code for this work will be made publicly available upon acceptance.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[5] Fabian H Sinz, Xaq Pitkow, Jacob Reimer, Matthias Bethge, and Andreas S Tolias. Engineering a less artificial intelligence. *Neuron*, 103(6):967–979, 2019.

[6] Thomas Serre. Deep learning: the good, the bad, and the ugly. *Annual review of vision science*, 5:399–426, 2019.

[7] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

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

[10] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)*, May 2019. URL `https://openreview.net/forum?id=Bygh9j09KX`.

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

[12] R. Geirhos, K. Meding, and F. A. Wichmann. Beyond accuracy: quantifying trial-by-trial behaviour of CNNs and humans by measuring error consistency. In *Advances in Neural Information Processing Systems 33*, 2020.

[13] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

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

[15] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., New York, NY, USA, 1982. ISBN 0716715678.

[16] Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365, 2016.

[17] S. A. Cadena, G. H. Denfield, E. Y. Walker, L. A. Gatys, A. S. Tolias, M. Bethge, and A. S. Ecker. Deep convolutional models improve predictions of macaque V1 responses to natural images. *PLoS Computational Biology*, 2019. doi: 10.1101/201764.

[18] Konstantin-Klemens Lurz, Mohammad Bashiri, Konstantin Willeke, Akshay Jagadish, Eric Wang, Edgar Y. Walker, Santiago A Cadena, Taliah Muhammad, Erick Cobos, Andreas S. Tolias, Alexander S Ecker, and Fabian H. Sinz. Generalization in data-driven models of primary visual cortex. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Tp7kI90Htd`.

[19] E Y Walker, F H Sinz, E Cobos, T Muhammad, E Froudarakis, P G Fahey, A S Ecker, J Reimer, X Pitkow, and A S Tolias. Inception loops discover what excites neurons most using deep predictive models. *Nat. Neurosci.*, 2019.

[20] Ruth C. Fong, Walter J. Scheirer, and David D. Cox. Using human brain activity to guide machine learning. *Scientific Reports*, 8(1):5397, Mar 2018. doi: 10.1038/s41598-018-23618-6. URL `https://doi.org/10.1038/s41598-018-23618-6`.

[21] Zhe Li, Wieland Brendel, Edgar Walker, Erick Cobos, Taliah Muhammad, Jacob Reimer, Matthias Bethge, Fabian Sinz, Zachary Pitkow, and Andreas Tolias. Learning from brains how to regularize machines. In *Advances in Neural Information Processing Systems*, pages 9529–9539, 2019.

[22] 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.

[23] Richard A. Caruana. Multitask Learning: A Knowledge-Based Source of Inductive Bias. In *Machine Learning Proceedings 1993*. 1993. doi: 10.1016/b978-1-55860-307-3.50012-5.

[24] Li Zhaoping and Zhaoping Li. *Understanding vision: theory, models, and data*. Oxford University Press, USA, 2014.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[27] John S Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer, 1990.

[28] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[30] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. In *Machine Learning for Autonomous Driving Workshop, NeurIPS 2019*, volume 190707484, Jul 2019.

[31] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[32] Joel Dapello, Tiago Marques, Martin Schrimpf, Franziska Geiger, David Cox, and James J DiCarlo. Simulating a primary visual cortex at the front of CNNs improves robustness to image perturbations. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*, volume 33, pages 13073–13087. Curran Associates, Inc., 2020.

[33] Matthias Kummerer, Thomas S. A. Wallis, Leon A. Gatys, and Matthias Bethge. Understanding low- and high-level contributions to fixation prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[34] Patrick McClure and Nikolaus Kriegeskorte. Representational distance learning for deep neural networks. *Frontiers in computational neuroscience*, 10:131, 2016.

[35] Zhaoping Li. A saliency map in primary visual cortex. *Trends in cognitive sciences*, 6(1):9–16, 2002.

[36] Li Zhaoping and Li Zhe. Primary visual cortex as a saliency map: a parameter-free prediction and its test by behavioral data. *PLoS Comput Biol*, 11(10):e1004375, 2015.

[37] Xilin Zhang, Li Zhaoping, Tiangang Zhou, and Fang Fang. Neural activities in v1 create a bottom-up saliency map. *Neuron*, 73(1):183–192, 2012.

[38] Nobuhiko Wagatsuma, Akinori Hidaka, and Hiroshi Tamura. Correspondence between monkey visual cortices and layers of a saliency map model based on a deep convolutional neural network for representations of natural images. *Eneuro*, 8(1), 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] briefly in Discussion and conclusion
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] briefly in Discussion and conclusion
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We will provide the code as a URL in case the paper gets accepted
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] When needed
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] Citations were provided
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]