Exploring the perceptual straightness of adversarially robust and biologically-inspired visual representations

Anonymous Author(s)
Affiliation
Address
email

Abstract

Humans have been shown to use a "straightened" encoding to represent the natural visual world as it evolves in time (Hénaff et al. 2019). In the context of discrete video sequences, "straightened" means that changes between frames follow a more linear path in representation space at progressively deeper levels of processing. While deep convolutional networks are often proposed as models of human visual processing, many do not straighten natural videos. In this paper, we explore the relationship between robustness, biologically-inspired filtering mechanisms, and representational straightness in neural networks in response to time-varying input, and identify curvature as a useful way of evaluating neural network representations. We find that (1) adversarial training leads to straighter representations in both CNN and transformer-based architectures and (2) biologically-inspired elements increase straightness in the early stages of a network, but do not guarantee increased straightness in downstream layers of CNNs. Our results suggest that constraints like adversarial robustness bring computer vision models closer to human vision, but when incorporating biological mechanisms such as V1 filtering, additional modifications are needed to more fully align human and machine representations.

1 Intro

Visual input from the natural world evolves over time, and we can think of that change over time as a trajectory in some representation space. This trajectory changes at different levels of processing from input at the retina to brain regions such as V1, and finally to perception as figure 1 illustrates. We can ask about the shape of that trajectory and consider that there might be advantages to a straighter, less curved, trajectory. Hénaff et al. [18] observed that trajectories are straighter in human perceptual space than in pixel space. They suggested that a straighter representation may be useful for visual tasks that require predicting the future.

Learning a useful visual representation is also a goal in computer vision. Properties like robustness to transformations and task flexibility that characterize human vision are often desirable in computer vision representations. Yet, many existing computer vision models still fail to capture aspects of human vision, despite achieving high accuracy on visual tasks like recognition [15, 18]. In Hénaff et al. [18] it was found that, while biologically-inspired V1-like transformations yield straighter representations compared to the input domain, popular computer vision models such as ImageNet-trained AlexNet do not.

In this paper, we explore what makes some learned visual representations straighter than others. We evaluate how training for adversarial robustness in both CNN and transformer-based architectures may lead to the straight representations generated by human vision. Because DNNs learn an early representation that differs from what's known about human vision, we also ask if hard-coding that

Submitted to 4th Workshop on Shared Visual Representations in Human and Machine Visual Intelligence (SVRHM) at NeurIPS 2022. Do not distribute.

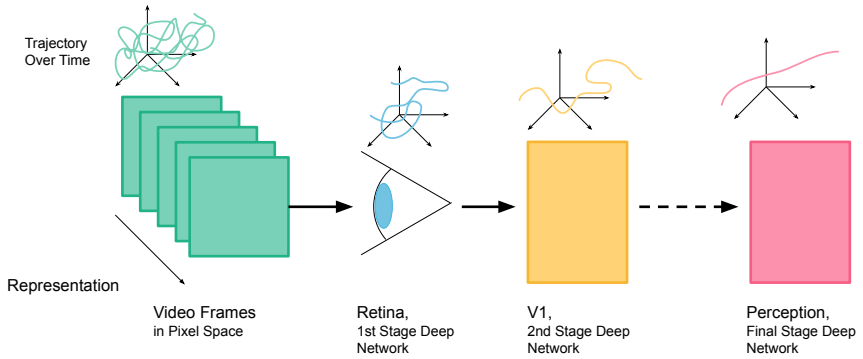

Figure 1: Illustration of the representation of a discrete video sequence becoming progressively straighter as information is processed from pixel space through a visual processing pipeline.

early representation might lead to a trained network with more straightening. Overall, we find that deep learning models are capable of relying on a straightened representation which may be useful for learning more robust and stable image and video processing systems.

2 Previous Work

Deep Neural Networks (DNNs) have been proposed as models of human visual processing, owing to their ability to predict neural response patterns [33, 26, 21]. Comparing human and DNN perception [2, 15, 28] has shown that adversarial examples are an important area in understanding how humans and DNNs differ [12, 20, 16, 8]. With adversarial examples, changes that are imperceptible to humans can cause a network to misclassify an image. Adversarial training [23] improves the misclassfication problem and has been suggested to help networks learn visual representations that are more perceptually aligned with humans. [14, 20]. Given adversarial training schemes are not biologically plausible however, recent work has identified mechanisms that are better supported by vision science [7, 17, 8].

Adversarially robust models have also been shown to do better at transfer learning than their non-adversarially robust counterparts [9], and adversarially robust features can be used directly for tasks like image generation and in-painting [27]. The potential for adversarial training to improve learned representations has led to new adversarial training frameworks that extend to flow-based generative models [25] and semantic segmentation models [32]. In this paper, we build on work around adversarial robustness by evaluating if increasing this robustness leads to straightened representations like those found in human spatiotemporal processing.

3 Methods

Representational straightness can be evaluated as a reduction in curvature. For a temporal sequence, such as a video, curvature is defined as the angle between the vectors representing the *difference* between consecutive frames. Let $\mathbf{x}$ refer to a representation of a video of length T, with x_t being a representation of one frame of a video at time step t. The representation may be at any stage of the processing pipeline, from a vector of raw input pixels from the video frame, to the activations of a network's hidden layer. Then v_t represents the difference between successive frames:

$$v_t = x_t - x_{t-1} \tag{1}$$

$$\hat{v}_t = \frac{v_t}{\|v_t\|} \tag{2}$$

We can find the curvature at time t by finding the angle between successive $\hat{v}_t$, which we call c_t:

$$c_t = \arccos\left(\hat{v}_t \cdot \hat{v}_{t+1}\right) \tag{3}$$

The global curvature of a video sequence is then simply the mean angle over all time steps:

$$\text{Global curvature} := \sum_{t=0}^{T-1} \frac{c_t}{T-1} \tag{4}$$

This is the formulation proposed by Hénaff et al. [18]. One can compute this global curvature for any representation of a video sequence over time, either on the vector of pixels (likely not very straight), or one can apply it to a representation of that video, e.g. at any layer of a neural network model.

4 Effect of Architecture and Training Scheme on Straightness

We tested a variety of models for output curvature, to investigate the relationship between model type and curvature of the output layer. As shown in Fig 2, we found non-adversarially trained image models to have the highest output curvature. All adversarially trained models have lower curvature than their non-adversarially trained counterparts, as well as overall. We also tested a family of non-parametric biological network models: Brucenet [4], based on summary statistics of a steerable pyramid, and Henaffbio [18], a two-stage model based on center-surround filters followed by oriented Gabor filters. These biological models showed the lowest output curvatures. Self-supervised DINO [5] models have similar output curvature values to their supervised counterparts – despite DINO models having been shown to have more semantically meaningful feature correspondences. In addition, we investigated a next-frame prediction model, PredNet [22]. As PredNet is a predictive coding model for frames in pixel space, it produces output predictions with similar curvature to its video input.

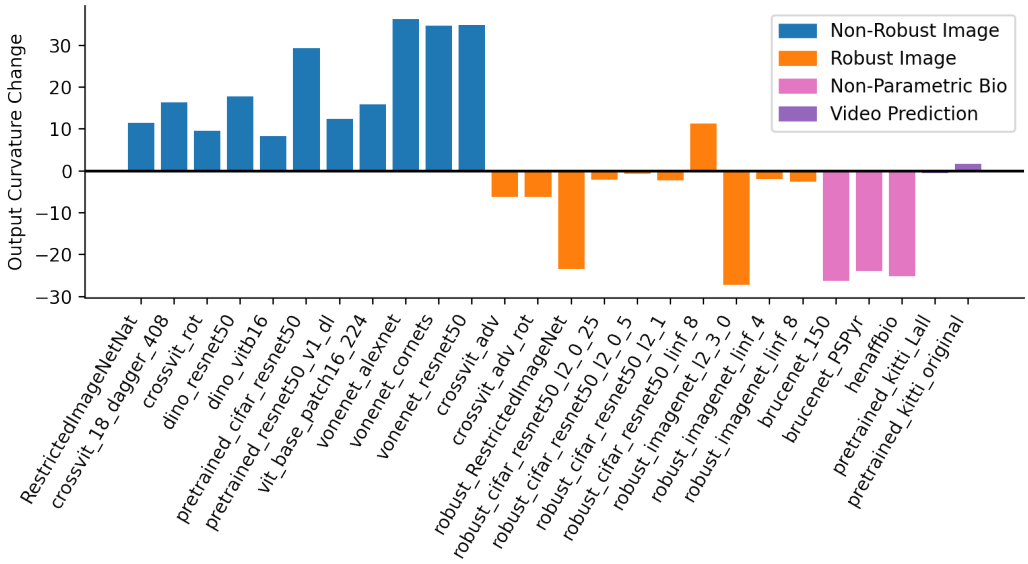

Figure 2: **Final output layer curvature for a variety of network architecture and training schemes.** Non-adversarially trained image models increase curvature. Most adversarially trained image models reduce curvature, resulting in a more straightened output. Non-parametric biological models, without learned filters, produce highly straightened output representations. Video models trained on next-frame prediction show output curvatures very similar to that of input pixels.

5 Adversarial Attacks in Image Models

Given the increased straightness seen in adversarially robust models, we investigated the relationship between the type and strength of adversarial attack and the resulting curvature of the model's output. To evaluate the effect of these attacks on curvature, we compared a set of ResNet50 networks, trained on CIFAR-10, ImageNet [10], and Restricted ImageNet (a subset of ImageNet [14, 20]), trained both without adversarial attacks, as well as trained on both l_2 and l_∞ norms at a range of attack

strengths [23, 13], and measured the output curvature of each resulting model (Figure 3). We find that output curvature is unsurprisingly highest for non-adversarially trained networks. l_∞ attacks decrease output curvature, with larger values of ϵ leading to decreased curvature. l_2 attacked networks have the straightest output curvatures, however strength of attack does not affect the output curvature.

Differences between the l_2 and l_∞ norm may be the reason that we found l_2 norm training to lead to more straight representations. The l_∞ norm takes the maximum entry of a vector whereas the l_2 norm is the square root of the sum of squares and is informed by the all components of a vector. Thus, the l_2 norm may allow updates in training to affect more weights and change the over the whole representation space to produce a more straight and stable representation over time.

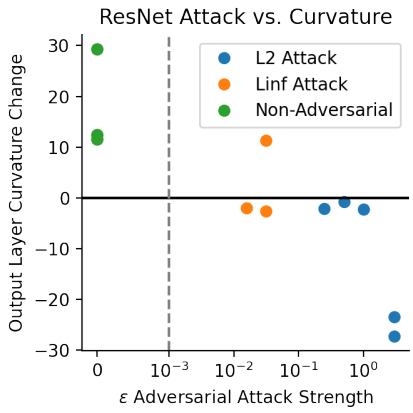

Figure 3: White-box adversarial attacks reduce curvature in the output layer of ResNet. Increased attack strength (ϵ) decreases curvature, with l_2 attacks most reducing curvature. Data on symlog x-scale to show $\epsilon = 0$: data to left of line on linear scale, and right of line on log scale.

6 Curvature Across Model Layers

We next investigated the evolution of layer curvature of various deep network models over major model blocks from input to output (Figure 4). As in Figure 2, we again see that non-adversarially trained models increase curvature in the representation, while adversarially robust models show a straightened output. For most models, this trend is consistent throughout all layers of the network. For ResNet50 however, the adversarially trained model begins by curving the output in earlier layers, then reducing curvature strongly. Conversely, in the non-adversarially trained CrossVIT network, the curvature is reduced in middle layers, then highly curved in later layers.

We also tested the layer-evolution of curvature for PredNet (Fig 4), a network inspired by predictive coding trained to predict the next frame in a video sequence. Since PredNet's trained to predict a future video frame, it's input and output domains are both in pixel space, so it is unsurprising that the output layer maintains a very similar curvature value to that of its input frame. Interestingly however, the model's curvature strongly increases in the representation of the first model block, then re-straightens the representation throughout the rest of the network before returning to its starting curvature.

7 Biologically Inspired Models

We investigated straightness for a variety of both parametric and non-parametric biologically-inspired models. Given straightness is thought to increase over progressively deeper layers of visual processing,

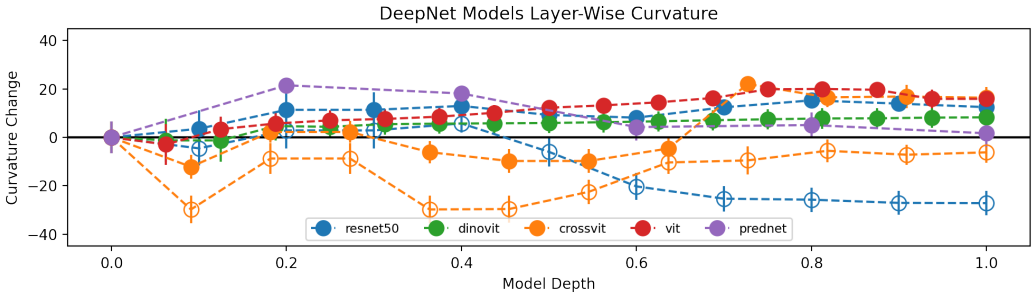

Figure 4: **Deep network models vary in curvature for each layer over model depth.** Curvature is shown for a variety of model types. Filled circles indicate non-adversarially trained models, and open circles indicate adversarially-trained robust models.

we aligned these networks along the visual processing areas they are most closely matched to (Figure 5). For the non-parametric Henaffbio [19] and Brucenet [4] models, as well as VOneNetCornets [7] these are implicit in the design of the network. For the adversarially trained Visual Transformer network CrossVITRotAdv [3], these layers are those best matched by BrainScore for V4 and IT layers [28]. For all biologically-inspired models except for VOneNetCornets, curvature progressively decreases through deeper network layers. For VOneNetCornets, curvature decreases up until the V1 layer in which a noise term is added; curvature then strongly increases, far above the pixel-curvature baseline. To determine if this increase in curvature was due to the added noise, we tested the same model with the noise term set to zero. While this reduced the downstream curvature after the V1 layer, this change did not eliminate the curvature increase present in the V1 layer.

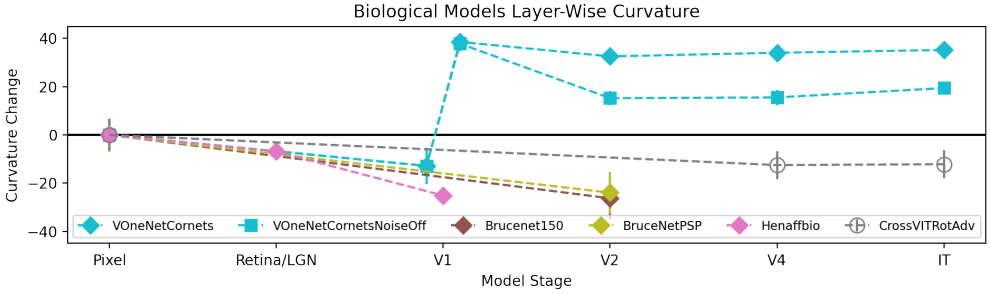

Figure 5: **Biologically-inspired network models result in straightened representations compared to input pixels.** Non-parametric multi-stage filter models Brucenet and Henaffbio progressively straighten. The VOneNetCornets model straightens the representation in its V1-block up until its noise layer, where curvature increases dramatically and does not recover in downstream convolutional blocks. Adversarially-trained CrossVIT model shows progressive straightening.

The increase in curvature for VOneNet at later layers suggest that making the front-end of deep network more like biologically-inspired models is not enough to get a straightened representation downstream in a deep network. This is interesting because VOneNet is is reported to be more adversarially robust to white-box attacks than a standard trained ResNet. It suggests that adversarial *training*, not the property of adversarial robustness itself, leads to straightened representations in deep neural networks. Our finding supports Dapello et al. [8] who found that the neural population geometry of adversarial trained models was noticeable differences from VOneNet and other models trained with neural stochasticity mechanisms. However, it is puzzling that adversarial training, which is biologically implausible, would lead to more straight representations than biologically inspired mechanisms. More constraints or modifications may be need to get straight representations with biologically plausible methods.

8 Discussion

In conclusion, we show change in model representational curvature to be a simple and computationally cheap metric for evaluating both image and video models across a variety of tasks. We show that for a variety of image classification models, output curvature is reduced when models are trained with strong white-box adversarial attacks. This property of straightness over time may lead to more stable predictions over both input space and for temporal sequences. Although Hénaff et al. [18] found that ImageNet-trained DNNs did not yield perceptually straight representations when tested on videos, we find that this is not a limitation of the model but rather of the training procedure. Moreover, we find evidence to suggest that a model's ability to straighten input stimuli may be a useful and easily computed measure of its ability to produce similar visual representations to humans. In evaluating curvature over layers in biologically inspired models, we show that biologically inspired mechanisms work to reduce curvature in a model's representation, even more so than adversarial training. However, the simple addition of non-parametric biologically inspired filtering mechanisms at the input of a model are insufficient to maintain output curvature. These results identify representational curvature as a common thread between biologically inspired and adversarially robust models, and highlight the benefits and limitations of these techniques in creating temporally-stable representations.

References

[1] R. C. C. at University of Chicago. Chicago motion database, 2022. URL `https://cmd.rcc.uchicago.edu`.

[2] A. Berardino, V. Laparra, J. Ballé, and E. Simoncelli. Eigen-distortions of hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

[3] W. Berrios and A. Deza. Joint rotational invariance and adversarial training of a dual-stream transformer yields state of the art brain-score for area v4. *arXiv preprint arXiv:2203.06649*, 2022.

[4] R. Brown, V. DuTell, B. Walter, R. Rosenholtz, P. Shirley, M. McGuire, and D. Luebke. Efficient dataflow modeling of peripheral encoding in the human visual system. *arXiv preprint arXiv:2107.11505*, 2021.

[5] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[6] C.-F. R. Chen, Q. Fan, and R. Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021.

[7] J. Dapello, T. Marques, M. Schrimpf, F. Geiger, D. Cox, and J. J. DiCarlo. Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations. *Advances in Neural Information Processing Systems*, 33:13073–13087, 2020.

[8] J. Dapello, J. Feather, H. Le, T. Marques, D. Cox, J. McDermott, J. J. DiCarlo, and S. Chung. Neural population geometry reveals the role of stochasticity in robust perception. *Advances in Neural Information Processing Systems*, 34:15595–15607, 2021.

[9] T. Davchev, T. Korres, S. Fotiadis, N. Antonopoulos, and S. Ramamoorthy. An empirical evaluation of adversarial robustness under transfer learning. *arXiv preprint arXiv:1905.02675*, 2019.

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

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

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

[13] L. Engstrom, A. Ilyas, H. Salman, S. Santurkar, and D. Tsipras. Robustness (python library), 2019. URL `https://github.com/MadryLab/robustness`.

[14] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, B. Tran, and A. Madry. Learning perceptually-aligned representations via adversarial robustness. In *ArXiv preprint arXiv:1906.00945*, 2019.

[15] J. Feather, A. Durango, R. Gonzalez, and J. McDermott. Metamers of neural networks reveal divergence from human perceptual systems. *Advances in Neural Information Processing Systems*, 32, 2019.

[16] J. Feather, G. Leclerc, A. Mądry, and J. H. McDermott. Model metamers illuminate divergences between biological and artificial neural networks. *bioRxiv*, 2022.

[17] C. Guo, M. J. Lee, G. Leclerc, J. Dapello, Y. Rao, A. Madry, and J. J. DiCarlo. Adversarially trained neural representations may already be as robust as corresponding biological neural representations. *arXiv preprint arXiv:2206.11228*, 2022.

[18] O. J. Hénaff, R. L. Goris, and E. P. Simoncelli. Perceptual straightening of natural videos. *Nature neuroscience*, 22(6):984–991, 2019.

[19] O. J. Hénaff, Y. Bai, J. A. Charlton, I. Nauhaus, E. P. Simoncelli, and R. L. Goris. Primary visual cortex straightens natural video trajectories. *Nature communications*, 12(1):1–12, 2021.

[20] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

[21] A. J. Kell and J. H. McDermott. Deep neural network models of sensory systems: windows onto the role of task constraints. *Current opinion in neurobiology*, 55:121–132, 2019.

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

[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[25] P. Pope, Y. Balaji, and S. Feizi. Adversarial robustness of flow-based generative models. In *International Conference on Artificial Intelligence and Statistics*, pages 3795–3805. PMLR, 2020.

[26] R. Rajalingham, K. Schmidt, and J. J. DiCarlo. Comparison of object recognition behavior in human and monkey. *Journal of Neuroscience*, 35(35):12127–12136, 2015.

[27] S. Santurkar, D. Tsipras, B. Tran, A. Ilyas, L. Engstrom, and A. Madry. Computer vision with a single (robust) classifier. In *ArXiv preprint arXiv:1906.09453*, 2019.

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

[29] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack. Study of subjective and objective quality assessment of video. *IEEE transactions on Image Processing*, 19(6): 1427–1441, 2010.

[30] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack. A subjective study to evaluate video quality assessment algorithms. In *Human Vision and Electronic Imaging XV*, volume 7527, pages 128–137. SPIE, 2010.

[31] R. Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[32] X. Xu, H. Zhao, and J. Jia. Dynamic divide-and-conquer adversarial training for robust semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7486–7495, 2021.

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

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

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] See Appendix C.2
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix C.4
   (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)? [Yes] See Supplemental
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (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)? [Yes] See Appendix
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? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (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]

A Appendix

B Curvature Definition

Note that curvature is distinct from simple cosine similarity in that curvature is calculated on frame differences (v_t), whereas cosine similarity depends on the angle between the frame vectors themselves (x_t). Curvature can be thought of as a first order variant cosine similarity.

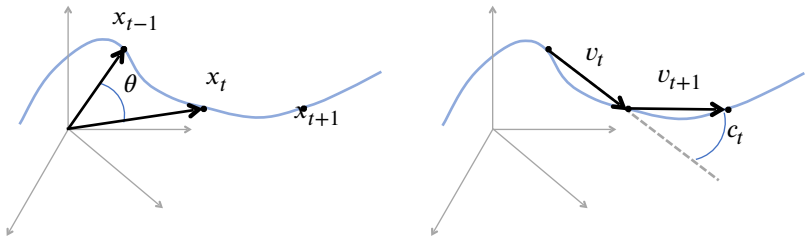

Figure 6: Illustration of how the curvature measure is distinct from cosine similarity. (Left) Three points are sampled along a trajectory in time (x_{t-1}, x_t, x_{t+1}). The angle θ between neighboring x samples is their cosine similarity. (Right) Curvature c_t is the angle between v_1 and v_{t+1}. v_t is the difference between x_t and x_{t-1}

$$\text{cosine similarity (vectors)} := \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \tag{5}$$

$$\text{cosine similarity (video frames)} := \cos(\theta) = \frac{x_t \cdot x_{t+1}}{\|x_t\|\|x_{t+1}\|} \tag{6}$$

$$\text{curvature} := c_t = \arccos(\hat{v}_t \cdot \hat{v}_{t+1}) \tag{7}$$

$$\text{cosine curvature} := \hat{v}_t \cdot \hat{v}_{t+1} = \frac{v_t \cdot v_{t+1}}{\|v_t\|\|v_{t+1}\|} = \cos(c_t) \tag{8}$$

C Models

C.1 Model Sources

All Deep neural networks we analyzed were pretrained. The standard-trained ImageNet model was downloaded from PyTorch's model zoo [24]. Adversarially robust ResNet models were all downloaded from [13]. The adversarially robust ResNets were trained using projected gradient descent. All ViT [11] and standard trained CrossViT daggar [6] models were downloaded from the timm library [31]. All CrossViT daggar adversarially robust and rotationally invariant models were downloaded from the repository of Berrios and Deza [3]. The adversarially robust CrossViTs were trained with fast gradient sign method as stated in Berrios and Deza [3]. DINO models were downloaded from the DINO repository [5], while PredNet models were downloaded from Lotter et al. [22].

C.2 Model Analysis Procedure

We showed each model the same 12 natural videos that were used in the psychophysics experiments of [18]. The videos were taken from the Chicago Motion Database [1], the film 'Dogville', Lions Gate Entertainment (2003), and LIVE Video Quality Database [29, 30]. The videos were grayscale, consisting of 11 frames each of 512×512 pixels, capturing natural motion such as rippling ocean water or a person walking through a crowded street. We resized the video frames to be 224×224 for all deep networks and 256×256 for bio-models that use steerable pyramids. One limitation of this work is that we did not evaluate models on larger video datasets, but we wanted to use psychophysical validated stimuli for our analyses. For each model, we recorded its activations at intermediate and

final layers for each video. We then found the global curvature for each stage of the model using equation 4 where we used the flattened model activations as the input x_t to the curvature procedure.

We compared the global curvature at each layer of the model to the curvature of the video in pixel space. Models that straighten are defined as models that have a lower global curvature at deeper layers. When comparing the curvature of different model layers, we chose not to reduce the dimensionality of each layer activation to be the same across stages. Although principle components analysis (PCA) was sometimes used in Hénaff et al. [18] when expressing curvature, they did not use it in their analysis of deep networks. Furthermore, while an architecture's inherent dimensionality is likely relevant to a representation's curvature, we preferred not to introduce any additional transformations that would influence the measured curvature.

C.3 Compute

Our methods do not require large compute. All individual model analyses can be run on CPU. We used a single GPU to speed up getting the features activations at each layer to the order of minutes per model.

C.4 Adversarial Accuracy and Curvature

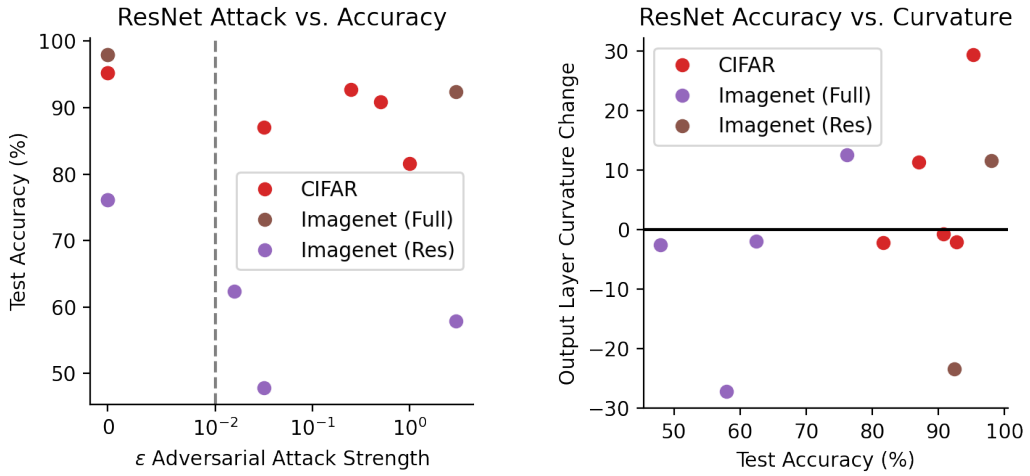

Figure 7: (Left): While adversarial attacks with greater strength impart many desirable robustness properties on a network, adversarial training does not improve test accuracy, often decreasing test accuracy on the within-domain test set for a given model. Data plotted on symlog scale. (Right): While stronger adversarial attacks decrease curvature, improved test accuracy for a model is not predictive of output curvature reduction. Rather, within a given model training/test set, increased test accuracy predicts a smaller curvature reduction in the output layer.

C.5 Negative Societal Impacts

We believe there are few negative societal impacts of this paper. Our work was exploratory and did not introduce any new models. The only negative impacts may be the general loss of jobs and industries that may result from artificial intelligence replacing human workers