CURVED REPRESENTATION SPACE OF VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

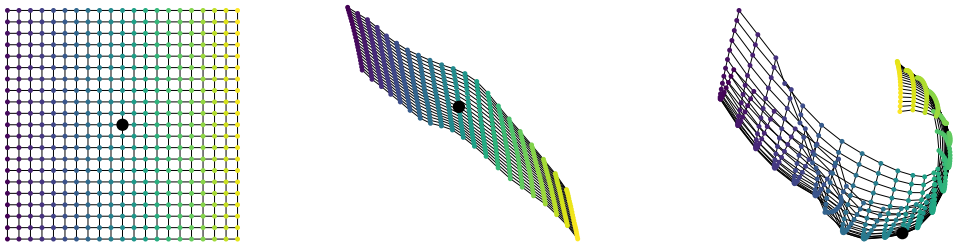
Neural networks with self-attention (a.k.a. Transformers) like ViT and Swin have emerged as a better alternative to traditional convolutional neural networks (CNNs) for computer vision tasks. However, our understanding of how the new architecture works is still limited. In this paper, we focus on the phenomenon that Transformers show higher robustness against corruptions than CNNs, while not being overconfident (in fact, we find Transformers are actually underconfident). This is contrary to the intuition that robustness increases with confidence. We resolve this contradiction by investigating how the output of the penultimate layer moves in the representation space as the input data moves within a small area. In particular, we show the following. (1) While CNNs exhibit fairly linear relationship between the input and output movements, Transformers show nonlinear relationship for some data. For those data, the output of Transformers moves in a curved trajectory as the input moves linearly. (2) When a data is located in a curved region, it is hard to move it out of the decision region since the output moves along a curved trajectory instead of a straight line to the decision boundary, resulting in high robustness of Transformers. (3) If a data is slightly modified to jump out of the curved region, the movements afterwards become linear and the output goes to the decision boundary directly. Thus, Transformers can be attacked easily after a small random jump and the perturbation in the final attacked data remains imperceptible. In other words, there does exist a decision boundary near the data, which is hard to find only because of the curved representation space. This also explains the underconfident prediction of Transformers. (4) The curved regions in the representation space start to form at an early training stage and grow throughout the training course. Some data are trapped in the regions, obstructing Transformers from reducing the training loss.

1 INTRODUCTION

Self-attention-based neural network architectures, including Vision Transformers (Dosovitskiy et al., 2021), Swin Transformers (Liu et al., 2021), etc. (hereinafter referred to as Transformers), have shown to outperform traditional convolutional neural networks (CNNs) in various computer vision tasks. The success of the new architecture has prompted a question, how Transformers work, especially compared to CNNs, which would also shed light on the deeper understating of CNNs and eventually neural networks.

In addition to the improved task performance (e.g., classification accuracy) compared to CNNs, Transformers also show desirable characteristics in other aspects. It has been shown that Transformers are more robust to adversarial perturbations than CNNs (Bai et al., 2021; Naseer et al., 2021; Paul & Chen, 2022). In addition, Transformers are reported not overconfident in their predictions unlike CNNs (Minderer et al., 2021) (and we show that Transformers are actually underconfident in this paper).

The high robustness, however, does not comport with underconfidence. Intuitively, a data that is correctly classified by a model with lower confidence is likely to be located closer to the decision boundary. Then, a smaller amount of perturbation would move the data out of the decision region, which translates into lower robustness of the model. However, the previous results claim the opposite.



(a) Input space (b) Representation space (ResNet50) (c) Representation space (Swin-T)

Figure 1: 2D projected movements of (a) the data (black dot) in the input space and corresponding output features in the representation space for (b) ResNet50 and (c) Swin-T.

To mitigate the contradiction of the robustness and the underconfidence, we explore the representation space of Transformers and CNNs. We examine the input-output relationship of the models, i.e., how the output feature of the penultimate layer (which is simply referred to as output in this paper) changes with respect to the linear change of the input data. We find that for certain data, Transformers move the output along a nonlinear trajectory within a certain range, whereas CNNs show fairly linear movements of the output. In other words, the representation space of Transformers is *curved*. We show that this curved representation space results in the aforementioned contradiction.

Fig. 1 visualizes the representation spaces of CNNs and Transformers comparatively (see Appendix A for implementation details). An image data from ImageNet (Russakovsky et al., 2015), marked with the black dot in Fig. 1a, is gradually modified by a fixed amount along two mutually orthogonal directions. The corresponding outputs of ResNet50 and Swin-T are obtained, which are shown after two-dimensional projection in Figs. 1b and 1c, respectively. While the gradual changes of the input produce almost linear changes in the output of ResNet50, the output trajectory of Swin-T is curved around the original output (and then becomes linear when the change of the output is large).

Our research questions and findings can be summarized as follows.
1. *How does the representation space of Transformers look like?* To answer this, we analyze the output movement of the penultimate layer with respect to the linear movement of the input. We find that for some input data, the directions of successive movements of the output significantly change in the case of Transformers unlike CNNs, indicating that **the representation space of Transformers is locally curved**.
2. *What makes Transformers robust to input perturbation?* **We find that the curved regions in the representation space account for the robustness of Transformers**. When a data is located in a curved region, the movement of the output by the perturbation of the input data follows a curved trajectory. This makes it hard to move the data out of its decision region with a small amount of perturbation, which explains high robustness of Transformers for the data.
3. *Then, why is the prediction of Transformers underconfident?* Although it takes many steps to escape from a curved decision region and reach a decision boundary, we find that a decision boundary is actually located closely to the original output. We demonstrate a simple trick to reach the decision boundary quickly. I.e., when a small amount of random noise is added to the input data, its output can jump out of the locally curved region and arrive at a linear region, from which a closely located decision boundary can be reached by adding only a small amount of perturbation. This reveals that **the decision boundary exits near the original data in the representation space, which explains the underconfident predictions of Transformers**.
4. *When are the curved regions created and how do they influence the training process?* To answer this, we examine the curvedness of the representation space during training. Interestingly, curved regions already start to form at an early training stage. This causes difficulty of training of Transformers, because **once a training data is trapped in a curved region, it hardly escapes the region and its training loss is hardly reduced for the rest of the training process**. This explains larger training loss values observed in training of Transformers in comparison to those of CNNs.

2 RELATED WORK

Since the first application of the self-attention mechanism to vision tasks (Dosovitskiy et al., 2021), a number of studies have shown that the models built with traditional convolutional layers are outperformed by Transformers utilizing self-attention layers in terms of task performance (Liu et al., 2021; Chu et al., 2021; Huang et al., 2021; Li et al., 2021; Touvron et al., 2021; Wang et al., 2021; Xiao et al., 2021; Yang et al., 2021; Yuan et al., 2021; Liu et al., 2022). There have been efforts to compare CNNs and Transformers in various aspects. Empirical studies show that Transformers have higher adversarial robustness than CNNs (Paul & Chen, 2022; Naseer et al., 2021; Aldahdooh et al., 2021; Bhojanapalli et al., 2021), which seems to be due to the reliance of Transformers on lower frequency information than CNNs (Park & Kim, 2022; Benz et al., 2021). Minderer et al. (2021) conclude that Transformers are calibrated better than CNNs yielding overconfident predictions (Guo et al., 2017; Thulasidasan et al., 2019; Wen et al., 2021). However, there has been no clear explanation encompassing both higher robustness and lower confidence of Transformers.

Understanding how neural networks work has been an important research topic. A useful way for this is to investigate the input-output mapping formed by a model. Since models with piecewise linear activation functions (e.g., ReLU) implement piecewise linear mappings, several studies investigate the characteristics of linear regions, e.g., counting the number of linear regions as a measure of model expressivity (or complexity) (Montufar et al., 2014; Hanin & Rolnick, 2019a;b; Telgarsky, 2015; Serra et al., 2018; Raghu et al., 2017) and examining local properties of linear regions (Zhang & Wu, 2020). Some studies examine the length of the output curve for a given unit-length input (Raghu et al., 2017; Price & Tanner, 2021; Hanin et al., 2022). There also exist some works that relate the norm of the input-output Jacobian matrix to generalization performance (Sokolić et al., 2017; Novak et al., 2018). However, the input-output relationship of Transformers has not been explored previously, which is focused in this paper.

3 ON THE OSTENSIBLE CONTRADICTION OF HIGH ROBUSTNESS AND UNDERCONFIDENCE

3.1 MODEL CALIBRATION

It is desirable that a trained classifier is well-calibrated by making prediction with reasonable certainty, e.g., for data that a classifier predicts with confidence of 80%, its accuracy should also be 80% in average. A common measure to evaluate model calibration is the expected calibration error (ECE) defined as (Naeini et al., 2015)

$$\mathrm{ECE} = \sum_{i=1}^{K} P(i) \cdot |o_i - e_i|, \tag{1}$$

where K is the number of bins of confidence, $P(i)$ is the fraction of data falling into bin i, o_i is the accuracy of the data in bin i, and e_i is the average confidence of the data in bin i. A limitation of ECE is that it does not distinguish between overconfidence and underconfidence because the sign of the difference between the accuracy and the confidence is ignored. Therefore, we define a modified version of ECE, called signed ECE (sECE), as follows.

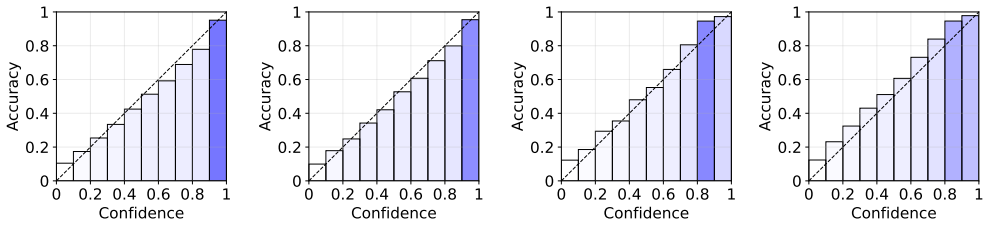
$$\mathrm{sECE} = \sum_{i=1}^{K} P(i) \cdot (o_i - e_i). \tag{2}$$

An overconfident model will have higher confidence than accuracy, resulting in a negative sECE value. An underconfident model, in contrast, will show a positive value of sECE.

We compare the calibration of CNNs, including ResNets (He et al., 2016), DenseNets (Huang et al., 2017), MobileNets (Sandler et al., 2018; Howard et al., 2019), and Transformers, including ViT (Dosovitskiy et al., 2021) and Swin (Liu et al., 2021), on the ImageNet validation set using ECE and sECE in Table 1 and Fig. 2. CNNs show negative ECE values in Table 1 and bar plots below the 45° line in Fig. 2, indicating overconfidence in prediction, which is consistent with the previous studies (Guo et al., 2017). On the other hand, Transformers are underconfident, showing positive sECE and bar plots over the 45° line. This comparison result is interesting: Transformers reportedly show higher classification accuracy than CNNs, but in fact with lower confidence.

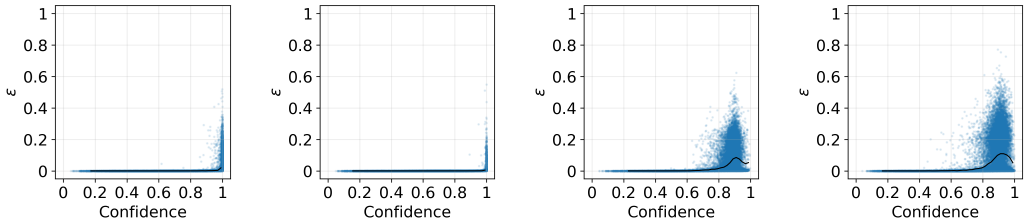
Table 1: ECE and sECE of CNNs (left side) and Transformers (right side) with $K = 10$.

Models	ECE	sECE	Models	ECE	sECE
ResNet50	3.67%	-3.57%	ViT-B/16	5.54%	+5.54%
ResNet101	4.91%	-4.89%	ViT-B/32	6.35%	+6.35%
ResNet152	5.01%	-5.00%	ViT-L/16	4.93%	+0.92%
DenseNet121	2.50%	-2.30%	ViT-L/32	4.15%	+2.74%
DenseNet169	5.50%	-5.49%	Swin-T	7.08%	+7.08%
MobileNetV2	2.80%	-2.63%	Swin-S	4.34%	+4.26%
MobileNetV3	2.45%	-2.42%	Swin-B	4.88%	+4.23%



(a) ResNet50 (b) MobileNetV2 (c) ViT-B/16 (d) Swin-T

Figure 2: Reliability diagrams of CNNs and Transformers. Transparency of bars represent the number of data in each confidence bin. Results for other models can be found in Appendix B.



(a) ResNet50 (b) MobileNetV2 (c) ViT-B/16 (d) Swin-T

Figure 3: Lengths (ϵ) of the travel to decision boundaries for the ImageNet validation data. Black lines represent average values. Results for other models can be found in Appendix C.

3.2 Passage to Decision Boundary

It is a common intuition that if a model classifies a data with low confidence, the data is likely to be located near a decision boundary. Based on the above results, therefore, the decision boundaries of Transformers are assumed to be formed near the data compared to CNNs. To validate this, we formulate a procedure to examine the distance to a decision boundary from a data through a linear travel. Concretely, we aim to solve the following optimization problem.

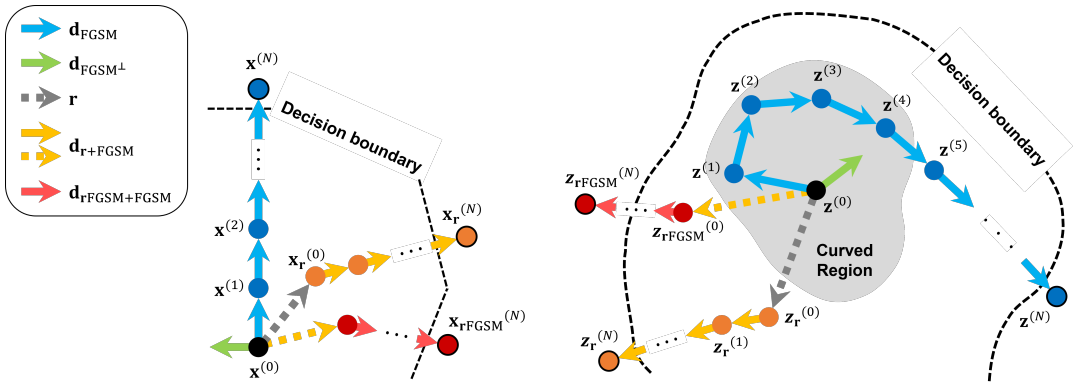
$$\arg\min_{\epsilon} \mathcal{C}(\mathbf{x}') \neq y, \qquad \mathbf{x}' = \mathbf{x} + \epsilon \cdot \mathbf{d}, \tag{3}$$

where $\mathbf{x}$ is an input data, y is the true class label of $\mathbf{x}$, $\mathcal{C}$ is the classifier, $\mathbf{d}$ is the travel direction, ϵ is a positive real number, and $\mathbf{x}'$ is the traveled result of $\mathbf{x}$. Here, we want to set the direction $\mathbf{d}$ in such a way that the travel reaches to a closely located decision boundary and ϵ becomes as small as possible. For this, we adopt the idea of the fast gradient sign method (FGSM) (Goodfellow et al., 2015) used for adversarial attack. We find the direction to which the classification loss function J (i.e., cross-entropy) increases, i.e.,

$$\mathbf{d} = \text{sign}(\nabla_{\mathbf{x}} J(\mathcal{C}(\mathbf{x}), y)). \tag{4}$$

Note that $\|\mathbf{d}\|_2 = \sqrt{D}$, where D is the dimension of $\mathbf{x}$. Refer to Alg. 1 in Appendix C for the detailed procedure to solve the optimization problem in Eq. 3.

Fig. 3 shows the obtained values of ϵ with respect to the confidence values for the ImageNet validation data. It is observed that decision boundaries are farther from the data in the input space for



(a) Input space (b) Representation space

Figure 4: Illustration of the input-output relationship of Transformers in terms of the trajectories in the input space and the representation space.

Transformers than CNNs, which is the opposite to our expectation. This contradiction is resolved in the following section.

4 Resolving the Contradiction

4.1 Shape of Representation Space

As an attempt to resolve the contradiction, we examine the input-output relationship of the models, i.e., how linear movements in the input space appear in the representation space of a model.

We divide the travel to the decision boundary into N steps as

$$\mathbf{x}^{(n)} = \mathbf{x}^{(0)} + n \cdot \frac{\epsilon}{N}\mathbf{d}_{\mathbf{x}}, \quad (n = 0, 1, \cdots, N) \tag{5}$$

where $\mathbf{x}^{(0)} = \mathbf{x}$ and $\mathbf{x}^{(N)}$ are the initial and final data points, respectively. For each $\mathbf{x}^{(n)}$, we obtain its output feature at the penultimate layer, which is denoted as $\mathbf{z}^{(n)}$. Unlike the travel in the input space, the magnitude and direction of the travel appearing in the representation space may change at each step. Thus, the movement at step n is defined as

$$\mathbf{d}_{\mathbf{z}}^{(n)} = \mathbf{z}^{(n)} - \mathbf{z}^{(n-1)} \tag{6}$$

from which the magnitude ($\omega^{(n)}$) and relative direction ($\theta^{(n)}$) are obtained as

$$\omega^{(n)} = \|\mathbf{d}_{\mathbf{z}}^{(n)}\|, \qquad \theta^{(n)} = \cos^{-1}\left(\frac{\mathbf{d}_{\mathbf{z}}^{(n)} \cdot \mathbf{d}_{\mathbf{z}}^{(n+1)}}{\|\mathbf{d}_{\mathbf{z}}^{(n)}\| \cdot \|\mathbf{d}_{\mathbf{z}}^{(n+1)}\|}\right). \tag{7}$$

We consider four different ways to determine $\mathbf{d}_{\mathbf{x}}$:

- $\mathbf{d}_{\text{FGSM}}$ (blue-colored trajectory in Fig. 4): FGSM direction for $\mathbf{x}^{(0)}$ (as in Sec. 3.2),
- $\mathbf{d}_{\mathbf{r}}$: random direction,
- $\mathbf{d}_{\text{FGSM}^\perp}$ (green-colored trajectory in Fig. 4): random direction that is orthogonal to the FGSM direction,
- $\mathbf{d}_{\mathbf{r}+\text{FGSM}}$ (yellow-colored trajectory in Fig. 4): FGSM direction for the randomly perturbed data $\mathbf{x}_{\mathbf{r}}^{(0)} = \mathbf{x}^{(0)} + \epsilon_{\mathbf{r}} \cdot \mathbf{r}$, where $\mathbf{r}$ is a random vector ($\|\mathbf{r}\|_2 = \sqrt{D}$) and $\epsilon_{\mathbf{r}}$ controls the amount of the "random jump."

Fig. 5 shows the direction changes in early steps of travel (i.e., $\theta^{(1)}, \cdots, \theta^{(4)}$) for ResNet50 and Swin-T when ϵ/N=.002 and $\epsilon_{\mathbf{r}}$=.05 (see Appendix D.1 for the magnitude distributions, which shows

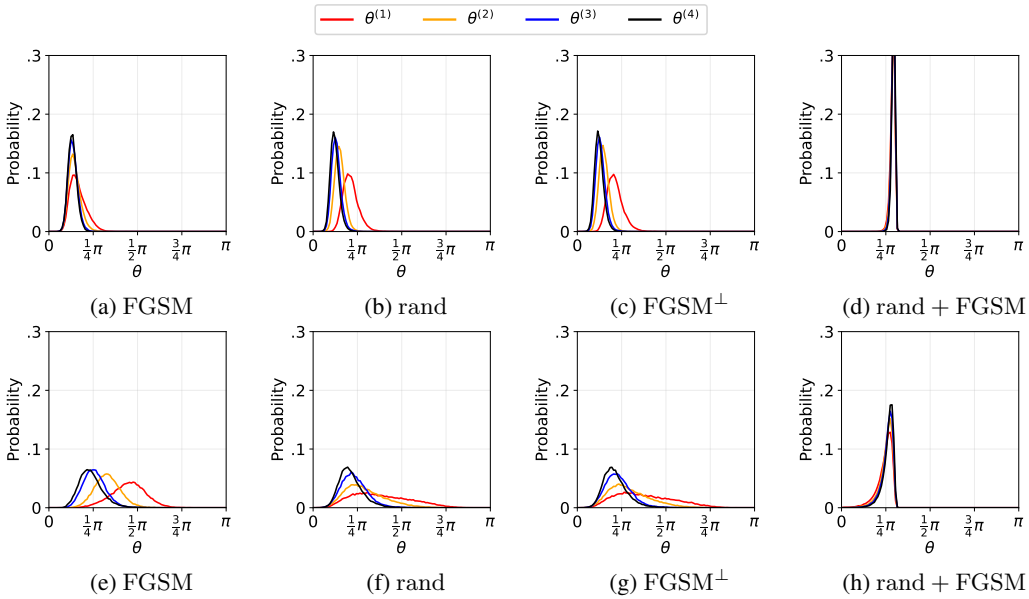


Figure 5: Direction changes of output features in early steps of travel for ResNet50 (top row) and Swin-T (bottom row).

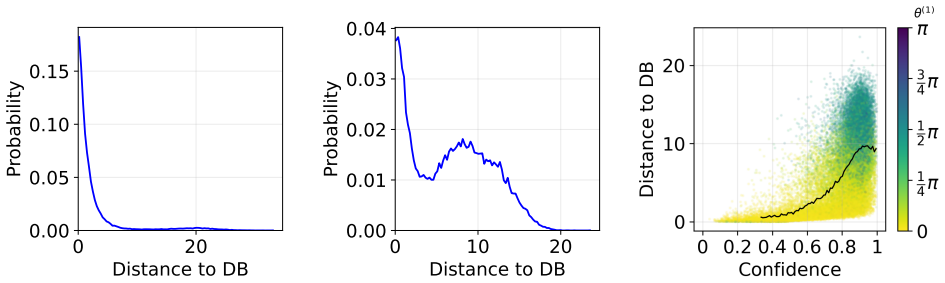
a similar trend; the results for the whole travel are shown in Appendix D.2). The following observations can be made (also illustrated in Fig. 4). 1) For ResNet50, the direction does not change much, no matter how the travel direction in the input space is set (Figs. 5a, 5b, 5c). This indicates that the input-output relationship of CNNs is fairly linear around the data. 2) In contrast, Swin-T shows locally nonlinear input-output relationship; $\theta^{(n)}$ is significantly large no matter how the travel direction in the input space is set (Figs. 5e, 5f, 5g). I.e., the representation space of Transformers is *curved* around the data. 3) Fig. 5h shows that the direction does not change much after the random jump. By moving $\mathbf{x}^{(0)}$ in a random direction $\mathbf{r}$, $\mathbf{z}^{(0)}$ can pass over the curved region without meandering in the early steps and make linear movements afterwards ($\mathbf{z_r}^{(n)}$ in Fig. 4).

4.2 Robustness and Underconfidence of Transformers

Figs. 6a and 6b show the distribution of the Euclidean distance from the original output to the decision boundary in the representation space (i.e., $||\mathbf{z}^{(N)} - \mathbf{z}^{(0)}||$) for ResNet50 and Swin-T. Note that the distance scale is different between the models. Interestingly, the distance distribution for Swin-T is *bimodal*, i.e., the data are grouped into those having small distances (around zero) and those having large distances (around 10).

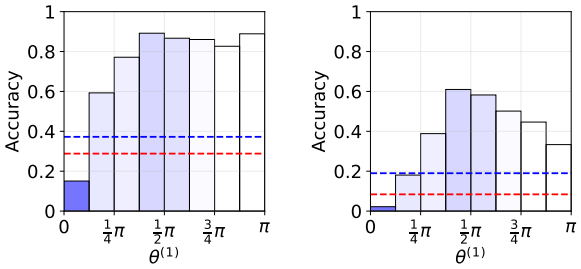
We examine this phenomenon further in Fig. 6c, which is a scatter plot between the confidence and the distance for Swin-T, where the colors represent $\theta^{(1)}$ of the corresponding data. Note that $\theta^{(1)}$ is highly correlated to the total direction change ($\sum_{n=1}^{N-1} \theta^{(n)}$), and thus is used as a measure of curvedness of the representation space around the data (see Appendix D.3). It is clear that the curvedness dichotomizes the data: those associated with small values of $\theta^{(1)}$ are located in linear regions (marked with yellowish colors), while those associated with large values of $\theta^{(1)}$ are located in curved regions (marked with greenish colors). In particular, the data in the latter group show larger distances to the decision boundaries, and thus become more robust against adversarial attacks. In other words, since they are located in curved regions, an attack on them becomes challenging.

To validate this, we apply the iterative FGSM attack (I-FGSM) (Kurakin et al., 2017), which is one of the strong attacks, to the correctly classified ImageNet validation data for Swin-T. We set the maximum amount of perturbation to ϵ_{IFGSM}=.001 or .002, the number of iterations to T=10, and the step size to $\epsilon_{\text{IFGSM}}/T$. Fig. 7 shows the classification accuracy after attack with respect to $\theta^{(1)}$. We can observe that the data having large values of $\theta^{(1)}$ show high robustness (i.e., high accuracy after



(a) ResNet50 (b) Swin-T (c) Swin-T

Figure 6: Distance between the original output and the decision boundary in the representation space. (a) Distribution for ResNet50. (b) Distribution for Swin-T. (c) Distance with respect to confidence for Swin-T. Colors indicate $\theta^{(1)}$. Black lines correspond to average values.



(a) $\epsilon_{\text{IFGSM}} = .001$ (b) $\epsilon_{\text{IFGSM}} = .002$

Figure 7: Accuracy after the I-FGSM attack for Swin-T with respect to $\theta^{(1)}$. Transparency of the bars represents the number of samples in each bin of $\theta^{(1)}$. Blue and red dashed lines indicate the overall accuracy of Swin-T and ResNet50 after the attack, respectively.

the attack), which makes the overall robustness of Swin-T (blue line) higher than that of ResNet50 (red line).

We hypothesize that the curved representation space also causes the underconfident prediction of Transformers. As shown as the blue-colored trajectory in Fig. 4b, the decision boundary is actually close to the data point but the curved travel reaches the decision boundary at a farther location. To validate this hypothesis, we add a small amount of noise to the input data in order to check if the decision boundary at a closer location can be found if the data jumps out of the curved region (i.e., reaching $\mathbf{z_r}^{(N)}$ from $\mathbf{z_r}^{(0)}$ in Fig. 4b).

Fig. 8 shows the distribution of the distance to the decision boundary in the representation space. It can be observed that when the FGSM direction is used after random jump (orange line), the bimodality of the distribution is significantly reduced. As shown in Fig. 5h, the travel becomes less curved and thus the decision boundary can be reached effectively. The 2D projected movements after the jump in Fig. 9 also supports this. Furthermore, the random jump can be made even more effective by setting the jump direction to the FGSM direction found after the random jump (red-colored trajectory in Fig. 4), resulting in further reduction in the bimodality (red line in Fig. 8).

The reduced distance to the boundary by random jump implies that the jumped input data can be made misclassified by adding a smaller amount of perturbation than the original input data. Fig. 10 demonstrates that this actually works. The figure shows example images perturbed linearly in the FGSM direction (i.e., $\mathbf{x}^{(N)}$) and those first undergone random jump ($\epsilon_{\mathbf{r}}$=.05) and then perturbed linearly in the FGSM direction (i.e., $\mathbf{x_r}^{(N)}$). It is clear that the images are easily misclassified with significantly reduced amounts of perturbation (smaller ϵ and higher PSNR) after the random jump passing over curved regions.

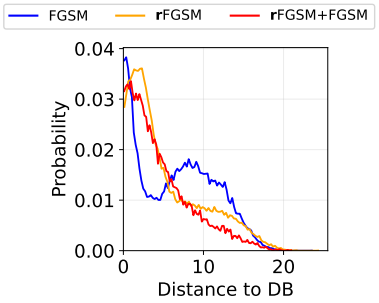


Figure 8: Distribution of the distance between the original output and the decision boundary in the representation space of Swin-T for different travel directions set in the input space.

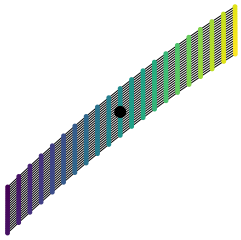


Figure 9: 2D projected movements of the output in the representation space after random jump for Swin-T.

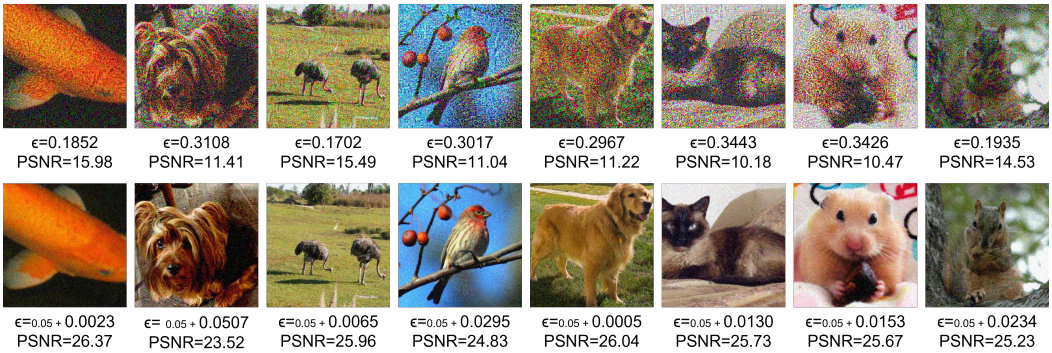


Figure 10: Example images that are perturbed by FGSM and reach decision boundaries. The total amount of perturbation (ϵ) and the peak signal-to-noise ratio (PSNR) in dB are also shown. *Top*: Perturbed images. *Bottom*: Images perturbed after random jump ($\epsilon_{\mathbf{r}} = .05$).

5 Curved Space During Training

For deeper understanding of the curved regions in the representation space, we look into the training process of Transformers. Each row of Fig. 11 shows the loss change of each of the 10,000 ImageNet training data at every 10 epochs during training of Swin-T (see Appendix F for training details). The rows are sorted in an ascending order of $\theta^{(1)}$ after training. It is observed that for the data located in curved regions (showing large values of $\theta^{(1)}$), the loss does not change much from the early training stage (bottom rows in the figure). Fig. 12 provides another view of this phenomenon in terms of the relationship between the loss at a certain epoch and the loss change from the epoch until the end of training. The loss values for the data residing in curved regions (dark-colored points in the figure) are hardly reduced already from 30 epochs. In other words, certain training data seem to be *trapped* in curved regions, which obstructs the training of the network.

When do curved regions start to form? We check the relationship of $\theta^{(1)}$ at a certain training stage and $\theta^{(1)}$ after training in Fig. 13. The data points are mostly above the 45° line, indicating that once a data is trapped in a curved region, it hardly escapes the region. It can be also seen that $\theta^{(1)}$ becomes larger during training, i.e., the curvedness gets severer.

6 Conclusion

We studied the input-output relationship of Transformers by examining the trajectory of the output in the representation space with respect to linear movements in the input space. The experimental results indicated that the representation space of Transformers is curved around some data, which

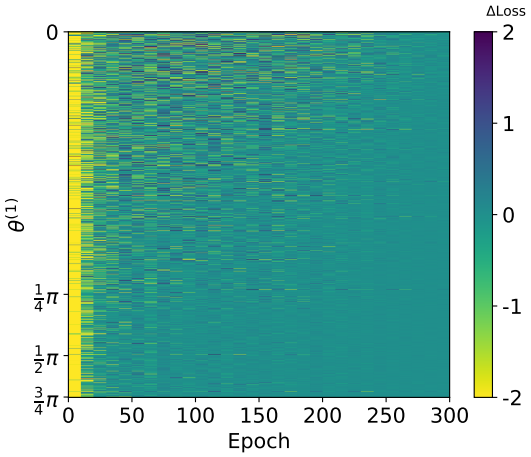


Figure 11: Loss changes of 10,000 training data from ImageNet during training of Swin-T. Each row represents the loss change at every 10 epochs for each data. The rows are sorted in an ascending order of $\theta^{(1)}$ after training. The loss change value is clipped within [-2,2] for visualization.

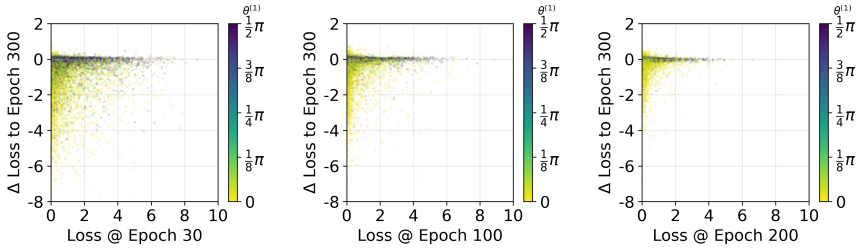


Figure 12: Scatter plots showing the loss at a certain training epoch with respect to the loss change till the end of training for Swin-T. Colors indicate $\theta^{(1)}$ after training.

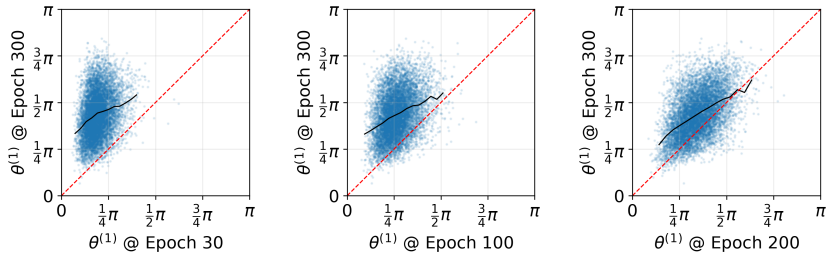


Figure 13: Relationship of $\theta^{(1)}$ at a certain training stage and the final $\theta^{(1)}$ for Swin-T. Black lines indicate average values.

explains high robustness and underconfident prediction of Transformers. It was also shown that the curved regions are formed at the early stage of training and can cause difficulty in training.

Based on our findings, many interesting research questions can follow towards better understanding of the working mechanism of neural networks, such as which architectural components and optimization techniques promote curved regions, how curved regions can be reduced or intensified during or after training, how curvedness can be quantified, etc.

REFERENCES

Ahmed Aldahdooh, Wassim Hamidouche, and Olivier Deforges. Reveal of vision transformers robustness against adversarial attacks. *arXiv preprint arXiv:2106.03734*, 2021.

Yutong Bai, Jieru Mei, Alan Yuille, and Cihang Xie. Are transformers more robust than CNNs? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Philipp Benz, Soomin Ham, Chaoning Zhang, Adil Karjauv, and In So Kweon. Adversarial robustness comparison of vision transformer and MLP-mixer to CNNs. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2021.

Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019a.

Boris Hanin and David Rolnick. Deep ReLU networks have surprisingly few activation patterns. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019b.

Boris Hanin, Ryan Jeong, and David Rolnick. Deep ReLU networks preserve expected length. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for MobileNetV3. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. LocalViT: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, 2014.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Roman Novak, Yasaman Bahri, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Namuk Park and Songkuk Kim. How do vision transformers work? In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

Ilan Price and Jared Tanner. Trajectory growth lower bounds for random sparse deep ReLU networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, , and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, 2015.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Thiago Serra, Christian Tjandraatmadja, and Srikumar Ramalingam. Bounding and counting linear regions of deep neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

Jure Sokolić, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16), 2017.

Matus Telgarsky. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101*, 2015.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2021.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Yeming Wen, Ghassen Jerfel, Rafael Muller, Michael W Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. Combining ensembles and data augmentation can harm your calibration. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token ViT: Training vision transformers from scratch on ImageNet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Xiao Zhang and Dongrui Wu. Empirical studies on the properties of linear regions in deep neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

A REPRESENTATION SPACE VISUALIZATION

Fig. 1 is produced as follows. The FGSM direction $\mathbf{d}_\mathbf{x}$ is found for the given data $\mathbf{x}$, and we choose a random direction $\mathbf{d}_\mathbf{x}^\perp$ orthogonal to $\mathbf{d}_\mathbf{x}$. Then, we move $\mathbf{x}$ in the input space using $\mathbf{d}_\mathbf{x}$ and $\mathbf{d}_\mathbf{x}^\perp$ by

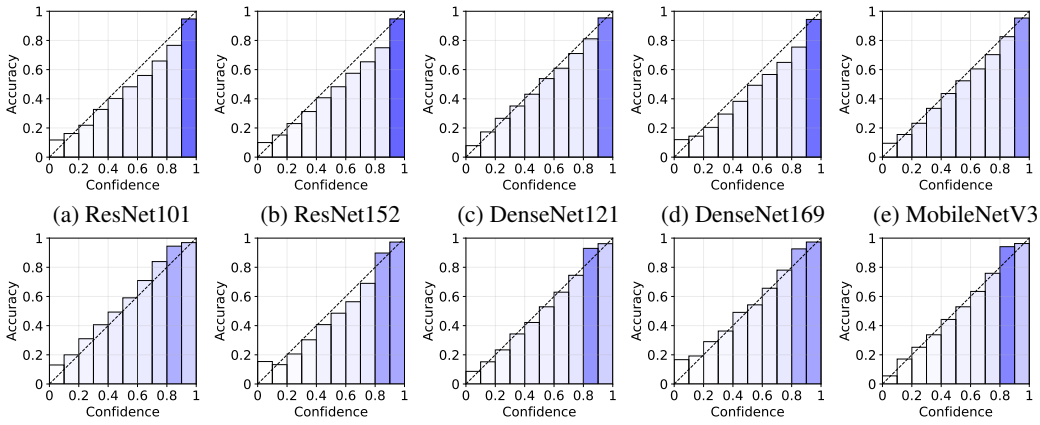
$$\mathbf{x}_{ij} = \mathbf{x} + \frac{\alpha \cdot i}{N}\mathbf{d}_\mathbf{x} + \frac{\alpha \cdot j}{N}\mathbf{d}_\mathbf{x}^\perp \tag{8}$$

where α is a positive real number, N determines the total number of points in the grid of the input space as $(2N+1)^2$ (Fig. 1a), and i and j are integers within $[-N, N]$.

For each $\mathbf{x}_{ij}$, we obtain its output at the penultimate layer of a model, $\mathbf{z}_{ij}$, i.e., the feature in the representation space. Then, $\mathbf{z}_{ij}$ is projected onto the two-dimensional plane determined by two arbitrary mutually orthogonal vectors $\mathbf{d}_\mathbf{z}$ and $\mathbf{d}_\mathbf{z}^\perp$, which produces Figs. 1b and 1c.

B MODEL CALIBRATION

Fig. 14 provides reliability diagrams of the models that are listed in Table 1 but not included in Fig. 2.



(a) ResNet101 (b) ResNet152 (c) DenseNet121 (d) DenseNet169 (e) MobileNetV3

(f) ViT-B/32 (g) ViT-L/16 (h) ViT-L/32 (i) Swin-S (j) Swin-B

Figure 14: Reliability diagrams of CNNs (top row) and Transformers (bottom row). Transparency of bars represent the number of data in each bin of confidence.

C PASSAGE TO DECISION BOUNDARY

Algorithm 1 provides the detailed procedure to travel to the decision boundary linearly in the input space (Sec. 3.2).

Fig. 15 extends Fig. 3 for other Transformer models.

D REPRESENTATION SPACE ANALYSIS

D.1 MAGNITUDE OF TRAVEL

Fig. 16 shows the step-wise magnitudes of in early steps of travel (i.e., $\omega^{(1)}, \cdots, \omega^{(4)}$), which shows a similar trend to Fig 5.

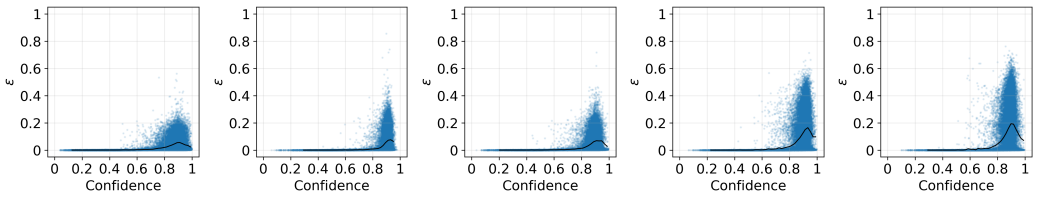
D.2 WHOLE TRAVEL TRAJECTORY

Fig. 17 shows the step-wise magnitude ($\omega^{(n)}$) and direction change ($\theta^{(n)}$) during the whole travel for N=50 steps to the decision boundary for Swin-T. The travel direction is determined by FGSM. Each

Algorithm 1 Travel to decision boundary

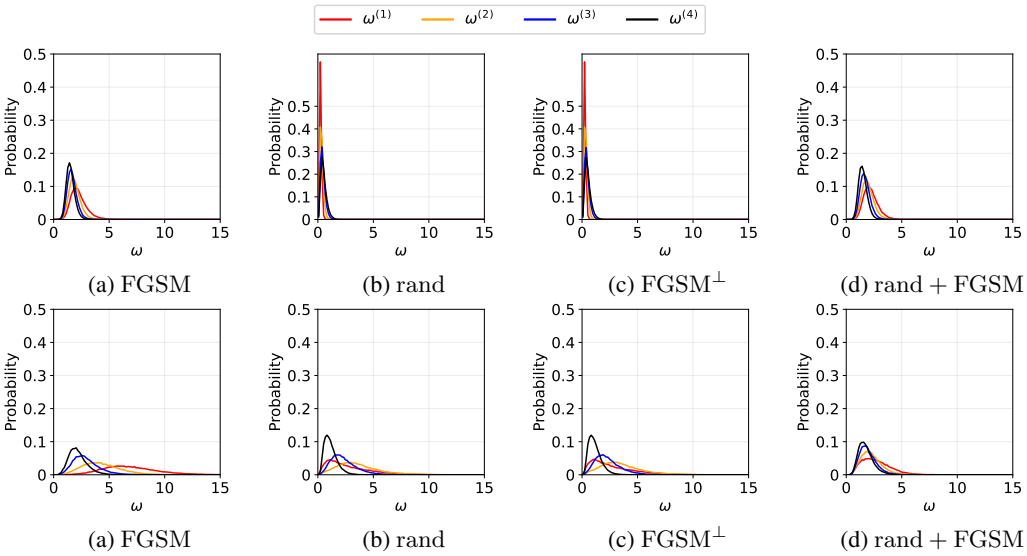
Input: A correctly classified image $\mathbf{x} \in [0,1]^D$, true class label y, classifier $\mathcal{C}$, travel direction $\mathbf{d}$
Parameters: Initial length of travel ϵ_i, travel length ϵ, length decay ϵ_d ($0 < \epsilon_d < 1$), tolerance ϵ_t
Output: Traveled image $\mathbf{x}'$

$\epsilon \leftarrow \epsilon_i$
while $\epsilon < \epsilon_t$ **do**
  Generate perturbation $\mathbf{p} = \epsilon \cdot \mathbf{d}$.
  $\mathbf{x}' \leftarrow \mathrm{clip}_{0,1}(\mathbf{x} + \mathbf{p})$ where $\mathrm{clip}_{a,b}(z) = \min(\max(z, a), b)$.
  if $\mathcal{C}(\mathbf{x}') \neq \mathbf{y}$ **then**
    $\epsilon \leftarrow \epsilon \times \epsilon_d$
  else
    $\epsilon \leftarrow \epsilon + \epsilon$
  end if
end while
Return traveled image $\mathbf{x}'$.



(a) ViT-B/32 (b) ViT-L/16 (c) ViT-L/32 (d) Swin-S (e) Swin-B

Figure 15: Lengths (ϵ) of the travel to decision boundaries for the ImageNet validation data. Black lines represent average values.

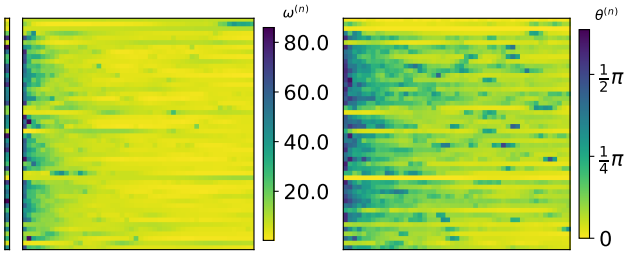


(a) FGSM (b) rand (c) FGSM$^{\perp}$ (d) rand + FGSM

(a) FGSM (b) rand (c) FGSM$^{\perp}$ (d) rand + FGSM

Figure 16: Magnitude distributions of ImageNet features in early steps of travel for ResNet50 (top row) and Swin-T (bottom row).

row in Figs. 17a and 17b corresponds to each of 50 images from ImageNet. Note that Figs. 17a depicts the sum-normalized magnitudes (i.e., $\omega^{(n)} / \sum_{n=1}^{N} \omega^{(n)}$); the sum values are shown on the left side. Largely nonlinear movements mostly appear in early steps of the travel. Afterwards, the trajectory becomes fairly linear.



(a) $\omega^{(n)}$ (b) $\theta^{(n)}$

Figure 17: $\omega^{(n)}$ and $\theta^{(n)}$ during the whole travel of 50 ImageNet data to decision boundaries for Swin-T.

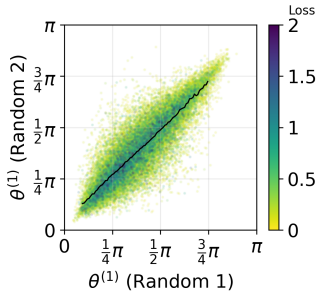


Figure 18: Relationship of $\theta^{(1)}$ in two different random travel directions.

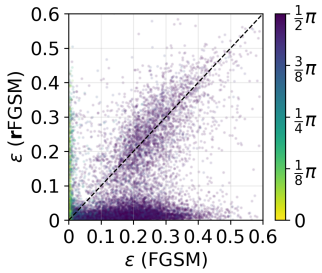


Figure 19: Relationship of the length of travel (ϵ) to decision boundaries for original images (x-axis) and randomly jumped images (y-axis). Colors indicate $\theta^{(1)}$.

D.3 Measure of curvedness

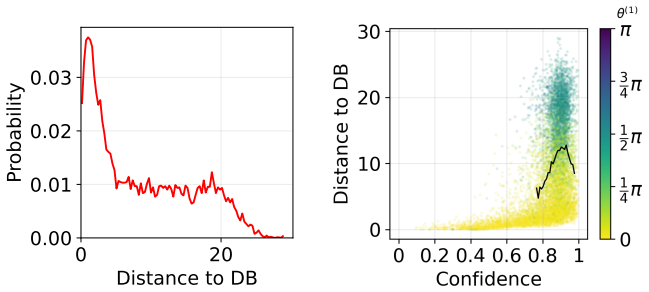
Table 2 shows the correlation coefficients between $\theta^{(1)}$ (the first direction change) and $\sum_{n=1}^{N-1}\theta^{(n)}$ (the total direction change). While the latter is a measure of curvedness of the representation space in an overall sense, the two values are highly correlated. Thus, we use $\theta^{(1)}$ instead; in addition, it can provide the information about the local curvedness near $\mathbf{z}_0$.

Table 2: Correlation coefficients between $\theta^{(1)}$ and $\sum_{n=1}^{N-1}\theta^{(n)}$ for four different types of travel.

Model	FGSM	rand	$\text{FGSM}^{\perp}$	rand + FGSM
ResNet50	.8983	.7833	.7877	.8827
Swin-T	.7397	.7978	.7941	.7698

D.4 Curvedness in different directions

Fig. 18 shows the relationship of $\theta^{(1)}$ in two different random directions, implying that no matter which random direction is chosen, the result remains consistent.



(a) (b)

Figure 20: Distance between the output of the ImageNet training data and the decision boundary in the representation space for Swin-T. (a) Distribution of distance. (b) Distance with respect to confidence. Colors indicate $\theta^{(1)}$.

E Distance to decision boundaries after jump

Fig. 19 shows the relationship of the length of travel (ϵ) to decision boundaries for original images (x-axis) and random jumped images (y-axis) for Swin-T. The dark-colored points on the bottom part of the figure correspond to the images that escape curved regions with random jump and then can reach nearby located decision boundaries with imperceptible noise (Fig. 10).

F Model training

We follow the original training implementation code of the Swin-T Transformer for model training in Sec. 5, e.g., 300 training epochs, batch size of 128, learning rate of 5×10^{-4}, weight decay parameter of .05, and AdamW (Loshchilov & Hutter, 2019) as the optimizer.

Fig. 20 examines the distance from the original output to the decision boundary in the representation space for the ImageNet training data in the case of Swin-T. The observed trend is similar to that for the validation data (Figs. 6b and 6c).