Curved Representation Space of Vision Transformers

Juyeop Kim, Junha Park, Songkuk Kim*, Jong-Seok Lee*

School of Integrated Technology / BK21 Graduate Program in Intelligent Semiconductor Technology
Yonsei University, Korea
{juyeopkim, junha.park, songkuk, jong-seok.lee}@yonsei.ac.kr

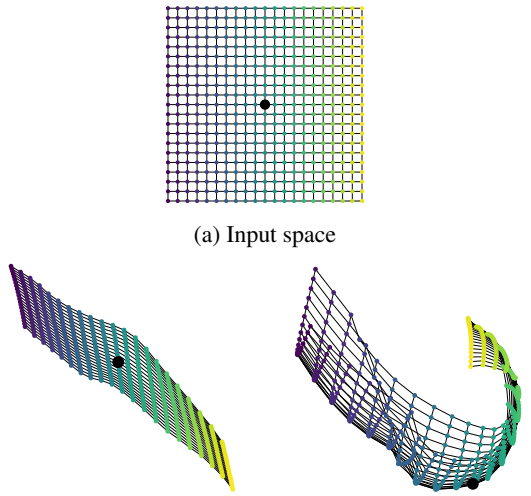
Abstract

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). 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. This is contrary to the intuition that robustness increases with confidence. We resolve this contradiction by empirically investigating how the output of the penultimate layer moves in the representation space as the input data moves linearly 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. 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 explains the underconfident prediction of Transformers. Also, we examine mathematical properties of the attention operation that induce nonlinear response to linear perturbation. Finally, we share our additional findings, regarding what contributes to the curved representation space of Transformers, and how the curvedness evolves during training.

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 deeper understanding of CNNs and eventually neural networks.

*Corresponding authors



(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.

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 and Chen 2022). Moreover, 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 (see Appendix for detailed discussion). 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.

To mitigate the contradiction of robustness and underconfidence, this paper presents our empirical study to explore

the representation space of Transformers and CNNs. More specifically, we focus on the *linearity of the models*, i.e., the change of the output feature (which is simply referred to as output in this paper) with respect to the linear change of the input data. It is known that adversarial examples are a result of models being too linear, based on which the fast gradient sign method (FGSM) was introduced to show that deep neural networks can be easily fooled (Goodfellow, Shlens, and Szegedy 2015). Motivated by this, we examine the input-output relationship of Transformers through the course that the input is gradually perturbed along the direction determined by FGSM.

Fig. 1 visualizes the representation spaces of CNNs and Transformers comparatively (see Appendix 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 (He et al. 2016) and Swin-T (Liu et al. 2021) 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 nonlinear around the original output (and then becomes linear when the change of the output is large), i.e., the representation space is locally *curved*. We empirically show that this curved representation space results in the aforementioned contradiction.

Our main 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 movement of the penultimate layer's output with respect to the linear movement of the input. We use the adversarial gradient produced by FGSM (Goodfellow, Shlens, and Szegedy 2015) as the direction of movement in the input space, to investigate the linearity of the feature space of the models. We find that 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, a series of linear perturbations to the input move the output point along a curved trajectory. This makes it hard to move the data out of its decision region along a short and straight line, 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 exists near the original data in the representation space, which explains the underconfident predictions of Transformers**.

We also present additional observations examining what contributes to the curved representation space of Transformers and when the curvedness is formed during training.

The Appendix of this paper can be found in the following link: https://arxiv.org/abs/2210.05742.

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. 2022a). There have been efforts to compare CNNs and Transformers in various aspects. Empirical studies show that Transformers have higher adversarial robustness than CNNs (Paul and Chen 2022; Naseer et al. 2021; Aldahdooh, Hamidouche, and Deforges 2021; Bhojanapalli et al. 2021), which seems to be due to the reliance of Transformers on lower frequency information than CNNs (Park and Kim 2022; Benz et al. 2021). Other studies conclude that Transformers are calibrated better than CNNs yielding overconfident predictions (Guo et al. 2017; Thulasidasan et al. 2019; Wen et al. 2021; Minderer 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 and Rolnick 2019a,b; Telgarsky 2015; Serra, Tjandraatmadja, and Ramalingam 2018; Raghu et al. 2017) and examining local properties of linear regions (Zhang and Wu 2020). Some studies examine the length of the output curve for a given unit-length input (Raghu et al. 2017; Price and Tanner 2021; Hanin, Jeong, and Rolnick 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.

On the Ostensible Contradiction of High Robustness and Underconfidence

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 (i.e., probability of the predicted class) 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, Cooper, and Hauskrecht 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. One 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 signed ECE (sECE) to augment ECE, 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 ResNet50 (He et al. 2016) and MobileNetV2 (Sandler et al. 2018; Howard et al. 2019), and Transformers, including ViT-B/16 (Dosovitskiy et al. 2021) and Swin-T (Liu et al. 2021), on the ImageNet validation set using ECE and sECE in Fig. 2 (see Fig. 11 in Appendix for the results of other models). CNNs show negative ECE values and bar plots below the 45° line, 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.

Passage to Decision Boundary

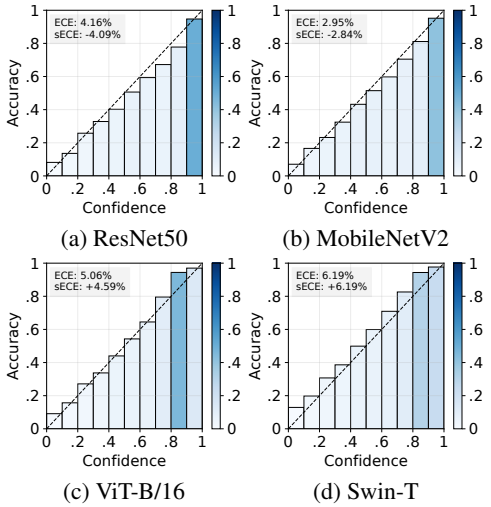
It is a common intuition that if a model classifies a data with low confidence, the data is likely to be located near a decision boundary (see Appendix for detailed discussion). Based on the above results, therefore, the decision boundaries of Transformers are assumed to be formed near the data compared 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 the 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 indicating the travel length, and $\mathbf{x}'$ is the traveled result of $\mathbf{x}$. We set the travel direction $\mathbf{d}$ as the adversarial gradient produced by FGSM, i.e.,

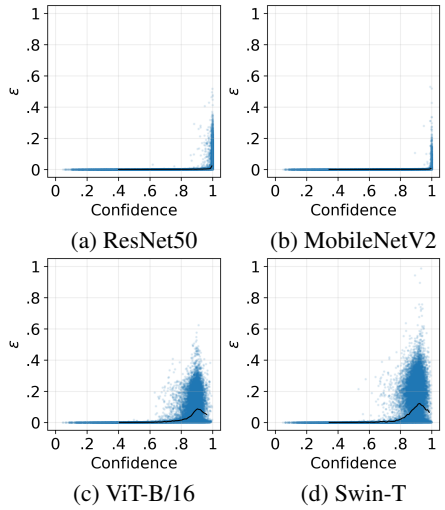
$$\mathbf{d} = \mathrm{sign}(\nabla_{\mathbf{x}} J(\mathcal{C}(\mathbf{x}), y)), \tag{4}$$

where J is the classification loss function (i.e., cross-entropy). Note that $\|\mathbf{d}\|_2 = \sqrt{D}$, where D is the dimension of $\mathbf{x}$. Refer to Algorithm 1 in Appendix for the detailed procedure to solve the optimization problem in Eq. 3.



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

Figure 2: Reliability diagrams of CNNs and Transformers. Transparency of bars represent the ratio of the number of data in each confidence bin. ECE and sECE values are also shown in each case.



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

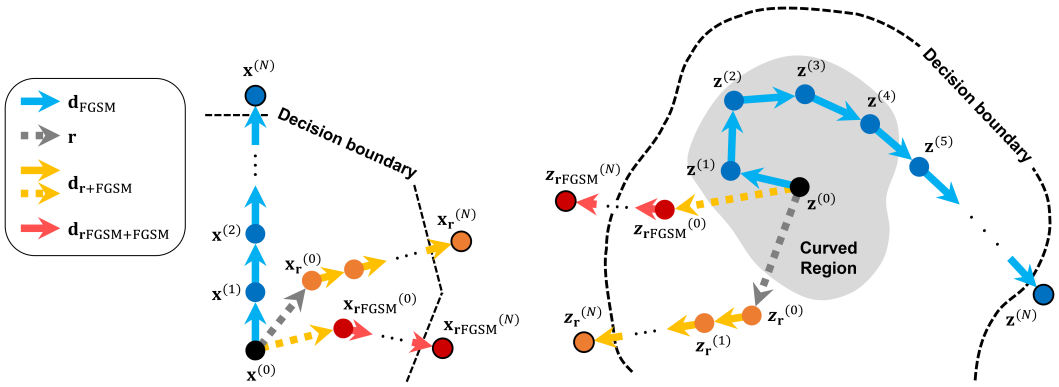
Figure 3: Lengths (ϵ) of the travel to decision boundaries with respect to the confidence for the ImageNet validation data. Black lines represent average values.

Fig. 3 shows the obtained values of ϵ with respect to the confidence values for the ImageNet validation data (see Fig. 12 in Appendix for the results of other models). On the contrary to our expectation, decision boundaries seem to be located farther from the data in the input space for Transformers than CNNs. This contradiction is resolved in the following section.

Resolving the Contradiction

Shape of Representation Space

As mentioned in the **Introduction**, the FGSM attack was first introduced to show that the linearity of a model causes



(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.

its vulnerability to adversarial perturbations (Goodfellow, Shlens, and Szegedy 2015). To resolve the contradiction between high robustness (a large distance to the decision boundary) and underconfidence (a small distance to the decision boundary) of Transformers in the previous section, therefore, we examine the degree of linearity of the input-output relationship, i.e., how linear movements in the input space appear in the representation space of Transformers.

We divide the travel into N steps as

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

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

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

We consider three different ways to determine $\mathbf{d}$:

- $\mathbf{d}_{\mathrm{FGSM}}$ (blue-colored trajectory in Fig. 4): FGSM direction for $\mathbf{x}^{(0)}$ (as in Eq. 4).
- $\mathbf{d}_{\mathbf{r}+\mathrm{FGSM}}$ (yellow-colored trajectory in Fig. 4): FGSM direction determined 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 this "random jump."
- $\mathbf{d}_{\mathbf{r}\mathrm{FGSM}+\mathrm{FGSM}}$ (red-colored trajectory in Fig. 4): FGSM direction determined for the data perturbed in the direction of $\mathbf{d}_{\mathbf{r}+\mathrm{FGSM}}$, i.e., $\mathbf{x}_{\mathbf{r}\mathrm{FGSM}}^{(0)} = \mathbf{x}^{(0)} + \epsilon_{\mathbf{r}+\mathrm{FGSM}} \cdot \mathbf{d}_{\mathbf{r}+\mathrm{FGSM}}$, where $\epsilon_{\mathbf{r}+\mathrm{FGSM}}$ controls the amount of this jump.

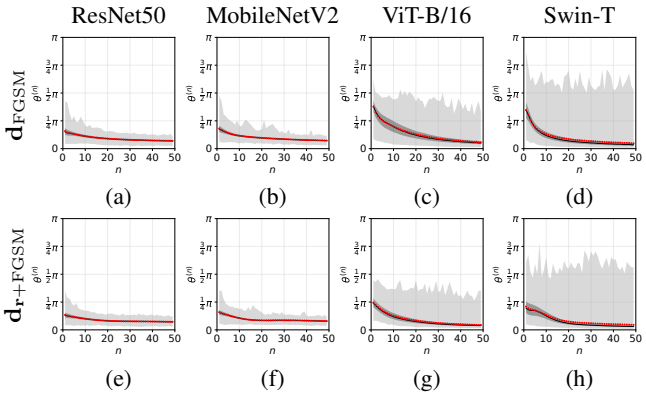


Figure 5: Direction changes of output features with respect to the travel step (n). Light-gray regions: Range between the minimum and maximum values. Dark-gray regions: Range between the first quartile (Q1) and the third quartile (Q3). Black lines: Medians (Q2). Red dots: Mean values.

Figs. 5a-5d show the direction changes in travel for ResNet50, MobileNetV2, ViT-B/16 and Swin-T when $\mathbf{d} = \mathbf{d}_{\mathrm{FGSM}}$, $\epsilon = .05$, and $N = 50$. See Fig. 13 in Appendix for the results of other travel directions, which shows a similar trend. For ResNet50 and MobileNetV2, the direction does not change much (Figs. 5a and 5b), which in fact holds regardless of the travel direction in the input space (see Figs. 13a and 13b in Appendix). This indicates that **the input-output relationship of CNNs is fairly linear around the data**. In contrast, ViT-B/16 and Swin-T shows locally nonlinear input-output relationship; $\theta^{(n)}$ is significantly large in early steps of travel (Figs. 5c and 5d), even for other travel directions in the input space (see Figs. 13c and 13d in Appendix). I.e., **Transformers generate *nonlinear* response to linear perturbation and the representation space of Transformers is *curved* around the data**.

Figs. 5e-5h show the direction changes in travel when $\mathbf{d} = \mathbf{d}_{\mathbf{r}+\mathrm{FGSM}}$, with $\epsilon_{\mathbf{r}} = .05$ except for ViT-B/16 using

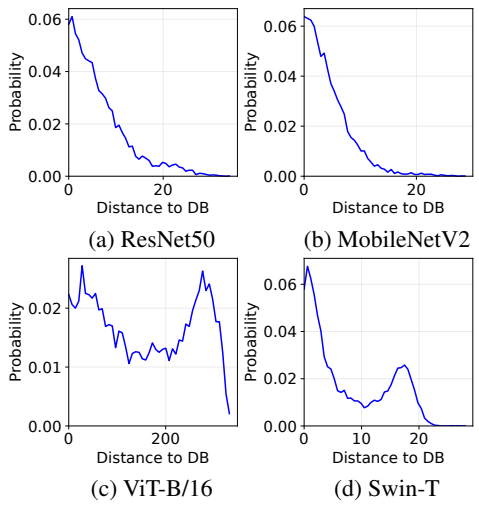


Figure 6: Distribution of distance from the original output to the decision boundary (DB) in the representation space.

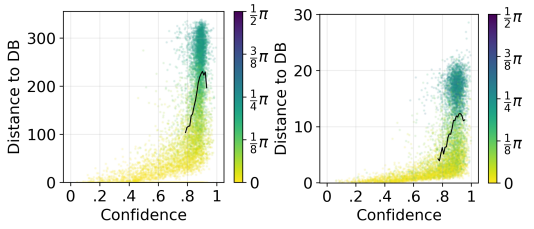


Figure 7: Distance from the original output to the decision boundary (DB) in the representation space with respect to confidence. Colors indicate $\theta^{(1)}$. Black lines represent average values.

$\epsilon_{\mathbf{r}} = .20$ (see Appendix for discussion). It can be observed that the direction does not change much after the random jump. I.e., **the curvedness of the representation space is localized around the data**. Therefore, by making $\mathbf{x}^{(0)}$ jump a certain distance 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. 4b).

Robustness and Underconfidence of Transformers

Fig. 6 shows the distribution of the Euclidean distance from the original output to the decision boundary in the representation space (i.e., $||\mathbf{z}^{(N)} - \mathbf{z}^{(0)}||$). Note that the distance scale is different between the models. Interestingly, the distance distributions for ViT-B/16 and Swin-T are *bimodal*, i.e., the data are grouped into those having small distances and those having large distances.

We examine this phenomenon further in Fig. 7, which shows scatter plots between the confidence and the distance, 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

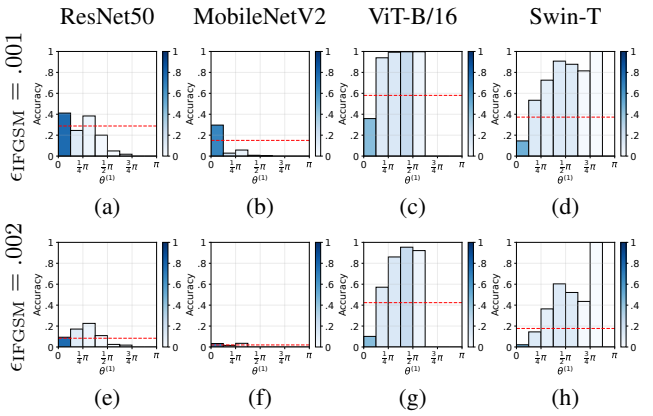


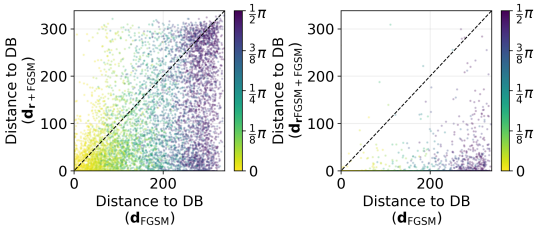
Figure 8: Accuracy after the I-FGSM attack with respect to $\theta^{(1)}$. Transparency of the bars represents the ratio of the number of samples in each bin of $\theta^{(1)}$. Red dashed lines indicate the overall accuracy after the attack.

curvedness of the representation space around the data (see Appendix). 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.

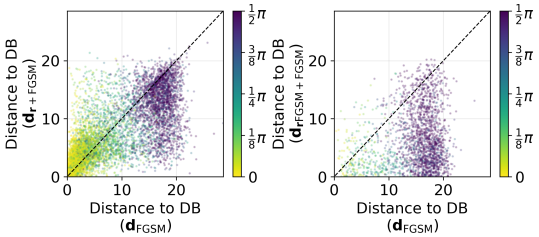
To validate this, we apply the iterative FGSM attack (I-FGSM) (Kurakin, Goodfellow, and Bengio 2017), which is one of the strong attacks, to the correctly classified ImageNet validation data. We set the maximum amount of perturbation to $\epsilon_{\mathrm{IFGSM}}$=.001 or .002, the number of iterations to T=10, and the step size to $\epsilon_{\mathrm{IFGSM}}/T$. Fig. 8 shows the classification accuracy after the 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 even after the attack), which makes the overall robustness of Transformers higher than that of CNNs.

We hypothesize that the curved representation space also causes the underconfident prediction of Transformers. That is, as shown in Fig. 4b, the decision boundary is actually close to the data point (on the left side of the data), but the curved travel (blue-colored trajectory in Fig. 4b) 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).

Figs. 9a and 9b show the relationship of the distance to decision boundaries for original outputs (x-axis) and jumped images (y-axis) for Swin-T. The direction for travel is indicated in the axis. It can be observed that when the FGSM direction is computed and used after random jump ($\mathbf{d} = \mathbf{d}_{\mathbf{r}+\mathrm{FGSM}}$; yellow-colored trajectory in Fig. 4), the distance



(a) ViT-B/16



(b) Swin-T

Figure 9: Relationship of the length of travel (ϵ) to decision boundaries (DB) for original images (x-axis) and jumped images (y-axis). Colors indicate $\theta^{(1)}$.

is significantly reduced (left figures in Figs. 9a and 9b; most data points under the 45° line). As shown in Figs. 5g and 5h, the travel becomes less curved and thus the decision boundary can be reached effectively. The 2D projected movements after the jump in Fig. 16 in Appendix also supports this. Furthermore, the random jump can be made even more effective by setting the jump direction to the FGSM direction that would have been found if random jump was applied ($\mathbf{d} = \mathbf{d}_{\mathbf{r}\mathrm{FGSM}+\mathrm{FGSM}}$; red-colored trajectory in Fig. 4), resulting in further reduction in distance (right figures in Figs. 9a and 9b).

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., ${\mathbf{x}_{\mathbf{r}}}^{(N)}$) for Swin-T. 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.

Nonlinearity of Attention Operation

Why do curves tend to appear in the representation space of Transformers only, and not in CNNs? In this section, we explain this theoretically by revisiting convolution and self-attention operations. Note that we use matrices to denote inputs and outputs instead of vectors for better explanation of the operations. Empirical results of this section can be found in the next section and Table 1 in Appendix.

Deep neural networks transform data points through contiguous blocks that perform similar operations. CNNs, for instance, comprise layers of a convolution operation and activation. As well known, a convolution is a linear operation (Hayes 1996), i.e., an increment $\mathbf{P}$ to the input $\mathbf{X}$ converts into the addition of separate responses:

$$\mathrm{Conv}(\mathbf{X}+\mathbf{P}) = \mathrm{Conv}(\mathbf{X}) + \mathrm{Conv}(\mathbf{P}). \tag{8}$$

Activation functions may imbue the transformation with nonlinearity in theory, which is very limited in reality. ReLUs are linear until the input data travels to the negative region. In the case of sigmoid functions, the input data is supposed to linger in the non-saturated region, which is pseudo-linear. Therefore, the main building block of CNNs is a linear transformation.

At the heart of Transformers, an attention block transforms an input ***query*** into the weighted sum of neighbor ***values***, which is a linear projection of input tokens. The weights are calculated as softmax of attention scores $\mathbf{A}$, which is an inner product of ***query*** and ***key***:

$$\mathbf{A}(\mathbf{X}) = \mathbf{X}\mathbf{W}_{\mathbf{q}}\mathbf{W}_{\mathbf{k}}^{\top}\mathbf{X}^{\top}, \tag{9}$$

$$\mathrm{Attn}(\mathbf{X}) = \mathrm{softmax}(\mathbf{A}/\sqrt{D_k})\mathbf{X}\mathbf{W}_{\mathbf{v}}, \tag{10}$$

where $\mathbf{W}_{\mathbf{q}}$, $\mathbf{W}_{\mathbf{k}}$, and $\mathbf{W}_{\mathbf{v}}$ are the projection heads for query, key, and value, respectively, and D_k is the column dimension of $\mathbf{W}_{\mathbf{k}}$. If $\mathbf{X}$ is moved by $\mathbf{P}$, $\mathbf{A}$ will change as follows:

$$\begin{aligned}\mathbf{A}(\mathbf{X}+\mathbf{P}) =& (\mathbf{X}+\mathbf{P})\mathbf{W}_{\mathbf{q}}\mathbf{W}_{\mathbf{k}}^{\top}(\mathbf{X}^{\top}+\mathbf{P}^{\top}) && (11)\\ =& \mathbf{X}\mathbf{W}_{\mathbf{q}}\mathbf{W}_{\mathbf{k}}^{\top}\mathbf{X}^{\top} + \mathbf{P}\mathbf{W}_{\mathbf{q}}\mathbf{W}_{\mathbf{k}}^{\top}\mathbf{P}^{\top} + \\ & \mathbf{X}\mathbf{W}_{\mathbf{q}}\mathbf{W}_{\mathbf{k}}^{\top}\mathbf{P}^{\top} + \mathbf{P}\mathbf{W}_{\mathbf{q}}\mathbf{W}_{\mathbf{k}}^{\top}\mathbf{X}^{\top} && (12)\\ =& \mathbf{A}(\mathbf{X}) + \mathbf{A}(\mathbf{P}) + \\ & \underbrace{\mathbf{X}\mathbf{W}_{\mathbf{q}}\mathbf{W}_{\mathbf{k}}^{\top}\mathbf{P}^{\top} + \mathbf{P}\mathbf{W}_{\mathbf{q}}\mathbf{W}_{\mathbf{k}}^{\top}\mathbf{X}^{\top}}_{\text{residual}}. && (13)\end{aligned}$$

As shown in Eq. 13, the attention score is not linear and the deviation from the linear response is the combination of the projection heads and input data. During the inference operation, the projection heads are fixed and the linear perturbation to the input data will generate a varying degree of nonlinearity depending on the magnitude of the input and the angle between the projection head and the input (see Fig. 17 in Appendix for detailed discussion). Additionally, the softmax function in Eq. 10 augments the nonlinearity of the attention operation.

Additional Intriguing Observations

In this section, in addition to the aforementioned main discoveries, we share our additional intriguing observations, for which we leave further detailed analysis as future work.

Contribution of Components to Curvedness

Which component in Transformers fortifies the curvedness of the representation space? When ResNet50 and Swin-T are compared (Table 1 in Appendix), we find that in both models the activation functions contribute the most to the increase of $\theta^{(1)}$. GELU causes curvedness more than ReLU because the former is more nonlinear than the latter. In the case of ResNet50, the convolutional layers and batch normalization (BatchNorm) do not cause curvedness of the representation space. In contrast, for Swin-T, the layer normalization (LayerNorm) and self-attention layers intensify the

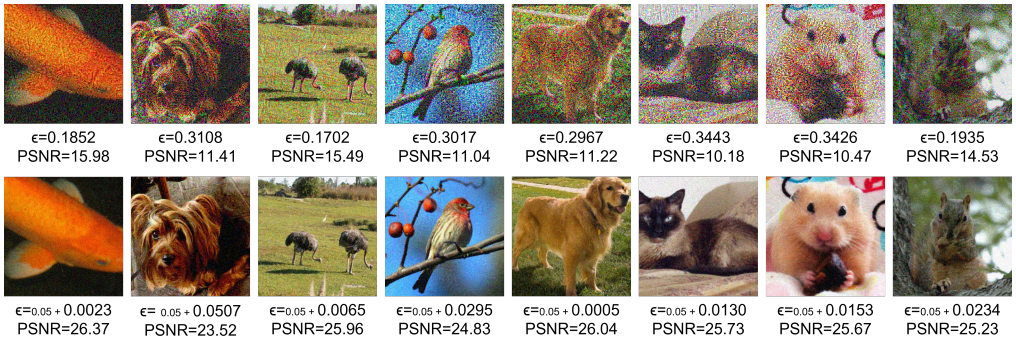


Figure 10: Example images that are perturbed by FGSM so as to reach decision boundaries and become misclassified. The total amount of perturbation (ϵ) and the peak signal-to-noise ratio (PSNR) in dB are also shown. Top: Perturbed images. Bottom: Images perturbed after random jump ($\epsilon_{\mathbf{r}}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

curvedness. The result of these compound contributions of different components appears as the curvedness of the representation space of Transformers.

This observation raises an interesting question about ConvNeXt (Liu et al. 2022b), which is a CNN but uses GELU and LayerNorm instead of ReLU and BatchNorm, respectively: Which trend will it follow, CNNs or Transformers? Surprisingly, we observe that ConvNeXt-Tiny follows the trend of *Transformers*, rather than CNNs (Table 1 and Figs. 18a and 19a in Appendix). This indicates that the curvedness in the representation space highly depends on the particular components used in models, and is not just a problem of models being CNNs or Transformers.

Knowledge Distillation and Curvedness

Another interesting model we find is DeiT-Ti (Touvron et al. 2021), a convolution-free Transformer, and its distilled version, which we refer to as DeiT-Ti-Distilled. We observe that as expected, DeiT-Ti follows the trend of Transformers, i.e., underconfidence with high robustness (Figs. 18b and 19b in Appendix). However, DeiT-Ti-Distilled, knowledge-distilled DeiT-Ti with a CNN teacher, tends to follow the trend of CNNs, i.e., overconfidence with low robustness (Figs. 18c and 19c in Appendix). The results in Table 1 also coincides with this observation, where the values of $\theta^{(1)}$ are reduced for DeiT-Ti-Distilled compared to DeiT-Ti. This indicates that knowledge distillation can also affect the non-linearity of Transformers.

Curved Space During Training

For deeper understanding of the curved regions in the representation space, we look into the training process of Transformers. We observe that for the data located in curved regions, the loss does not change much from the early training stage (Fig. 20 in Appendix; no change in loss for bottom rows in the figure, which show large values of $\theta^{(1)}$). This phenomenon can also be observed from another view, in terms of the relationship between the loss at a certain epoch and the loss change from the epoch until the end of training (Fig. 21 in Appendix; 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? When we check the relationship of $\theta^{(1)}$ at a certain training stage and $\theta^{(1)}$ after training, we observe that once a data is trapped in a curved region, it hardly escapes the region and $\theta^{(1)}$ becomes larger during training, i.e., the curvedness gets severer (Fig. 22 in Appendix; data points mostly above the 45° line).

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 explains high robustness and underconfident prediction of Transformers.

Future Work

In general, understanding the behavior of a certain neural network model, either analytically or empirically, is a difficult task, which cannot be accomplished by a single paper but requires a lot of research efforts. We have focused on the input-output relationship along the adversarial direction generated by FGSM, which revealed the existence of curvedness in the representation space of Transformers. We believe that we have opened a new perspective of understanding Transformers, and many derivative research questions will naturally follow, e.g., consideration of different travel directions, input-output relationship of various building blocks of neural networks, effects of different training recipes, effects of training datasets, etc., which we leave as future works. It is also our hope that our work promotes further interesting research topics (e.g., ways to reduce/intensify curvedness during or after training, measures of local/-global curvedness, theoretical analysis of curvedness, etc.) and applications (e.g., effective adversarial attacks considering curvedness, robust model architectures, robust training methods, etc.).

Acknowledgements

This work was supported in part by the fund (IITP 2022-0-00117) from the Korea government, MSIT, and in part by the NRF grant funded by the Korean government, MSIT (2021R1A2C2011474).

References

Aldahdooh, A.; Hamidouche, W.; and Deforges, O. 2021. Reveal of vision transformers robustness against adversarial attacks. *arXiv preprint arXiv:2106.03734*.

Bai, Y.; Mei, J.; Yuille, A.; and Xie, C. 2021. Are transformers more robust than CNNs? In *NIPS*.

Benz, P.; Ham, S.; Zhang, C.; Karjauv, A.; and Kweon, I. S. 2021. Adversarial robustness comparison of vision transformer and MLP-Mixer to CNNs. In *Proceedings of the British Machine Vision Conference (BMVC)*.

Bhojanapalli, S.; Chakrabarti, A.; Glasner, D.; Li, D.; Unterthiner, T.; and Veit, A. 2021. Understanding robustness of transformers for image classification. In *ICCV*.

Chu, X.; Tian, Z.; Wang, Y.; Zhang, B.; Ren, H.; Wei, X.; Xia, H.; and Shen, C. 2021. Twins: Revisiting the design of spatial attention in vision transformers. In *NIPS*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *ICLR*.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern meural networks. In *ICML*.

Hanin, B.; Jeong, R.; and Rolnick, D. 2022. Deep ReLU networks preserve expected length. In *ICLR*.

Hanin, B.; and Rolnick, D. 2019a. Complexity of linear regions in deep networks. In *ICML*.

Hanin, B.; and Rolnick, D. 2019b. Deep ReLU networks have surprisingly few activation patterns. In *NIPS*.

Hayes, M. H. 1996. *Statistical digital signal processing and modeling*. John Wiley & Sons, Inc., 1st edition.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; Le, Q. V.; and Adam, H. 2019. Searching for MobileNetV3. In *ICCV*.

Huang, Z.; Ben, Y.; Luo, G.; Cheng, P.; Yu, G.; and Fu, B. 2021. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017. Adversarial machine learning at scale. In *ICLR*.

Li, Y.; Zhang, K.; Cao, J.; Timofte, R.; and Van Gool, L. 2021. LocalViT: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*.

Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. 2022a. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*.

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022b. A ConvNet for the 2020s. In *CVPR*.

Minderer, M.; Djolonga, J.; Romijnders, R.; Hubis, F.; Zhai, X.; Houlsby, N.; Tran, D.; and Lucic, M. 2021. Revisiting the calibration of modern neural networks. In *NIPS*.

Montufar, G. F.; Pascanu, R.; Cho, K.; and Bengio, Y. 2014. On the number of linear regions of deep neural networks. In *NIPS*.

Naeini, M. P.; Cooper, G.; and Hauskrecht, M. 2015. Obtaining well calibrated probabilities using Bayesian binning. In *AAAI*.

Naseer, M.; Ranasinghe, K.; Khan, S.; Hayat, M.; Khan, F.; and Yang, M.-H. 2021. Intriguing properties of vision transformers. In *NIPS*.

Novak, R.; Bahri, Y.; Abolafia, D. A.; Pennington, J.; and Sohl-Dickstein, J. 2018. Sensitivity and generalization in neural networks: An empirical study. In *ICLR*.

Park, N.; and Kim, S. 2022. How do vision transformers work? In *ICLR*.

Paul, S.; and Chen, P.-Y. 2022. Vision transformers are robust learners. In *AAAI*.

Price, I.; and Tanner, J. 2021. Trajectory growth lower bounds for random sparse deep ReLU networks. In *ICML*.

Raghu, M.; Poole, B.; Kleinberg, J.; Ganguli, S.; and Sohl-Dickstein, J. 2017. On the expressive power of deep neural networks. In *ICML*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; ; and Fei-Fei, L. 2015. ImageNet large scale visual recognition challenge. *IJCV*, 115(3): 211–252.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR*.

Serra, T.; Tjandraatmadja, C.; and Ramalingam, S. 2018. Bounding and counting linear regions of deep neural networks. In *ICML*.

Sokolić, J.; Giryes, R.; Sapiro, G.; and Rodrigues, M. R. 2017. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*.

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

Thulasidasan, S.; Chennupati, G.; Bilmes, J. A.; Bhattacharya, T.; and Michalak, S. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *NIPS*.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *ICML*.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*.

Wen, Y.; Jerfel, G.; Muller, R.; Dusenberry, M. W.; Snoek, J.; Lakshminarayanan, B.; and Tran, D. 2021. Combining ensembles and data augmentation can harm your calibration. In *ICLR*.

Xiao, T.; Singh, M.; Mintun, E.; Darrell, T.; Dollár, P.; and Girshick, R. 2021. Early convolutions help transformers see better. In *NIPS*.

Yang, J.; Li, C.; Zhang, P.; Dai, X.; Xiao, B.; Yuan, L.; and Gao, J. 2021. Focal attention for long-range interactions in vision transformers. In *NIPS*.

Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.-H.; Tay, F. E.; Feng, J.; and Yan, S. 2021. Tokens-to-token ViT: Training vision transformers from scratch on ImageNet. In *ICCV*.

Zhang, X.; and Wu, D. 2020. Empirical studies on the properties of linear regions in deep neural networks. In *ICLR*.