An Impartial Take to the CNN vs Transformer Robustness Contest

Francesco Pinto[1,2], Philip H. S. Torr[1], and Puneet K. Dokania[1,2]
{francesco.pinto}@eng.ox.ac.uk

[1]University of Oxford & [2]Five AI Ltd., UK

Abstract. Following the surge of popularity of Transformers in Computer Vision, several studies have attempted to determine whether they could be more robust to distribution shifts and provide better uncertainty estimates than Convolutional Neural Networks (CNNs). The almost unanimous conclusion is that they are, and it is often conjectured more or less explicitly that the reason of this supposed superiority is to be attributed to the self-attention mechanism. In this paper we perform extensive empirical analyses showing that recent state-of-the-art CNNs (particularly, ConvNeXt [20]) can be as robust and reliable or even sometimes more than the current state-of-the-art Transformers. However, there is no clear winner. Therefore, although it is tempting to state the definitive superiority of one family of architectures over another, they seem to enjoy similar extraordinary performances on a variety of tasks while also suffering from similar vulnerabilities such as texture, background, and simplicity biases.

Keywords: Transformers, CNNs, Robustness, Calibration

1 Introduction

Transformers are a family of neural network architectures that became extremely popular in natural language processing, and are primarily characterised by the extensive use of the attention mechanisms as defined in [37]. Before Vision Transformers (ViT) [8] were introduced, Transformers were considered difficult to use for computer vision applications due to the prohibitive computational complexity and memory requirements of the self-attention mechanism. Since then, several transformer variants that are efficient to train with performance more competitive with the state-of-the-art CNNs like BiT [17] (e.g. [19, 36, 41]) have been proposed.

The effectiveness of transformers compared to CNNs in computer vision applications has led to recent interest in comparing them in obtaining reliable predictive uncertainty and robustness to distribution shifts. The almost unanimous conclusion in the literature is that transformers exhibit: (1) better calibration [22], (2) better robustness to covariate shift [3, 23, 28, 42], and (3) better uncertainty estimation for tasks like out-of-distribution detection (OoD) [3, 9].

Currently, these conclusions are mostly misleading as (1) the recent convolutional architectures (ConNeXt) were not available for proper comparisons; (2) the comparisons are often performed with questionable assumptions (e.g. comparing model capacity solely based on their parameter count) or training procedures (e.g. trying to make the training as similar as possible for both the families at the cost of damaging the performance of either); and (3) the choice of the evaluation metrics is often not carefully justified and the most subtle aspects of the interpretation of the results were not identified. Additionally, when it comes to explaining the outcome of the analysis, which mostly leads to concluding that Transformers are superior, the credit is often given (more or less explicitly) to the most prominent feature that distinguishes Transformers from CNNs: the self-attention mechanism. Yet, a fair comparison and an understanding of whether and how self-attention modules would allow learning superior features compared to convolutional models is needed before providing a definitive answer regarding the superiority of one over another.

Taking a step in this direction, we thoroughly evaluate the robustness and reliability of most recent state-of-the-art Transformers (ViT [8] and SwinT [19]) and CNN architectures (BiT [17] and ConvNeXt [20]) on ImageNet-1K [6]. We would like to highlight that we do not modify the training recipes of CNNs and Transformers to ensure that they are at their current best during comparisons. The main takeaways of our work are:

1. **Simplicity bias experiment** [34]. Transformers, just like CNNs, also suffer from the so-called simplicity bias. They are somewhat similar to CNNs in finding shortcuts (undesirable) to solve the desired task. Therefore, as opposed to the common notion, despite the capability of the self-attention modules to communicate globally, Transformers as well tend to focus on easy-to-discriminate parts of the input and conveniently ignore other complex-yet-discriminative ones. Hence, similar to CNNs, they might just be learning to combine sets of simple and potentially spurious features, rather than more complex and invariant ones. Based on this experiment, we discourage the common trend in the literature to give unnecessary praise to the self-attention module of Transformers anytime these perform better against CNNs. More theoretical developments, analyses, and well-thought experiments are needed to support such claims.
2. We show that for out-of-distribution detection task, CNNs and Transformers **perform equally well**. We also highlight why, unless domain-specific assumptions are made, preferring AURP over AUROC in situations of data imbalance (which generally is the case) might give the false impression of one model being significantly superior to others.
3. In-distribution calibration of the best performing CNN model (in terms of accuracy) is better than the best performing Transformer. However, there is **no clear winner** that performs the best in all the experiments including covariate shift.
4. Again, there is **no clear winner** in detecting misclassified inputs.

These takeaways also suggest that the inductive biases induced in CNNs by using the design components popularised by Transformers (e.g. GeLU [13] activations, LN normalization [2] etc.), but without using the self-attention mechanism, might be highly effective in bridging the gap between the two in terms of robustness. However, this speculation requires further analysis as there are too many variables involved in designing a model (from architectural design choices to optimization algorithms) and the interplay between them is not well understood yet.

2 Experimental Design and Choices

2.1 Setup

Models. We consider state-of-the-art convolutional and non-convolutional models for our analysis.

1. **BiT** [17]: It is a very commonly used family of fully convolutional architectures. Its members are ResNet variants that have been shown to achieve state-of-the-art accuracy on ImageNet classification and that, with an appropriate fine-tuning procedure, transfer well to many other datasets. In this paper we consider BiT-R50x1, BiT-R50x3, BiT-R101x1, BiT-R101x3, BiT-R152x2, BiT-R152x4 (where R50/101/152 indicates the ResNet variant, and the multiplicative factor scales the number of channels).
2. **ConvNeXt** [20]: A recent family of fully convolutional architecture that is very close to the non-convolutional Transformer models in terms of training recipes and design choices. Its members have been shown to produce either comparable or superior performance to Transformers on several large-scale datasets. ConvNeXt exemplifies how advancing state-of-the-art in one family of networks can yield architecture design choices that, if adapted properly, can benefit other families of networks too. Our conclusions heavily rely on the careful architecture design process of ConvNeXt. We consider ConvNeXt-B, ConvNeXt-L, ConvNeXt-XL variants. Here and also for other models, B, L and XL indicate the capacity (B = Base, L = Large, XL = Extra Large).
3. **ViT** [8]: First successful use of Transformers on vision tasks. Its members still exhibit state-of-the-art performances. We consider ViT-B/16 and ViT-L/16[1], where 16 indicates the input token patch size.
4. **SwinTransformer** [19]: A family of transformers implementing a hierarchical architecture employing a shifting window mechanism. We consider the Swin-B and Swin-L variants. We use patch size of 4 pixels and shifted windows of size 7 as they provide highly competitive performance.

Training. Unless stated otherwise, all the considered architectures have been pre-trained on ImageNet-21k [32] and fine-tuned on ImageNet-1k [6]. We use the

[1] We omit ViT-B/32 ViT-L/32 as we find them to always underperform with respect to ViT-B/16 and ViT-L/16 (a similar observation was made in [28]). Similarly, we also omit DeiT [36] as it underperforms compared to SwinTransformers.

trained checkpoints available in the `timm library` [39] except only for the simplicity bias experiments where we fine-tune the models on our own. Additional results showing the impact of pre-training are shown in Appendix A.
Datasets. Since the in-distribution dataset is ImageNet-1K, we use ImageNet-A [14], ImageNet-R [12], ImageNetv2 [31], ImageNet-Sketch [38] for the *domain-shift* experiments. For *out-of-distribution* detection experiments, we use ImageNet-O [14]. For our preliminary analyses to understand *existing biases* in Transformers and CNNs, we use ImageNet9 [40], the Cue-Conflict Stimuli dataset [38], and also *synthesize* a dataset by combining MNIST and CIFAR-10 datasets. For Imagenet experiments, we apply the standard preprocessing pipeline. Additional results showing the impact of input preprocessing are shown in Appendix A.

2.2 Yet Another Analysis?

Before we begin discussing our analyses, we would like to mention how we differ from the existing ones.

Closest to ours is a recent analysis presented by [3] which involves rather simpler architectures for both Transformers (DeiT) and CNNs (ResNet-50), and also drops transformer-specific training techniques (for instance, reducing training epochs to 100 from 300, removing augmentations and regularisation techniques etc.). This indeed brings DeiT down to CNNs in terms of training procedure, however, makes DeiT underperform significantly. Although they derive interesting insights, the applicability of these insights for a practitioner with an intent to identify the most robust and best performing model is somehow limited. Therefore, we not only consider a wider variety of CNNs and Tranformers in our analysis, we also do not modify their standard training recipes so that their best performance is being compared. In [29], authors do provide a partial and preliminary analysis questioning the existing literature, however, solid evidence is still lacking. Another work [35] showed superiority of CNNs over Transformers on natural covariate-shift datasets. Differently from them, our analysis not only considers these metrics, but also the performance in terms of calibration, misclassification detection, and out-of-distribution detection. Other recent work [23, 42] performs partially overlapping analyses reaching the same conclusion about the superiority of Transformers. However, [23] do not consider recent CNN models, and also compare Transformers pre-trained on ImageNet-21K with CNNs that are trained from scratch on ImageNet-1K. Instead, [42] only compares with the extremely simple CNN variants.

We would also like to highlight that comparing different models based on their capacity (determined solely based on their number of parameters) might lead to wrong conclusions. How well a model would preform in practice is heavily dependent on the nature and the composition (hierarchy, depth etc.) of the underlying functions, not just on the number of parameters. To provide a widely known example, an MLP with one hidden layer and enough hidden units (large number of parameters) can theoretically fit most functions of interest, and it is known to be a universal function approximator [5, 15]. However, in practice, they underperform compared to a deep network (with same or even less number

of parameters). The interaction of inductive biases and training procedures plays an important role towards finding solutions that generalise well.

Therefore, although the number of parameters can be a proxy for comparing model capacity, in practice, it can be misleading. Indeed, when compute and memory constraints are imposed, a practitioner will always find the best performing model satisfying such constraints rather than choosing a model based on the parameter count[2]. We provide discussions and empirical findings (using standard complexity measures) to support our arguments above in Appendix B.

3 Empirical Evaluation and Analysis

3.1 Are Transformer Features More Robust than CNN ones?

There is no clear answer to this question in the literature. It is known that for a model to generalise to previously unseen domains, its predictions should not depend on spurious features that are specific to the distribution from which the training and test in-domain sets are sampled from, but on robust features that generalise across other domains under covariate-shift [30]. Typical examples of spurious features described in literature are the background's colour, textures and generally any simple pattern that correlates strongly with the labels in the training set but not in the test set [1].

It is usually conjectured in the literature that Transformers might be learning more robust features than CNNs because of the ability of their self-attention modules to communicate globally within a given input [28]. Which, in fact, is equivalent to implicitly criticizing the convolutional inductive biases of CNNs for their relatively poor robustness. Before we begin comparing these two families in terms of robustness, here we first present a few experiments to analyse their vulnerabilities. These experiments show that *the sole presence of the self-attention mechanism is not sufficient for Transformers to neglect spurious features*, and they result to be as biased as CNNs towards them.

Simplicity Bias Experiment. The intent of this experiment is to understand what Transformers and CNNs prefer to learn in situations where it is possible to focus only on the simple discriminative features of the input and ignore the complex discriminative ones in order to perform well on the task. This experiment was proposed and analysed on CNNs by [34]. Following their work, we first create a binary classification task where the input $X = [\mathbf{x}, \bar{\mathbf{x}}]$ is composed of the concatenation of $\mathbf{x}$ and $\bar{\mathbf{x}}$, both discriminative, and learning features for *either or both* will lead to an accurate classifier. We design this task such that, say, $\bar{\mathbf{x}}$ is more complex[3] than $\mathbf{x}$. Then, under this setting, a trained classifier suffers from simplicity bias if (1) fixing $\mathbf{x}$ and randomly modifying $\bar{\mathbf{x}}$ in the input

[2] Consider that ViT-L/32 has about 307M parameters, ViT-L/16 has 305M, yet ViT-L/32 requires about 15GFLOPS, while ViT-L/16 requires about 61GFLOPS, and ViT-L/32 exhibits lower accuracy and robustness than ViT-B/32 [28]

[3] We understand that defining complexity is subjective. Here we assume that something that is visually more complex (having more colors, shapes, textures etc.) across the training set would require learning more complex features.

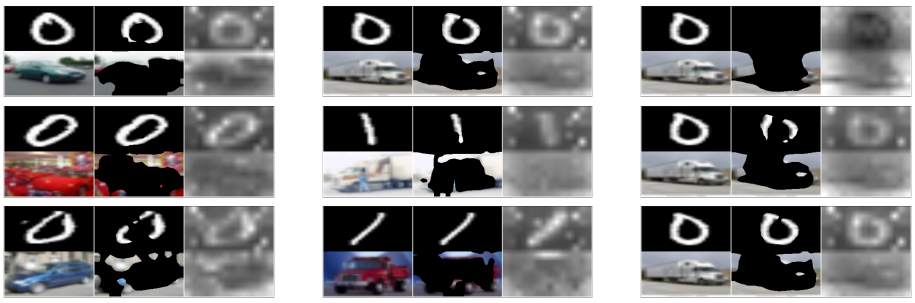
		SB			BB				TB
	# params (M)	In-domain	R-MNIST	R-CIFAR	O (↑)	MS (↑)	MR (↑)	BG-Gap (↓)	CCS(↑)
BiT-R50×1	25	100	48.39	100	94.57	83.21	76.2	7.00	31.09
BiT-R50×3	217	100	48.14	100	95.14	85.14	80.22	4.92	33.12
BiT-R101×1	44	100	48.50	99.94	94.17	81.28	75.19	6.09	32.81
BiT-R101×3	387	100	48.19	99.89	94.32	81.19	76.67	4.52	32.58
BiT-R152×2	232	100	48.39	99.94	94.64	80.05	75.09	4.95	35.47
BiT-R152×4	936	100	48.19	100	95.01	81.16	75.33	5.83	37.19
ConvNeXt-B	88	100	48.29	99.94	97.95	93.95	90.42	3.53	30.63
ConvNeXt-L	196	100	48.20	99.89	98.2	95.19	91.63	3.56	35.16
ConvNeXt-XL	348	100	48.75	99.69	98.49	95.23	92.3	2.93	36.95
ViT-B/16	86	100	48.59	99.79	97.36	92.35	88	4.34	30.78
ViT-L/16	304	100	52.79	95.66	98.02	94.05	90.05	4	47.19
Swin-B	87	100	48.75	99.64	97.75	90.94	86.47	4.47	26.95
Swin-L	195	100	48.69	99.74	98.02	92.99	88.47	4.52	30.08

Table 1: **Simplicity bias (SB), Background bias (BB) and Texture bias (TB) experiments**. For **SB**, in-domain indicates the accuracy when MNIST and CIFAR images are associated as in the training set. *A model suffers from SB if R-MNIST accuracy is close to random whereas R-CIFAR accuracy is close to the in-domain.* For **BB**, we report the absolute accuracy on the original (O), mixed-same (MS), and mixed-random (MR) datasets, respectively. **BG-Gap** defined as the difference in accuracy between MS and MR, quantifies the impact of background in producing correct classifications. For **TB** we report the CCS accuracy. All quantities in the table are percentages (%).

does not change its prediction, and (2) fixing $\bar{\mathbf{x}}$ and randomly modifying $\mathbf{x}$ in the input drops the test accuracy to the random prediction baseline.

To create the dataset for the above experiment, $\mathbf{x}$ is taken from the MNIST dataset [7] (randomly sampled image of a certain digit) while $\bar{\mathbf{x}}$ from the relatively more complex CIFAR-10 (randomly sampled image of a certain label). For instance, say digit **0** is associated to **car** and the whole concatenated image is assigned label $+1$, and digit **1** is associated to **truck** and the concatenated image is labelled -1. Refer to the top left of Figure 1. During training, this relationship holds true for all the examples (in-domain). We fine-tune our classifiers on this dataset for 3 epochs (it is easy to converge on this dataset). At test time, we either randomise the MNIST part of the image (R-MNIST) or the CIFAR part of the image (R-CIFAR) for the analysis. Results are reported in Table 1.

As it can be seen, the accuracy is almost the same for all the models (except in ViT-L/16) even if the CIFAR (more complex) part of the input is completely randomized (R-CIFAR). However, the accuracy drops to nearly random (50%) when the MNIST part of the input is randomized (R-MNIST). This shows that both families, Transformers and CNNs, rely on MNIST for classification and are agnostic to the CIFAR component. Hence, both are prone to simplicity bias. To understand which are the most prominent features leveraged by the Transformer, we visualize the pixels that fall above the 70% quantile of the intensity values in the attention map, and blacken the ones that fall below it in Figure 1. This figure confirms that Transformer's self-attention mechanism neglects complex features in favour of simple features. Figure 1 (c) also shows how the



(a) **Cars**, target label **-1** (b) **Trucks**, target label **1** (c) Top to bottom: attention at layer 1, 4 and 12.

Fig. 1: **Simplicity Bias Experiment for Transformers**: For each triplet of images, from left to right: input image, test image without pixels on which the attention value is below the 70% quantile and the attention map visualization. The attention maps show that the Transformer (ViT-B/16) gives high attention values to simple features and neglect the complex ones.

self-attention changes through the layers of the transformer. At the first layer there is no specific focus on the MNIST digit, but as the layers progress (i.e. as the features specialise to be useful for the classification), the attention values increase around the digit.

Reliance on Backgrounds and Texture. Here we measure the performance of several architectures on a benchmark that measures the reliance of features on backgrounds and textures: ImageNet9 [40] and the Cue-Conflict Stimuli [38].

The **ImageNet9** dataset selects a subset of labels and images from the original ImageNet dataset. In our experiments we measure the accuracy on the full images of the dataset (*original split*), images in which the background has been swapped with another image of same class (*mixed-same*), images in which the background has been swapped with another image of different class chosen at random (*mixed-random*). Sample images are provided in Appendix C. The authors of this dataset suggest taking the gap between the accuracy on mixed-same and mixed-random as a quantifier of the reliance on background information to produce accurate predictions. As it can be seen from Table 1, some of the highest capacity BiT models do not rely more on the background than ViT-B/16, SWIN-B and SWIN-L. ConvNeXt models rely on backgrounds even less than Transformers, suggesting that the self-attention mechanism might not be the only factor responsible for the difference observed between low-capacity ResNets and Transformers.

The **Cue-Conflict Stimuli** dataset alters the texture information of an image using style transfer: given an image of a certain class, it uses as style-image a sample from another class (sample images in Appendix C). The purpose is to deceive classifiers that overly rely on textures to make predictions. As it can be seen in Table 1, although the top performing model is ViT-L/16 (with a sig-

nificant margin), Swin Transformers exhibit an even heavier reliance on texture than ConvNeXt models, and ViT-B/16 performs comparably to ConvNeXt-B. This suggests that the sole presence of the self-attention in an architecture is not sufficient for the model to not be biased towards texture information.

Conclusion 1

- Transformers can leverage spurious features just like CNNs. They can be comparably prone to various biases such as simplicity bias, background bias, and texture bias. The sole presence of self-attention might not be sufficient to avoid such biases.

3.2 Out-of-Distribution Detection

Current notion in the literature is that Transformers are better than CNNs at detecting OoD samples [28].

We compare various CNN and Transformer models at the task of detecting ImageNet-O samples from ImageNet-1K. ImageNet-O contains 2K samples in 200 classes, while the subset of ImageNet-1K used as the corresponding in-distribution set contains 10K samples [14] (therefore, there is a stark imabalance in the number of samples belonging to the two sets). Both ImageNet-O and ImageNet-1K (test) samples are fed to the classifier, for each point an uncertainty score is computed and a binary threshold-based classifier is used to distinguish between them. Since the choice of the threshold depends on the risk exposure desired for a certain application, a standard evaluation procedure considers all the risk thresholds and computes the AUROC (Area Under the Receiver Operating Characteristic curve) and the AUPR (Area Under the Precision-Recall curve).

AUPR vs AUROC? We start by observing that the apparent complexity in distinguishing ImageNet-O samples from ImageNet-1K observed in the literature (e.g. [14, 28]) mostly depends on the interaction between specific evaluation choices. The AUPR, in the case of an imbalanced number of samples belonging to the positive and negative classes, is known to prefer one class over another. However, for out-of-distribution evaluation, unless additional domain-specific assumptions are made, there is no preferred mistake: confusing an in-distribution sample with an out-of-distribution sample or viceversa are both equally important mistakes. To exemplify why the AUPR can yield misleading conclusions, in Table 2 we consider different possible assignments of the positive class and apply a rebalancing technique as well. Recent work [14, 28] concluding that there exist a dramatic gap between CNNs and Transformers on OoD detection performance report values when OoD samples are considered as positives (third column from the right). In this setting, for instance, the performance of BiT-R50x1 is less than half of the performance of ViT-L/16, and extremely low (with respect to the attainable maximum of 100). However, only rebalancing the number of samples[4]

[4] We oversample OoD samples (4×) so that both in-distribution and OoD datasets have 10000 samples each. We could rebalance them also by randomly sampling 2000

	IND=1,OoD=0				IND=0,OoD=1			
	Imbalanced		**Balanced**		**Imbalanced**		**Balanced**	
	AUROC (↑)	AUPR (↑)	AUROC (↑)	AUPR (↑)	AUROC (↑)	AUPR (↑)	AUROC (↑)	AUPR (↑)
BiT-R50x1	65.17	90.15	65.17	65.81	65.17	23.30	65.17	60.13
BiT-R50x3	74.56	92.30	74.56	71.28	74.56	36.26	74.56	72.49
BiT-R101x1	70.34	91.35	70.34	68.75	70.34	28.53	70.34	66.11
BiT-R101x3	77.32	93.40	77.32	74.84	77.32	38.74	77.32	74.66
BiT-R152x2	77.46	93.51	77.46	75.23	77.46	38.24	77.46	74.43
BiT-R152x4	80.07	94.39	80.07	78.10	80.07	44.25	80.07	78.17
ConvNeXt-B	85.72	95.53	85.72	81.74	85.72	59.15	85.72	85.53
ConvNeXt-L	89.07	96.90	89.07	86.96	89.07	65.33	89.07	88.55
ConvNeXt-XL	<u>90.04</u>	97.19	<u>90.04</u>	88.11	<u>90.04</u>	68.50	<u>90.04</u>	89.75
ViT-B/16	79.89	95.26	79.89	82.30	79.89	36.77	79.89	73.77
ViT-L/16	**90.60**	97.85	**90.60**	91.27	**90.60**	64.58	**90.60**	88.90
Swin-B	83.74	95.29	83.74	81.01	83.74	52.93	83.74	82.80
Swin-L	87.76	96.55	87.76	85.67	87.76	62.51	87.76	87.27

Table 2: ImageNet-O: **OoD** performance analysis when in-distribution samples are assigned label 1 and OoD label 0, and vice-versa (with and without rebalancing). AUROC (%) is invariant whereas AUPR (%) is extremely sensitive to these design choices. The best performing method based on AUROC is in bold and the second best is underlined. The gap between the two is marginal.

the performance of BiT-R50x1 rises to more than two thirds of the performance of ViT-L/16 (last column on the right). Alternatively, if the choice of the positive and negative class is flipped, in an imbalance condition, one can obtain an absolute gap between the performance of BiT-R50x1 and ViT-L/16 of less than 8% (third column from the left). If one drew conclusions solely based on this column, one would think there is only a marginal difference between the performance of the two models. This gap widens when rebalancing the number of samples (fourth column from the left). This exemplifies how widely the AUPR can vary based on evaluation choices that, in the lack of domain-specific assumptions, are arbitrary. On the other hand, the AUROC does not vary across all the considered evaluation setups, because it gives the same importance to both types of errors that can occur. These results allow us to conclude that, for the considered models, ImageNet-O is evidently not as hard to distinguish from ImageNet as it is believed to be.

For completeness, in Appendix D we provide a proof to show that AUROC is invariant to the choice of positive and negative classes.

Comparing Transformers and CNNs From the AUROC values in Table 2 it is clear that the top-performing CNN (ConvNeXt-XL) is competitive to the top-performing Transformer (ViT-L/16). ConvNeXt-L outperforms Swin-L, and ConvNeXt-B outperforms Swin-B. The best performing BiT (BiT-R152×4) outperforms ViT-B.

out of the 10000 in-distribution samples, but this could induce some variance in the metrics; we also observed that the average of this strategy coincides with the balancing strategy.

Conclusion 2

- CNNs can perform as well as Transformers for OoD detection.
- With no domain-specific assumptions regarding the importance of one category over another (in-distribution vs OoD), AUROC should be preferred over AUPR as it is stable across evaluation choices.

3.3 Calibration on In-Distribution and Domain-Shift

A model is said to be calibrated if its confidence (i.e. the maximum probability score of the softmax output) and its accuracy match. The idea is to attribute to the confidence the frequentist probabilistic meaning of counting the amount of times the model is correct. Several measures have been proposed targeted specifically towards quantifying the said mismatch between a classifier's confidence and its accuracy. These measures are primarily the variants of the well-known Expected Calibration Error (ECE) [25] such as the recently proposed Adaptive Calibration Error (AdaECE) [24].

Comparing Transformers and CNNs On in-domain data (Table 3), ViTs produce the lowest calibration error and Swin transformers are outperformed by ConvNeXts. On covariate-shifted inputs (Table 4), ViTs produce higher calibration error than ConvNeXts and Swin transformers, and the model producing the lowest calibration error is the Swin-L. Consistently with [22], within a family of models, the ECE typically decreases as the number of parameters (and also the accuracy) increases.

	ImageNet-1K (Test)		
	Acc (↑)	**ECE** (↓)	**AdaECE** (↓)
BiT-R50x1	74.03	3.49	3.45
BiT-R50x3	77.92	6.56	6.51
BiT-R101x1	75.85	5.10	5.10
BiT-R101x3	78.20	7.63	7.63
BiT-R152x2	78.00	6.37	6.37
BiT-R152x4	78.16	9.38	9.38
ConvNeXt-B	85.53	2.87	2.82
ConvNeXt-L	86.29	2.27	2.34
ConvNeXt-XL	**86.58**	2.38	2.29
ViT-B/16	78.01	**1.40**	**1.41**
ViT-L/16	84.38	1.81	1.83
SWIN-B	84.71	8.40	8.40
SWIN-L	85.83	5.50	5.50

Table 3: **In-distribution** accuracy (%) and **calibration** (%) for ImageNet-1K.

	Domain-Shift											
	ImageNet-R			ImageNet-A			ImageNet-V2			ImageNet-SK		
	Acc (↑)	ECE (↓)	AdaECE (↓)	Acc (↑)	ECE (↓)	AdaECE (↓)	Acc (↑)	ECE (↓)	AdaECE (↓)	Acc (↑)	ECE (↓)	AdaECE (↓)
BiT-R50x1	39.87	15.50	15.50	10.97	42.94	42.94	62.70	8.49	8.45	27.34	24.87	24.87
BiT-R50x3	46.39	14.65	14.65	24.08	34.48	34.48	66.36	13.13	13.13	33.47	28.55	28.55
BiT-R101x1	41.72	12.24	12.24	16.29	36.68	36.68	64.61	10.21	10.21	28.69	24.37	24.37
BiT-R101x3	47.00	15.80	15.80	27.11	32.92	32.92	66.44	14.44	14.39	34.15	30.67	30.67
BiT-R152x2	48.02	15.38	15.38	27.15	32.25	32.25	66.76	12.14	12.13	35.70	28.41	28.41
BiT-R152x4	47.57	15.32	15.32	30.84	29.93	29.93	67.12	15.75	15.67	35.08	31.45	31.45
ConvNeXt-B	62.46	2.57	2.51	52.63	8.28	8.31	75.43	**2.91**	**2.78**	48.64	8.85	8.84
ConvNeXt-L	64.57	3.00	3.08	58.23	7.57	7.26	76.77	3.72	3.85	50.08	10.31	10.31
ConvNeXt-XL	**66.01**	2.92	2.90	**61.11**	7.54	7.21	**77.20**	4.00	4.24	**52.67**	11.16	11.15
ViT-B16	43.15	5.21	5.21	24.17	22.89	22.89	66.25	4.71	4.68	18.18	13.02	13.02
ViT-L16	61.54	3.07	3.07	47.08	11.99	11.99	74.28	5.34	5.22	45.96	10.67	10.67
Swin-B	59.63	2.18	2.17	49.72	8.77	8.76	74.74	4.92	4.81	45.07	**7.75**	**7.75**
Swin-L	64.24	**2.14**	**2.11**	59.52	**6.19**	**6.33**	76.65	3.03	3.14	48.87	8.72	8.71

Table 4: **Domain-shift** accuracy (%) and **calibration** (%) for ImageNet-1K.

Conclusion 3

- There is no one model that performs the best in all the covariate shift experiments in terms of calibration. Transformers or CNNs either can be better or worse depending on the experiment.
- The best performing model in terms of accuracy is not the most calibrated one.

Is Low Calibration Error Enough for a Classifier to be Reliable? A perfectly calibrated classifier can still be highly inaccurate and unreliable. For example, consider the binary case where there are 70 negative test samples and 30 positives. A classifier that has learned to classify every sample to a negative class with a confidence of 0.7 will be perfectly calibrated, however, only 70% accurate. Since neural networks trained on cross-entropy loss are known to be overconfident [11], even if we somehow manage to calibrate them well, they still might be assigning higher confidence to the wrongly predicted samples than the correctly predicted ones. If the minority class samples (positives in the above example) are as important as the majority ones, this behaviour raises concerns relating to their reliability. Analysing and quantifying such behaviour is necessary to complement our understanding in terms of the reliability of neural networks. In the next section, we discuss this aspect as well.

3.4 Misclassified Input Detection

One of the tasks a reliable classifier should be good at is to reject samples on which they are likely to be wrong. This particular task did not receive much attention in the Transformers vs CNNs comparisons performed by the existing literature. Several ways to evaluate a model at this task are available (e.g. metrics based on ROC [18] or Rejection-Accuracy curves [10, 14]), however, it has already been observed that these metrics favour models that have higher test accuracy [4, 21]. A recently proposed metric that allows comparison of different models in this aspect, agnostic to their individual accuracy, is Prediction Rejection Ratio (PRR) [21]. The sign of this metric is an indication of whether a model tend to

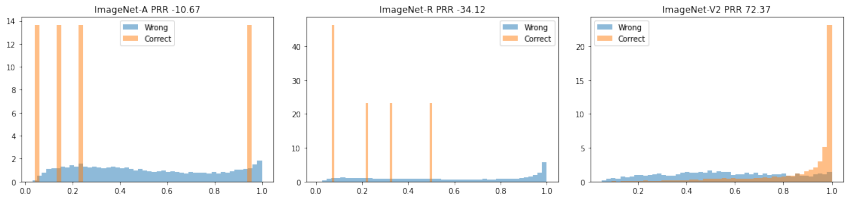


Fig. 2: From left to right: the distribution of the confidence values for wrong and right samples for ViT-L/16 on ImageNet-A, ImageNet-R and ImageNet-V2. As it can be seen, in several cases wrong samples are given higher confidence than correct samples (PRR < 0). In cases when PRR > 0, some wrong samples are still given higher confidences (similar to correct samples), but to a lesser extent.

provide lower confidence to correctly classified samples and higher to wrongly classified ones or not. The PRR ranges from -1 to 1. It is 0 if the rejection choice is performed at random, negative if the network is more confident on misclassified samples than on correctly classified ones, and positive viceversa. The optimal value of 1 is achieved when the classifier rejects only the misclassified samples while rejecting the most uncertain ones.

For instance, consider Figure 2 where we show the distribution of the confidence values for samples that a ViT-L/16 wrongly and correctly classified on a few datasets. As observed, in many cases the network is more confident on wrongly classified samples than on the correctly classified ones. This is captured by the sign of PRR (reported in %). However, the corresponding miscalibration error values as shown in Table 4 are particularly low (especially on ImageNet-R). Therefore, as discussed in Section 3.3, low miscalibration solely can be misleading in providing a deep understanding of the reliability of different models.

	In-Distribution	**Domain-Shift**			
	ImageNet-1K (Test)	**ImageNet-A**	**ImageNet-R**	**ImageNet-SK**	**ImageNet-V2**
		PRR (↑)			
BiT-R50x1	68.38	54.90	-25.60	58.70	63.13
BiT-R50x3	67.61	28.58	-42.72	60.25	64.09
BiT-R101x1	69.82	-0.42	-25.94	60.08	64.60
BiT-R101x3	68.93	29.50	-34.52	60.13	65.00
BiT-R152x2	68.03	31.56	-35.12	59.26	63.26
BiT-R152x4	67.00	**92.04**	-46.05	59.34	61.48
ConvNeXt-B	73.43	16.03	-39.91	67.44	69.84
ConvNeXt-L	73.48	40.56	-23.60	69.03	69.50
ConvNeXt-XL	74.37	35.96	-19.32	69.29	70.07
ViT-B16	74.17	11.54	-46.01	63.94	70.51
ViT-L16	**76.03**	-10.67	-34.12	**69.79**	**72.37**
Swin-B	72.04	32.65	-32.95	64.23	67.35
Swin-L	72.89	56.54	**36.53**	63.52	68.49

Table 5: **Misclassification detection** results using the PRR (%) metric.

Comparing Transformers and CNNs As it can be seen form Table 5, in in-distribution ViT-L/16 is the best model, immediately followed by ConvNeXt-XL. ViT-B/16 slightly outperforms ConvNeXt-B and L, which in turn outperform Swin-B and L. On ImageNet-A, the best model is BiT-R152x4, with a significant margin with respect to any other model. The second best model is Swin-L, and the third best is BiT-R50x1. On ImageNet-R, the only model with positive PRR is Swin-L, and the models with highest negative PRR are ConvNeXt-XL and L, followed by BiT-R50x1 and 101x1. On ImageNet-Sketches ViT-L/16 and ConvNeXt-XL perform comparably, immediately followed by ConvNeXt-L and B. On ImageNetV2, ViT-L/16 is the best model, immediately followed by ViT-B/16 and all the ConvNeXts.

Conclusion 4

- No single model is the winner in detecting misclassified samples.
- The fact that several models are severely overconfident and wrong on ImageNet-R (PRR) while showing low calibration errors indicate that the calibration analysis should be complemented with experiments such as misclassification detection to understand their reliability.

4 Concluding Remarks

We performed an extensive analysis comparing current state-of-the-art Transformers and CNNs. With simple experiments, we have shown that Transformers, just like CNNs, are vulnerable to picking spurious or simple discriminative features in the training set instead of focusing on robust features that generalise under covariate shift conditions. Therefore, the presence of the self-attention mechanism might not be facilitating learning more complex and robust features. To show it is not even necessary, we observed that ConvNeXt models exhibit even superior robustness with respect to current Transformers without leveraging the self-attention mechanism in a few cases. We also conducted an in-depth analysis about the out-of-distribution, calibration, and misclassification detection properties of these models. We hope that our work will encourage development of modules within Transformers and CNNs that can avoid various biases. Additionally, our analysis in Appendix B regarding the lack of reliable metrics to quantify a model's capacity to open new avenues for future work.

Acknowledgements

This work is supported by the UKRI grant: Turing AI Fellowship EP/W002981/1 and EPSRC/MURI grant: EP/N019474/1. We would like to thank the Royal Academy of Engineering and FiveAI. Francesco Pinto's PhD is funded by the European Space Agency (ESA). PD would like to thank Anuj Sharma and Kemal Oksuz for their comments on the draft.

Bibliography

[1] Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant Risk Minimization. arXiv e-prints arXiv:1907.02893 (Jul 2019)

[2] Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)

[3] Bai, Y., Mei, J., Yuille, A., Xie, C.: Are transformers more robust than cnns? NeurIPS (2021)

[4] Condessa, F., Kovacevic, J., Bioucas-Dias, J.: Performance measures for classification systems with rejection. Pattern Recognition (2015)

[5] Cybenko, G.: Approximation by superpositions of a sigmoidal function. Math. Control Signal Systems 2, 303–314 (1989)

[6] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 CVPR. pp. 248–255 (2009)

[7] Deng, L.: The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine **29**(6), 141–142 (2012)

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

[9] Fort, S., Ren, J., Lakshminarayanan, B.: Exploring the limits of Out-of-Distribution detection. NeurIPS (2021)

[10] Fumera, G., Roli, F.: Support vector machines with embedded reject option. In: Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines. p. 68–82. SVM '02, Springer-Verlag, Berlin, Heidelberg (2002)

[11] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. p. 1321–1330. ICML'17, JMLR.org (2017)

[12] Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization. ICCV (2021)

[13] Hendrycks, D., Gimpel, K.: Bridging nonlinearities and stochastic regularizers with gaussian error linear units. CoRR **abs/1606.08415** (2016), http://arxiv.org/abs/1606.08415

[14] Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. CVPR (2021)

[15] Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks **2**(5), 359–366 (1989)

[16] Jiang*, Y., Neyshabur*, B., Mobahi, H., Krishnan, D., Bengio, S.: Fantastic generalization measures and where to find them. In: ICLR (2020)

[17] Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., Houlsby, N.: Big transfer (BiT): General visual representation learning. ECCV (2020)

[18] Landgrebe, T.C.W., Tax, D.M.J., Paclík, P., Duin, R.P.W.: The interaction between classification and reject performance for distance-based reject-option classifiers. Pattern Recogn. Lett. **27**(8), 908–917 (jun 2006)
[19] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. ICCV (2021)
[20] Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A ConvNet for the 2020s. CVPR (2022)
[21] Malinin, A., Mlodozeniec, B., Gales, M.: Ensemble distribution distillation. ICLR (2020)
[22] Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., Lucic, M.: Revisiting the calibration of modern neural networks. NeurIPS (2021)
[23] Morrison, K., Gilby, B., Lipchak, C., Mattioli, A., Kovashka, A.: Exploring corruption robustness: Inductive biases in vision transformers and mlp-mixers. vol. abs/2106.13122 (2021), https://arxiv.org/abs/2106.13122
[24] Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P.H., Dokania, P.K.: Calibrating deep neural networks using focal loss. NeurIPS (2020)
[25] Naeini, M.P., Cooper, G.F., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. Proc. Conf. AAAI Artif. Intell. **2015**, 2901–2907 (Jan 2015)
[26] Neyshabur, B., Bhojanapalli, S., Mcallester, D., Srebro, N.: Exploring generalization in deep learning. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) NeurIPS. vol. 30. Curran Associates, Inc. (2017)
[27] Neyshabur, B., Bhojanapalli, S., Srebro, N.: A PAC-bayesian approach to spectrally-normalized margin bounds for neural networks. In: ICLR (2018)
[28] Paul, S., Chen, P.Y.: Vision transformers are robust learners. AAAI (2022)
[29] Pinto, F., Torr, P., Dokania, P.: Are vision transformers always more robust than convolutional neural networks? NeurIPS Workshop on Distribution Shifts: Connecting Methods and Applications (2021)
[30] Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset Shift in Machine Learning. The MIT Press (2009)
[31] Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? ICML (2019)
[32] Ridnik, T., Ben-Baruch, E., Noy, A., Zelnik-Manor, L.: Imagenet-21k pretraining for the masses (2021)
[33] Sanyal, A., Torr, P.H.S., Dokania, P.K.: Stable rank normalization for improved generalization in neural networks and GANs. ICLR (2020)
[34] Shah, H., Tamuly, K., Raghunathan, A., Jain, P., Netrapalli, P.: The pitfalls of simplicity bias in neural networks. NeurIPS (2020)
[35] Tang, S., Gong, R., Wang, Y., Liu, A., Wang, J., Chen, X., Yu, F., Liu, X., Song, D., Yuille, A., Torr, P.H., Tao, D.: Robustart: Benchmarking robustness on architecture design and training techniques. arxiv (2021)
[36] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. ICML (2021)

[37] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł.U., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017)
[38] Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning robust global representations by penalizing local predictive power. NeurIPS pp. 10506–10518 (2019)
[39] Wightman, R.: Pytorch image models. https://github.com/rwightman/pytorch-image-models (2019)
[40] Xiao, K., Engstrom, L., Ilyas, A., Madry, A.: Noise or signal: The role of image backgrounds in object recognition. ICLR (2021)
[41] Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F.E.H., Feng, J., Yan, S.: Tokens-to-Token ViT: Training vision transformers from scratch on ImageNet. ICCV 2021
[42] Zhang, C., Zhang, M., Zhang, S., Jin, D., Zhou, Q., Cai, Z., Zhao, H., Yi, S., Liu, X., Liu, Z.: Delving deep into the generalization of vision transformers under distribution shifts. CVPR (2022)