ImageNet-E: Benchmarking Neural Network Robustness via Attribute Editing

Xiaodan Li[1], Yuefeng Chen[1*], Yao Zhu[2], Shuhui Wang[3*], Rong Zhang[1], Hui Xue[1]
[1]Alibaba Group [2]Zhejiang University [3]Inst. of Comput. Tech., CAS, China
{fiona.lxd, yuefeng.chenyf, stone.zhangr, hui.xueh}@alibaba-inc.com
ee_zhuy@zju.edu.cn, wangshuhui@ict.ac.cn

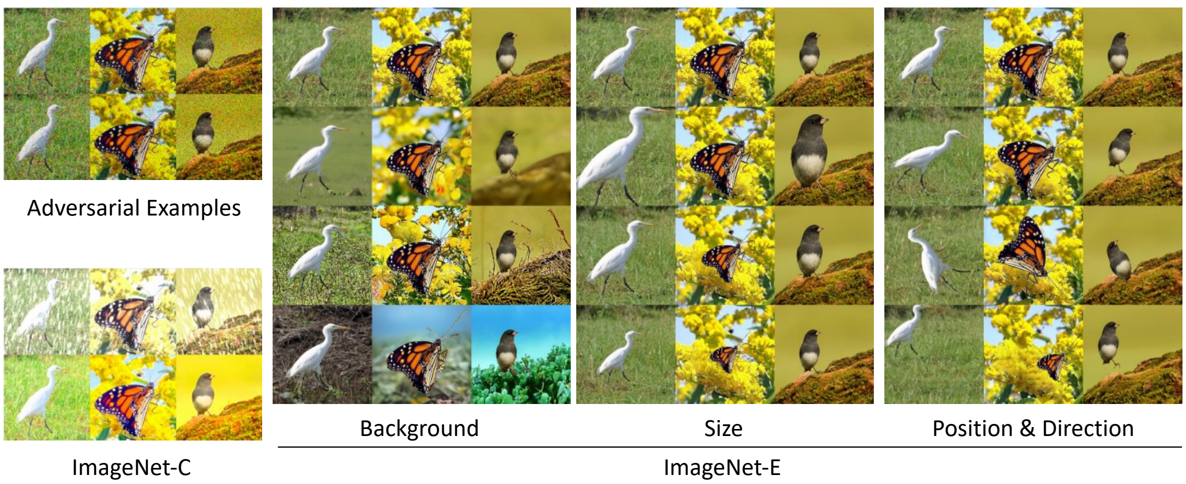

Figure 1. Examples of the proposed ImageNet-E dataset. In contrast to adversarial examples or datasets like ImageNet-C [22] who add perturbation or corruptions to original images, we edit the object attributes with controls of backgrounds, sizes, positions and directions.

Abstract

Recent studies have shown that higher accuracy on ImageNet usually leads to better robustness against different corruptions. Therefore, in this paper, instead of following the traditional research paradigm that investigates new out-of-distribution corruptions or perturbations deep models may encounter, we conduct model debugging in in-distribution data to explore which object attributes a model may be sensitive to. To achieve this goal, we create a toolkit for object editing with controls of backgrounds, sizes, positions, and directions, and create a rigorous benchmark named ImageNet-E(diting) for evaluating the image classifier robustness in terms of object attributes. With our ImageNet-E, we evaluate the performance of current deep learning models, including both convolutional neural networks and vision transformers. We find that most models are quite sensitive to attribute changes. A small change in the background can lead to an average of 9.23% drop on top-1 accuracy. We also evaluate some robust models including both adversarially trained models and other robust trained models and find that some models show worse robustness against attribute changes than vanilla models. Based on these findings, we discover ways to enhance attribute robustness with preprocessing, architecture designs, and training strategies. We hope this work can provide some insights to the community and open up a new avenue for research in robust computer vision. The code and dataset are available at https://github.com/alibaba/easyrobust.

∗ Corresponding author.
This research is supported in part by the National Key Research and Development Progrem of China under Grant No.2020AAA0140000.

1. Introduction

Deep learning has triggered the rise of artificial intelligence and has become the workhorse of machine intelligence. Deep models have been widely applied in various fields such as autonomous driving [28], medical science [33], and finance [38]. With the spread of these techniques, the robustness and safety issues begin to be essential, especially after the finding that deep models can be easily fooled by negligible noises [16]. As a result, more researchers contribute to building datasets for benchmarking model robustness to spot vulnerabilities in advance.

Most of the existing work builds datasets for evaluating the model robustness and generalization ability on out-of-distribution data [7, 22, 30] using adversarial examples and common corruptions. For example, the ImageNet-C(orruption) dataset conducts visual corruptions such as Gaussian noise to input images to simulate the possible processors in real scenarios [22]. ImageNet-R(enditions) contains various renditions (*e.g.*, paintings, embroidery) of ImageNet object classes [21]. As both studies have found that higher accuracy on ImageNet usually leads to better robustness against different domains [22,50]. However, most previous studies try to achieve this in a top-down way, such as architecture design, exploring a better training strategy, *etc*. We advocate that it is also essential to manage it in a bottom-up way, that is, conducting model debugging with the in-distribution dataset to provide clues for model repairing and accuracy improvement. For example, it is interesting to explore whether a bird with a water background can be recognized correctly even if most birds appear with trees or grasses in the training data. Though this topic has been investigated in studies such as causal and effect analysis [9], the experiments and analysis are undertaken on domain generalization datasets. How a deep model generalizes to different backgrounds is still unknown due to the vacancy of a qualified benchmark. Therefore, in this paper, we provide a detached object editing tool to conduct the model debugging from the perspective of object attribute and construct a dataset named ImageNet-E(diting).

The ImageNet-E dataset is a compact but challenging test set for object recognition that contains controllable object attributes including backgrounds, sizes, positions and directions, as shown in Fig. 1. In contrast to ObjectNet [5] whose images are collected by their workers via posing objects according to specific instructions and differ from the target data distribution. This makes it hard to tell whether the degradation comes from the changes of attribute or distribution. Our ImageNet-E is automatically generated with our object attribute editing tool based on the original ImageNet. Specifically, to change the object background, we provide an object background editing method that can make the background simpler or more complex based on diffusion models [25, 46]. In this way, one can easily evaluate how much the background complexity can influence the model performance. To control the object size, position, and direction to simulate pictures taken from different distances and angles, an object editing method is also provided. With the editing toolkit, we apply it to the large-scale ImageNet dataset [42] to construct our ImageNet-E(diting) dataset. It can serve as a general dataset for benchmarking robustness evaluation on different object attributes.

With the ImageNet-E dataset, we evaluate the performance of current deep learning models, including both convolutional neural networks (CNNs), vision transformers as well as the large-scale pretrained CLIP [40]. We find that deep models are quite sensitive to object attributes. For example, when editing the background towards high complexity (see Fig. 1, the 3rd row in the background part), the drop in top-1 accuracy reaches 9.23% on average. We also find that though some robust models share similar top-1 accuracy on ImageNet, the robustness against different attributes may differ a lot. Meanwhile, some models, being robust under certain settings, even show worse results than the vanilla ones on our dataset. This suggests that improving robustness is still a challenging problem and the object attributes should be taken into account. Afterward, we discover ways to enhance robustness against object attribute changes. The main contributions are summarized as follows:

- We provide an object editing toolkit that can change the object attributes for manipulated image generation.
- We provide a new dataset called ImageNet-E that can be used for benchmarking robustness to different object attributes. It opens up new avenues for research in robust computer vision against object attributes.
- We conduct extensive experiments on ImageNet-E and find that models that have good robustness on adversarial examples and common corruptions may show poor performance on our dataset.

2. Related Work

The literature related to attribute robustness benchmarks can be broadly grouped into the following themes: robustness benchmarks and attribute editing datasets. Existing robustness benchmarks such as ImageNet-C(orruption) [22], ImageNet-R(endition) [21], ImageNet-Stylized [14] and ImageNet-3DCC [30] mainly focus on the exploration of the corrupted or out-of-distribution data that models may encounter in reality. For instance, the ImageNet-R dataset contains various renditions (*e.g.*, paintings, embroidery) of ImageNet object classes. ImageNet-C analyzes image models in terms of various simulated image corruptions (*e.g.*, noise, blur, weather, JPEG compression, *etc.*). Attribute editing dataset creation is a new topic and few studies have explored it before. Among them, ObjectNet [5] and ImageNet-9 (*a.k.a.* background challenge) [50] can be the representative. Specifically, ObjectNet collects a large real-world test set for object recognition with controls where object backgrounds, rotations, and imaging viewpoints are random. The images in ObjectNet are collected by their workers who image objects in their homes. It consists of 313 classes which are mainly household objects. ImageNet-9 mainly creates a suit of datasets that help disentangle the impact of foreground and background signals on classification. To achieve this goal, it uses coarse-grained classes with corresponding rectangular bounding boxes to remove the foreground and then paste the cut area with other backgrounds. It can be observed that there lacks a dataset that can smoothly edit the object attribute.

3. Preliminaries

Since the editing tool is developed based on diffusion models, let us first briefly review the theory of denoising diffusion probabilistic models (DDPM) [25,46] and analyze how it can be used to generate images.

According to the definition of the Markov Chain, one can always reach a desired stationary distribution from a given distribution along with the Markov Chain [15]. To get a generative model that can generate images from random Gaussian noises, one only needs to construct a Markov Chain whose stationary distribution is Gaussian distribution. This is the core idea of DDPM. In DDPM, given a data distribution $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, a forward noising process produces a series of latents $\mathbf{x}_1, ..., \mathbf{x}_T$ of the same dimensionality as the data $\mathbf{x}_0$ by adding Gaussian noise with variance $\beta_t \in (0, 1)$ at time t:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), s.t.\ 0 < \beta_t < 1, \tag{1}$$

where β_t is the diffusion rate. Then the distribution $q(\mathbf{x}_t|\mathbf{x}_0)$ at any time t is:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}, (1-\bar{\alpha}_t)\mathbf{I}),\ \mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{2}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_t)$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. It can be proved that $\lim_{t\to\infty} q(\mathbf{x}_t) = \mathcal{N}(0, \mathbf{I})$. In other words, we can map the original data distribution into a Gaussian distribution with enough iterations. Such a stochastic forward process is named as diffusion process since what the process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ does is adding noise to $\mathbf{x}_{t-1}$.

To draw a fresh sample from the distribution $q(\mathbf{x}_0)$, the Markov process is reversed. That is, beginning from a Gaussian noise sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, a reverse sequence is constructed by sampling the posteriors $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$. To approximate the unknown function $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, in DDPMs, a deep model p_θ is trained to predict the mean and the covariance of $\mathbf{x}_{t-1}$ given $\mathbf{x}_t$ instead. Then the $\mathbf{x}_{t-1}$ can be sampled from the normal distribution defined as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)). \tag{3}$$

In stead of inferring $\mu_\theta(\mathbf{x}_t, t)$ directly, [25] propose to predict the noise $\epsilon_\theta(\mathbf{x}_t, t)$ which was added to $\mathbf{x}_0$ to get $\mathbf{x}_t$ with Eq. (2). Then $\mu_\theta(\mathbf{x}_t, t)$ is:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right). \tag{4}$$

[25] keep the value of $\Sigma_\theta(\mathbf{x}_t, t)$ to be constant. As a result, given a sample $\mathbf{x}_t$ at time t, with a trained model that can predict the noise $\epsilon_\theta(\mathbf{x}_t, t)$, we can get $\mu_\theta(\mathbf{x}_t, t)$ according to Eq. (4) to reach the $\mathbf{x}_{t-1}$ with Equation (3) and eventually we can get to $\mathbf{x}_0$.

Previous studies have shown that diffusion models can achieve superior image generation quality compared to the current state-of-the-art generative models [1]. Besides, there have been plenty of works on utilizing the DDPMs to generate samples with desired properties, such as semantic image translation [37], high fidelity data generation from low-density regions [45], *etc*. In this paper, we also choose the DDPM adopted in [1] as our generator.

4. Attribute Editing with Diffusion Models and ImageNet-E

Most previous robustness-related work has focused on the important challenges of robustness on adversarial examples [7], common corruptions [22]. They have found that higher clean accuracy usually leads to better robustness. Therefore, instead of exploring a new corruption that models may encounter in reality, we pay attention to the model debugging in terms of object attributes, hoping to provide new insights to clean accuracy improvement. In the following, we describe our object attribute editing tool and the generated ImageNet-E dataset in detail.

4.1. Object Attribute Editing with Diffusion Models

Background editing. Most existing corruptions conduct manipulations on the whole image, as shown in Fig. 1. Compared to adding global corruptions that may hinder the visual quality, a more likely-to-happen way in reality is to manipulate the backgrounds to fool the model. Besides, it is shown that there exists a spurious correlation between labels and image backgrounds [13]. From this point, a background corruption benchmark is needed to evaluate the model's robustness. However, the existing background challenge dataset achieves background editing with copy-paste operation, resulting an obvious artifacts in generated images [50]. This may leave some doubts about whether the evaluation is precise since the dataset's distribution may have changed. To alleviate this concern, we adopt DDPM approach to incorporate background editing by adding a guiding loss that can lead to backgrounds with desired properties to make the generated images stay in/close to the original distribution. Specifically, we choose to manipulate the background in terms of texture complexity due to the hypothesis that an object should be observed more easily from simple backgrounds than from complicated ones. In general, the texture complexity can be evaluated with the gray-level co-occurrence matrix (GLCM) [17], which calculates the gray-level histogram to show the texture characteristic. However, the calculation of GLCM is non-differentiable, thus it cannot serve as the conditional guidance of image generation. We hypothesize that a complex image should contain more frequency components in its spectrum and higher amplitude

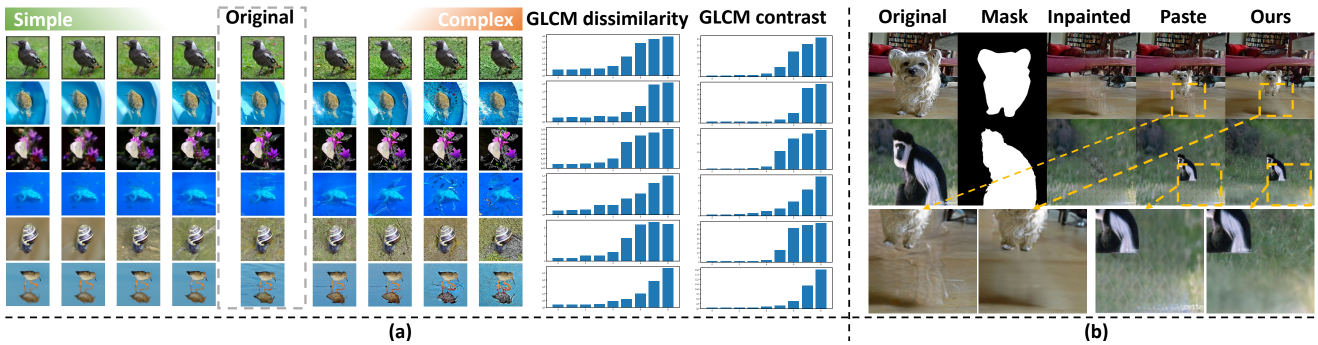

Figure 2. (a) Images generated with the proposed background complexity editing method. (b) Edited images with size changing. The Fréchet inception distance (FID) for pasting is 50.64 while it is 32.59 for ours, indicating the effectiveness of the leveraging of DDPMs.

indicates greater complexity. Thus, we define the objective of complexity as:

$$\mathcal{L}_c = \sum \left| \mathcal{A}(\mathcal{F}(\mathbf{x})) \right|, \tag{5}$$

where $\mathcal{F}$ is the Fourier transform [6], $\mathcal{A}$ extracts the amplitude of the input spectrum. $\mathbf{x}$ is the evaluated image. Since minimizing this loss helps us generate an image with desired properties and should be conducted on the $\mathbf{x}_0$, we need a way of estimating a clean image $\mathbf{x}_0$ from each noisy latent representation $\mathbf{x}_t$ during the denoising diffusion process. Recall that the process estimates at each step the noise $\epsilon_\theta(\mathbf{x}_t, t)$ added to $\mathbf{x}_0$ to obtain $\mathbf{x}_t$. Thus, $\hat{\mathbf{x}}_0$ can be estimated via Equation (6) [1]. The whole optimization procedure is shown in Algorithm 1.

$$\hat{\mathbf{x}}_0 = \frac{\mathbf{x}_t}{\sqrt{\bar{\alpha}_t}} - \frac{\sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}. \tag{6}$$

As shown in Fig. 2(a), with the proposed method, when we guide the generation procedure with the proposed objective towards the complex direction, it will return images with visually complex backgrounds. We also provide the GLCM dissimilarity and contrast of each image to make a quantitative analysis of the generated images. A higher dissimilarity/contrast score indicates a more complex image background [17]. It can be observed that the complexity is consistent with that calculated with GLCM, indicating the effectiveness of the proposed method.

Controlling object size, position and direction. In general, the human vision system is robust to position, direction and small size changes. Whether the deep models are also robust to these object attribute changes is still unknown to researchers. Therefore, we conduct the image editing with controls of object sizes, positions and directions to find the answer. For a valid evaluation on different attributes, all other variables should remain unchanged, especially the background. Therefore, we first disentangle the object and background with the in-painting strategy provided by [54]. Specifically, we mask the object area in input image $\mathbf{x}$. Then we conduct in-painting to remove the object and get the pure background image $\mathbf{x}^b$, as shown in Fig. 2(b) column 3. To realize the aforementioned object attribute controlling, we adopt the orthogonal transformation. Denote P as the pixel locations of object in image $\mathbf{x}$ where $P \in \mathbb{R}^{3\times N_o}$. N_o is the number of pixels belong to object and $p_i = [x_i, y_i, 1]^T$ is the position of object's i-th pixel. $h' \in [0, H-h], w' \in [0, W-w]$ where $[x, y, w, h]$ stand for the enclosing rectangle of the object with mask M. Then the newly edited $\mathbf{x}[T_{\text{attribute}} \cdot P] = \mathbf{x}[P]$ and $M[T_{\text{attribute}} \cdot P] = M[P]$, where

$$T_{\text{size}} = \begin{bmatrix} s & 0 & \Delta x \\ 0 & s & \Delta y \\ 0 & 0 & 1 \end{bmatrix}, T_{\text{position}} = \begin{bmatrix} 1 & 0 & w' \\ 0 & 1 & h' \\ 0 & 0 & 1 \end{bmatrix}, T_{\text{direction}} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{7}$$

where s is the resize scale. θ is the rotation angle. $\Delta x = (1-s)\cdot(x+w/2), \Delta y = (1-s)\cdot(y+h/2)$.

With the background image $\mathbf{x}^b$ and edited object $\mathbf{x}^o$, a naive way is to place the object in the original image to the corresponding area of background image $\mathbf{x}^b$ as $M \odot \mathbf{x}^o + (1-M) \odot \mathbf{x}^b$. However, the result generated in this manner may look disharmonic, lacking a delicate adjustment to blending them together. Besides, as shown in Fig. 2(b) column 3, the object-removing operation may leave some artifacts behind, failing to produce a coherent and seamless result. To deal with this problem, we leverage DDPM models to blend them at different noise levels along the diffusion process. Denote the image with desired object attribute as $\mathbf{x}^o$. Starting from the pure background image $\mathbf{x}^b$ at time t_0, at each stage, we perform a guided diffusion step with a latent $\mathbf{x}_t$ to obtain the $\mathbf{x}_{t-1}$ and at the same time, obtain a noised version of object image $\mathbf{x}^o_{t-1}$. Then the two latents are blended with the mask M as $\mathbf{x}_{t-1} = M \odot \mathbf{x}^o_{t-1} + (1-M) \odot \mathbf{x}_{t-1}$. The DDPM denoising procedure may change the background. Thus a proper initial timing is required to maintain a high resemblance to the original background. We set the iteration steps t_0 as 50 and 25 in Algorithm 1 and 2 respectively.

Algorithm 1: Background editing

input : source image $\mathbf{x}$, mask M, diffusion model $(\mu_\theta(\mathbf{x}_t), \Sigma_\theta(\mathbf{x}_t))$, $\bar{\alpha}_t$, λ, iteration steps t_0

output: edited image $\mathbf{x}_0$

$\mathbf{x}_{t_0} \sim \mathcal{N}(\sqrt{\bar{\alpha}_{t_0}}\mathbf{x}, (1-\bar{\alpha}_{t_0})\mathbf{I})$;

for $t \leftarrow t_0$ **to** 1 **do**

  $\hat{\mathbf{x}}_0 \leftarrow \frac{\mathbf{x}_t}{\sqrt{\bar{\alpha}_t}} - \frac{\sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t,t)}{\sqrt{\bar{\alpha}_t}}$;

  $\nabla_{bg} \leftarrow \nabla_{\hat{\mathbf{x}}_0}\mathcal{L}_c(\hat{\mathbf{x}}_0)$;

  $\mathbf{x}^b_{t-1} \sim \mathcal{N}(\mu_\theta(\mathbf{x}_t) + \lambda\Sigma_\theta(\mathbf{x}_t)\nabla_{bg}, \Sigma_\theta(\mathbf{x}_t))$;

  $\mathbf{x}^o \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}, (1-\bar{\alpha}_t)\mathbf{I})$;

  $\mathbf{x}_{t-1} \leftarrow M \odot \mathbf{x}^o + (1-M) \odot \mathbf{x}^b_{t-1}$;

end

Algorithm 2: Object size controlling

input : source image $\mathbf{x}$, mask M, diffusion model $(\mu_\theta(\mathbf{x}_t), \Sigma_\theta(\mathbf{x}_t))$, $\bar{\alpha}_t$, iteration steps t_0, ratio s

output: edited image $\mathbf{x}_0$

$\mathbf{x}^b \leftarrow$ *ObjectRemoving*$(\mathbf{x}, M)$;

$\mathbf{x}, M \leftarrow$ *Rescale* $(\mathbf{x}, M, s)$;

$\mathbf{x}_{t_0} \sim \mathcal{N}(\sqrt{\bar{\alpha}_{t_0}}\mathbf{x}^b, (1-\bar{\alpha}_{t_0})\mathbf{I})$;

for $t \leftarrow t_0$ **to** 1 **do**

  $\mathbf{x}^b_{t-1} \sim \mathcal{N}(\mu_\theta(\mathbf{x}_t), \Sigma_\theta(\mathbf{x}_t))$;

  $\mathbf{x}^o \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}, (1-\bar{\alpha}_t)\mathbf{I})$;

  $\mathbf{x}_{t-1} \leftarrow M \odot \mathbf{x}^o + (1-M) \odot \mathbf{x}^b_{t-1}$;

end

4.2. ImageNet-E dataset

With the tool above, we conduct object attribute editing including background, size, direction and position changes based on the large-scale ImageNet dataset [42] and ImageNet-S [12], which provides the mask annotation. To guarantee the dataset quality, we choose the animal classes from ImageNet classes such as dogs, fishes and birds, since they appear more in nature without messy backgrounds. Classes such as stove and mortarboard are removed. Finally, our dataset consists of 47872 images with 373 classes based on the initial 4352 images, each of which is applied 11 transforms. Detailed information can be found in Appendix A. For background editing, we choose three levels of the complexity, including $\lambda = -20, \lambda = 20$ and $\lambda =$ 20-adv with adversarial guidance (see Sec.B for details) instead of complexity. Larger λ indicates stronger guidance towards high complexity. For the object size, we design four levels of sizes in terms of the object pixel rates (= $\text{sum}(M > 0.5)/\text{sum}(M \geq 0)$): $[\text{Full}, 0.1, 0.08, 0.05]$ where 'Full' indicates making the object as large as possible while maintaining its whole body inside the image. Smaller rates indicate smaller objects. For object position, we find that some objects hold a high object pixel rate in the whole image, resulting in a small $H - h$. Take the first picture in Fig. 2 for example, the dog is big and it will make little visual differences after position changing. Thus, we adopt the data whose pixel rate is 0.05 as the initial images for the position-changing operation.

In contrast to benchmarks like ImageNet-C [22] giving images from different domains so that the model robustness in these situations may be assessed, our effort aims to give an editable image tool that can conduct model debugging with in-distribution (ID) data, in order to identify specific shortcomings of different models and provide some insights for clean accuracy improving. Thus, the data distribution should not differ much from the original ImageNet. We choose the out-of-distribution (OOD) detection methods Energy [34] and GradNorm [27] to evaluate whether our editing tool will move the edited image out of its original distribution. These OOD detection methods aim to distinguish the OOD examples from the ID examples. The results are shown in Fig. 3. x-axis is the ID score in terms of the quantities in Energy and GradNorm and y-axis is the frequency of each ID score. A high ID score indicates the detection method takes the input sample as the ID data. Compared to other datasets, our method barely changes the data distribution under both Energy (the 1st row) and GradNorm (the 2nd row) evaluation methods. Besides, the Fréchet inception distance (FID) [24] for our ImageNet-E is 15.57 under the random background setting, while it is 34.99 for ImageNet-9 (background challenge). These all imply that our editing tool can ensure the proximity to the original ImageNet, thus can give a controlled evaluation on object attribute changes. To find out whether the DDPM will induce some degradation to our evaluation, we have conducted experiment in Tab. 1 with the setting $\lambda = 0$ during background editing. This operation will first add noises to the original and then denoise them. It can be found in "Inver" column that the degradation is negligible compared to degradation induced by attribute changes.

5. Experiments

We conduct evaluation experiments on various architectures including both CNNs (ResNet (RN) [20], DenseNet [26], EfficientNet (EF) [47], ResNest [53], ConvNeXt [36]) and transformer-based models (Vision-Transformer (ViT) [10], Swin-Transformer (Swin) [35]). Other state-of-the-art models that trained with extra data such as CLIP [40], EfficientNet-L2-Noisy-Student [51] are also evaluated in the Appendix. Apart from different sizes of these models, we have also evaluated their adversarially trained versions for comprehensive studies. We report the drop of top-1 accuracy as metric based on the idea that the attribute changes should induce little influence to a robust trained model. More experimental details and results of top-1 accuracy can be found in the Appendix.

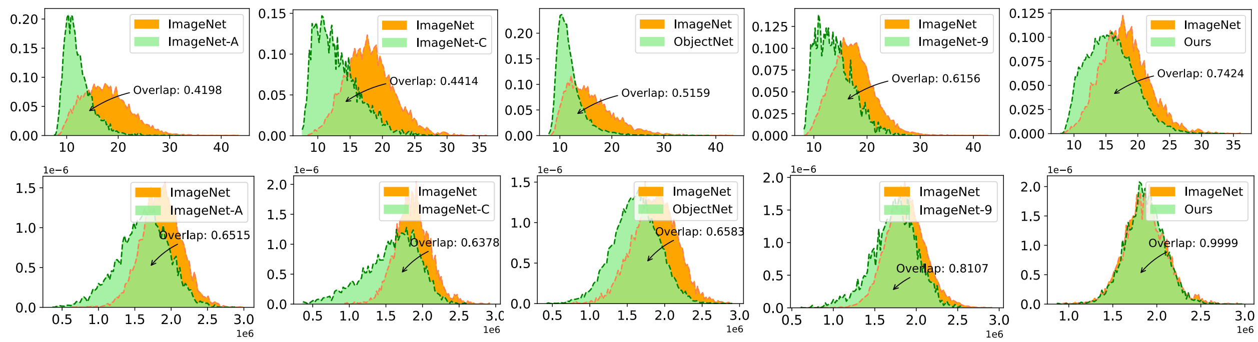

Figure 3. Distributions of ID score of different datasets in terms of the quantities in Energy (the first row) and GradNorm (the second row) for in-distribution (ImageNet) and other datasets. Higher overlap indicates greater proximity to ImageNet.

5.1. Robustness evaluation

Normally trained models. To find out whether the widely used models in computer vision have gained robustness against changes on different object attributes, we conduct extensive experiments on different models. As shown in Tab. 1, when only the background is edited towards high complexity, the average drop in top-1 accuracy is 9.23% ($\lambda = 20$). This indicates that most models are sensitive to object background changes. Other attribute changes such as size and position can also lead to model performance degradation. For example, when changing the object pixel rate to 0.05, as shown in Fig. 1 row 4 in the 'size' column, while we can still recognize the image correctly, the performance drop is 18.34% on average. We also find that the robustness under different object attributes is improved along with improvements in terms of clean accuracy (Original) on different models. Accordingly, a switch from an RN50 (92.69% top-1 accuracy) to a Swin-S (96.21%) leads to the drop in accuracy decrease from 15.72% to 10.20% on average. By this measure, models have become more and more capable of generalizing to different backgrounds, which implies that they indeed learn some robust features. This shows that object attribute robustness can be a good way to measure future progress in representation learning. We also observe that larger networks possess better robustness on the attribute editing. For example, swapping a Swin-S (96.21% top-1 accuracy) with the larger Swin-B (95.96% top-1 accuracy) leads to the decrease of the dropped accuracy from 10.20% to 8.99% when $\lambda = 20$. In a similar fashion, a ConvNeXt-T (9.32% drop) is less robust than the giant ConvNeXt-B (7.26%). Consequently, models with even more depth, width, and feature aggregation may attain further attribute robustness. Previous studies [31] have shown that zero-shot CLIP exhibits better out-of-distribution robustness than the finetuned CLIP, which is opposite to our ImageNet-E as shown in Tab. 1. This may serve as the evidence that our ImageNet-E has a good proximity to ImageNet. We also find that compared with fully-supervised trained model under the same backbone (ViT-B), the CLIP fails to show a better attribute robustness. We think this may be caused by that the CLIP has spared some capacity for OOD robustness.

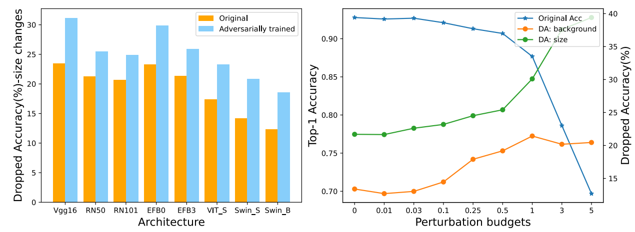

Figure 4. Comparisons between vanilla models and adversarially trained models across different architectures in terms of size changes (left). Evaluation of adversarial models (RN50) trained with different perturbation budgets is provided in the right figure.

Adversarially trained models. Adversarial training [43] is one of the state-of-the-art methods for improving the adversarial robustness of deep models and has been widely studied [2]. To find out whether they can boost the attribute robustness, we conduct extensive experiments in terms of different architectures and perturbation budgets (constraints of l_2 norm bound). As shown in Fig. 4, the adversarially trained ones are not robust against attribute changes including both backgrounds and size-changing situations. The dropped accuracies are much greater compared to normally trained models. As the perturbation budget grows, the situation gets worse. This indicates that adversarial training can do harm to robustness against attributes.

5.2. Robustness enhancements

Based on the above evaluations, we step further to discover ways to enhance the attribute robustness in terms of preprocessing, network design and training strategies. More details including training setting and numerical experimental results can be found in Appendix C.5.

Preprocessing. Given that an object can be inconspicuous due to its small size or subtle position, viewing an object at

Table 1. Evaluations with different state-of-the-art models in terms of Top-1 accuracy and the corresponding drop of accuracy under background changes, size changes, random position (rp) and random direction (rd).

Models	Original	Background changes					Size changes				Position	Direction	Avg.
		Inver	$\lambda=-20$	$\lambda=20$	$\lambda=20$-adv	Random	Full	0.1	0.08	0.05	rp	rd	
RN50	92.69%	1.97%	7.30%	13.35%	29.92%	13.34%	2.71%	7.25%	10.51%	21.26%	26.46%	25.12%	15.72%
DenseNet121	92.10%	1.49%	6.29%	9.00%	29.20%	12.43%	3.50%	7.00%	10.68%	21.55%	26.53%	23.64%	14.98%
EF-B0	92.85%	1.07%	7.10%	10.71%	34.88%	15.64%	3.03%	8.00%	11.57%	23.28%	27.91%	19.11%	16.12%
ResNest50	95.38%	1.44%	6.33%	8.98%	26.62%	11.28%	2.53%	5.27%	8.01%	18.03%	21.37%	17.32%	12.57%
ViT-S	94.14%	**0.82%**	6.42%	8.98%	31.12%	13.06%	**0.80%**	5.37%	8.59%	17.37%	22.86%	17.13%	13.17%
Swin-S	**96.21%**	1.13%	5.18%	7.33%	23.50%	9.31%	1.27%	4.21%	6.29%	14.16%	17.35%	**13.42%**	10.20%
ConvNeXt-T	96.07%	1.43%	**4.69%**	**6.26%**	**19.83%**	**7.93%**	1.75%	**3.28%**	**5.18%**	**12.76%**	**15.71%**	15.78%	**9.32%**
RN101	94.00%	2.11%	7.05%	11.62%	29.47%	13.57%	2.57%	6.81%	10.12%	20.65%	25.85%	24.42%	15.21%
DenseNet169	92.37%	1.12%	5.81%	8.43%	27.51%	11.61%	2.25%	6.90%	10.41%	20.59%	24.93%	20.68%	13.91%
EF-B3	94.97%	1.87%	7.77%	8.40%	29.90%	12.92%	1.36%	6.80%	10.16%	21.36%	24.98%	17.24%	14.09%
ResNest101	95.54%	1.10%	5.58%	6.65%	23.03%	10.40%	1.35%	3.97%	6.53%	15.44%	19.11%	14.31%	10.64%
ViT-B	95.38%	0.83%	5.32%	8.43%	26.60%	10.98%	**0.62%**	4.00%	6.30%	14.51%	18.82%	14.95%	11.05%
Swin-B	95.96%	0.79%	4.46%	6.23%	21.44%	8.25%	0.99%	3.16%	5.04%	12.34%	15.38%	**12.60%**	8.99%
ConvNeXt-B	**96.42%**	**0.69%**	**3.75%**	**4.86%**	**16.49%**	**6.04%**	0.99%	**2.25%**	**3.36%**	**9.47%**	**12.40%**	13.01%	**7.26%**
CLIP-zeroshot	80.01%	4.88%	11.56%	15.28%	36.14%	20.09%	3.33%	12.67%	15.77%	25.31%	28.87%	21.57%	19.06%
CLIP-finetuned	93.68%	2.17%	9.82%	11.83%	38.33%	18.19%	9.06%	9.25%	12.67%	23.32%	28.56%	22.00%	18.30%

several different locations may lead to a more stable prediction. Having this intuition in mind, we perform the classical Ten-Crop strategy to find out if this operation can help to get a robustness boost. The Ten-Crop operation is executed by cropping all four corners and the center of the input image. We average the predictions of these crops together with their horizontal mirrors as the final result. We find this operation can contribute a 0.69% and 1.24% performance boost on top-1 accuracy in both background and size changes scenarios on average respectively.

Network designs. Intuitively, a robust model should tend to focus more on the object of interest instead of the background. Therefore, recent models begin to enhance the model by employing attention modules. Of these, the ResNest [53] can be a representative. The ResNest is a modularized architecture, which applies channel-wise attention on different network branches to leverage their success in capturing cross-feature interactions and learning diverse representations. As it has achieved a great boost in the ImageNet dataset, it also shows superiority on ImageNet-E compared to ResNet. For example, a switch from RN50 decreases the average dropped accuracy from 15.72% to 12.57%. This indicates that the channel-wise attention module can be a good choice to improve the attribute robustness. Another representative model can be the vision transformer, which consists of multiple self-attention modules. To study whether incorporating transformer's self-attention-like architecture into the model design can help attribute robustness generalization, we establish a hybrid architecture by directly feeding the output of res_3 block in RN50 into ViT-S as the input feature like [3]. The dropped accuracy decreases by 1.04% compared to the original RN50, indicating the effectiveness of the self-attention-like architectures.

Training strategy. a) *Robust trained.* There have been plenty of studies focusing on the robust training strategy to improve model robustness. To find out whether these works can boost the robustness on our dataset, we further evaluate these state-of-the-art models including SIN [14], Debiased-CNN [32], Augmix [23], ANT [41], DeepAugment [21] and model trained with lots of standard augmentations (RN50-T) [48]. As shown in Tab. 2, apart from the RN50-T, while the Augmix model shows the best performance against the background change scenario, the Debiased model holds the best in the object size change scenario. What we find unexpectedly is the SIN performance. The SIN method features the novel data augmentation scheme where ImageNet images are stylized with style transfer as the training data to force the model to rely less on textural cues for classification. Though the robustness boost is achieved on ImageNet-C (mCE 69.32%) compared to its vanilla model (mCE 76.7%), it fails to improve the robustness in both object background and size-changing scenarios. The drops of top-1 accuracy for vanilla RN50 and RN50-SIN are 21.26% and 24.23% respectively, when the object size rate is 0.05, though they share similar accuracy on original ImageNet. This indicates that existing benchmarks cannot reflect the real robustness in object attribute changing. Therefore, a dataset like ImageNet-E is necessary for comprehensive evaluations on deep models. b) *Masked image modeling.* Considering that masked image modeling has demonstrated impressive results in self-supervised representation learning by recovering corrupted image patches [4], it may be robust to the attribute changes. Therefore, we choose the Masked AutoEncoder (MAE) [18] as the training strategy since its objective is recovering images with only 25% patches. Specifically, we adopt the MAE training strategy with ViT-B backbone and then finetune it with ImageNet training data. We find that the robustness is improved. For example, the dropped accuracy decreases from 10.62% to 9.05% on average compared to vanilla ViT-B.

Table 2. Evaluations with different robust models in terms of Top-1 accuracy and the corresponding dropped accuracy.

Architectures	Ori	Background changes					Size changes				Position	Direction	Avg.
		Inver	$\lambda=-20$	$\lambda=20$	$\lambda=$ 20-adv	Random	Full	0.1	0.08	0.05	rp	rd	
RN50	92.69%	1.97%	7.30%	13.35%	29.92%	13.34%	2.71%	7.25%	10.51%	21.26%	26.46%	25.12%	15.72%
RN50-Adversarial	81.96%	**0.66%**	**4.75%**	13.62%	37.87%	15.25%	4.87%	9.62%	13.94%	25.51%	32.51%	31.96%	18.99%
RN50-SIN	91.57%	2.23%	7.61%	12.19%	33.16%	13.58%	1.68%	8.30%	12.60%	24.23%	29.16%	27.24%	16.98%
RN50-Debiased	93.34%	1.43%	6.09%	11.45%	27.99%	12.12%	1.98%	5.53%	8.76%	19.27%	24.01%	24.97%	14.22%
RN50-Augmix	93.50%	0.98%	6.26%	8.38%	30.49%	12.94%	1.61%	6.40%	9.97%	21.42%	27.14%	22.42%	14.70%
RN50-ANT	91.87%	1.68%	6.62%	11.94%	35.66%	15.36%	1.57%	7.12%	10.62%	21.49%	26.66%	25.23%	16.23%
RN50-DeepAugment	92.88%	1.50%	6.62%	12.37%	32.40%	13.32%	**1.36%**	7.27%	10.62%	21.28%	26.28%	21.29%	15.28%
RN50-T	**94.55%**	1.05%	5.65%	**7.38%**	**21.89%**	**10.42%**	2.11%	**4.74%**	**7.83%**	**17.46%**	**21.12%**	**19.60%**	**11.82%**

5.3. Failure case analysis

To explore the reason why some robust trained models may fail, we leverage the LayerCAM [29] to generate the heat map for different models including vanilla RN50, RN50+SIN and RN50+Debiased for comprehensive studies. As shown in Fig. 5, the heat map of the Debiased model aligns better with the objects in the image than that of the original model. It is interesting to find that the SIN model sometimes makes wrong predictions even with its attention on the main object. We suspect that the SIN model relies too much on the shape. for example, the ‘sea urchin’ looks like the ‘acron’ with the shadow. However, its texture clearly indicates that it is the ‘sea urchin’. In contrast, the Debiased model which is trained to focus on both the shape and texture can recognize it correctly. More studies can be found in Appendix C.4.

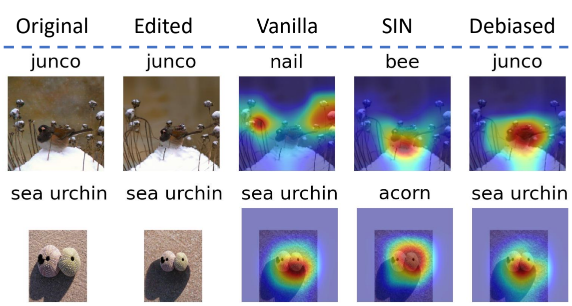

Figure 5. Heat maps for explaining which parts of the image dominate the model decision through LayerCAM [29].

5.4. Model repairing

To validate that the evaluation on ImageNet (IN)-E can help to provide some insights for model’s applicability and enhancement, we conduct a toy example for model repairing. Previous evaluation shows that the ResNet50 is vulnerable to background changes. Based on this observation, we randomly replace the backgrounds of objects with others during training and get a validation accuracy boost from 77.48% to 79.00%. Note that the promotion is not small as only 8781 training images with mask annotations are available in ImageNet. We also step further to find out if the improved model can get a boost the OOD robustness, as shown in the Tab. 3. It can be observed that with the insights provided by the evaluation on ImageNet-E, one can explore the model’s attribute vulnerabilities and manage to repair the model and get a performance boost accordingly.

Table 3. Model repairing results. Top-1 accuracy (%) is reported except for IN-C, which is mCE (mean Corruption Error). Higher top-1 accuracy and lower mCE indicate better performance. IN-E reports the average accuracy on ImageNet-E.

Models	IN	IN-v2	IN-A	IN-C$\downarrow$	IN-R	IN-Sketch	IN-E
RN50	77.5	65.7	6.5	68.6	39.6	27.5	83.7
RN50-repaired	**79.0**	**67.2**	**9.4**	**65.8**	**40.7**	**29.4**	**85.0**

6. Conclusion and Future work

In this paper, we put forward an image editing toolkit that can take control of object attributes smoothly. With this tool, we create a new dataset called ImageNet-E that can serve as a general dataset for benchmarking robustness against different object attributes. Extensive evaluations conducted on different state-of-the-art models show that most models are vulnerable to attribute changes, especially the adversarially trained ones. Meanwhile, other robust trained models can show worse results than vanilla models even when they have achieved a great robustness boost on other robustness benchmarks. We further discover ways for robustness enhancement from both preprocessing, network designing and training strategies.

Limitations and future work. This paper proposes to edit the object attributes in terms of backgrounds, sizes, positions and directions. Therefore, the annotated mask of the interest object is required, resulting in a limitation of our method. Besides, since our editing toolkit is developed based on diffusion models, the generalization ability is determined by DDPMs. For example, we find synthesizing high-quality person images is difficult for DDPMs. Under the consideration of both the annotated mask and data quality, our ImageNet-E is a compact test set. In our future work, we would like to explore how to leverage the edited data to enhance the model’s performance, including both the validation accuracy and robustness.

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 3, 4

[2] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*, 2021. 6

[3] Yutong Bai, Jieru Mei, Alan L Yuille, and Cihang Xie. Are transformers more robust than cnns? *Advances in Neural Information Processing Systems*, 34:26831–26843, 2021. 7, 15

[4] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 7, 15

[5] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019. 2

[6] Salomon Bochner, Komaravolu Chandrasekharan, and K Chandrasekharan. *Fourier transforms*. Number 19. Princeton University Press, 1949. 4

[7] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 3–14, 2017. 2, 3

[8] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021. 15

[9] Peng Cui and Susan Athey. Stable learning establishes some common ground between causal inference and machine learning. *Nature Machine Intelligence*, 4(2):110–115, 2022. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5

[11] Noam Eshed. Novelty detection and analysis in convolutional neural networks. Master's thesis, Cornell University, 2020. 12

[12] Shanghua Gao, Zhong-Yu Li, Ming-Hsuan Yang, Ming-Ming Cheng, Junwei Han, and Philip Torr. Large-scale unsupervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2022. 5, 12

[13] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 3

[14] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 2, 7, 18

[15] Charles J Geyer. Practical markov chain monte carlo. *Statistical science*, pages 473–483, 1992. 3

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1

[17] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973. 3, 4

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 7

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 15

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[21] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 2, 7, 19

[22] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019. 1, 2, 3, 5

[23] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 7, 19

[24] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 3

[26] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5

[27] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 34:677–689, 2021. 5, 16

[28] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37:100270, 2020. 1

[29] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021. 8

[30] Oguzhan Fatih Kar, Teresa Yeo, and Amir Zamir. 3d common corruptions for object recognition. In *ICML 2022 Shift Happens Workshop*. 2

[31] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022. 6, 17

[32] Yingwei Li, Qihang Yu, Mingxing Tan, Jieru Mei, Peng Tang, Wei Shen, Alan Yuille, and Cihang Xie. Shape-texture debiased neural network training. *arXiv preprint arXiv:2010.05981*, 2020. 7, 19

[33] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017. 1

[34] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020. 5, 16

[35] 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 IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 5

[36] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[37] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021. 3

[38] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications: A survey. *Applied Soft Computing*, 93:106384, 2020. 1

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 15

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 5, 17

[41] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In *European Conference on Computer Vision*, pages 53–69. Springer, 2020. 7, 19

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2, 5

[43] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020. 6, 18

[44] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33:11539–11551, 2020. 16

[45] Vikash Sehwag, Caner Hazirbas, Albert Gordo, Firat Ozgenel, and Cristian Canton. Generating high fidelity data from low-density regions using diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11492–11501, 2022. 3

[46] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2, 3

[47] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 5

[48] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021. 7

[49] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022. 17

[50] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *Proceedings of the International Conference on Learning Representations*, 2021. 2, 3

[51] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 5, 17

[52] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple

framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. 15

[53] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2736–2746, 2022. 5, 7

[54] Chuanxia Zheng, Tat-Jen Cham, Jianfei Cai, and Dinh Phung. Bridging global context interactions for high-fidelity image completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11512–11522, June 2022. 4

[55] Yao Zhu, Yuefeng Chen, Xiaodan Li, Kejiang Chen, Yuan He, Xiang Tian, Bolun Zheng, Yaowu Chen, and Qingming Huang. Toward understanding and boosting adversarial transferability from a distribution perspective. *IEEE Transactions on Image Processing*, 31:6487–6501, 2022. 16