IMAGENET-E: BENCHMARKING NEURAL NETWORK ROBUSTNESS VIA ATTRIBUTE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent studies have shown that higher accuracy on ImageNet usually leads to better robustness against different corruptions. In this paper, instead of following the traditional research paradigm that investigates new out-of-distribution corruptions or perturbations deep models may encounter, we aim to conduct model debugging with 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. An imperceptible change in the background can lead to an average of 10.15% drop rate 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 will be publicly available.

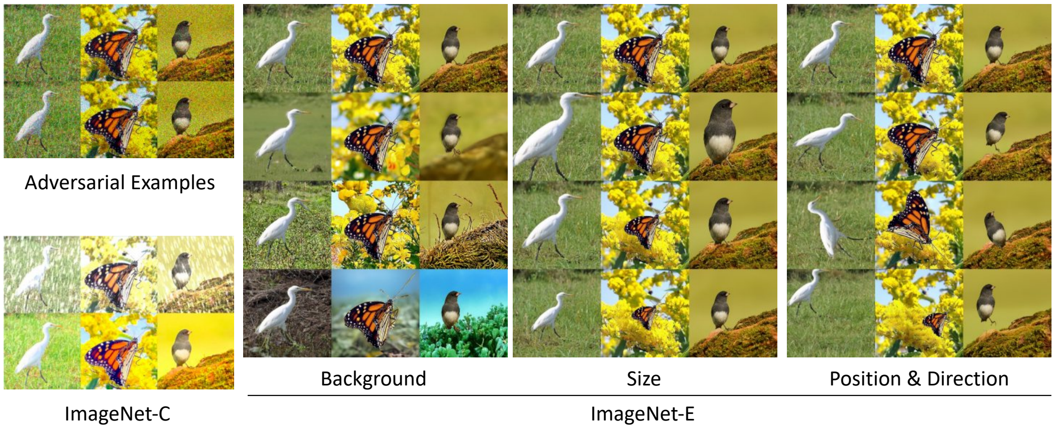


Figure 1: Examples of the proposed ImageNet-E dataset. In contrast to adversarial examples or datasets like ImageNet-C who add perturbation or corruptions to original images, we edit the object attributes with controls of backgrounds, sizes, positions and directions.

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 (Huang et al., 2020), medical science (Litjens et al., 2017), and finance (Ozbayoglu et al., 2020). With the spread of these techniques, the robustness and safety issue begins to be more essential, especially after the finding that deep models can be easily mistaken by negligible noises (Goodfellow et al., 2014). 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 (Carlini & Wagner, 2017; Hendrycks & Dietterich, 2019; Kar et al.) 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 (Hendrycks & Dietterich, 2019). ImageNet-R(enditions) contains various renditions (*e.g.*, paintings, embroidery) of ImageNet object classes (Hendrycks et al., 2021). As both studies have found that higher accuracy on ImageNet usually leads to better robustness against different domains (Hendrycks & Dietterich, 2019; Xiao et al., 2021). We advocate that it is essential to conduct model debugging with the in-distribution data to provide clues for model accuracy improvement, besides exploring a new domain that models may confront. 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 (Cui & Athey, 2022), 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).

Specifically, the ImageNet-E is a compact but challenging test set for object recognition that contains controllable object attributes including backgrounds, sizes, positions and directions, as shown in Figure 1. In contrast to ObjectNet (Barbu et al., 2019) whose images are collected by their workers via posing objects according to specific instructions and differ from the target data 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 (Sohl-Dickstein et al., 2015; Ho et al., 2020). 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 above editing toolkit, we apply it to the large-scale ImageNet dataset (Russakovsky et al., 2015) to construct our ImageNet-E(diting) dataset. It can serve as a general dataset for benchmarking robustness evaluation on different object attributes.

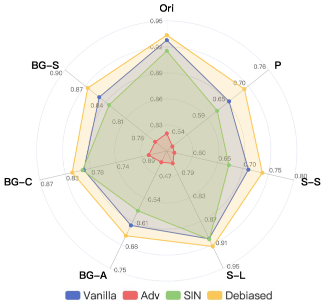


Figure 2: Top-1 accuracies on the original images (Ori) and corresponding edited ones, including simple (BG-S), complex (BG-C) and adversarial (BG-A) backgrounds, different object sizes including small (S-S) and large (S-L), and position (P) editing.

With the generated ImageNet-E, we evaluate the performance of current deep learning models, including both convolutional neural networks (CNNs) and vision transformers. We find that deep models are quite sensitive to object attributes. For example, when editing the background towards high complexity (see Figure 1, the 3rd row in the background part), the drop rate of top-1 accuracy reaches 10.15% 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, as shown in Figure 2. 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 smoothly for manipulated image generation.
- We provide a new dataset called ImageNet-E that can serve as a general dataset 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 with good robustness to other corruptions may show poor performance on our dataset.

2 Preliminaries

We first briefly review the theory of denoising diffusion probabilistic models (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) and analysis how it can be used to generate the desired image.

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 (Geyer, 1992). 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 $\mathbf{x}_t$. 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, Ho et al. (2020) propose to predict the noise $\epsilon_\theta(\mathbf{x}_t, t)$ which was added to $\mathbf{x}_0$ to get $\mathbf{x}_t$ with Equation 2. Then $\mu_\theta(\mathbf{x}_t, t)$ is:

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

Ho et al. (2020) 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 Equation 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 (Avrahami et al., 2022). Besides, there have been plenty of works on utilizing the DDPMs to generate samples with desired properties, such as semantic image translation (Meng et al., 2021), high fidelity data generation from low-density regions (Sehwag et al., 2022), *etc*. In this paper, we also choose DDPMs as our generator.

3 Attribute Editing with Diffusion Models and ImageNet-E

Most previous work on robustness in deep vision models has focused on the important challenges of robustness on adversarial examples (Carlini & Wagner, 2017), common corruptions (Hendrycks & Dietterich, 2019), unknown unknowns (Hendrycks et al., 2018). 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. We develop an object attribute editing tool for generating images while maintaining their semantic meaning. In the following, we describe our editing tool and the generated ImageNet-E dataset in detail.

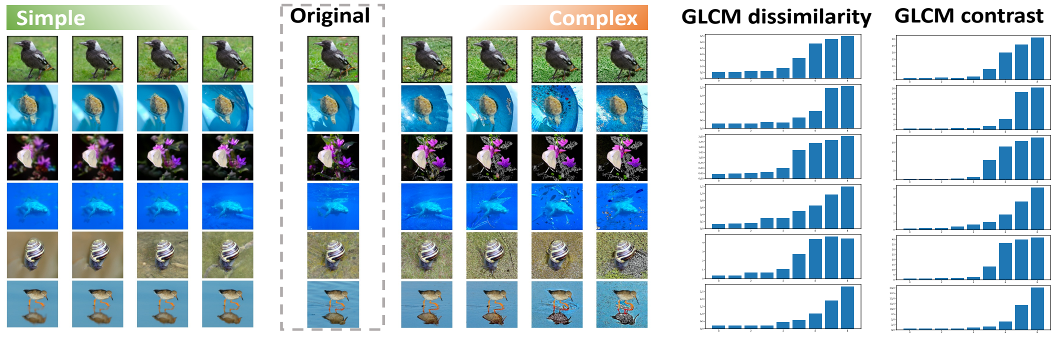


Figure 3: Images generated with the proposed background complexity editing method.

3.1 Object Attribute Editing with Diffusion Models

Background editing. Most existing corruptions conduct manipulations on the whole image, as shown in Figure 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 attack the model. Besides, it is shown that there exists a spurious correlation between labels and image backgrounds (Geirhos et al., 2020). From this point, a background corruption benchmark is needed to evaluate the model's robustness. In this work, 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) (Haralick et al., 1973), 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 indicates greater complexity. Thus, we define the objective of complexity as:

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

where $\mathcal{F}$ is the Fourier transformation (Bochner et al., 1949), $\mathcal{A}$ 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 (Avrahami et al., 2022). The whole optimization procedure is shown in Algorithm 1.

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

Algorithm 1: Background editing

```
input : source image $\mathbf{x}$, input mask $M$, diffusion
        model $(\mu_\theta(\mathbf{x}_t), \Sigma_\theta(\mathbf{x}_t))$,
        hyperparameter $\lambda$, 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 0 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 $\mathbf{x}$, input mask $M$, diffusion
        model $(\mu_\theta(\mathbf{x}_t), \Sigma_\theta(\mathbf{x}_t))$, iteration steps
        $t_0$, target 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 0 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
```

As shown in Figure 3, 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 (Haralick et al., 1973). It can be observed that the complexity is consistent with the complex value 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 Zheng et al. (2022). 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 Figure 4 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

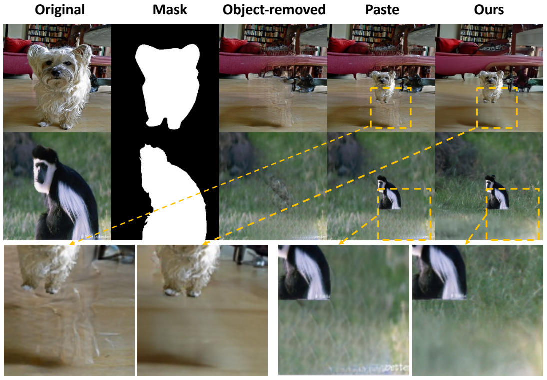


Figure 4: Edited images with size changing. The Fréchet inception distance (FID) for pasting is 50.64 while it is 32.59 for ours, indicating the effectiveness of the leveraging of DDPMs.

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

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 Figure 4 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.

3.2 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 (Russakovsky et al., 2015) and ImageNet-S Gao et al. (2022), 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. Detailed information can be found in Appendix A. For background editing, we choose five levels of the complexity, including $\lambda = 0, \lambda = -20, \lambda = 20, \lambda = 100$ and $\lambda =$ 20-adv with adversarial guidance 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 Figure 4 for example, the dog is big and it will make little visual differences after position changing. Therefore, we adopt the data whose pixel rate is 0.05 as our initial images and run the position-changing operation.

In contrast to benchmarks like ImageNet-C (Hendrycks & Dietterich, 2019) 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 edit the object's attribute in the given image while maintaining it in the original distribution for model debugging, in order to identify specific shortcomings of different models and provide some insights for clean accuracy improving. Thus, we choose the out-of-distribution (OOD) detection method Energy (Liu et al., 2020) and GradNorm (Huang et al., 2021) as the evaluation methods to find out whether our editing tool will move the edited image out of its original distribution. In contrast to FID which indicates the divergence of two datasets, the OOD detection is used to indicate the extent of the deviance of a single input image from the in-distribution dataset. The results are shown in Figure 5. x-axis is the in-distribution (ID) score 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, therefore, the ImageNet data are on the right side. Compared to other datasets, our method barely changes the data distribution under both Energy (the 1st row) and GradNorm (the 2nd row) evaluation methods. This implies that our editing tool can ensure the proximity to the original ImageNet, thus can give a controlled evaluation on object attribute changes. Besides, to find out whether the DDPM will induce some degradation to our evaluation, we have conducted experiment in Table 1 with the setting $\lambda = 0$ during background editing. This operation will first add noises to the original and denoise them. It can be found in "Inver" column that the degradation is negligible compared to degradation induced by attribute changes.

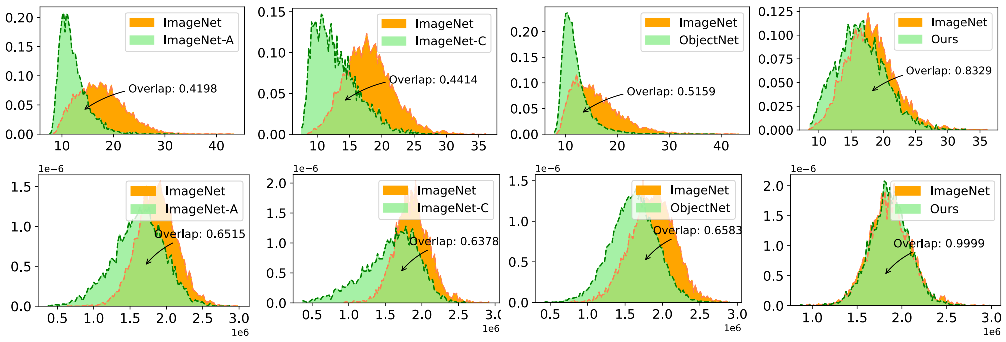


Figure 5: The distribution of the OOD scores for in-distribution (ImageNet) and other datasets. Higher overlap indicates greater proximity to ImageNet.

4 Experiments

We conduct evaluation experiments on various architectures including both CNNs (ResNet (RN) (He et al., 2016), DenseNet (Huang et al., 2017), EfficientNet (EF) (Tan & Le, 2019), ResNest (Zhang et al., 2022), ConvNeXt (Liu et al., 2022)) and transformer-based models (Vision-Transformer (ViT) (Dosovitskiy et al., 2020), Swin-Transformer(Swin) (Liu et al., 2021)). Apart from different sizes of these models, we have also evaluated their adversarially trained versions for comprehensive studies. More details can be found in Appendix D.

4.1 Robustness evaluation of state-of-the-art models

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 Table 1, when only the background is edited towards high complexity, the average drop rate of top-1 accuracy is 10.15% ($\lambda = 20$). This indicates that most models are quite 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

Table 1: Evaluations with different state-of-the-art models in terms of Top-1 accuracy drop rate.

Models	Ori	Inver	Background changes				Size changes				Position	Direction
			$\lambda = -20$	$\lambda = 20$	$\lambda = 100$	$\lambda =$ 20-adv	Full	0.1	0.08	0.05	rp	rd
RN50	0.9278	2.13%	7.88%	14.40%	31.54%	32.28%	3.02%	7.82%	11.34%	22.94%	28.54%	27.10%
DenseNet121	0.9205	1.62%	6.83%	9.77%	20.53%	31.70%	3.80%	7.60%	11.59%	23.40%	28.80%	25.67%
EF-B0	0.9285	1.15%	7.65%	11.54%	27.08%	37.57%	3.54%	8.61%	12.46%	25.08%	30.06%	20.58%
ResNest50	0.9531	1.51%	6.64%	9.41%	18.06%	27.91%	2.55%	5.53%	8.39%	18.90%	22.41%	18.16%
ViT-S	0.9474	1.75%	7.73%	11.23%	19.21%	33.96%	**1.29%**	5.53%	8.39%	18.90%	22.41%	18.16%
Swin-S	**0.9621**	**1.17%**	5.38%	7.62%	13.36%	24.42%	1.34%	4.38%	6.54%	14.72%	18.04%	**13.95%**
ConvNeXt-T	0.9602	1.49%	**4.88%**	**6.52%**	**9.91%**	**20.64%**	1.77%	**3.42%**	**5.39%**	**13.28%**	**16.35%**	16.43%
RN101	0.9400	2.25%	7.50%	12.37%	29.59%	31.35%	2.88%	7.25%	10.76%	21.97%	27.50%	25.98%
DenseNet169	0.9239	1.21%	6.29%	9.13%	21.54%	29.78%	2.44%	7.50%	11.27%	22.29%	26.99%	22.39%
EF-B3	0.9499	1.97%	8.19%	8.84%	18.31%	31.49%	1.58%	7.16%	10.70%	22.49%	26.30%	18.15%
ResNest101	0.9557	1.16%	5.84%	6.96%	11.32%	24.11%	1.55%	4.16%	6.84%	16.16%	20.00%	14.98%
ViT-B	0.9570	0.71%	5.44%	8.37%	12.42%	25.35%	**0.52%**	4.91%	6.87%	16.40%	20.35%	**11.96%**
Swin-B	0.9593	0.82%	4.65%	6.49%	12.26%	22.34%	0.94%	3.29%	5.25%	12.86%	16.03%	13.13%
ConvNeXt-B	**0.9646**	**0.71%**	**3.89%**	**5.04%**	**7.64%**	**17.10%**	1.14%	**2.34%**	**3.48%**	**9.82%**	**12.86%**	13.49%

to 0.05, as shown in Figure 1 row 4 in the 'size' column, while we can still recognize the image correctly, the performance drop rate is 22.27% 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 (92.78% top-1 accuracy) to a Swin-S (96.21%) leads to the drop rate decrease from 14.43% to 7.64% when $\lambda = 20$. 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 RN50 (92.78% top-1 accuracy) with the larger RN101 (94.00% top-1 accuracy) leads to the decrease of the drop rates from 14.43% to 12.68% when $\lambda = 20$. In a similar fashion, a ViT-S (11.91% drop rate) is less robust than the giant ViT-B (8.59% drop rate). Consequently, models with even more depth, width, and feature aggregation may attain further attribute robustness.

Adversarially trained models. Adversarial training (Salman et al., 2020) is one of the state-of-the-art methods for improving the adversarial robustness of deep neural networks and has been widely studied (Bai et al., 2021a). 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 Figure 6, it is surprising to find that the adversarially trained ones are not robust against attribute changes including both backgrounds and size-changing situations. The drop rates are much greater compared to normally trained models. As the perturbation budget grows, the model gets worse on our ImageNet-E. This indicates that adversarial training can do harm to robustness against attributes.

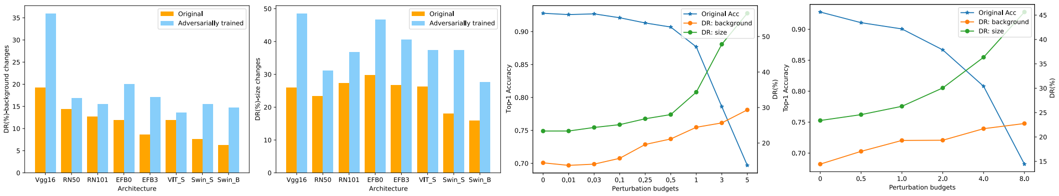


Figure 6: Drop rate comparison of vanilla models and adversarially trained models across different architectures in terms of background changes and size changes (left two). Evaluation of adversarial models trained with different perturbation budgets is also provided in the right figures (l_2, l_∞).

4.2 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 D.5.

Preprocessing. Given that an object can be inconspicuous due to its small size or subtle position, viewing an object at 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, some recent models begin to enhance the model by employing some attention modules. Of these, the state-of-the-art ResNest (Zhang et al., 2022) 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 in our ImageNet-E compared to ResNet. For example, a switch from RN50 decreases the average drop rate from 17.53% to 12.59%. 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 create a hybrid architecture by directly feeding the output of res_3 block in RN50 into ViT-S as the input feature like Bai et al. (2021b). The drop rate decreases by 4.16% 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 (Geirhos et al., 2018), DebiasedCNN (Li et al., 2020), Augmix (Hendrycks et al., 2020), ANT (Rusak et al., 2020), DeepAugment (Hendrycks et al., 2021). As shown in Table 2, 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 performance boost is achieved on ImageNet-C (mCE 69.32%) compared to its vanilla model (mCE 76.7%), they fail to improve the robustness in both object background and size-changing scenarios. The drop rates for vanilla RN50 and RN50-SIN are 23.38% and 26.56% 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 (MIM) has demonstrated impressive results in self-supervised representation learning by recovering corrupted image patches (Bao et al., 2022), it may be robust to the attribute changes. Therefore, we choose the Masked AutoEncoder (MAE) (He et al., 2022a) 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 greatly improved. For example, the drop rate decreases from 8.69% to 6.70% when $\lambda = 20$ compared to vanilla ViT-B. Motivated by the success of MAE, we also test another classical MIM-based method SimMIM (Xie et al., 2022) and can also get a boost. These results validate the effectiveness of MIM training strategy in attribute robustness.

Table 2: Evaluations with different robust models in terms of Top-1 accuracy drop rate.

Models	Ori	Inver	Background changes				Size changes				Position	Direction
			$\lambda = -20$	$\lambda = 20$	$\lambda = 100$	$\lambda =$ 20-adv	Full	0.1	0.08	0.05	rp	rd
RN50	0.9278	2.13%	7.88%	14.40%	31.54%	32.28%	3.02%	7.82%	11.34%	22.94%	28.54%	27.10%
RN50-A	0.8202	0.81%	**5.79%**	16.62%	34.62%	46.21%	6.00%	11.74%	17.00%	31.12%	39.66%	38.99%
RN50-SIN	0.9154	2.43%	8.31%	13.31%	30.04%	36.21%	1.63%	9.06%	13.76%	26.46%	31.85%	29.75%
RN50-Debiased	0.9336	1.53%	6.53%	12.26%	29.58%	**29.98%**	2.22%	**5.92%**	**9.38%**	**20.65%**	**25.73%**	26.75%
RN50-Augmix	**0.9352**	**1.05%**	6.70%	**8.96%**	**13.60%**	32.61%	1.65%	6.84%	10.66%	22.91%	29.03%	23.97%
RN50-ANT	0.9186	1.82%	7.20%	13.00%	22.38%	38.81%	1.75%	7.75%	11.56%	23.40%	29.02%	27.46%
RN50-DeepAugment	0.9290	1.61%	7.13%	13.32%	35.24%	34.89%	**1.61%**	7.82%	11.43%	22.91%	28.29%	**22.93%**

4.3 BAD CASE ANALYSIS

To explore the reason why some robust trained models may fail, we leverage the LayerCAM (Jiang et al., 2021) to generate the heat map for different models including vanilla RN50, RN50+SIN and RN50+Debiased for comprehensive studies. As shown in Figure 7, 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 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 D.4.

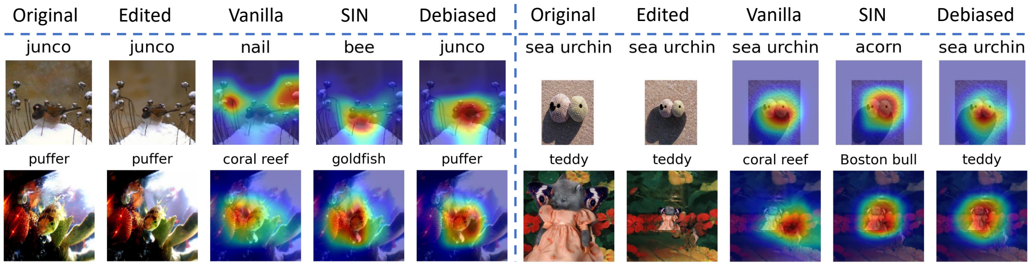


Figure 7: Heat maps for explaining which parts of the image dominate the model decision through LayerCAM (Jiang et al., 2021).

5 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) (Hendrycks & Dietterich, 2019), ImageNet-R(endition) (Hendrycks et al., 2021), ImageNet-Stylized (Geirhos et al., 2018) and ImageNet-3DCC (Kar et al.) mainly focus on the exploring of the corruption 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 (Barbu et al., 2019) and ImageNet-9 (Xiao et al., 2021) can be the representative. 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 edit the object attribute smoothly.

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. In considering 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.

Ethics Statement

In this paper, we provide a novel object attribute editing tool and create a dataset named ImageNet-E. With the proposed dataset ImageNet-E, one can easily conduct model debugging to find out current deep learning's weaknesses against different attributes, thus making deep models more robust. We hope this work can provide some insights to the community and open up a new avenue for research in robust computer vision. We did not use crowdsourcing and did not conduct research with human subjects in our experiments. We cited the creators when using existing assets (*e.g.*, code, data, models).

Reproducibility Statement

Our attribute editing tool is an appealingly simple method, which is developed based on the publicly available codes from DDPMs. We specify the settings of hyper-parameters and how they were chosen in our paper. The source code for our toolkit can be found at https://huggingface.co/spaces/Anonymous-123/ImageNet-Editing. Due to the requirement for GPU, the demo at this site is disabled and can only return the input image. Alternatively, we record a demo video locally, which can be found at https://drive.google.com/file/d/1h5EV3MHPGgkBww9grhlvrl--kSIrD5Lp/view?usp=sharing.

References

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*, pp. 18208–18218, 2022.

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

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, 2021b.

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

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.

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

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*, pp. 3–14, 2017.

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.

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.

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

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*, pp. 1–20, 2022. doi: 10.1109/TPAMI.2022.3218275.

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.

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.

Charles J Geyer. Practical markov chain monte carlo. *Statistical science*, pp. 473–483, 1992.

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

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.

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*, pp. 770–778, 2016.

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*, pp. 16000–16009, 2022a.

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*, pp. 16000–16009, 2022b.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

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.

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*, pp. 8340–8349, 2021.

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

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*, pp. 4700–4708, 2017.

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.

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.

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.

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

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.

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.

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.

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.

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*, pp. 10012–10022, 2021.

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.

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.

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

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.

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*, pp. 8748–8763. PMLR, 2021.

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*, pp. 53–69. Springer, 2020.

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.

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.

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.

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*, pp. 11492–11501, 2022.

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

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

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

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*, pp. 7959–7971, 2022.

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.

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*, pp. 10687–10698, 2020.

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*, pp. 9653–9663, 2022.

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*, pp. 2736–2746, 2022.

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)*, pp. 11512–11522, June 2022.

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. doi: 10.1109/TIP.2022.3211736.

A Details for ImageNet-E

To guarantee the visual quality of the generated examples, we choose the animal classes from ImageNet since they appear more in nature without messy backgrounds. Specifically, images whose coarse labels in [fish, shark, bird, salamander, frog, turtle, lizard, crocodile, dinosaur, snake, trilobite, arachnid, ungulate, monotreme, marsupial, coral, mollusk, crustacean, marine mammals, dog, wild dog, cat, wild cat, bear, mongoose, butterfly, echinoderms, rabbit, rodent, hog, ferret, armadillo,primate] are picked. The corresponding coarse labels of each class we refer to can be found in Eshed (2020)[1]. Finally, our ImageNet-E consists of 373 classes. Since the number of masks provided in ImageNet-S (Gao et al., 2022) in these classes is 4352, thus the number of images in each edited kind is 4352. The ImageNet-E contains 11 kinds of attributes editing, including 5 kinds of background editing and 4 kinds of size editing, as well as one kind of position editing and one kind of direction editing. Finally, our ImageNet-E contains 47872 images. Experiments on more images can be found in section D.3. The comprehensive comparisons with the state-of-the-art robustness benchmarks are shown in Figure 8. In contrast to other benchmarks that investigate new out-of-distribution corruptions or perturbations deep models may encounter, w conduct model debugging with in-distribution data to explore which object attributes a model may be sensitive to. The examples in ImageNet-E are shown in Figure 9. A demo video for our editing toolkit can be found at this url:`https://drive.google.com/file/d/1h5EV3MHPGgkBww9grhlvrl--kSIrD5Lp/view?usp=sharing`.

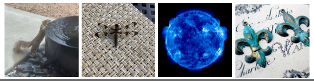
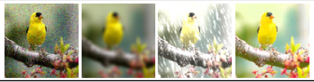
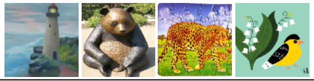
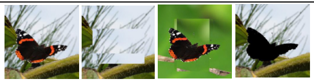
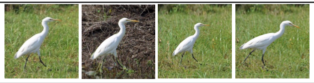
Benchmarks	Description	Classes	Samples
ImageNet-A	Challenging examples collected by-hand	200	
ImageNet-C	Corruptions added on images	1000	
ImageNet-R	Various renditions of ImageNet object classes	200	
ImageNet-3DCC	3D common corruptions	1000	
ImageNet-9	Images whose objects and backgrounds are disentangled with bbox	370	
ImageNet-E	Images with attribute-edited objects	373	

Figure 8: Benchmark comparison.

B Background Editing

Intuitively, an image with complicated background tends to contain more high-frequency components, such as edges. Therefore, a straight-forward way is to define the background complexity as the amplitude of high-frequency components. However, this operation can result in noisy backgrounds, instead of the ones with complicated textures. Therefore, we directly define complexity as the amplitude of all frequency components. The compared results are shown in Figure 10. It can be observed that the amplitude supervision on high-frequency components tends to make the model generate images with more noise. In contrast, amplitude supervision on all frequency components can help to generate images with texture-complex backgrounds. To edit the background adversarially, we set $\mathcal{L}_c = \mathrm{CE}(f(\mathbf{x}), y)$ where 'CE' is the cross entropy loss. f and y are the classifier and label of $\mathbf{x}$ respectively. We adopt the classifier f from guided-diffusion[2].

[1] https://github.com/noameshed/novelty-detection/blob/master/imagenet_categories_synset.csv

[2] https://github.com/openai/guided-diffusion

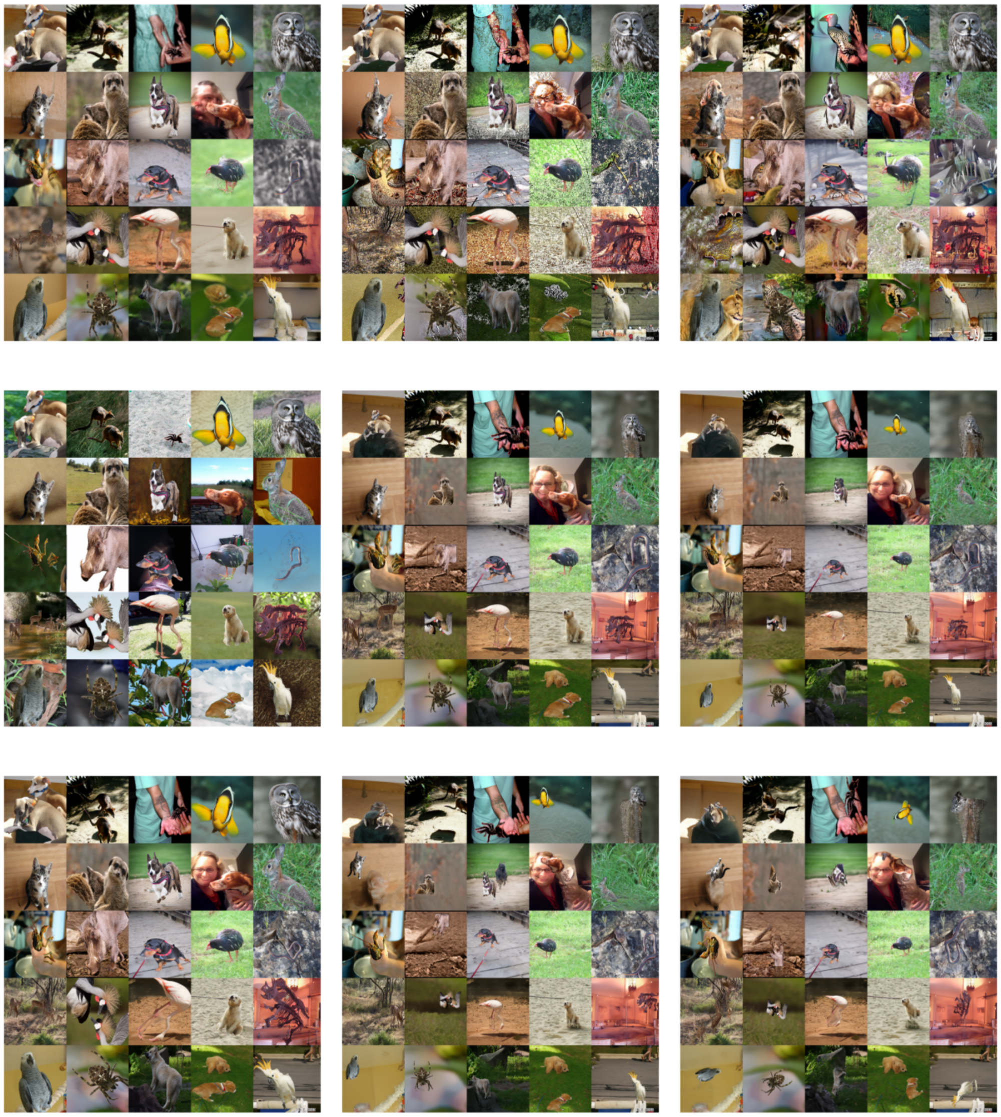


Figure 9: Samples from ImageNet-E. From left to right, top to bottom, the images stand for background editing with $\lambda = -20$, $\lambda = 20$, $\lambda = 20$-adv, randomly shuffled backgrounds, size editing with rate 0.1 and 0.05, randomly rotate, random position, randomly rotate based on images with object pixel rate 0.05 respectively.

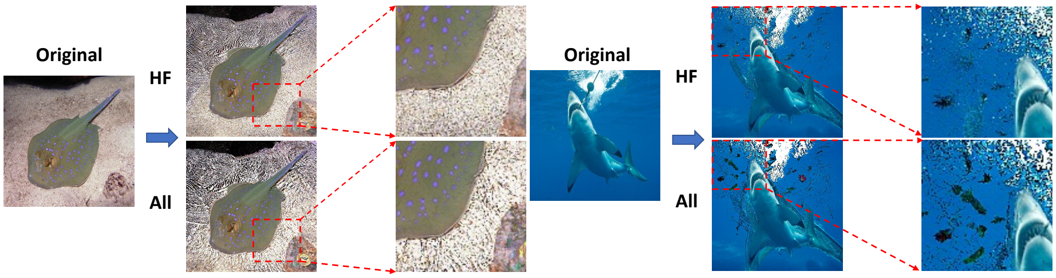


Figure 10: Comparisons between the amplitude supervision on high-frequency components (HF) and amplitude supervision on all frequency components (All).

C Image Editing with Denoising Diffusion Probabilistic Models

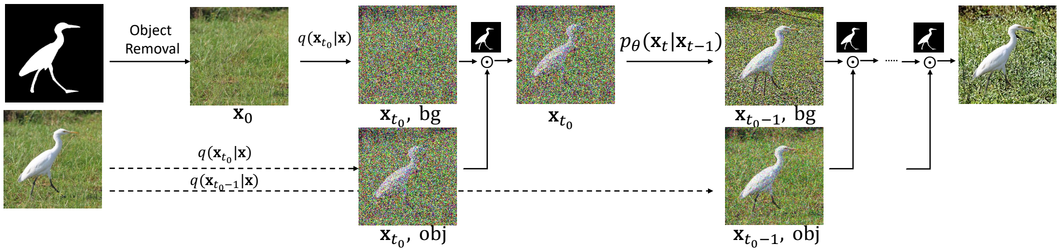

Object Removal
$q(\mathbf{x}_{t_0}|\mathbf{x})$
$\mathbf{x}_0$
$\mathbf{x}_{t_0}$, bg
$\mathbf{x}_{t_0}$
$p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_{t_0-1}$, bg
$q(\mathbf{x}_{t_0}|\mathbf{x})$
$q(\mathbf{x}_{t_0-1}|\mathbf{x})$
$\mathbf{x}_{t_0}$, obj
$\mathbf{x}_{t_0-1}$, obj

Figure 11: Attribute editing with DDPMs.

D Experimental details

D.1 Details for metrics

In this paper, we care more about how different attributes impact different models. Therefore, we choose the top-1 accuracy drop rate as our evaluation metric. A lower drop rate indicates higher robustness against our attribute changes. The drop rate (DR) is defined as:

$$\text{DR} = \frac{\text{acc}_{\text{original}} - \text{acc}}{\text{acc}_{\text{original}}}. \tag{8}$$

The detailed top-1 accuracy (Top-1) and drop rate (DR)on our ImageNet-E are listed in Table 3, Table 4 and Table 5, Table 6. RN50-T is the ResNet50 model from timm library which is trained with lots of training strategies (Wightman et al., 2021).

Table 3: Evaluations under different backgrounds.

Models	Ori	Inver		$\lambda = -20$		$\lambda = 20$		$\lambda = 100$		$\lambda = 20$-Adv	
	Top-1	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR
RN50	0.9278	0.9072	2.13%	0.8539	7.88%	0.7934	14.40%	0.6345	31.54%	0.6277	32.28%
DenseNet121	0.9205	0.9061	1.62%	0.8581	6.83%	0.8310	9.77%	0.7319	20.53%	0.6290	31.70%
EFB0	0.9285	0.9178	1.15%	0.8575	7.65%	0.8214	11.54%	0.6770	27.08%	0.5797	37.57%
ViT-S	0.9474	0.9308	1.75%	0.8742	7.73%	0.8410	11.23%	0.7654	19.21%	0.6257	33.96%
Swin-S	**0.9621**	**0.9508**	**1.17%**	**0.9103**	**5.38%**	**0.8888**	**7.62%**	**0.8336**	**13.36%**	**0.7271**	**24.42%**
RN101	0.9400	0.9189	2.25%	0.8695	7.50%	0.8238	12.37%	0.6619	29.59%	0.6453	31.35%
DenseNet169	0.9239	0.9125	1.21%	0.8656	6.29%	0.8394	9.13%	0.7247	21.54%	0.6486	29.78%
EFB3	0.9499	0.9310	1.97%	0.8720	8.19%	0.8657	8.84%	0.7758	18.31%	0.6507	31.49%
ViT-B	0.9570	0.9498	**0.71%**	0.9045	5.44%	0.8765	8.37%	0.8378	12.42%	0.7141	25.35%
Swin-B	**0.9593**	**0.9517**	0.82%	**0.9150**	**4.65%**	**0.8973**	**6.49%**	**0.8420**	**12.26%**	**0.7452**	**22.34%**

Table 4: Evaluations with different robust models under different backgrounds.

Models	Ori	Inver		$\lambda = -20$		$\lambda = 20$		$\lambda = 100$		$\lambda = 20$-adv	
	Top-1	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR
RN50	0.9278	0.9072	2.13%	0.8539	7.88%	0.7934	14.40%	0.6345	31.54%	0.6277	32.28%
RN50-A	0.8202	0.8130	**0.81%**	0.7721	**5.79%**	0.6834	16.62%	0.5359	34.62%	0.4409	46.21%
RN50-SIN	0.9154	0.8934	2.43%	0.8396	8.31%	0.7938	13.31%	0.6406	30.04%	0.5841	36.21%
RN50-debiased	0.9336	0.9191	1.53%	0.8725	6.53%	0.8189	12.26%	0.6573	29.58%	0.6535	29.98%
RN50-Augmix	0.9352	0.9252	1.05%	0.8724	6.70%	0.8512	8.96%	0.8079	13.60%	0.6301	32.61%
RN50-ANT	0.9186	0.9019	1.82%	0.8525	7.20%	0.7993	13.00%	0.7131	22.38%	0.5621	38.81%
RN50-DeepAugment	0.9290	0.9138	1.61%	0.8626	7.13%	0.8051	13.32%	0.6015	35.24%	0.6048	34.89%
RN50-T	**0.9455**	**0.9350**	1.11%	**0.8890**	5.98%	**0.8717**	**7.81%**	**0.8224**	**13.02%**	**0.7266**	**23.15%**

D.2 Classes whose Top-1 accuracy drops the greatest

To find out which class gets the worst robustness against attribute changes, we plot the dropped accuracy in Figure 12. The evaluated models are vanilla RN50 and its Debiased model. It can be observed that objects that have tentacles with simple backgrounds are more easily to be attacked.

Table 5: Evaluations under different object sizes.

Models	Ori	Full		0.10		0.08		0.05		0.05-rp		rd	
	Top-1	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR
RN50	0.9278	0.8998	3.02%	0.8544	7.82%	0.8218	11.34%	0.7143	22.94%	0.6623	28.54%	0.6757	27.10%
DenseNet121	0.9205	0.8855	3.80%	0.8510	7.60%	0.8142	11.59%	0.7055	23.40%	0.6557	28.80%	0.6846	25.67%
EF-B0	0.9285	0.8956	3.54%	0.8485	8.61%	0.8128	12.46%	0.6957	25.08%	0.6494	30.06%	0.7374	20.58%
ViT-S	0.9474	0.9352	**1.29%**	0.8764	7.50%	0.8410	11.23%	0.7445	21.42%	0.6966	26.48%	0.7752	18.18%
Swin-S	**0.9621**	**0.9492**	1.34%	**0.9200**	**4.38%**	**0.8992**	**6.54%**	**0.8205**	**14.72%**	**0.7886**	**18.04%**	**0.8279**	**13.95%**
RN101	0.9400	0.9129	2.88%	0.8719	7.25%	0.8388	10.76%	0.7335	21.97%	0.6815	27.50%	0.6958	25.98%
DenseNet169	0.9239	0.9014	2.44%	0.8547	7.50%	0.8196	11.27%	0.7178	22.29%	0.6744	26.99%	0.7169	22.39%
EF-B3	0.9499	0.9349	1.58%	0.8817	7.16%	0.84881	10.70%	0.7361	22.49%	0.6999	26.30%	0.7773	18.15%
ViT-B	0.9570	**0.9520**	**0.52%**	0.9097	4.91%	0.8909	6.87%	0.7998	16.40%	0.7619	20.35%	**0.8422**	**11.96%**
Swin-B	**0.9593**	0.9503	0.94%	**0.9280**	**3.29%**	**0.9092**	**5.25%**	**0.8362**	**12.86%**	**0.8058**	**16.03%**	0.8336	13.13%

Table 6: Evaluations with different robust models under different object sizes.

Models	Ori	Full		0.10		0.08		0.05		0.05-rp		rd	
	Top-1	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR
RN50	0.9278	0.8998	3.02%	0.8544	7.82%	0.8218	11.34%	0.7134	22.94%	0.6623	28.54%	0.6757	27.10%
RN50-A	0.8202	0.7710	6.00%	0.7234	11.74%	0.6802	17.00%	0.5645	31.12%	0.4945	39.66%	0.5000	38.99%
RN50-SIN	0.9154	0.9005	1.63%	0.8327	9.06%	0.7897	13.76%	0.6734	26.46%	0.6241	31.85%	0.6433	29.75%
RN50-debiased	0.9336	0.9129	2.22%	0.8781	5.92%	0.8458	9.38%	0.7407	20.65%	0.6933	25.73%	0.6837	26.75%
RN50-Augmix	0.9352	0.9198	1.65%	0.8710	6.84%	0.8353	10.66%	0.7208	22.91%	0.6636	29.03%	0.7108	23.97%
RN50-ANT	0.9186	0.9025	1.75%	0.8475	7.75%	0.8125	11.56%	0.7038	23.40%	0.6521	29.02%	0.6664	27.46%
RN50-DeepAugment	0.9290	0.9140	**1.61%**	0.8561	7.82%	0.8226	11.43%	0.7160	22.91%	0.6660	28.29%	0.7159	22.93%
RN50-T	**0.9455**	**0.9285**	1.80%	**0.8981**	**5.02%**	**0.8672**	**8.28%**	**0.7709**	**18.46%**	**0.7343**	**22.34%**	**0.7495**	**20.73%**

For example, the dropped accuracy of the ‘black widow’ class reaches 47% for both vanilla and Debiased models. In contrast, the impact is smaller for images with complicated backgrounds such as pictures from ‘squirrel monkey’.

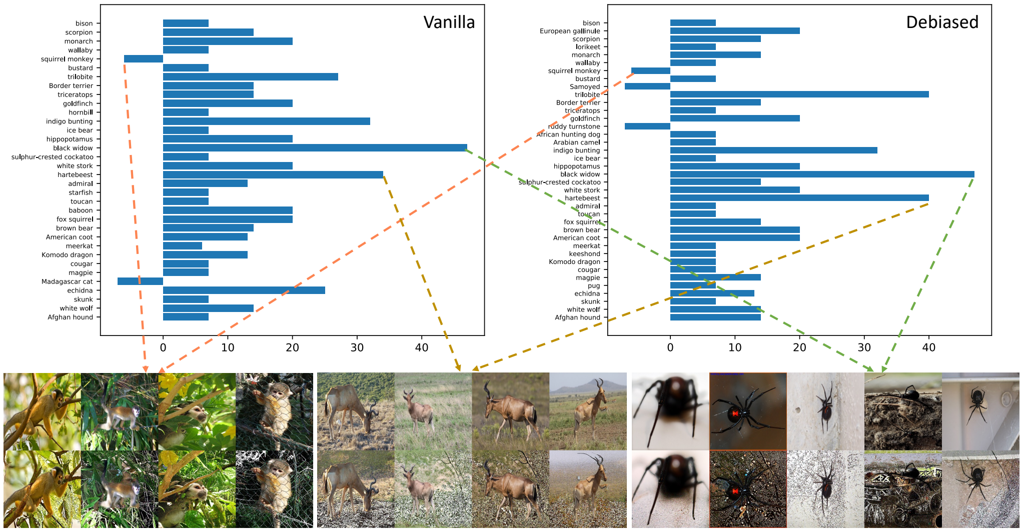


Figure 12: Dropped accuracy (%) in each class. Classes whose number of images is less than 15 or drop rate is zero are removed.

D.3 Experiments on more data

To explore the model robustness against object attributes on large-scale datasets, we step further to conduct the image editing on all the images in the ImageNet-S validation set. Finally, the edited dataset ImageNet-E-L shares the same size as ImageNet-S, which consists of 919 classes and 10919 images. We conduct both background editing and size editing to them. The evaluation results are shown in Table 7. The same conclusion can also be observed. For instance, most models show vulnerability against attribute changing since the average drop rates reach 15.52% and 24.80% in background and size changes respectively. When the model gets larger, the robustness is improved.

The consistency implies that using our ImageNet-E can already reflect the model robustness against object attribute changes.

Table 7: Evaluations with more data.

Models	Original	Background		Size-0.05		Models	Original	Background		Size-0.05	
	Top-1	Top-1	DR	Top-1	DR		Top-1	Top-1	DR	Top-1	DR
DenseNet121	0.8661	0.7473	13.71%	0.6148	29.01%	DenseNet169	0.8766	0.7603	13.27%	0.6331	27.78%
RN50	0.8815	0.7164	18.70%	0.6313	28.36%	RN101	0.8951	0.7533	15.85%	0.6511	27.27%
EF-B0	0.8855	0.7564	14.57%	0.6216	29.79%	EF-B3	0.9212	0.8081	12.28%	0.6618	28.17%
ResNest50	0.9209	0.8061	12.49%	0.7005	23.96%	ResNest101	0.9279	0.8346	10.05%	0.7267	21.67%
ViT-S	0.9214	0.7894	14.34%	0.6930	24.80%	ViT-B	**0.9412**	0.8304	11.77%	0.7565	19.62%
Swin-S	**0.9310**	0.8298	10.88%	0.7536	19.06%	Swin-B	0.9316	0.8411	9.73%	0.7699	17.37%
ConvNeXt-T	0.9272	**0.8400**	**9.43%**	**0.7641**	**17.62%**	ConvNeXt-B	0.9406	**0.8641**	**8.12%**	**0.8034**	**14.58%**

D.4 BAD CASE ANALYSIS

To make a comprehensive study of how the model behaves, we step further to make a comparison of the heat maps of the originals and edited ones. We choose the images that are recognized correctly at first but misclassified after editing. All the attributes editing including background, size, directions are explored. The heat maps are visualized in Figure 13. It can be observed that compared to the SIN and Debiased models, the vanilla RN50 is more likely to lose its focus on the interest area, especially in the size change scenario. For example, in the second row, as it puts his focus on the background, it returns a result with the 'nail' label. The same fashion is also observed in the background change scenario. The predicted label of 'night snake' turns into 'spider web' as the complex background has attracted its attention. In contrast, the SIN and Debiased models have robust attention mechanisms. The quantitative results in Table 4 also validate this. The drop rate of RN50 (14.43%) is higher than SIN (13.04%) and Debiased (12.82%) even though the original accuracy of SIN (0.9154) is lower than vanilla RN50 (0.9278). However, the SIN also has its weakness. We find that though the SIN pays attention to the desired region, it can also make wrong predictions. As shown in the second row of Figure 13, when the object size gets smaller, the shape-based SIN model tends to make wrong predictions, *e.g.*, mistaking the 'sea urchin' as 'acorn' due to the lack of texture analysis. As a result, the drop rate in the size change scenario is 26.56% for SIN, even lower than vanilla RN50, whose drop rate is 23.38%. On the contrary, the Debiased model can recognize it correctly, profiting from its shape and texture-biased module. From the above observation, we can conclude that the texture matters in the small object scenario.

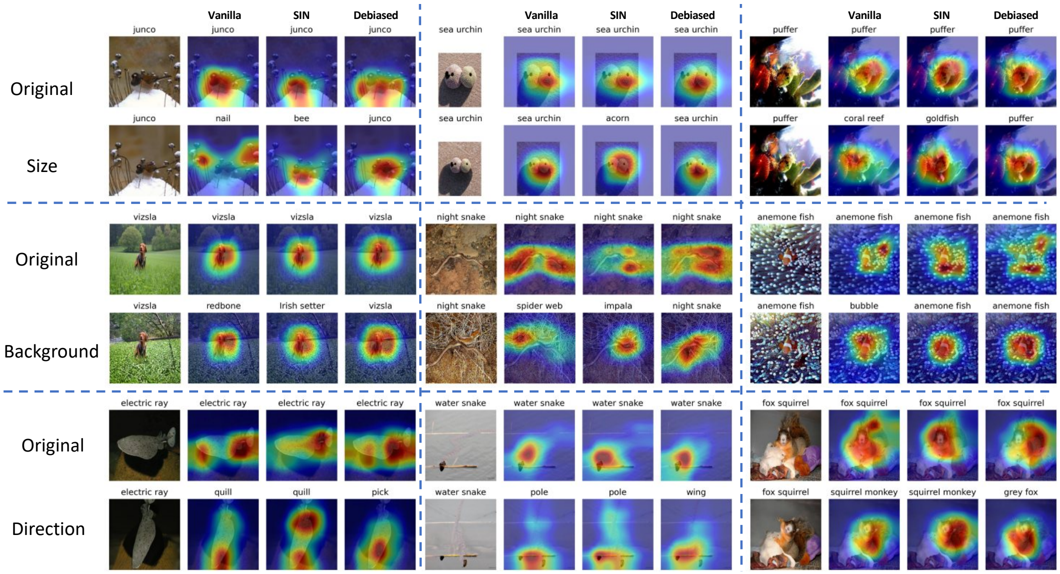


Figure 13: The heat map comparisons between original images and edited ones.

D.5 Details for robustness enhancements

Network design—-self-attention-like architecture. The results in Table 1 show that most vision transformers show better robustness than CNNs in our scenario. Previous study has shown that the self-attention-like architecture may be the key to robustness boost (Bai et al., 2021b). Therefore, to ablate whether incorporating this module can help attribute robustness generalization, we create a hybrid architecture (RN50d-hybrid) by directly feeding the output of res_3 block in RN50d into ViT-S as the input feature. The results are shown in Table 8. As we can find that while the added module maintains the robustness on background changes, it can help to boost the robustness against size changes. Moreover, the RN50-hybrid can also boost the overall performance compared to ViT-S.

Table 8: Ablation study of the self-attention-like architecture.

Models	Ori	Inver	Background changes				Size changes				Position	Direction
			$\lambda = -20$	$\lambda = 20$	$\lambda = 100$	$\lambda =$ 20-adv	Full	0.1	0.08	0.05	rp	rd
RN50d	0.9375	1.31%	**5.12%**	**6.91%**	12.37%	**20.68%**	2.97%	4.65%	7.54%	18.08%	21.85%	20.60%
ViT-S	0.9474	1.75%	7.73%	11.23%	19.21%	33.96%	**1.29%**	7.50%	11.23%	21.42%	26.48%	18.18%
R50d-hybrid	**0.9540**	**1.09%**	5.92%	7.51%	**10.95%**	22.57%	1.42%	**3.70%**	**6.21%**	**14.59%**	**18.06%**	**14.80%**

Training strategy—-Masked image modeling. Considering that masked image modeling has demonstrated impressive results in self-supervised representation learning by recovering corrupted image patches (Bao et al., 2022), it may be robust to the attribute changes. Thus, we test the Masked AutoEncoder (MAE) (He et al., 2022b) and SimMIM (Xie et al., 2022) training strategy based on ViT-B backbone. As shown in Table 9, the drop rates decrease a lot compared to vanilla ViT-B, validating the effectiveness of the masked image modeling strategy.

Table 9: Ablation study of the self-supervised models including MAE and MoCo-V3.

Models	Ori	Inver	Background changes				Size changes				Position	Direction
			$\lambda = -20$	$\lambda = 20$	$\lambda = 100$	$\lambda =$ 20-adv	Full	0.1	0.08	0.05	rp	rd
ViT-B	0.9570	**0.71%**	5.44%	8.37%	12.42%	25.35%	**0.52%**	4.91%	6.87%	16.40%	20.35%	**11.96%**
MAE-ViT-B	0.9612	0.81%	4.96%	**6.46%**	**10.17%**	**21.94%**	0.82%	**3.13%**	**5.06%**	**12.59%**	**16.10%**	14.56%
SimMIM-ViT-B	**0.9614**	0.78%	**3.30%**	7.03%	16.03%	24.52%	0.96%	3.35%	5.55%	13.70%	17.81%	14.17%

D.6 Hardware

Our experiments are implemented by PyTorch (Paszke et al., 2019) and runs on RTX-3090TI.

E Further exploration on backgrounds changing

Motivated by the models' vulnerability against background changes, especially for those complicated backgrounds. Apart from randomly picking the backgrounds from the ImageNet dataset as final backgrounds (random_bg), we also collect background templates with abundant textures, including leopard, eight diagrams, checker and stripe to explore the performance on out-of-distribution backgrounds. The evaluation results are shown in Table 10. It can be observed that the background changes can lead to a 14.70% drop rate. When the background is set to be a leopard or other images, the drop rates can even reach 39.60%. Sometimes the robust models even show worse robustness. For example, when the background is eight diagrams, all the robust models show worse results than the vanilla RN50, which is quite unexpected. To comprehend the behaviour behind it, we visualize the heat maps of the different models in Figure 8. An interesting finding is that deep models tend to make decisions with dependency on the backgrounds, especially when the background is complicated and can attract some attention. For example, when the background is the eight diagrams, the SIN takes the goldfish as a dishwasher. We suspect it has mistaken the background as dishes. In the same fashion, the Debiased model and ANT take the 'sea slug' with eight diagrams as a 'shopping basket', which seems to make sense since the 'sea slug' looks like a vegetable.

Table 10: Evaluation of images generated with different backgrounds.

Models	Ori	Random_bg		Leopard		Eight diagrams		Checker		Stripe	
RN50	0.9278	0.7935	14.39%	0.5717	38.32%	**0.6432**	**30.61%**	0.6513	29.73%	0.6290	32.14%
RN50-A	0.8202	0.6671	18.60%	0.2505	69.43%	0.3725	54.59%	0.3247	60.39%	0.4696	42.72%
RN50-SIN	0.9154	0.7799	14.83%	0.6274	31.48%	0.4874	46.78%	0.5115	44.14%	0.5265	42.50%
RN50-debiasd	0.9336	**0.8122**	**12.98%**	**0.6858**	**26.52%**	0.6268	32.85%	0.6710	28.11%	0.6316	32.32%
RN50-Augmix	**0.9352**	0.8056	13.84%	0.5735	38.67%	0.5620	39.89%	**0.6878**	**26.44%**	**0.6568**	**29.77%**
RN50-ANT	0.9186	0.7651	16.72%	0.5811	36.75%	0.5904	35.74%	0.5191	43.49%	0.5469	40.46%
RN50-DeepAugment	0.9290	0.7956	14.34%	0.6283	32.35%	0.5771	37.86%	0.5946	35.99%	0.6180	33.46%

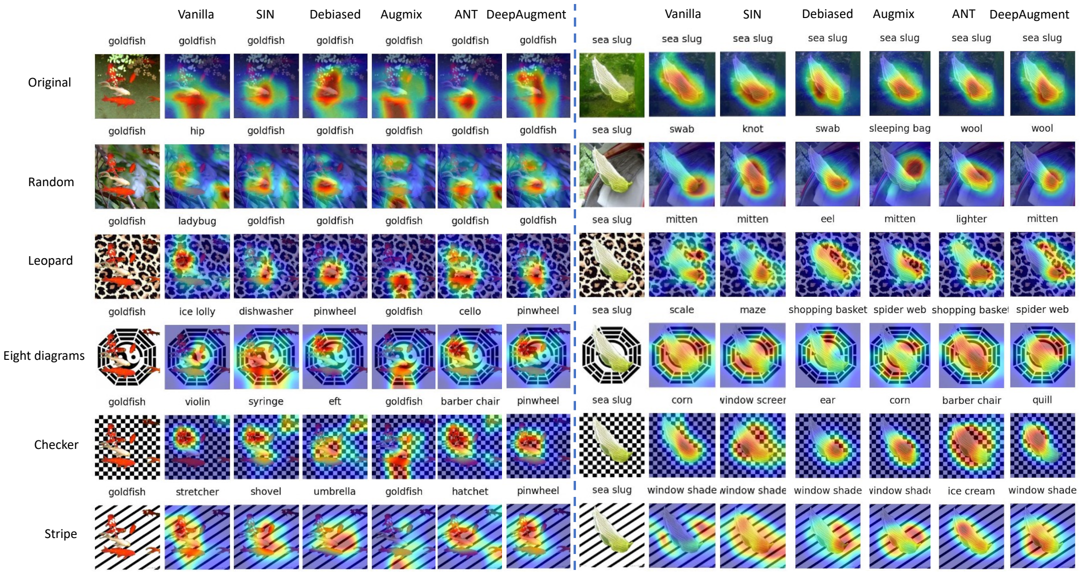


Figure 14: Heat maps under different backgrounds.

F Further discussion on the distribution

In this paper, our effort aims to give an editable image tool that can conduct model debugging with in-distribution data. One way for evaluating the similarity of two distributions can be KL divergence. However, the data distribution $p(x)$ is hard to be measured. As an alternative, we adopt some existing out-of-distribution (OOD) detection methods including Energy (Liu et al., 2020) and GradNorm (Huang et al., 2021) following DRA (Zhu et al., 2022). These OOD detection methods aim to distinguish the OOD examples from the in-distribution examples. Here We provide further comparisons with other datasets including ImageNet-V2, inpainted ImageNet-S, adversarial examples and ImageNet-9. With the results in Figure 15, we can find that the ImageNet-E holds the best proximity to the ImageNet. This implies that our editing tool can give a controlled evaluation on object attribute changes.

Covariate shift adaptation(*a.k.a* batch-norm adaptation, BNA) is a way for improving robustness against common corruptions (Schneider et al., 2020). Thus, it can help to get a top-1 accuracy performance boost in OOD data. Thus, one can easily find out if the provided dataset is OOD by checking whether the BNA can get a performance boost on its data. We have tested the full adaptation results using BNA on ResNet50. In contrast to the promotion on other out-of-distribution dataset, we find that this operation induces little changes to top-1 accuracy on both ImageNet validation set (0.7615 → 0.7613) and our ImageNet-E (0.7934 → 0.7933 under $\lambda = 20$, 0.6521 → 0.6514 under random position scenario, mean accuracy of 5 runs). This similar tendency implies that our ImageNet-E shares a similar property with ImageNet.

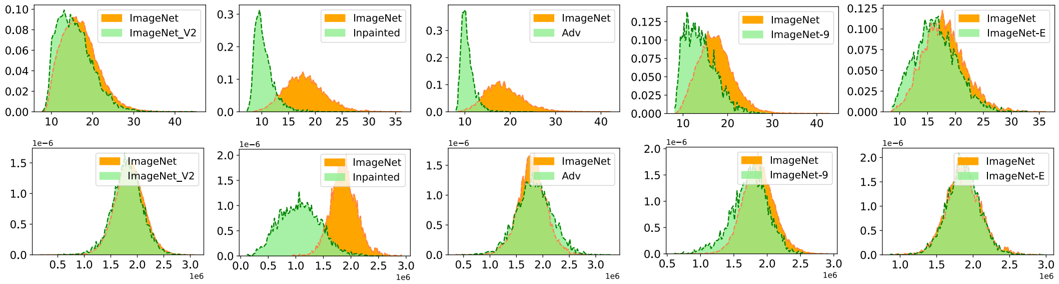


Figure 15: The distribution of the OOD scores for in-distribution (ImageNet) and other datasets. Higher overlap indicates greater proximity to ImageNet.

Table 11: Evaluations on different robustness benchmarks. All results are top-1 accuracies(%) on corresponding datasets except for ImageNet-C, which is mCE (mean Corruption Error). Higher top-1 accuracy and lower mCE indicate better performance.

Models	IN	IN-V2	IN-A	IN-C	IN-R	IN-Sketch	IN-E-bg-random	IN-E-size-005
CLIP-zero-shot	68.3	61.9	**50.1**	**43.1**	**77.6**	**48.3**	60.0	54.7
CLIP-FT	**81.2**	**70.7**	35.3	47..9	65.0	44.9	**75.5**	**70.1**

G Further evaluation on more state-of-the-art models

To provide evaluations on more state-of-the-art models, we step further to evaluate the CLIP (Radford et al., 2021) and EfficientNet-L2-Noisy-Student (Xie et al., 2020). CLIP shows a good robustness to out-of-distribution data (Kumar et al., 2022). Therefore, to find out whether the CLIP can also show a good robustness against attribute editing, we evaluate the all the CLIP models with both the zero-shot and end-to-end finetuned version. To achieve this, we finetune the pretrained CLIP on the ImageNet training dataset based on prompt-initialized model following Wortsman et al. (2022). It acquires 81.2 (ViT-B/16) and 87.7 (ViT-L/14@336px) top-1 accuracies on ImageNet validation set while it is 68.3 (ViT-B/16) and 76.6 (ViT-L/14@336px) for zero-shot version. The evaluation on ImageNet-E is shown in Table 11 and Table 12. Though previous studies have shown that the zero-shot CLIP model exhibits better out-of-distribution robustness than the finetuned ones, the finetuned CLIP shows better attribute robustness on ImageNet-E, as shown in Table 11 and Table 12. The tendency on ImageNet-E is the same with ImageNet (IN) validation set and ImageNet-V2 (IN-V2). This implies that the ImageNet-E shows a better proximity to ImageNet than other out-of-distribution benchmarks such as ImageNet-C (IN-C), ImageNet-A (IN-A). Another finding is that the CLIP model fails to show better robustness than ViT-B/16 while they share the same architectures. We suspect that this is caused by that CLIP may have spared some capacity for out-of-distribution robustness.

While EfficientNet-L2-Noisy-Student is one of the top models on ImageNet-A benchmark (Xie et al., 2020), it also shows superiority on ImageNet-E. To delve into the reason behind this, we test EfficientNet-L2-Noisy-Student-475 (EF-L2-NT-475) and EfficientNet-B0-Noisy-Student (EF-B0-NT). The EF-L2-NT-475 differs from EF-L2-NT in terms of input size, which former is 475 while it is 800 for the latter. It can be found that the input size can induce little improvement to the attribute robustness. In contrast, larger networks can benefit a lot to attribute robustness, which is consistent with the finding in Section 4.1.

H Model repair with ImageNet-E

To validate that the evaluation on ImageNet-E can help to provide some insights for model's applicability and enhancement, we conduct a toy example for model repairing. Our evaluation shows that the ResNet50 is vulnerable to background changes. Based on this observation, we randomly replace the backgrounds of objects during training and get a validation accuracy boost from 77.48% to 79.00%. Note that the promotion is not small as we only conduct this operation on 8781 training images since the ImageNet-S only provides 8781 annotated images with object masks in the training

Table 12: More evaluations on state-of-the-art models including CLIP and EfficientNet-L2-Noisy-Student.

Models	Ori	Inver	Background changes				Size changes				Position	Direction
			$\lambda = -20$	$\lambda = 20$	$\lambda = 100$	$\lambda = 20$-adv	Full	0.1	0.08	0.05	rp	rd
ViT-B/16	**0.9566**	**0.71%**	**5.44%**	**8.37%**	**12.42%**	**25.35%**	0.52%	**4.91%**	**6.87%**	**16.4%**	**20.35%**	**11.96%**
						Zero-shot						
CLIP_RN50	0.7238	8.32%	16.08%	23.10%	35.84%	48.45%	12.13%	19.88%	24.44%	36.59%	41.16%	34.97%
CLIP_RN101	0.7335	6.15%	14.68%	19.66%	31.21%	45.56%	8.71%	19.81%	24.79%	36.23%	41.00%	33.41%
CLIP_RN50x4	0.7718	6.00%	13.53%	17.19%	26.35%	40.67%	9.67%	16.03%	20.29%	31.39%	35.23%	31.41%
CLIP_RN50x16	0.8210	5.37%	12.30%	15.12%	19.89%	33.05%	8.06%	13.51%	16.48%	26.90%	30.78%	28.17%
CLIP_RN50x64	0.8566	5.57%	10.38%	**12.60%**	14.93%	**27.72%**	7.46%	10.74%	13.92%	**22.37%**	**25.23%**	24.01%
CLIP_ViT-B/32	0.7408	7.50%	17.88%	25.17%	37.56%	58.40%	4.04%	21.05%	26.64%	39.21%	45.04%	33.60%
CLIP_ViT-B/16	0.8001	6.10%	14.44%	19.10%	24.70%	45.17%	6.10%	15.85%	19.72%	31.63%	36.08%	26.96%
CLIP_ViT-L/14	**0.8761**	**4.97%**	12.60%	16.51%	20.04%	38.45%	**2.07%**	13.32%	17.22%	27.00%	31.03%	20.60%
CLIP_ViT-L/14@336px	**0.8801**	**3.59%**	**10.31%**	13.91%	**14.42%**	33.73%	3.59%	**10.46%**	**13.38%**	22.66%	26.00%	**18.35%**
						Finetune						
CLIP_ViT-B/16	0.9368	2.32%	10.49%	12.63%	18.31%	40.92%	4.97%	9.87%	13.52%	24.89%	30.48%	23.49%
CLIP_ViT-L/14@336px	**0.9697**	**1.33%**	**5.32%**	**6.37%**	**7.20%**	**20.55%**	**1.33%**	**3.58%**	**5.06%**	**11.32%**	**14.17%**	**11.30%**
EF-B0	0.9285	1.15%	7.65%	11.54%	27.08%	37.57%	3.54%	8.61%	12.46%	25.08%	30.06%	20.58%
EF-B0-NT	0.9430	2.08%	8.94%	11.15%	21.03%	37.05%	1.22%	8.39%	12.20%	24.35%	29.29%	20.22%
EF-L2-NT-475	**0.9784**	**1.10%**	3.68%	4.61%	5.99%	15.21%	**0.52%**	**2.26%**	**2.77%**	5.62%	7.52%	4.68%
EF-L2-NT	0.9763	1.29%	**3.58%**	**4.15%**	**5.32%**	**13.03%**	0.73%	2.32%	2.85%	**5.13%**	**6.17%**	**4.65%**

data of ImageNet. We also step further to find out if the improved model can get a boost on other robustness benchmarks, as shown in the Table 13. It can be observed that with the insights provided by the evaluation on ImageNet-E, one can explore the model's attribute vulnerabilities and struggle to repair the model and get a performance boost accordingly.

Table 13: Model repair results. All results are top-1 accuracies (%) on corresponding datasets except for ImageNet-C, which is mCE (mean Corruption Error). Higher top-1 accuracy and lower mCE indicate better performance.

Models	ImageNet-val	ImageNet-V2	ImageNet-A	ImageNet-C↓	ImageNet-R	ImageNet-Sketch
RN50	77.5	65.7	6.5	68.6	39.6	27.5
RN50-repaired	**79.0**	**67.2**	**9.4**	**65.8**	**40.7**	**29.4**

I Failure cases of generated images

The failure cases of generated images are shown in Figure 16. The diffusion model fails to generate high-quality person images. Though the object is reserved, the whole image looks quite wired. Therefore, we only keep the animal classes, resulting a compact set of ImageNet-E. However, extensive evaluations to 919 in Section D.3 have witnessed a same conclusion with evaluations on 373 classes. This implies that using our ImageNet-E can already reflect the model robustness against object attribute changes.

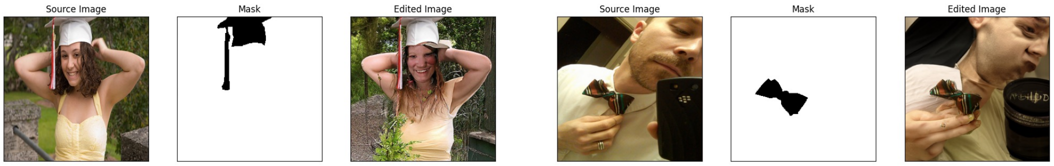


Figure 16: The failure cases of attribute editing

J Related literature to robustness enhancements

Adversarial training. Salman et al. (2020) focus on adversarially robust ImageNet classifiers and show that they yield improved accuracy on a standard suite of downstream classification tasks. It provides a strong baseline for adversarial training. Therefore, we choose their officially released ad-

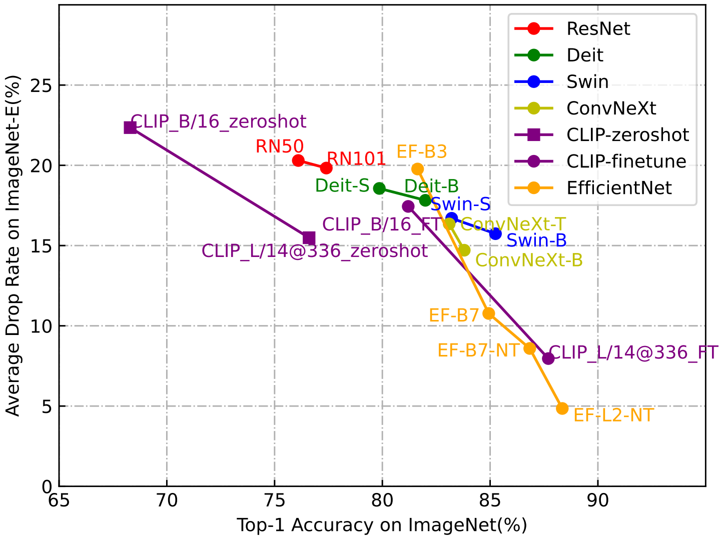


Figure 17: The overall performance (average of drop rate) of several state-of-the-art models.

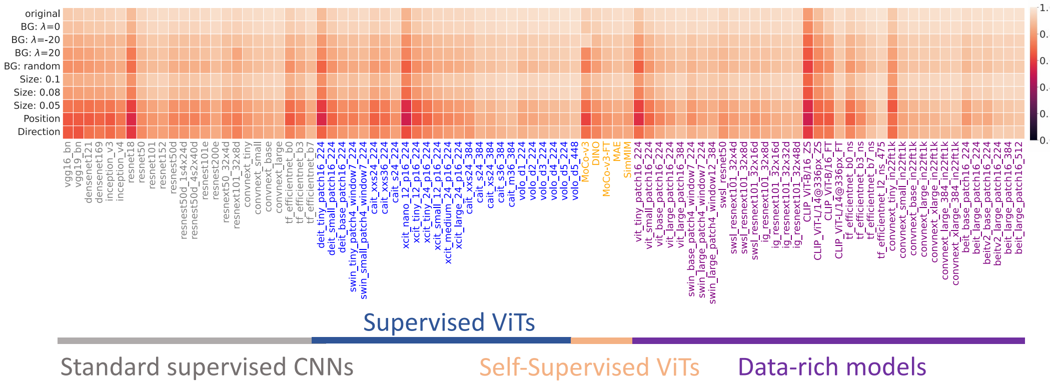


Figure 18: The top-1 accuracy performance under different editing scenarios of 91 state-of-the-art models.

versarially trained models[3] as the evaluation model. Models with different architectures are adopted here[4].

SIN (Geirhos et al., 2018) provides evidence that machine recognition today overly relies on object textures rather than global object shapes, as commonly assumed. It demonstrates the advantages of a shape-based representation for robust inference (using their Stylized-ImageNet dataset to induce such a representation in neural networks)

Debiased (Li et al., 2020) shows that convolutional neural networks are often biased towards either texture or shape, depending on the training dataset, and such bias degenerates model performance. Motivated by this observation, it develops a simple algorithm for shape-texture Debiased learning. To prevent models from exclusively attending to a single cue in representation learning, it augments training data with images with conflicting shape and texture information (*e.g.*, an image of chimpanzee shape but with lemon texture) and provides the corresponding supervision from shape and texture simultaneously. It empirically demonstrates the advantages of the shape-texture Debiased neural network training on boosting both accuracy and robustness.

[3] https://github.com/microsoft/robust-models-transfer

[4] https://github.com/alibaba/easyrobust

Augmix (Hendrycks et al., 2020) focuses on the robustness improvement to unforeseen data shifts encountered during deployment. It proposes a data processing technique named Augmix that helps to improve robustness and uncertainty measures on challenging image classification benchmarks.

ANT (Rusak et al., 2020) demonstrates that a simple but properly tuned training with additive Gaussian and Speckle noise generalizes surprisingly well to unseen corruptions, easily reaching the previous state of the art on the corruption benchmark ImageNet-C and on MNIST-C.

DeepAugment (Hendrycks et al., 2021). Motivated by the observation that using larger models and artificial data augmentations can improve robustness on real-world distribution shifts, contrary to claims in prior work. It introduces a new data augmentation method named DeepAugment, which uses image-to-image neural networks for data augmentation. It improves robustness on their newly introduced ImageNet-R benchmark and can also be combined with other augmentation methods to outperform a model pretrained on 1000× more labeled data.