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 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`.

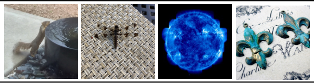
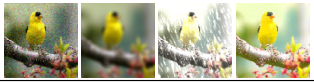
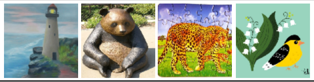
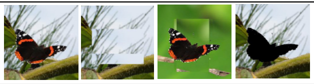
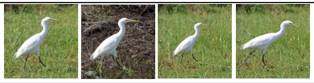
Benchmarks	Description	Classes	Samples	In-distribution
ImageNet-A	Challenging examples collected by-hand	200		X
ImageNet-C	Corruptions added on images	1000		X
ImageNet-R	Various renditions of ImageNet object classes	200		X
ImageNet-3DCC	3D common corruptions	1000		X
ImageNet-9	Images whose objects and backgrounds are disentangled with bbox	370		X
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 $\mathcal{L}_c = \text{CE}(f(\mathbf{x}), y)$ 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

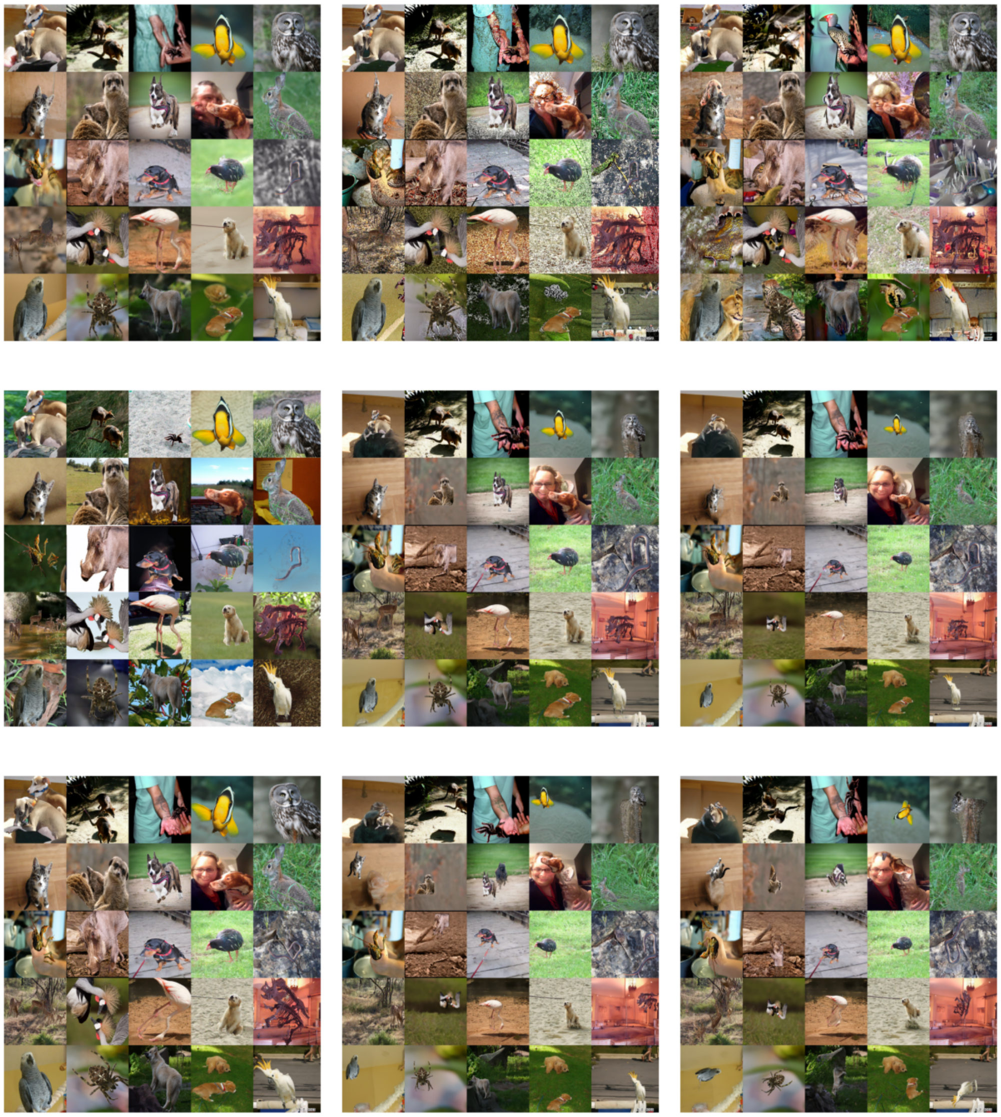


Figure 9: Samples from ImageNet-E. From left to right, top to bottom, the images stand for background editing with $\lambda = -20$, $\lambda = 20$, $\lambda = 20$-adv, randomly shuffled backgrounds, size editing with rate 0.1 and 0.05, randomly rotate, random position, randomly rotate based on images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

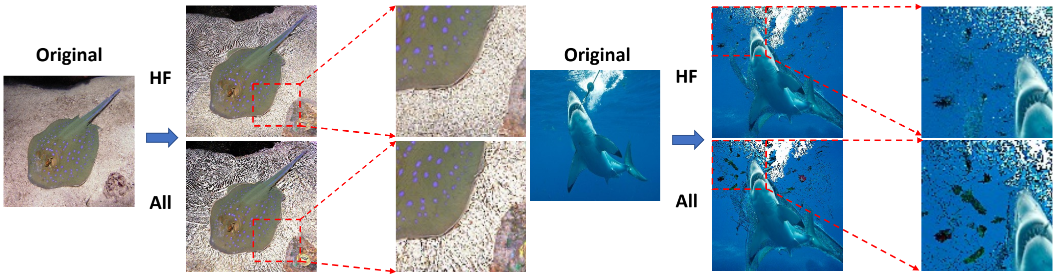


Figure 10: Comparisons between the amplitude supervision on high-frequency components (HF) and amplitude supervision on all frequency components (All).

C Image Editing with Denoising Diffusion Probabilistic Models

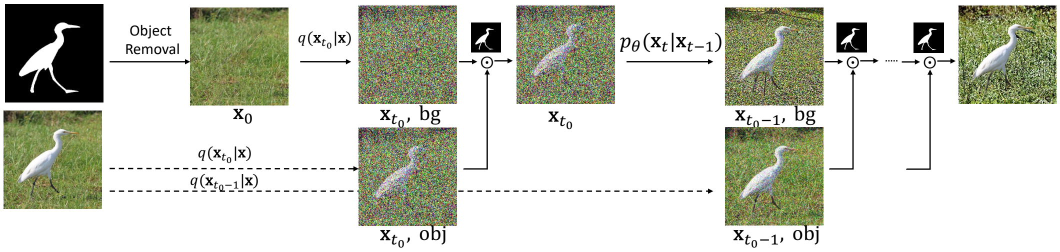

Object
Removal
$q(\mathbf{x}_{t_0}|\mathbf{x})$
$\mathbf{x}_0$
$\mathbf{x}_{t_0}$, bg
$\mathbf{x}_{t_0}$
$p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_{t_0-1}$, bg
$q(\mathbf{x}_{t_0}|\mathbf{x})$
$q(\mathbf{x}_{t_0-1}|\mathbf{x})$
$\mathbf{x}_{t_0}$, obj
$\mathbf{x}_{t_0-1}$, obj

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:

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

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.

Table 3: Evaluations under different backgrounds.

Background	Ori	$\lambda = 0$		$\lambda = -20$		$\lambda = 20$		$\lambda = 100$		$\lambda = 20$-Adv	
	Top-1	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR
RN50	0.9278	0.9069	2.25%	0.8495	8.44%	0.7939	14.43%	0.6278	32.33%	0.6222	32.94%
DenseNet121	0.9205	0.9049	1.69%	0.8552	7.09%	0.8311	9.71%	0.7301	20.68%	0.6268	31.91%
EFB0	0.9285	0.9148	1.48%	0.8490	8.56%	0.8180	11.90%	0.6722	27.60%	0.5793	37.61%
ViT-S	0.9474	0.9306	1.77%	0.8676	8.42%	0.8346	11.91%	0.7628	19.48%	0.6319	33.30%
Swin-S	**0.9621**	**0.9522**	**1.03%**	**0.9067**	**5.76%**	**0.8886**	**7.64%**	**0.8361**	**13.10%**	**0.7335**	**23.76%**
RN101	0.9400	0.9182	2.32%	0.8658	7.89%	0.8208	12.68%	0.6598	29.81%	0.6388	32.04%
DenseNet169	0.9239	0.9138	1.09%	0.8628	6.61%	0.8364	9.47%	0.7211	21.95%	0.6549	29.12%
EFB3	0.9499	0.9299	2.11%	0.8674	8.69%	0.8674	8.69%	0.7736	18.56%	0.6585	30.68%
ViT-B	0.9570	0.9492	0.82%	0.9007	5.88%	0.8748	8.59%	0.8377	12.47%	0.7128	25.52%
Swin-B	**0.9593**	**0.9524**	**0.72%**	**0.9108**	**5.06%**	**0.8996**	**6.22%**	**0.8446**	**11.96%**	**0.7511**	**21.70%**

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

Background	Ori	$\lambda = 0$		$\lambda = -20$		$\lambda = 20$		$\lambda = 100$		$\lambda = 20$-adv	
	Top-1	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR	Top-1	DR
RN50	0.9278	0.9069	2.25%	0.8495	8.44%	0.7939	14.43%	0.6276	32.36%	0.6222	32.94%
RN50-A	0.8202	0.8100	**1.24%**	0.7670	**6.49%**	0.6813	16.93%	0.5363	34.61%	0.4336	47.13%
RN50-SIN	0.9154	0.8920	2.56%	0.8350	8.78%	0.7960	13.04%	0.6370	30.41%	0.5786	36.79%
RN50-debiasd	0.9336	0.9196	1.50%	0.8665	7.19%	0.8139	12.82%	0.6600	29.31%	**0.6530**	**30.06%**
RN50-Augmix	**0.9352**	**0.9230**	1.30%	**0.8690**	7.08%	**0.8506**	**9.05%**	**0.8064**	**13.77%**	0.6275	32.90%
RN50-ANT	0.9186	0.9037	1.62%	0.8454	7.97%	0.8006	12.85%	0.7126	22.43%	0.5487	40.27%
RN50-DeepAugment	0.9290	0.9136	1.66%	0.8598	7.45%	0.7978	14.12%	0.5989	35.53%	0.5986	35.57%

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. 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'.

Table 5: Evaluations under different object sizes.

Object	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%	0.8563	7.71%	0.8225	11.35%	0.7109	23.38%	0.6514	29.79%	0.6756	27.18%
DenseNet121	0.9205	0.8855	3.80%	0.8497	7.69%	0.8117	11.82%	0.7073	23.16%	0.6533	29.03%	0.6852	25.56%
EF-B0	0.9285	0.8956	3.54%	0.8432	9.19%	0.8108	12.68%	0.6969	24.94%	0.6519	29.79%	0.7436	19.91%
ViT-S	0.9474	0.9352	**1.29%**	0.8798	7.14%	0.8407	11.26%	0.7429	21.59%	0.6990	26.22%	0.7773	17.95%
Swin-S	**0.9621**	**0.9492**	1.34%	**0.9193**	**4.45%**	**0.8970**	**6.77%**	**0.8157**	**15.22%**	**0.7881**	**18.09%**	**0.8238**	**14.37%**
RN101	0.9400	0.9129	2.88%	0.8763	6.78%	0.8384	10.81%	0.7302	22.32%	0.6824	27.40%	0.6880	26.81%
DenseNet169	0.9239	0.9014	2.44%	0.8501	7.99%	0.8230	10.92%	0.7162	22.48%	0.6700	27.48%	0.7146	22.65%
EF-B3	0.9499	0.9349	1.58%	0.8793	7.43%	0.8478	10.75%	0.7360	22.52%	0.6955	26.78%	0.7817	17.71%
ViT-B	0.9570	**0.9520**	**0.52%**	0.9101	4.90%	0.8864	7.38%	0.7996	16.45%	0.7642	20.15%	**0.8472**	**11.47%**
Swin-B	**0.9593**	0.9503	0.94%	**0.9262**	**3.45%**	**0.9122**	**4.91%**	**0.8343**	**13.03%**	**0.8068**	**15.90%**	0.8334	13.12%

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

Object size	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%	0.8563	7.71%	0.8225	11.35%	0.7109	23.38%	0.6514	29.79%	0.6756	27.18%
RN50-A	0.8202	0.7710	6.00%	0.7244	11.68%	0.6800	17.09%	0.5650	31.11%	0.4956	39.58%	0.4986	39.21%
RN50-SIN	0.9154	0.9005	1.63%	0.8280	9.55%	0.7823	14.54%	0.6723	26.56%	0.6188	32.40%	0.6441	29.64%
RN50-debiasd	0.9336	0.9129	2.22%	**0.8775**	**6.01%**	**0.8451**	**9.48%**	**0.7383**	**20.92%**	**0.6937**	**25.70%**	0.6850	26.63%
RN50-Augmix	**0.9352**	**0.9198**	1.65%	0.8743	6.51%	0.8308	11.16%	0.7190	23.12%	0.6558	29.88%	0.7093	24.16%
RN50-ANT	0.9186	0.9025	1.75%	0.8506	7.40%	0.8110	11.71%	0.7029	23.48%	0.6445	29.84%	0.6673	27.36%
RN50-DeepAugment	0.9290	0.9140	**1.61%**	0.8582	7.62%	0.8230	11.41%	0.7144	23.10%	0.6565	29.33%	**0.7171**	**22.81%**

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	0.7454	13.94%	0.6147	29.03%	DenseNet169	0.8766	0.7603	13.27%	0.6331	27.78%
RN50	0.8815	0.7112	19.32%	0.6295	28.59%	RN101	0.8951	0.7477	16.47%	0.6510	27.27%
EF-B0	0.8855	0.7521	15.06%	0.6197	30.02%	EF-B3	0.9212	0.8048	12.64%	0.6605	28.30%
ResNest50	0.9209	0.8031	12.79%	0.6998	24.01%	ResNest101	0.9279	0.8333	10.20%	0.7235	22.03%
ViT-S	0.9214	0.7845	14.86%	0.6930	24.79%	ViT-base	**0.9412**	0.8291	11.91%	0.7567	19.60%
Swin-S	**0.9310**	0.8288	10.98%	0.7520	19.23%	Swin-B	0.9316	0.8400	9.83%	0.7678	17.58%
ConvNeXt-T	0.9272	**0.8376**	**9.66%**	**0.7610**	**17.92%**	ConvNeXt-B	0.9406	**0.8603**	**8.54%**	**0.8024**	**14.69%**

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

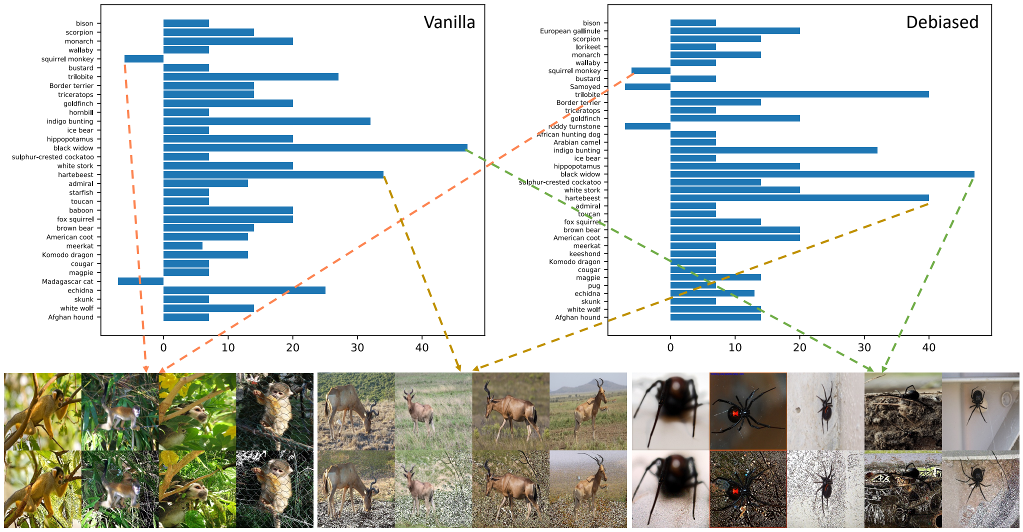


Figure 12: Dropped accuracy (%) in each class. Classes whose number of images is less than 15 or drop rate is zero are removed.

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.

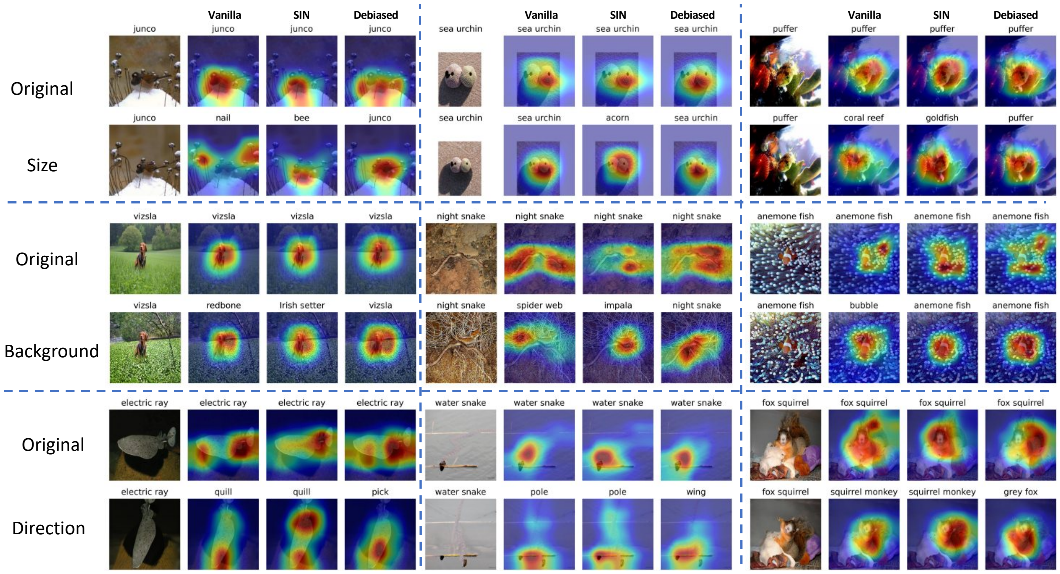


Figure 13: The heat map comparisons between original images and edited ones.

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., 2021). 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.

Architectures	Ori	Background changes					Size changes				Position	Direction
		$\lambda = 0$	$\lambda = -20$	$\lambda = 20$	$\lambda = 100$	$\lambda =$ 20-adv	Full	0.1	0.08	0.05	rp	rd
RN50d	0.9375	1.03%	**5.42%**	**7.23%**	12.83%	**20.32%**	2.77%	4.96%	7.66%	18.66%	21.62%	20.91%
ViT-S	0.9474	1.77%	8.42%	11.91%	19.48%	33.30%	**1.29%**	7.14%	11.26%	21.59%	26.22%	17.95%
R50d-hybrid	**0.9540**	**0.82%**	6.13%	7.70%	**11.08%**	21.87%	1.42%	**4.21%**	**6.65%**	**14.50%**	**18.25%**	**14.50%**

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) 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. Motivated by this success, we also test another kind of self-supervised-learning strategy. To be specific, we choose the representative method MoCo-V3 (Chen et al., 2021) in the contrastive learning family. However, it fails to get a boost. We suspect that the MoCo-V3 pays more attention to the global feature instead of the interested region since a small change in the background can lead to a high drop rate (19.29%) on accuracy.

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

Architectures	Ori	Background changes					Size changes				Position	Direction
		$\lambda = 0$	$\lambda = -20$	$\lambda = 20$	$\lambda = 100$	$\lambda =$ 20-adv	Full	0.1	0.08	0.05	rp	rd
ViT-B	0.9570	0.82%	5.88%	8.59%	12.47%	25.52%	0.52%	4.90%	7.38%	16.45%	20.15%	**11.47%**
MoCo-ViT-B	0.9318	2.47%	11.23%	19.29%	45.62%	38.74%	**-0.21%**	8.22%	12.32%	26.05%	35.17%	25.72%
MAE-ViT-B	**0.9612**	**0.75%**	**5.38%**	**6.70%**	**10.05%**	**21.75%**	0.82%	**3.07%**	**5.02%**	**12.80%**	**15.93%**	14.88%

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. 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.

F 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

Table 10: Evaluation of images generated with different backgrounds. The red ones in each row indicate the background with the worst performance of the corresponding models.

Models	Ori	Random_bg		Leopard		Eight diagrams		Checker		Stripe	
RN50	0.9278	0.7914	14.70%	0.5604	39.60%	**0.6438**	**30.61%**	0.6514	29.79%	0.6324	31.84%
RN50-A	0.8202	0.6673	18.64%	0.2553	68.87%	0.3725	54.58%	0.3244	60.45%	0.4701	42.68%
RN50-SIN	0.9154	0.7806	14.73%	0.6227	31.98%	0.4862	46.89%	0.5122	44.05%	0.5274	42.39%
RN50-debiasd	0.9336	**0.8104**	**13.20%**	**0.6893**	**26.17%**	0.6275	32.79%	0.6700	28.23%	0.6322	32.28%
RN50-Augmix	**0.9352**	0.8047	13.95%	0.5696	39.09%	0.5646	39.63%	**0.6866**	**26.58%**	**0.6566**	**29.79%**
RN50-ANT	0.9186	0.7642	16.81%	0.5717	37.76%	0.5898	35.79%	0.5198	43.41%	0.5444	40.74%
RN50-DeepAugment	0.9290	0.7971	14.20%	0.6312	32.06%	0.5779	37.79%	0.5960	35.84%	0.6182	33.46%

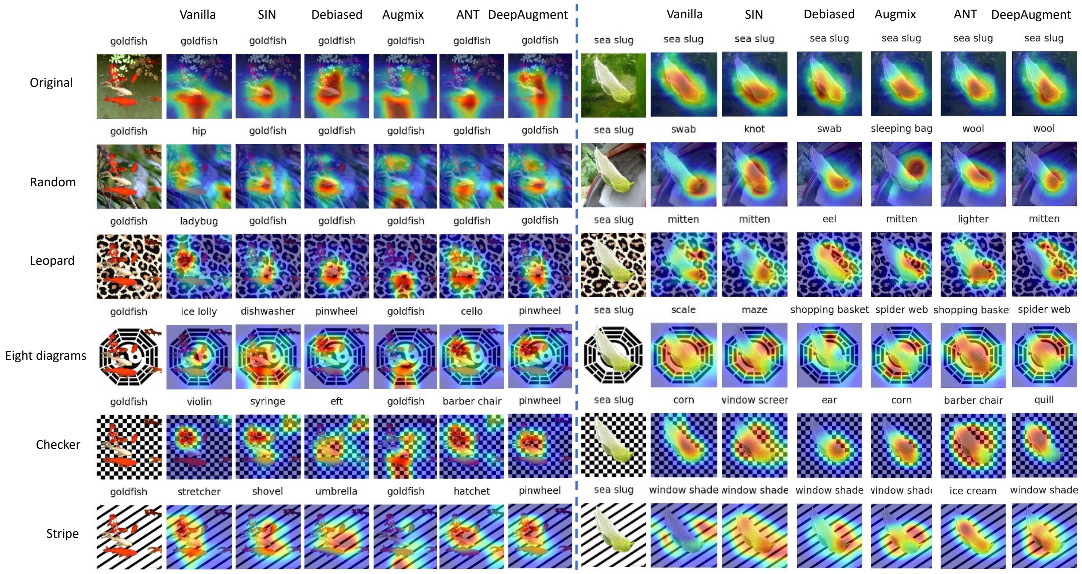


Figure 14: Heat maps under different backgrounds.

provides a strong baseline for adversarial training. Therefore, we choose their officially released adversarially 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.

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.

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

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

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.