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 [11][1]. Finally, our ImageNet-E consists of 373 classes. Since the number of masks provided in ImageNet-S [12] 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 C.3. The comprehensive comparisons with the state-of-the-art robustness benchmarks are shown in Figure 6. 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 8. A demo video for our editing toolkit can be found at this url:https://drive.google.com/file/d/1h5EV3MHPGgkBww9grhlvrl--kSIrD5Lp/view?usp=sharing. Our code can be found at an anonymous url: https://huggingface.co/spaces/Anonymous-123/ImageNet-Editing.

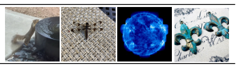
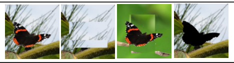
Benchmarks	Description	Classes	Samples
ImageNet-A	Challenging examples collected by-hand	200	
ImageNet-C	Corruptions added on images	1000	
ImageNet-R	Various renditions of ImageNet object classes	200	
ImageNet-3DCC	3D common corruptions	1000	
ImageNet-9	Images whose objects and backgrounds are disentangled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

Figure 6. Benchmark comparison.

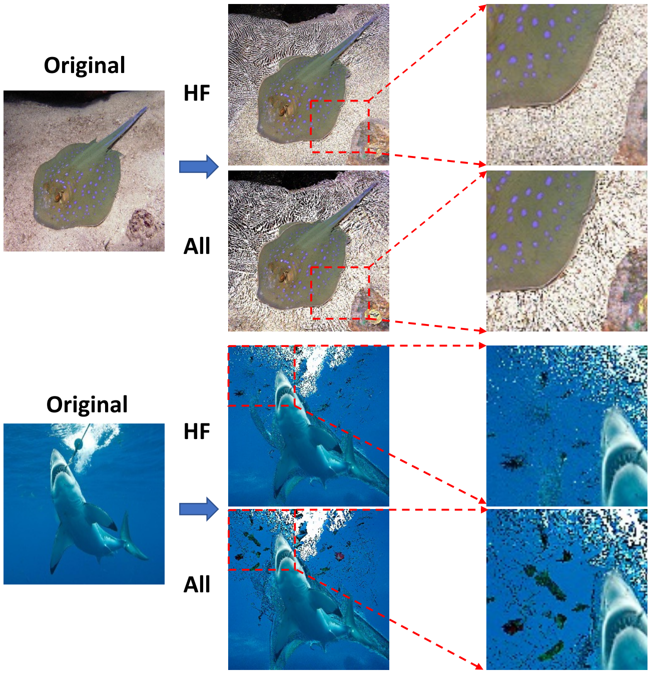


Figure 7. Comparisons between the amplitude supervision on high-frequency components (HF) and amplitude supervision on all frequency components (All).

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 7. 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].

C. Experimental details

C.1. Details for metrics

In this paper, we care more about how different attributes impact different models. Therefore, we choose the drop of top-1 accuracy as our evaluation metric. A lower dropped

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

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

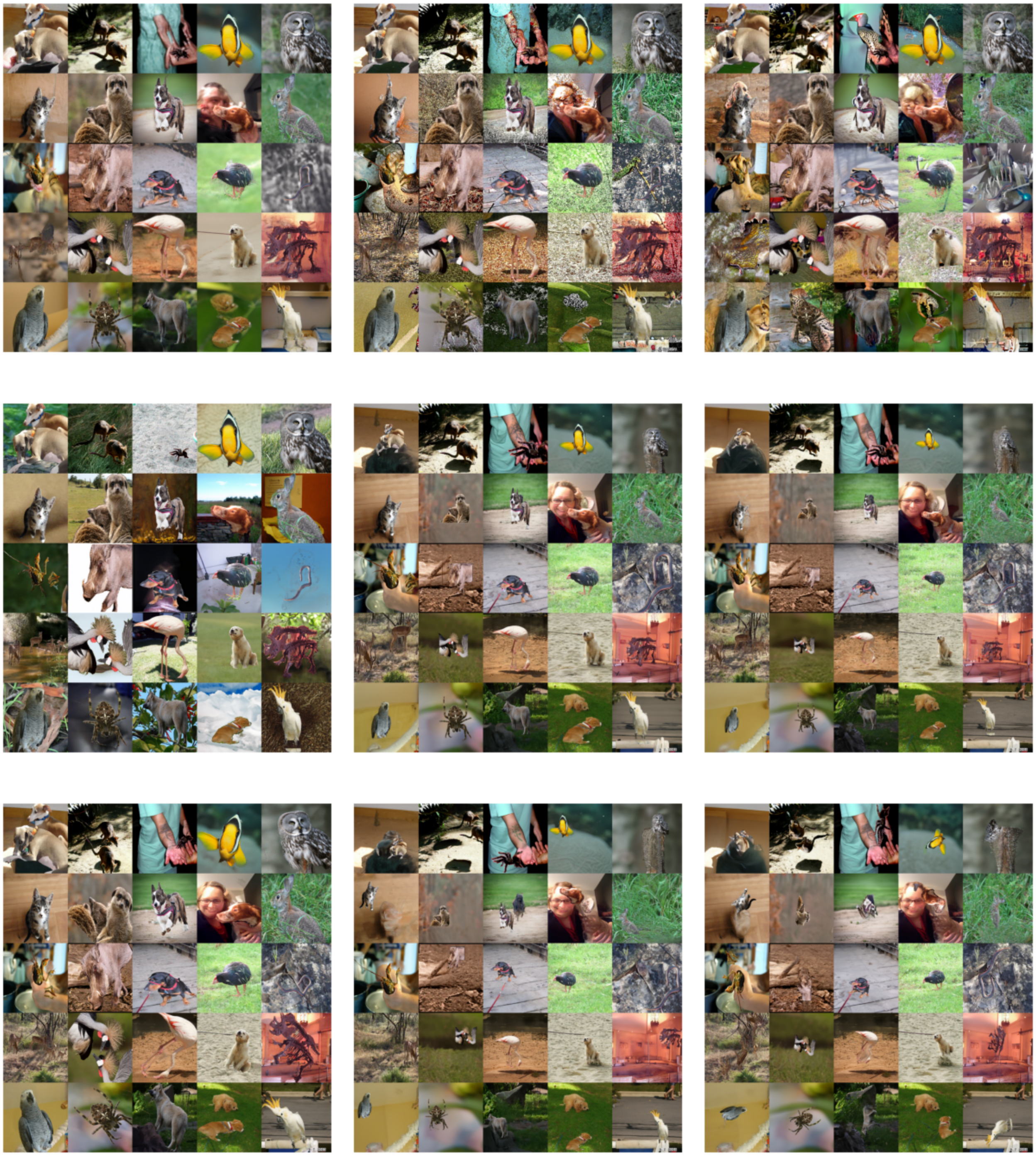


Figure 8. Samples from ImageNet-E. From left to right, top to bottom, the images stand for background editing with $\lambda = -20$, $\lambda = 20$, $\lambda = 20$-adv, randomly shuffled backgrounds, size editing with rate 0.1 and 0.05, randomly rotate, random position, randomly rotate based on images with object pixel rate 0.05 respectively.

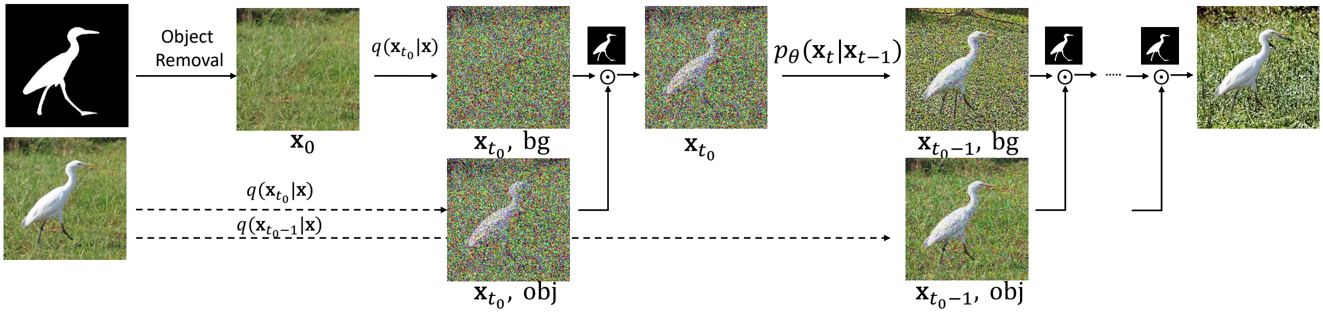

Object Removal
$q(\mathbf{x}_{t_0}|\mathbf{x})$
$p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$
$\mathbf{x}_{t_0}$, bg
$\mathbf{x}_{t_0}$
$\mathbf{x}_{t_0-1}$, bg
$q(\mathbf{x}_{t_0}|\mathbf{x})$
$q(\mathbf{x}_{t_0-1}|\mathbf{x})$
$\mathbf{x}_{t_0}$, obj
$\mathbf{x}_{t_0-1}$, obj

Figure 9. Attribute editing with DDPMs. Give an input image and its corresponding object mask, the object is firstly removed with inpainting operation to get the pure background image. Then, we leverage the diffusion process to edit the background image $\mathbf{x}_0$ and object image coherently. ⊙ denotes the element-wise blending of these two images using the object mask. For background editing, the background complexity objective function is added during the diffusion process (Alg. 1, line 5). For other object attributes editing, the object image needs to be transformed first (Alg. 2, line 1).

accuracy indicates higher robustness against our attribute changes. The dropped accuracy is defined as:

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

The detailed top-1 accuracy (Top-1) and dropped accuracy (DA)on our ImageNet-E are listed in Table 4, Table 5 and Table 6, Table 7. All the experiments are conducted for 5 runs and we report the mean value in the tables.

C.2. Classes whose accuracy drops the greatest

To find out which class gets the worst robustness against attribute changes, we plot the dropped accuracy in Figure 10. 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'.

C.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 8. The same conclusion can also be observed. For instance, most models show vulnerability against attribute changing since the average dropped accuracies reach 12.22% and 22.21% 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.

C.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 11. 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 5 also validate this. The dropped accuracy of RN50 (13.35%) is higher than SIN (12.19%) and Debiased (11.45%) even though the original accuracy of SIN (0.9157) is lower than vanilla RN50 (0.9269). 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 11, 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 dropped accuracy in the size change scenario is 24.23% for SIN, even lower than vanilla RN50, whose dropped accuracy is 21.26%. 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.

Table 4. Evaluations under different backgrounds.

Models	Ori	Inver		$\lambda = -20$		$\lambda = 20$		$\lambda = 20$-Adv		Random	
	Top-1	Top-1	DA	Top-1	DA	Top-1	DA	Top-1	DA	Top-1	DA
RN50	92.69%	90.72%	1.97%	85.39%	7.30%	79.34%	13.35%	62.77%	29.92%	79.35%	13.34%
DenseNet121	92.10%	90.61%	1.49%	85.81%	6.29%	83.10%	9.00%	62.90%	29.20%	79.67%	12.43%
EF-B0	92.85%	91.78%	1.07%	85.75%	7.10%	82.14%	10.71%	57.97%	34.88%	77.21%	15.64%
ResNest50	95.38%	93.94%	1.44%	89.05%	6.33%	86.40%	8.98%	68.76%	26.62%	84.10%	11.28%
ViT-S	94.14%	93.32%	**0.82%**	87.72%	6.42%	85.16%	8.98%	63.02%	31.12%	81.08%	13.06%
Swin-S	**96.21%**	**95.08%**	1.13%	91.03%	5.18%	88.88%	7.33%	72.71%	23.50%	86.90%	9.31%
ConvNeXt-T	96.07%	94.64%	1.43%	**91.38%**	**4.69%**	**89.81%**	**6.26%**	**76.24%**	**19.83%**	**88.14%**	**7.93%**
RN101	94.00%	91.89%	2.11%	86.95%	7.05%	82.38%	11.62%	64.53%	29.47%	80.43%	13.57%
DenseNet169	92.37%	91.25%	1.12%	86.56%	5.81%	83.94%	8.43%	64.86%	27.51%	80.76%	11.61%
EF-B3	94.97%	93.10%	1.87%	87.20%	7.77%	86.57%	8.40%	65.07%	29.90%	82.05%	12.92%
ResNest101	95.54%	94.44%	1.10%	89.96%	5.58%	88.89%	6.65%	72.51%	23.03%	85.14%	10.40%
ViT-B	95.38%	94.55%	0.83%	90.06%	5.32%	86.95%	8.43%	68.78%	26.60%	84.40%	10.98%
Swin-B	95.96%	95.17%	0.79%	91.50%	4.46%	89.73%	6.23%	74.52%	21.44%	87.71%	8.25%
ConvNeXt-B	**96.42%**	**95.73%**	**0.69%**	**92.67%**	**3.75%**	**91.56%**	**4.86%**	**79.93%**	**16.49%**	**90.38%**	**6.04%**

Table 5. Evaluations with different robust models under different backgrounds.

Models	Ori	Inver		$\lambda = -20$		$\lambda = 20$		$\lambda = 20$-Adv		Random	
	Top-1	Top-1	DA	Top-1	DA	Top-1	DA	Top-1	DA	Top-1	DA
RN50	92.69%	90.72%	1.97%	85.39%	7.30%	79.34%	13.35%	62.77%	29.92%	79.35%	13.34%
RN50-A	81.96%	81.30%	0.66%	77.21%	4.75%	68.34%	13.62%	44.09%	37.87%	66.71%	15.25%
RN50-SIN	91.57%	89.34%	2.23%	83.96%	7.61%	79.38%	12.19%	58.41%	33.16%	77.99%	13.58%
RN50-debiasd	93.34%	91.91%	1.43%	87.25%	6.09%	81.89%	11.45%	65.35%	27.99%	81.22%	12.12%
RN50-Augmix	93.50%	92.52%	**0.98%**	87.24%	6.26%	85.12%	8.38%	63.01%	30.49%	80.56%	12.94%
RN50-ANT	91.87%	90.19%	1.68%	85.25%	6.62%	79.93%	11.94%	56.21%	35.66%	76.51%	15.36%
RN50-DeepAugment	92.88%	91.38%	1.50%	86.26%	6.62%	80.51%	12.37%	60.48%	32.40%	79.56%	13.32%
RN50-T	**94.55%**	**93.50%**	1.05%	**88.90%**	**5.65%**	**87.17%**	**7.38%**	**72.66%**	**21.89%**	**84.13%**	**10.42%**

C.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 [3]. 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 9. 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.

Training strategy—-Masked image modeling. Considering that masked image modeling has demonstrated impressive results in self-supervised representation learning by recovering corrupted image patches [4], it may be robust to the attribute changes. Thus, we test the Masked AutoEncoder (MAE) [19] and SimMIM [52] training strategy based on ViT-B backbone. As shown in Table 10, the dropped accuracies 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 [8] in the contrastive learning family. After the end-to-end finetuning, it achieves top-1 83.0% accuracy on ImageNet. It can also improve the attribute robustness when compared to the vanilla ViT-B, showing the effectiveness of contrastive learning.

C.6. Hardware

Our experiments are implemented by PyTorch [39] and runs on RTX-3090TI.

D. Further exploration on backgrounds

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

Table 6. Evaluations under different object sizes.

Models	Ori	Full		0.10		0.08		0.05		0.05-rp		rd	
	Top-1	Top-1	DA	Top-1	DA	Top-1	DA	Top-1	DA	Top-1	DA	Top-1	DA
RN50	92.69%	89.98%	2.71%	85.44%	7.25%	82.18%	10.51%	71.43%	21.26%	66.23%	26.46%	67.57%	25.12%
DenseNet121	92.10%	88.60%	3.50%	85.10%	7.00%	81.42%	10.68%	70.55%	21.55%	65.57%	26.53%	68.46%	23.64%
EF-B0	92.85%	89.82%	3.03%	84.85%	8.00%	81.28%	11.57%	69.57%	23.28%	64.94%	27.91%	73.74%	19.11%
ResNest50	95.38%	92.85%	2.53%	90.11%	5.27%	87.37%	8.01%	77.35%	18.03%	74.01%	21.37%	78.06%	17.32%
ViT-S	94.14%	93.34%	0.80%	88.77%	5.37%	85.55%	8.59%	76.77%	17.37%	71.28%	22.86%	77.01%	17.13%
Swin-S	**96.21%**	**94.94%**	**1.27%**	92.00%	4.21%	89.92%	6.29%	82.05%	14.16%	78.86%	17.35%	**82.79%**	**13.42%**
ConvNeXt-T	96.07%	94.32%	1.75%	**92.79%**	**3.28%**	**90.89%**	**5.18%**	**83.31%**	**12.76%**	**80.36%**	**15.71%**	80.29%	15.78%
RN101	94.00%	91.43%	2.57%	87.19%	6.81%	83.88%	10.12%	73.35%	20.65%	68.15%	25.85%	69.58%	24.42%
DenseNet169	92.37%	90.12%	2.25%	85.47%	6.90%	81.96%	10.41%	71.78%	20.59%	67.44%	24.93%	71.69%	20.68%
EF-B3	94.97%	93.61%	1.36%	88.17%	6.80%	84.81%	10.16%	73.61%	21.36%	69.99%	24.98%	77.73%	17.24%
ResNest101	95.54%	94.19%	1.35%	91.57%	3.97%	89.01%	6.53%	80.10%	15.44%	76.43%	19.11%	81.23%	14.31%
ViT-B	95.38%	94.76%	**0.62%**	91.38%	4.00%	89.08%	6.30%	80.87%	14.51%	76.56%	18.82%	80.43%	14.95%
Swin-B	95.96%	94.97%	0.99%	92.80%	3.16%	90.92%	5.04%	83.62%	12.34%	80.58%	15.38%	83.36%	**12.60%**
ConvNeXt-B	**96.42%**	**95.43%**	0.99%	**94.17%**	**2.25%**	**93.06%**	**3.36%**	**86.95%**	**9.47%**	**84.02%**	**12.40%**	**83.41%**	13.01%

Table 7. 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	DA	Top-1	DA	Top-1	DA	Top-1	DA	Top-1	DA	Top-1	DA
RN50	92.69%	89.98%	2.71%	85.44%	7.25%	82.18%	10.51%	71.43%	21.26%	66.23%	26.46%	67.57%	25.12%
RN50-A	81.96%	77.09%	4.87%	72.34%	9.62%	68.02%	13.94%	56.45%	25.51%	49.45%	32.51%	50.00%	31.96%
RN50-SIN	91.57%	89.89%	1.68%	83.27%	8.30%	78.97%	12.60%	67.34%	24.23%	62.41%	29.16%	64.33%	27.24%
RN50-debiasd	93.34%	91.36%	1.98%	87.81%	5.53%	84.58%	8.76%	74.07%	19.27%	69.33%	24.01%	68.37%	24.97%
RN50-Augmix	93.50%	91.89%	1.61%	87.10%	6.40%	83.53%	9.97%	72.08%	21.42%	66.36%	27.14%	71.08%	22.42%
RN50-ANT	91.87%	90.30%	1.57%	84.75%	7.12%	81.25%	10.62%	70.38%	21.49%	65.21%	26.66%	66.64%	25.23%
RN50-DeepAugment	92.88%	91.52%	**1.36%**	85.61%	7.27%	82.26%	10.62%	71.60%	21.28%	66.60%	26.28%	71.59%	21.29%
RN50-T	**94.55%**	**92.44%**	2.11%	**89.81%**	**4.74%**	**86.72%**	**7.83%**	**77.09%**	**17.46%**	**73.43%**	**21.12%**	**74.95%**	**19.60%**

backgrounds. The evaluation results are shown in Table 12. It can be observed that the background changes can lead to a 13.34% accuracy drop. When the background is set to be a leopard or other images, the dropped accuracy can even reach 35.52%. 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 6. 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.

E. Further discussion on the distribution

In this paper, 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. Thus, we choose the out-of-distribution (OOD) detection methods including Energy [34] and GradNorm [27] following DRA [55] 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.

Covariate shift adaptation(*a.k.a* batch-norm adaptation, BNA) is a way for improving robustness against common corruptions [44]. Thus, it can help to get a top-1 accuracy performance boost in OOD data. 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 ($0.7615 \rightarrow 0.7613$) and our ImageNet-E ($0.7934 \rightarrow 0.7933$ under $\lambda = 20$, $0.6521 \rightarrow 0.6514$ under random position scenario, mean accuracy of 5 runs). This similar tendency implies that our ImageNet-E shares a similar property with ImageNet.

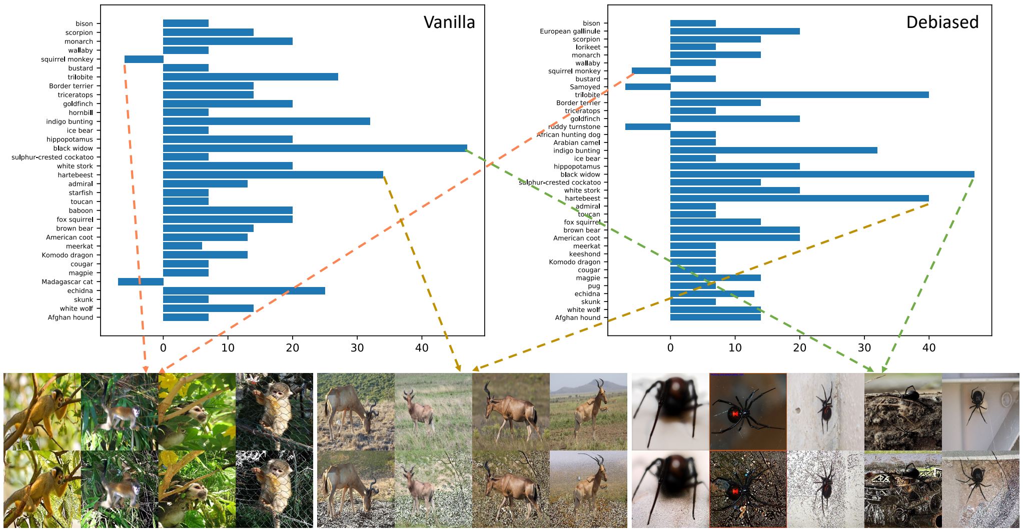


Figure 10. Dropped accuracy (%) in each class. Classes whose number of images is less than 15 or dropped accuracy is zero are removed.

Table 8. Evaluations with more data.

Models	Original	Background		Size-0.05		Models	Original	Background		Size-0.05	
	Top-1	Top-1	DA	Top-1	DA		Top-1	Top-1	DA	Top-1	DA
DenseNet121	86.60%	74.73%	11.87%	61.48%	25.12%	DenseNet169	87.66%	76.26%	11.40%	63.57%	24.09%
RN50	88.12%	71.64%	16.48%	63.13%	24.99%	RN101	89.52%	75.33%	14.19%	65.11%	24.41%
EF-B0	88.54%	75.64%	12.90%	62.16%	26.38%	EF-B3	92.12%	80.81%	11.31%	66.18%	25.96%
ResNest50	92.12%	80.61%	11.51%	70.05%	22.07%	ResNest101	92.78%	83.46%	9.32%	72.67%	20.11%
ViT-S	92.15%	78.94%	13.21%	69.30%	22.85%	ViT-B	**94.12%**	83.04%	11.08%	75.65%	18.47%
Swin-S	**93.11%**	82.98%	10.13%	75.36%	17.75%	Swin-B	93.18%	84.11%	9.07%	76.99%	16.19%
ConvNeXt-T	92.75%	**84.00%**	**9.43%**	**76.41%**	**16.34%**	ConvNeXt-B	94.05%	**86.41%**	**7.64%**	**80.34%**	**13.71%**

F. 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 [40] and EfficientNet-L2-Noisy-Student [51]. The average dropped accuracy in terms of different models can be found in Figure 12. CLIP shows a good robustness to out-of-distribution data [31]. Therefore, to find out whether the CLIP can also show a good robustness against attribute editing, we evaluate the CLIP model (Backbone ViT-B) 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 [49]. It acquires a 81.2% top-1 accuracy on ImageNet validation set while it is 68.3% for zero-shot version. The evaluation on ImageNet-E is shown in Table 11 and Table 13. 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 13. 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 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. As the network gets larger, its attribute robustness gets better.

While EfficientNet-L2-Noisy-Student is one of the top models on ImageNet-A benchmark [51], 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 5.1.

Evaluations on 91 state-of-the-art models can be found in Figure 13. All the evaluated models in this figure are all

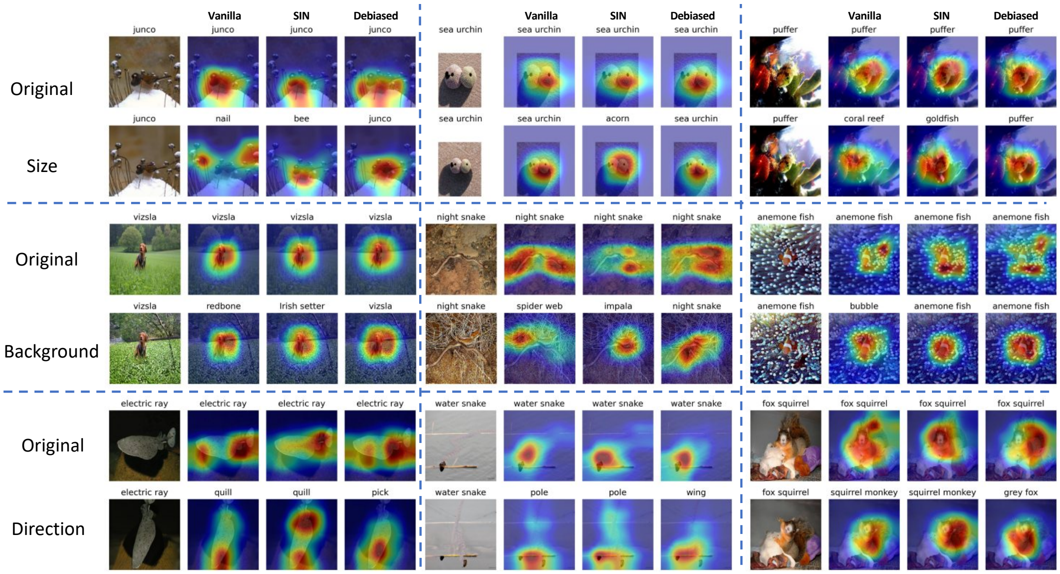


Figure 11. The heat map comparisons between original images and edited ones.

Table 9. Ablation study of the self-attention-like architecture.

Models	Ori	Background changes					Size changes				Position	Direction	Avg.
		Inver	$\lambda = -20$	$\lambda = 20$	$\lambda = 20$-Adv	Random	Full	0.1	0.08	0.05	rp	rd	
R50d	93.77%	1.23%	**4.80%**	**6.48%**	**19.39%**	**8.28%**	2.82%	4.36%	7.07%	16.95%	20.49%	19.31%	11.00%
ViT-S	94.74%	1.66%	7.32%	10.64%	32.17%	14.39%	**1.22%**	7.10%	10.64%	20.29%	25.08%	17.22%	14.61%
R50-hybrid	**95.40%**	**1.04%**	5.64%	7.16%	21.54%	9.19%	1.37%	**3.53%**	**5.92%**	**13.92%**	**17.23%**	**14.12%**	**9.96%**

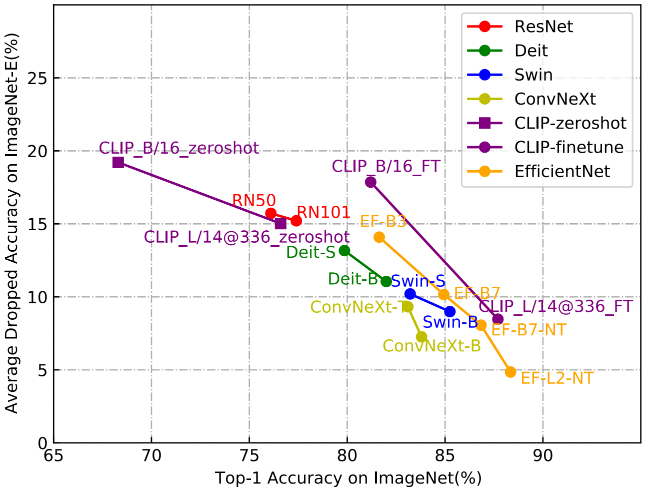


Figure 12. The average accuracy drop of different models. The x-axis is the model's top-1 accuracy on ImageNet.

provided by the timm library, except for the MoCo-V3-FT and CLIP-FT, which are finetuned by us.

G. Failure cases of generated images

The failure cases of generated images are shown in Figure 15. 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 C.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.

H. Related literature to robustness enhancements

Adversarial training. [43] 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 adversarially trained models[3] as the evaluation model. Models with different architectures are adopted here[4].

SIN [14] provides evidence that machine recognition to-

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

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

Table 10. Ablation study of the self-supervised models. All the compared models are end-to-end finetuned on ImageNet except for ViT-B, which is supervised trained from the early start.

Models	Ori	Background changes					Size changes				Position	Direction	Avg.
		Inver	$\lambda=-20$	$\lambda=20$	$\lambda=$ 20-Adv	Random	Full	0.1	0.08	0.05	rp	rd	
ViT-B	95.38%	0.83%	5.32%	8.43%	26.60%	10.98%	0.62%	4.00%	6.30%	14.51%	18.82%	14.95%	11.05%
CLIP_finetune	93.68%	2.17%	9.82%	11.83%	38.33%	18.19%	9.06%	9.25%	12.67%	23.32%	28.56%	22.00%	18.30%
MoCo-v3	95.70%	**0.55%**	4.91%	7.33%	24.33%	9.92%	0.92%	3.76%	5.62%	13.61%	17.85%	15.20%	10.35%
MAE-ViT-B	96.12%	0.78%	**4.77%**	**6.21%**	**21.09%**	**8.18%**	**0.78%**	**3.01%**	**4.86%**	**12.10%**	**15.47%**	14.00%	**9.05%**
SimMIM	**96.14%**	0.75%	5.13%	6.76%	23.58%	9.33%	0.97%	3.22%	5.33%	13.18%	17.12%	**13.62%**	9.82%

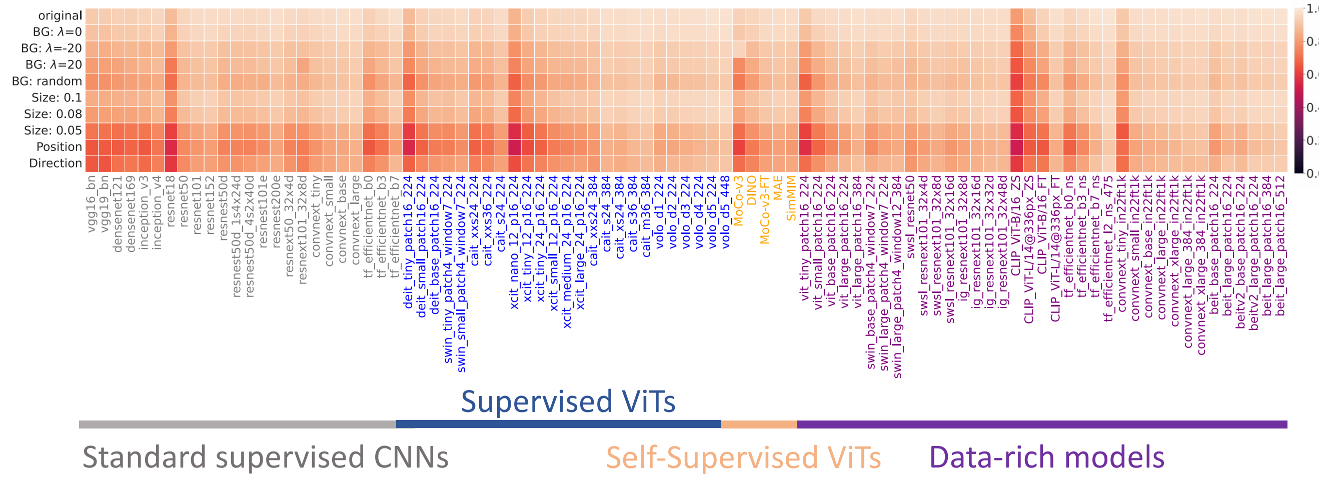


Figure 13. The top-1 accuracy performance under different editing scenarios of 91 state-of-the-art models.

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
CLIP-zero-shot	68.3	61.9	**50.1**	**43.1**	**77.6**	**48.3**	62.1
CLIP-FT	**81.2**	**70.7**	35.3	47..9	65.0	44.9	**77.2**

day 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 [32] 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 [23] 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 [41] 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 [21]. 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.

There are some more tables and figures in the next pages.

Table 12. Evaluation of images generated with different backgrounds.

Models	Original	Random_bg		Leopard		Eight diagrams		Checker		Stripe	
		Top-1	DA	Top-1	DA	Top-1	DA	Top-1	DA	Top-1	DA
RN50	92.69%	79.35%	13.34%	57.17%	35.52%	64.32%	28.37%	65.13%	27.56%	62.90%	29.79%
RN50-A	81.96%	66.71%	15.25%	25.05%	56.91%	37.21%	44.75%	32.47%	49.49%	46.96%	35.00%
RN50-SIN	91.57%	77.99%	13.58%	62.74%	28.83%	48.74%	42.83%	51.15%	40.42%	52.65%	38.92%
RN50-debiasd	93.34%	81.22%	12.12%	68.58%	24.76%	62.68%	30.66%	67.10%	26.24%	63.16%	30.18%
RN50-Augmix	93.50%	80.56%	12.94%	57.35%	36.15%	56.20%	37.30%	68.78%	24.72%	65.68%	27.82%
RN50-ANT	91.87%	76.51%	15.36%	58.11%	33.76%	59.04%	32.83%	51.91%	39.96%	54.69%	37.18%
RN50-DeepAugment	92.88%	79.56%	13.32%	62.83%	30.05%	57.71%	35.17%	59.46%	33.42%	61.80%	31.08%
R50-T	**94.55%**	**84.13%**	**10.42%**	**72.93%**	**21.62%**	**73.98%**	**20.57%**	**79.42%**	**15.13%**	**76.43%**	**18.12%**

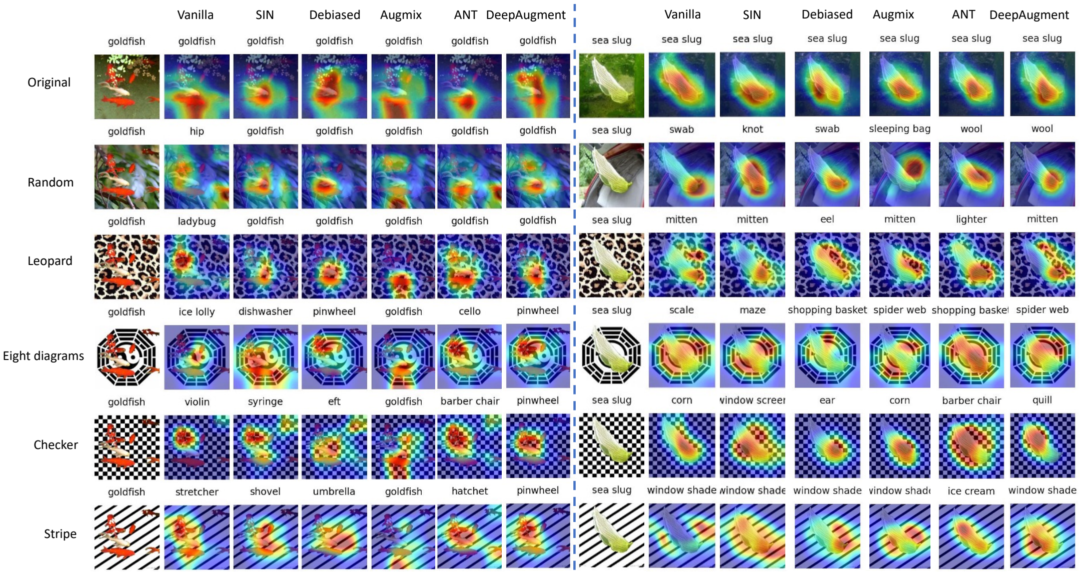


Figure 14. Heat maps under different backgrounds.

Table 13. More evaluations on state-of-the-art models including CLIP and EfficientNet-L2-Noisy-Student.

Models	Ori	Background changes					Size changes				Position	Direction	Avg.
		Inver	$\lambda = -20$	$\lambda = 20$	$\lambda = $ 20-Adv	Random	Full	0.1	0.08	0.05	rp	rd	
ViT-B/16	95.38%	**0.83%**	5.32%	8.43%	26.60%	10.98%	**0.62%**	4.00%	6.30%	14.51%	18.82%	14.95%	11.05%
Zero-shot													
CLIP_RN50	72.38%	6.03%	11.64%	16.72%	35.07%	21.82%	8.78%	14.39%	17.69%	26.48%	29.79%	25.31%	20.77%
CLIP_RN101	73.35%	4.51%	10.77%	14.42%	33.42%	19.63%	6.39%	14.53%	18.19%	26.58%	30.08%	24.51%	19.85%
CLIP_RN50x4	77.18%	4.64%	10.44%	13.27%	31.39%	18.51%	7.46%	12.37%	15.66%	24.23%	27.19%	24.25%	18.48%
CLIP_RN50x16	82.10%	4.39%	10.10%	12.41%	27.14%	16.62%	6.62%	11.10%	13.53%	22.09%	25.27%	23.13%	16.80%
CLIP_RN50x64	85.66%	4.77%	**8.89%**	**10.79%**	**23.75%**	**13.44%**	6.39%	9.20%	11.92%	**19.17%**	**21.62%**	20.57%	**14.57%**
CLIP_ViT-B/32	74.08%	5.55%	13.24%	18.64%	43.26%	26.39%	2.99%	15.59%	19.74%	29.05%	33.37%	24.89%	22.72%
CLIP_ViT-B/16	80.01%	4.88%	11.56%	15.28%	36.14%	20.09%	4.88%	12.67%	15.77%	25.31%	28.87%	21.57%	19.21%
CLIP_ViT-L/14	87.61%	4.35%	11.04%	14.46%	33.69%	18.35%	**1.81%**	11.67%	15.09%	23.66%	27.19%	18.05%	17.50%
CLIP_ViT-L/14-336	**88.01%**	**3.16%**	9.07%	12.25%	29.69%	16.08%	3.16%	**9.20%**	**11.78%**	19.94%	22.89%	**16.15%**	15.02%
CLIP_ViT-L/14-336	**88.01%**	**3.16%**	9.07%	12.25%	29.69%	16.08%	3.16%	**9.20%**	**11.78%**	19.94%	22.89%	**16.15%**	15.02%
Finetune													
CLIP_ViT-B/16-FT	93.68%	2.17%	9.82%	11.83%	38.33%	18.19%	4.66%	9.25%	12.67%	23.32%	28.56%	22.00%	17.86%
CLIP_ViT-L/14-336-FT	**96.97%**	1.29%	**5.16%**	**6.18%**	**19.93%**	**8.09%**	1.29%	**3.47%**	**4.90%**	**10.98%**	**13.74%**	**10.96%**	**8.47%**
EF-B0	92.85%	**1.07%**	7.10%	10.71%	34.88%	15.64%	3.03%	8.00%	11.57%	23.28%	27.91%	19.11%	16.12%
EF-B0-NT	94.30%	1.97%	8.43%	10.51%	34.93%	15.99%	1.79%	7.91%	11.50%	22.96%	27.62%	19.07%	16.07%
EF-B7	97.10%	1.80%	6.37%	7.20%	23.36%	10.78%	1.65%	4.16%	6.25%	14.13%	17.12%	10.56%	10.16%
EF-B7-NT	97.38%	1.30%	5.26%	6.10%	19.96%	9.15%	0.55%	3.31%	4.75%	10.67%	12.87%	7.98%	8.06%
EF-L2-NT-475	**97.84%**	1.08%	3.60%	4.51%	14.88%	7.14%	0.51%	2.21%	**2.71%**	5.50%	7.35%	4.58%	5.30%
EF-L2-NT	97.63%	1.26%	**3.50%**	**4.06%**	**12.73%**	**6.90%**	**0.71%**	**2.27%**	2.79%	**5.01%**	**6.03%**	**4.55%**	**4.85%**

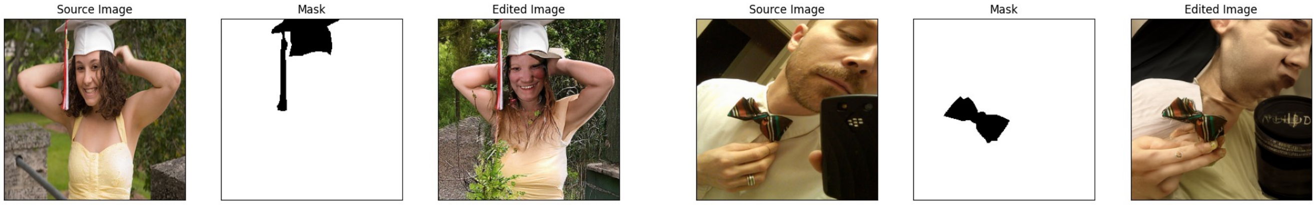


Figure 15. The failure cases of attribute editing.