A Appendix

Overview of Appendix

In the following, we provide a brief overview of the additional experiments reported in the Appendix.

- In App. B, we show how varying l_1 regularization weight (B.1), the starting point of the diffusion process T (B.2), diversity (B.3), and different levels of robustness (B.4) influence the DVCEs. In B.5, we add the ablation study of the angle used in the cone projection.
- In App. C, we show the results of our user study which shows that both quantitatively and qualitatively DVCEs generate more meaningful features compared to $l_{1.5}$-SVCEs [7] and BDVCEs [2].
- In App. D, we describe the hardware and resources used.
- In App. E, we explain the quantitative evaluation of the realism, validity, and closeness of our DVCEs.
- In App. F, we show, how our DVCEs can help to uncover spurious features.
- In App. G, we show some of the failure cases of our method.

B Ablation study of DVCEs

B.1 Regularization

In this section, we start by evaluating the impact of the distance regularization term on the diffusion process. We want VCEs to resemble the original image in the overall appearance and only change class-specific features to transfer the image into the target class. To achieve sparse changes, we use l_1-distance regularization. In Fig. 7, we vary the regularization strengths from 0.05 to 0.25. As can be seen, all regularization strengths generate meaningful target class-specific features, however, regularization C_d of 0.15 gives a good trade-off between being close to the original image as well as being realistic. For the rest of the evaluation, we, therefore, chose a weight of 0.15.

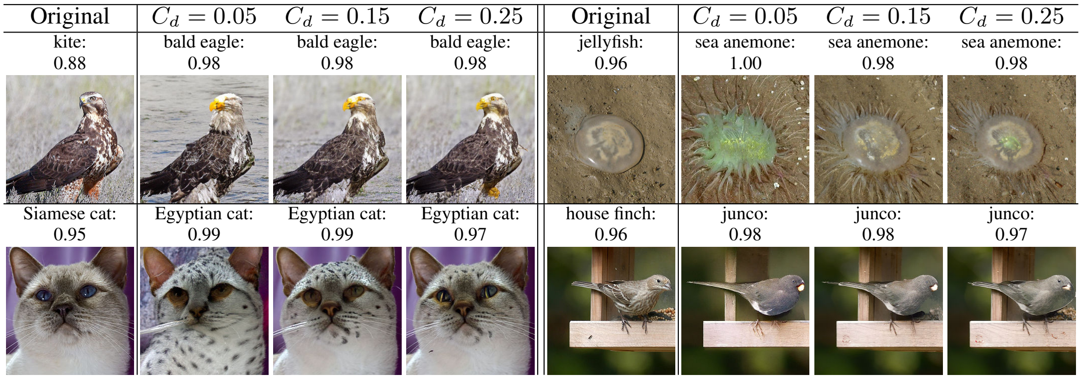


Figure 7: Ablation study for varying weights C_d for the l_1-distance term in our DVCEs for the non-robust Swin-TF using the cone projection introduced in Sec. 3.2.

B.2 Starting T_{start}

Next, we show how the starting time of the diffusion process influences the DVCEs. Diffusion-based sampling that is not conditioned on an original image starts the process at standard normally distributed noise. However, we show that it is easier to obtain high-quality VCEs that are similar to the original image by instead starting in the middle of the diffusion process by sampling from the closed-form probability distribution corresponding to timestep T_{start} (3). For this, we compare three different settings and define $\eta := \frac{T_{\text{start}}}{T}$ where we choose $\eta \in \{0.25, 0.5, 0.75\}$. We observe again that $\eta = 0.5$ gives a good trade-off between closeness, realism, and showing class-specific features. Thus, in all our experiments in this paper, we chose $\eta = 0.5$.

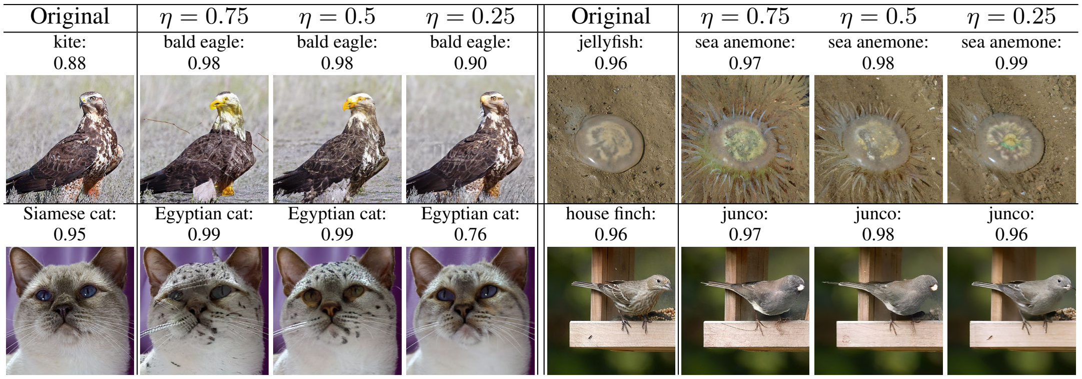


Figure 8: Comparison of different starting times $\eta := \frac{T_{\text{start}}}{T}$ of the diffusion process for our DVCEs for the non-robust Swin-TF using the cone projection.

B.3 Diversity

Note that the diffusion process is by design stochastic. This means that, unlike optimization-based VCEs, it is possible to generate a diverse set of VCEs from the same starting image. In Fig. 9, we visualize different DVCEs obtained for the non-robust Swin-TF using the cone projection.

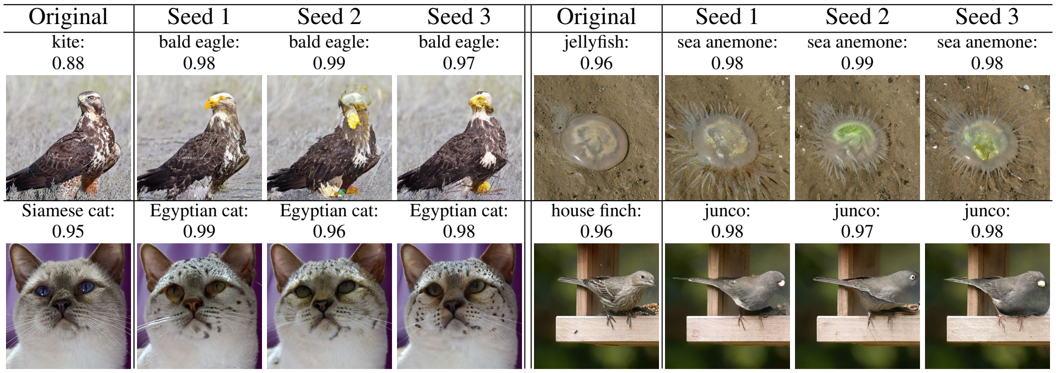


Figure 9: DVCEs for the non-robust Swin-TF using the cone projection across 3 different seeds for the standard parameters used in the paper. The DVCEs for different seeds show subtle variations of the generated images and satisfy the desired properties of VCEs introduced in Sec. 3.

B.4 Different robust models

In this section we first show more examples in Fig. 10 for the qualitative comparison of different robust models described in Sec. 4.2.

Further, in Fig. 11, we compare cone projection of 3 robust and 3 non-robust models.

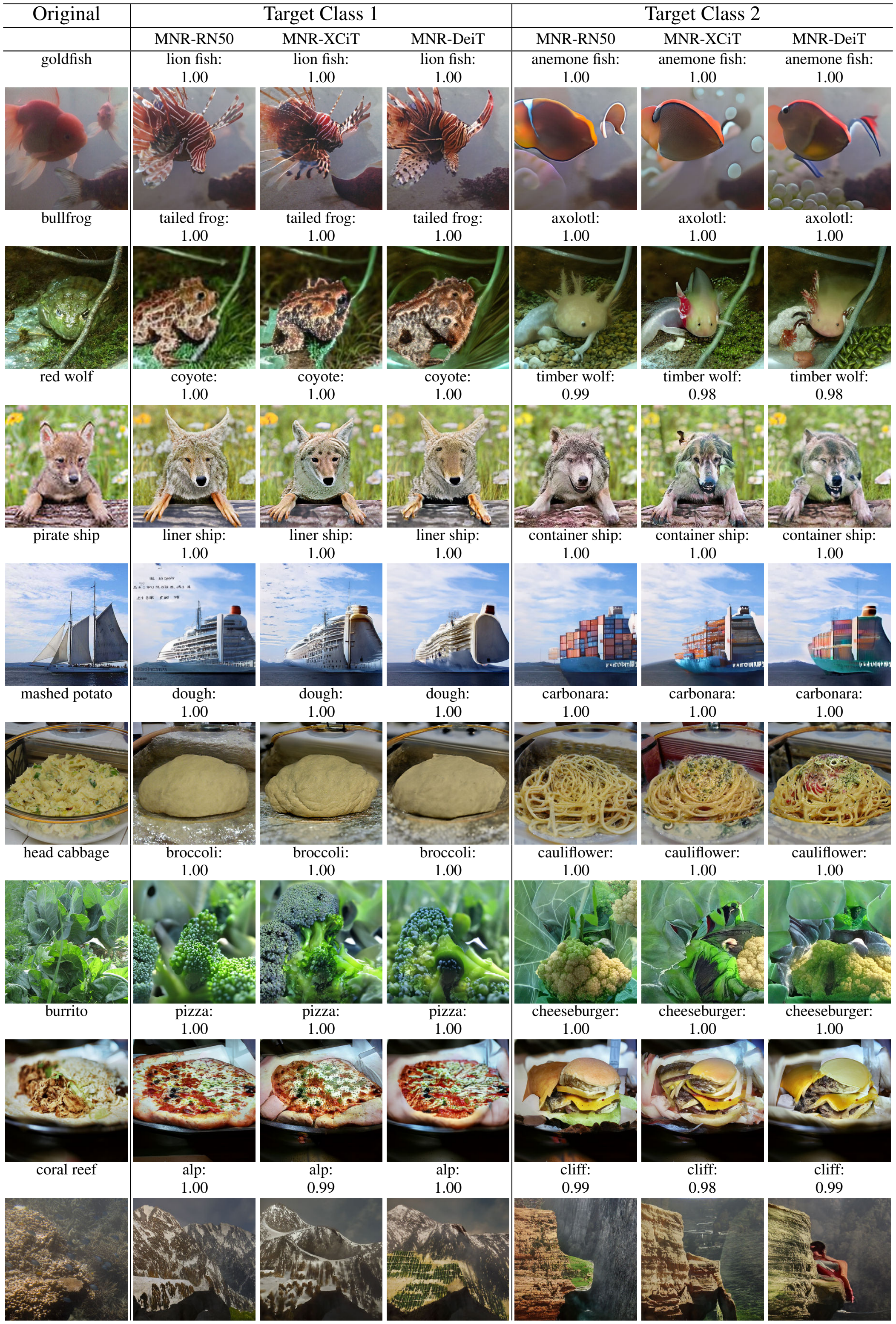


Figure 10: We compare DVCEs for three different robust models (no cone projection) which are all fine-tuned to be multiple-norm adversarially robust [11], that is againstl_1, l_2 and l_∞-perturbations.

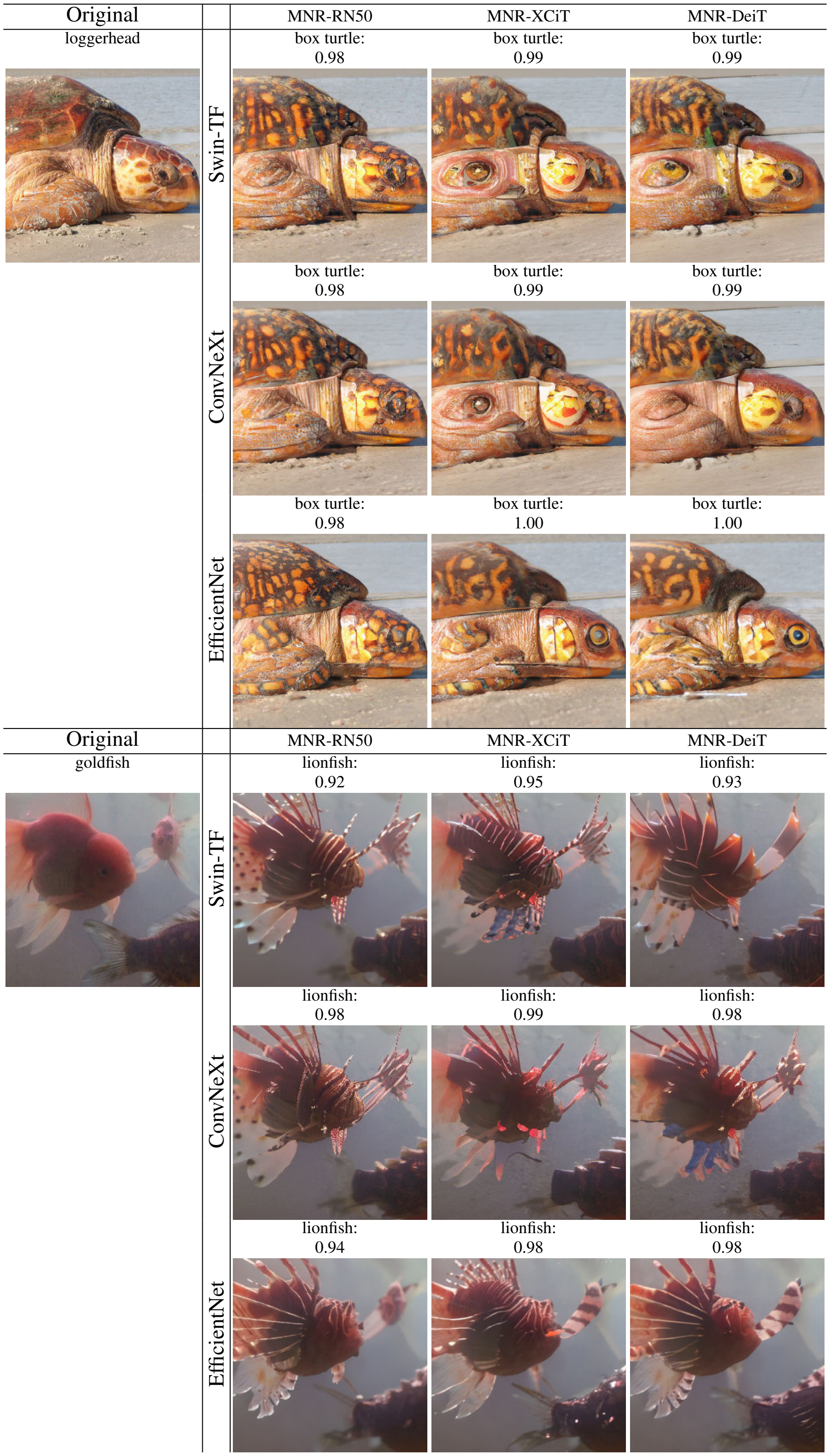


Figure 11: We compare our DVCEs for the non-robust classifiers used in Fig. 5 (rows) where we now vary additionally the adversarially robust model which is used for regularization (same as in Fig. 10 (columns)) for two different starting images. All non-robust classifiers show high confidence in the corresponding target classes and show meaningful target-specific changes. The variation of the DVCEs for the same classifier across different robust models used for the cone projection is small and on the level of the result of different seeds.

B.5 Cone projection

In this section, we are going to compare different parameters for the cone projection from Sec. 3.2, which allows us to explain non-robust classifiers. Remember that we project the gradient of the robust classifier onto a cone centered around the gradient of the target classifier. Thus, larger angles for the cone allow the method to deviate more and more from the target model gradient. In Fig. 12, we show the resulting DVCEs for various angles between 1° and 50°. As shown in Fig. 3, the pure gradient of the target model is unable to visually guide the diffusion process, thus angles smaller than or equal to 15° often do not produce class-specific features in the resulting images. For angles of at least 30°, we can see class-specific features in all images. Although it is possible to use even larger angles, we use 30° throughout the entire paper because this keeps the direction close to the target model's gradient, which is important as we want to explain the target model and not the robust one.

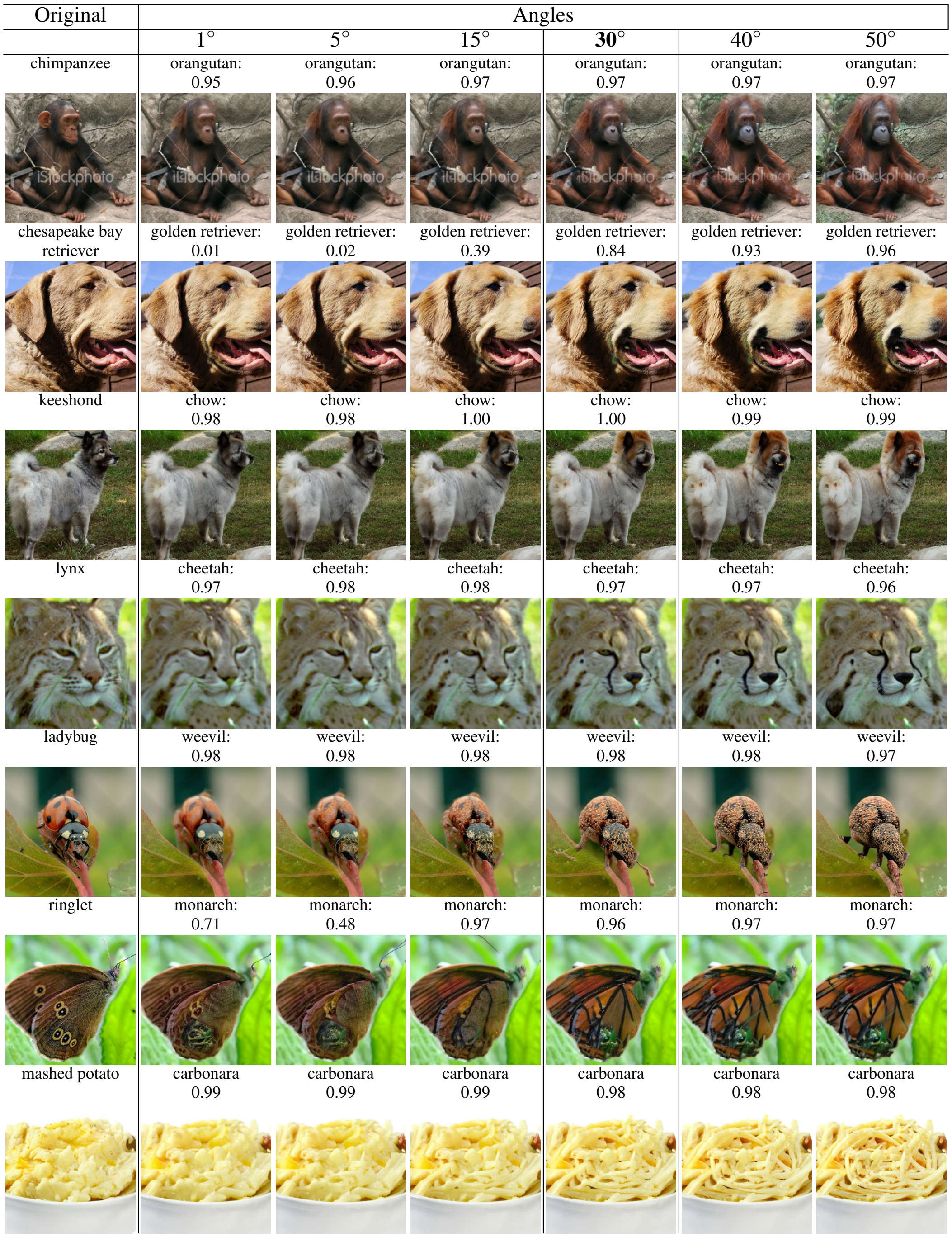


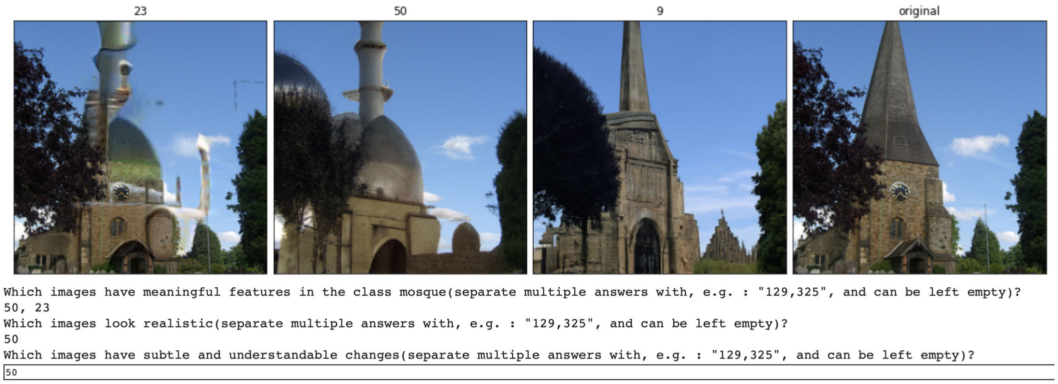
Figure 12: We compare different angles for the cone projection using the ConvNeXt model in combination with the MNR-RN50 robust model. For all images one observes that class-specific features (meaningful change) and high confidence (validity) are reached for more than 30°. As the goal is to stick to the gradient of the non-robust classifier as much as possible we have fixed in the rest of the paper the angle of of the cone to 30°.

C User study

In this section, we discuss a user study that we performed to compare the $l_{1.5}$-SVCEs [7], BDVCEs [2], and our DVCEs.

Whereas in Fig. 4 we provide a qualitative comparison showing the generated VCEs, the user study provides us also with a quantitative comparison. The images for DVCEs were generated the same as for Fig. 4, see Sec. 4.1 for details. However, for BDVCEs, maximizing confidence leads to images far away from the original one (images such as leopard, tiger, timber wolf, and white wolf in Fig. 4), thus we have selected one of the settings of the hyperparameters discussed in Sec. 4.1, where the changes were looking the closest on average while introducing the meaningful features of the target classes. For $l_{1.5}$-SVCEs we have selected the highest radius, $r = 150$, as this still leads to smaller changes (see Tab. 1) in all the metrics, compared to DVCEs and BDVCEs, while maximizes the confidence. To generate the images, we randomly selected images from the ImageNet test set and then generated VCEs for two target classes (different from the original class of the image) which belong same WordNet category as the original class.

In this study, the users, who participated voluntarily and without payment, were shown the original images and the VCEs of the three different methods in random order. The users were researchers in machine learning and related areas but none of them is working on VCEs themself or had seen the generated VCEs before. Following [7], we asked 20 users to rate 44 VCE, if the following three properties are satisfied for a given VCE (no or multiple answers are allowed): i) “Which images have meaningful features in the target class?” (**meaningful**), ii) “Which images look realistic?” **realism**, iii) “Which images show subtle, yet understandable changes?” (**subtle**). The screenshot with instructions and the shown images can be seen in Fig. 13. In the following we report the percentages of the images for DVCEs, $l_{1.5}$-SVCEs, and BDVCEs which were considered to have one of the three different properties: **meaningful** - **62.0%**, 48.4%, 38.7%; **realism** - 34.7%, 24.6%, **52.2%**; **subtle** - 45.0%, **50.6%**, 31.0%. In Fig. 14 we report additionally all original images together with their changes and provide the percentages for each image individually.



```
Which images have meaningful features in the class mosque(separate multiple answers with, e.g. : "129,325", and can be left empty)?
50, 23
Which images look realistic(separate multiple answers with, e.g. : "129,325", and can be left empty)?
50
Which images have subtle and understandable changes(separate multiple answers with, e.g. : "129,325", and can be left empty)?
50
```

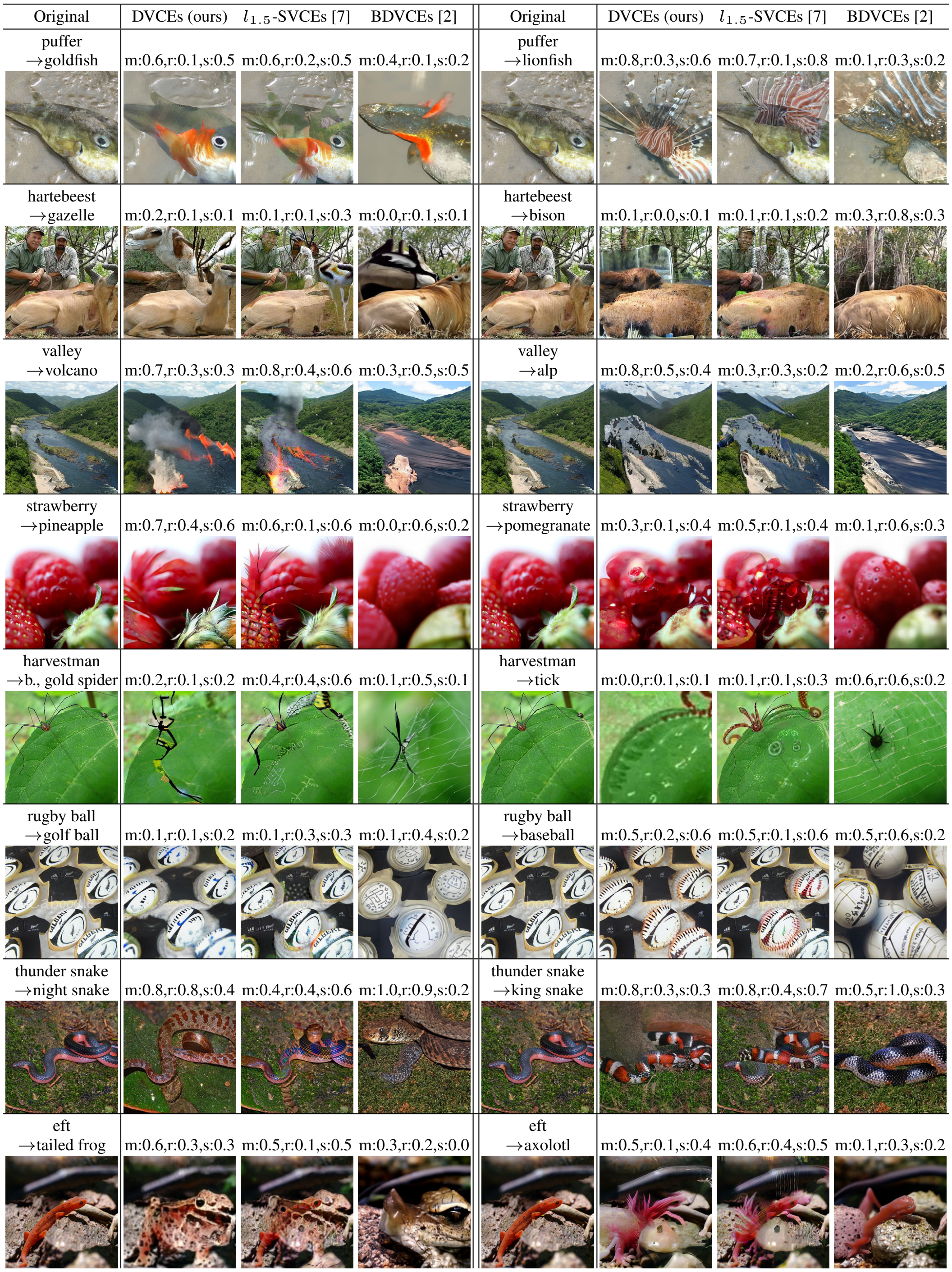
Figure 13: Screenshot of the instructions for the user study. The original image is shown on the right, then the users are shown three VCEs and are asked three questions about the changes concerning the desired target class.

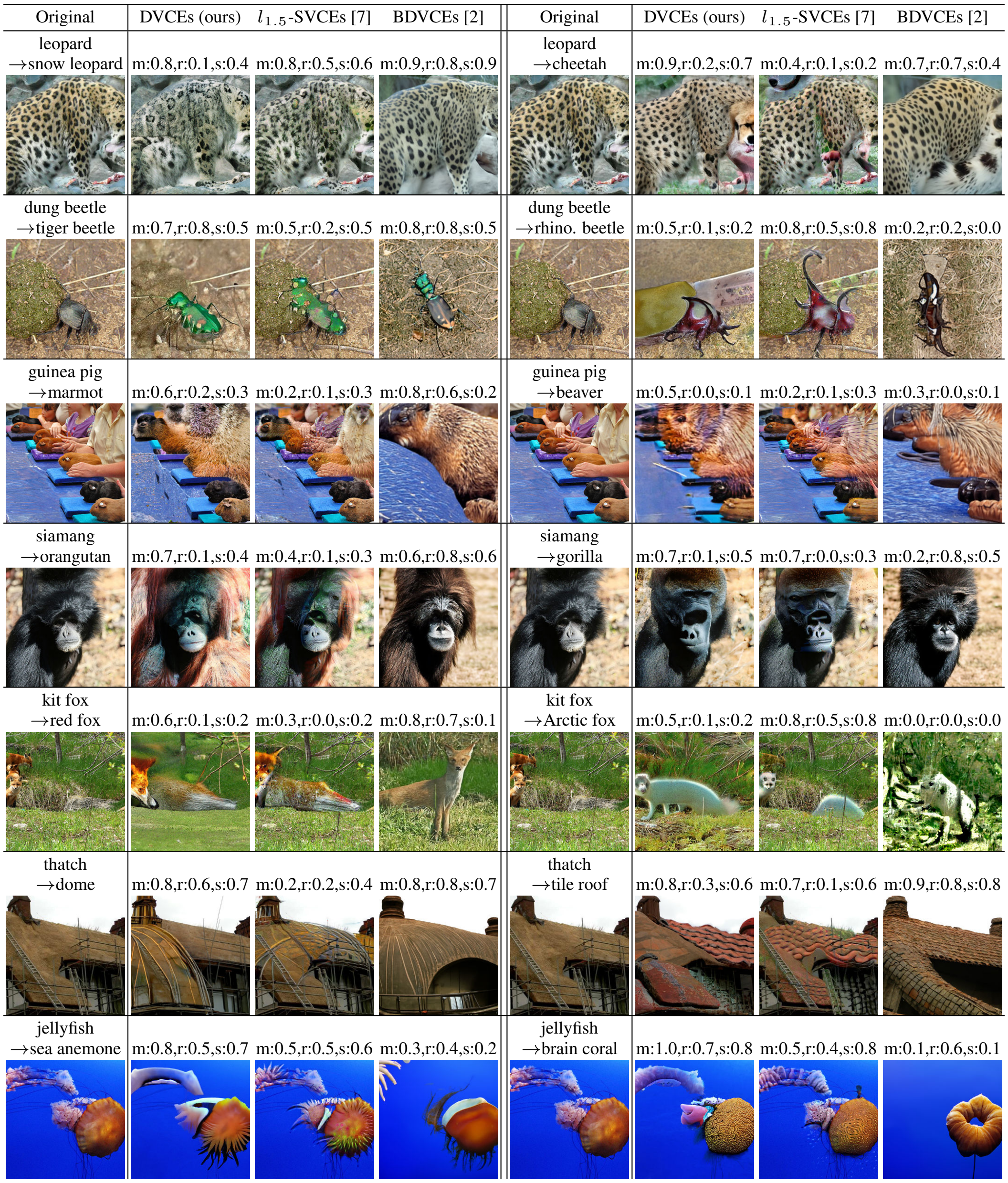
Our DVCEs achieve with 62.0% the highest percentage for meaningful changes, whereas BDVCEs achieve only 38.7%. However, BDVCEs are considered to be realistic in 52.2% of the cases whereas our DVCEs in 34.7%. The reason for this seemingly contradictory result is that the BDVCEs are often not able to realize a meaningful class change, in the sense that the generated images do not show the corresponding class-specific feature e.g. in Fig. 14 for the change siamang → gorilla, the BDVCEs look very good (realism 80%) but are considered to be only 20% meaningful (in fact they did not change the original image) or for jellyfish → brain coral the shown image looks realistic but shows no features of the target class. The reason is that it despite BD has the advantage that we select the image with the highest confidence from six parameter settings whereas for our method we only have one fixed parameter setting, it is sometimes not able to reach high confidence in the target class and does either too little changes (siamang → gorilla) or quite significant changes but meaningless ones (jellyfish → brain coral). Regarding $l_{1.5}$-SVCEs - they have the most subtle changes 50.6% vs.

45.0% for DVCEs but the images show often artefacts (please use zoom into images) which is the reason why they are considered the least realistic ones. In total, the user study shows that DVCE performs best among the three methods if one considers all three categories.

D Resources and hardware used

All the experiments were done on Tesla V100 GPUs, and for the generation of a batch of 6 DVCEs without cone projection and blended VCEs 3 minutes were required. For the batch of 6 DVCEs with cone projections, 5 minutes were required, as one additional model was loaded and the gradient with respect to it was calculated.


Original
DVCEs (ours)
$l_{1.5}$-SVCEs [7]
BDVCEs [2]
Original
DVCEs (ours)
$l_{1.5}$-SVCEs [7]
BDVCEs [2]
puffer →goldfish
m:0.6,r:0.1,s:0.5
m:0.6,r:0.2,s:0.5
m:0.4,r:0.1,s:0.2
puffer →lionfish
m:0.8,r:0.3,s:0.6
m:0.7,r:0.1,s:0.8
m:0.1,r:0.3,s:0.2
hartebeest →gazelle
m:0.2,r:0.1,s:0.1
m:0.1,r:0.1,s:0.3
m:0.0,r:0.1,s:0.1
hartebeest →bison
m:0.1,r:0.0,s:0.1
m:0.1,r:0.1,s:0.2
m:0.3,r:0.8,s:0.3
valley →volcano
m:0.7,r:0.3,s:0.3
m:0.8,r:0.4,s:0.6
m:0.3,r:0.5,s:0.5
valley →alp
m:0.8,r:0.5,s:0.4
m:0.3,r:0.3,s:0.2
m:0.2,r:0.6,s:0.5
strawberry →pineapple
m:0.7,r:0.4,s:0.6
m:0.6,r:0.1,s:0.6
m:0.0,r:0.6,s:0.2
strawberry →pomegranate
m:0.3,r:0.1,s:0.4
m:0.5,r:0.1,s:0.4
m:0.1,r:0.6,s:0.3
harvestman →b., gold spider
m:0.2,r:0.1,s:0.2
m:0.4,r:0.4,s:0.6
m:0.1,r:0.5,s:0.1
harvestman →tick
m:0.0,r:0.1,s:0.1
m:0.1,r:0.1,s:0.3
m:0.6,r:0.6,s:0.2
rugby ball →golf ball
m:0.1,r:0.1,s:0.2
m:0.1,r:0.3,s:0.3
m:0.1,r:0.4,s:0.2
rugby ball →baseball
m:0.5,r:0.2,s:0.6
m:0.5,r:0.1,s:0.6
m:0.5,r:0.6,s:0.2
thunder snake →night snake
m:0.8,r:0.8,s:0.4
m:0.4,r:0.4,s:0.6
m:1.0,r:0.9,s:0.2
thunder snake →king snake
m:0.8,r:0.3,s:0.3
m:0.8,r:0.4,s:0.7
m:0.5,r:1.0,s:0.3
eft →tailed frog
m:0.6,r:0.3,s:0.3
m:0.5,r:0.1,s:0.5
m:0.3,r:0.2,s:0.0
eft →axolotl
m:0.5,r:0.1,s:0.4
m:0.6,r:0.4,s:0.5
m:0.1,r:0.3,s:0.2


Original
DVCEs (ours)
$l_{1.5}$-SVCEs [7]
BDVCEs [2]
Original
DVCEs (ours)
$l_{1.5}$-SVCEs [7]
BDVCEs [2]
leopard →snow leopard
m:0.8,r:0.1,s:0.4
m:0.8,r:0.5,s:0.6
m:0.9,r:0.8,s:0.9
leopard →cheetah
m:0.9,r:0.2,s:0.7
m:0.4,r:0.1,s:0.2
m:0.7,r:0.7,s:0.4
dung beetle →tiger beetle
m:0.7,r:0.8,s:0.5
m:0.5,r:0.2,s:0.5
m:0.8,r:0.8,s:0.5
dung beetle →rhino. beetle
m:0.5,r:0.1,s:0.2
m:0.8,r:0.5,s:0.8
m:0.2,r:0.2,s:0.0
guinea pig →marmot
m:0.6,r:0.2,s:0.3
m:0.2,r:0.1,s:0.3
m:0.8,r:0.6,s:0.2
guinea pig →beaver
m:0.5,r:0.0,s:0.1
m:0.2,r:0.1,s:0.3
m:0.3,r:0.0,s:0.1
siamang →orangutan
m:0.7,r:0.1,s:0.4
m:0.4,r:0.1,s:0.3
m:0.6,r:0.8,s:0.6
siamang →gorilla
m:0.7,r:0.1,s:0.5
m:0.7,r:0.0,s:0.3
m:0.2,r:0.8,s:0.5
kit fox →red fox
m:0.6,r:0.1,s:0.2
m:0.3,r:0.0,s:0.2
m:0.8,r:0.7,s:0.1
kit fox →Arctic fox
m:0.5,r:0.1,s:0.2
m:0.8,r:0.5,s:0.8
m:0.0,r:0.0,s:0.0
thatch →dome
m:0.8,r:0.6,s:0.7
m:0.2,r:0.2,s:0.4
m:0.8,r:0.8,s:0.7
thatch →tile roof
m:0.8,r:0.3,s:0.6
m:0.7,r:0.1,s:0.6
m:0.9,r:0.8,s:0.8
jellyfish →sea anemone
m:0.8,r:0.5,s:0.7
m:0.5,r:0.5,s:0.6
m:0.3,r:0.4,s:0.2
jellyfish →brain coral
m:1.0,r:0.7,s:0.8
m:0.5,r:0.4,s:0.8
m:0.1,r:0.6,s:0.1

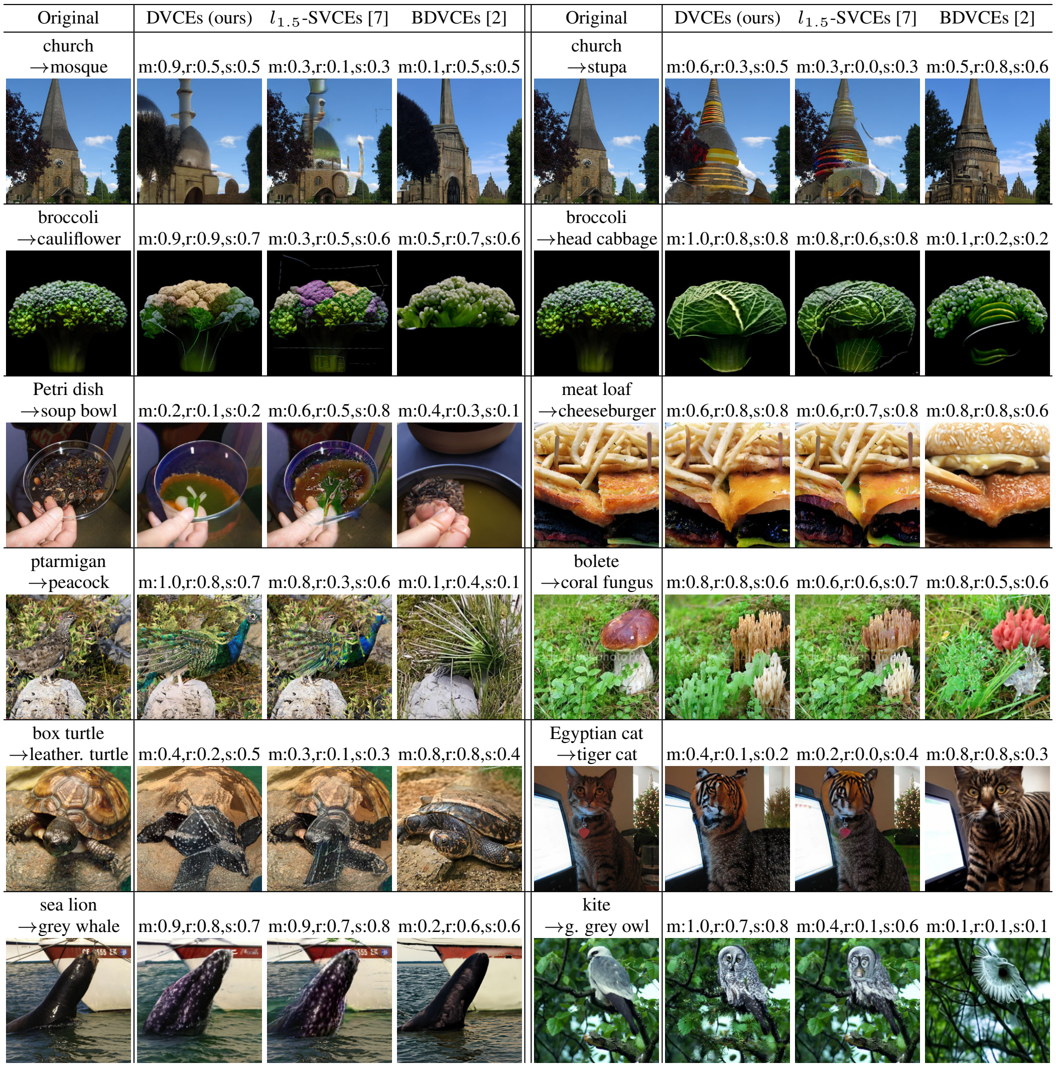


Figure 14: Comparison of different VCE methods for the robust MNR-RN50 (taken from [7]) for all images used for the user study from App. C. We show our DVCEs (left column), the $l_{1.5}$-SVCEs of [7] (middle), and BDVCEs [2] (right) with the respective percentage of time the respective metric (introduced in App. C) - **meaningful** (**m**), **realism** (**r**), **subtle** (**s**) - is chosen by the users. Here we show both the target class and the ground truth label for all the VCEs.

E Quantitative evaluation

In this section, we discuss the quantitative evaluation presented in Tab. 1 to complement the user study and the qualitative evaluation of VCEs. The images for DVCEs were generated the same as in App. C. For BDVCEs, because generating images for the FID evaluation is costly, we have chosen one of the settings of the hyperparameters (that achieves high confidence and such that the resulting images look similar to the original ones on average) discussed in Sec. 4.1. For $l_{1.5}$-SVCEs we have selected the highest radius, $r = 150$ for the same reasons as discussed in App. C.

Realism is assessed using Fréchet Inception Distance (FID) [21], which was also used in [7]. However, in [7], the authors noticed that there is a risk of having low FID scores for the VCEs that don't change the starting images significantly. To overcome this problem, we propose the following simple "crossover" evaluation:

1. divide the classes in the WordNet clusters into two disjoint sets A and B with resp. 5504 and 4576 images so that one gets a roughly balanced split of ImageNet classes.
2. compute VCEs for the test set images of classes A and B with targets (425 different classes) in B resp. A (352 different classes) (crossover). More precisely, as targets we use classes from the same WordNet cluster (see the first step) but which are in the other set. This ensures subtle changes of semantically similar classes and rules out meaningless class changes like "granny smith $\rightarrow$ container ship".
3. determine two subsets of the ImageNet trainset for the classes in A resp. B, such that in each subset the distribution of the labels corresponds to the one of A resp. B. Then we compute FID scores once between the training set corresponding to A and the VCEs generated with a target in A and for the training set corresponding to B and the VCEs with targets in B and report the average of the two FID scores.

This way, we make sure that the original images for VCEs don't come from the same distribution as the images from the subset of the training set of ImageNet. As a sanity check, we evaluated the FID scores of the training set of classes in A to the original images of the classes in B and vice versa which yields an average of 41.5. As all methods achieve a smaller FID score (lower is better), they are all able to produce features of the target classes. However, DVCE stands out with an FID score of 17.6 compared to 27.9 (BDVCEs) and 25.6 ($l_{1.5}$-SVCEs).

Validity is evaluated as the mean confidence of the model achieved on the VCEs for the selected target classes. The mean confidence of DVCEs is almost the same as that of $l_{1.5}$-SVCEs which maximize the confidence over the $l_{1.5}$-ball and thus are expected to have high mean confidence. So there is little difference in validity, whereas BDVCEs have significantly lower confidence and thus have worse performance regarding validity. This is again due to the problem that the parameters leading to high confidence of the classifier but still leading to an image related to the original class are extremely difficult to find (if they exist at all) as they are image-specific.

Closeness is assessed with the set of metrics - $l_1, l_{1.5}, l_2$, as well as the perceptual LPIPS metric [58] for the AlexNet model. As $l_{1.5}$-SVCEs directly manipulate the image without the modification by a generative model, it is not surprising that in terms of l_p-distances they outperform DVCEs and BDVCEs.

Summary From the qualitative and the quantitative results we deduce that DVCEs and $l_{1.5}$-SVCEs are equally valid but DVCEs outperform in terms of image quality (realism) all other approaches significantly while remaining close to the original images.

F Spurious features

In this section, we want to show how DVCEs can be used as a “debugging” tool for ImageNet classifiers (resp. the training set). First, we show that DVCE finds the same spurious features as discovered in [7] and we show also more spurious features found by our method.

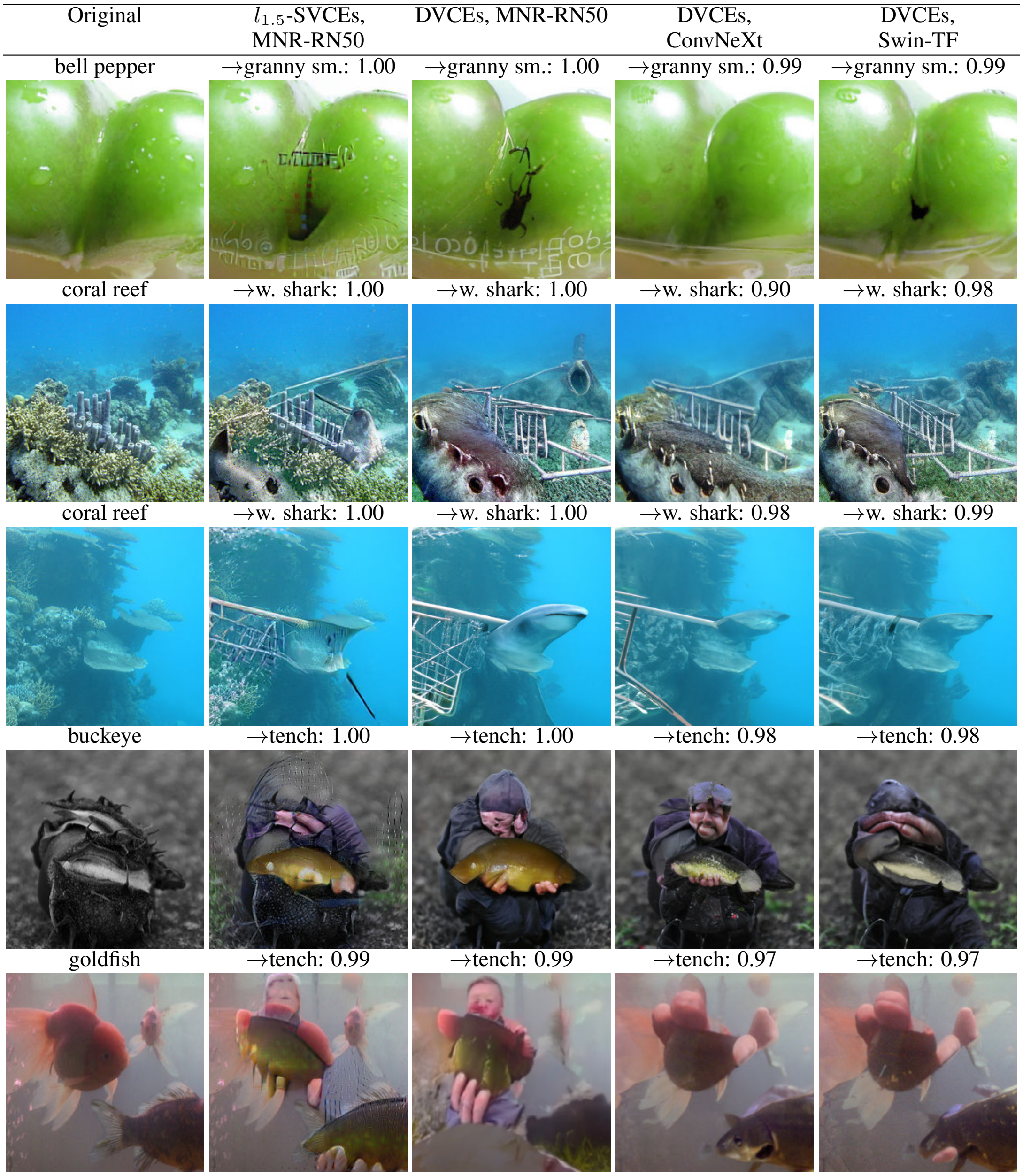


Figure 15: **Reproducing spurious features of [7] for the target classes “granny smith”, “white shark”, and “tench”.** $l_{1.5}$-SVCEs from [7] for $\epsilon_{1.5} = 150$ and DVCEs for MNR-RN50. DVCEs for ConvNeXt and Swin-TF have spurious features only in some cases, see last row, or second row for Swin-TF.

Reproducing spurious features found in [7]. In Fig. 15, we first reproduce the spurious features found by [7] with $l_{1.5}$-SVCE for $\epsilon_{1.5} = 150$ for the MNR-RN50 model. We generate the DVCEs for MNR-RN50, the non-robust Swin-TF [28], and ConvNeXt [29]. We find similar spurious features for the target classes “white shark” and “tench”. As discussed in [7], these spurious features are a consequence of the selection of the images in the training set. For “white shark” it contains a lot of images containing cages to protect the diver and thus the models pick up this “co-occurrence”. Even more extreme for “tench” where a large fraction of images show the angler holding the tench in their hands leading to the fact that counterfactuals for tench contain human faces and hands. The spurious text feature for the class “granny smith” shown by the MNR-RN50 model is not visible for Swin-TF, and ConvNeXt. One key difference is that they have been trained on ImageNet-22k and then fine-tuned to ImageNet which might have reduced this spurious feature but it requires a more detailed analysis to be sure about this.

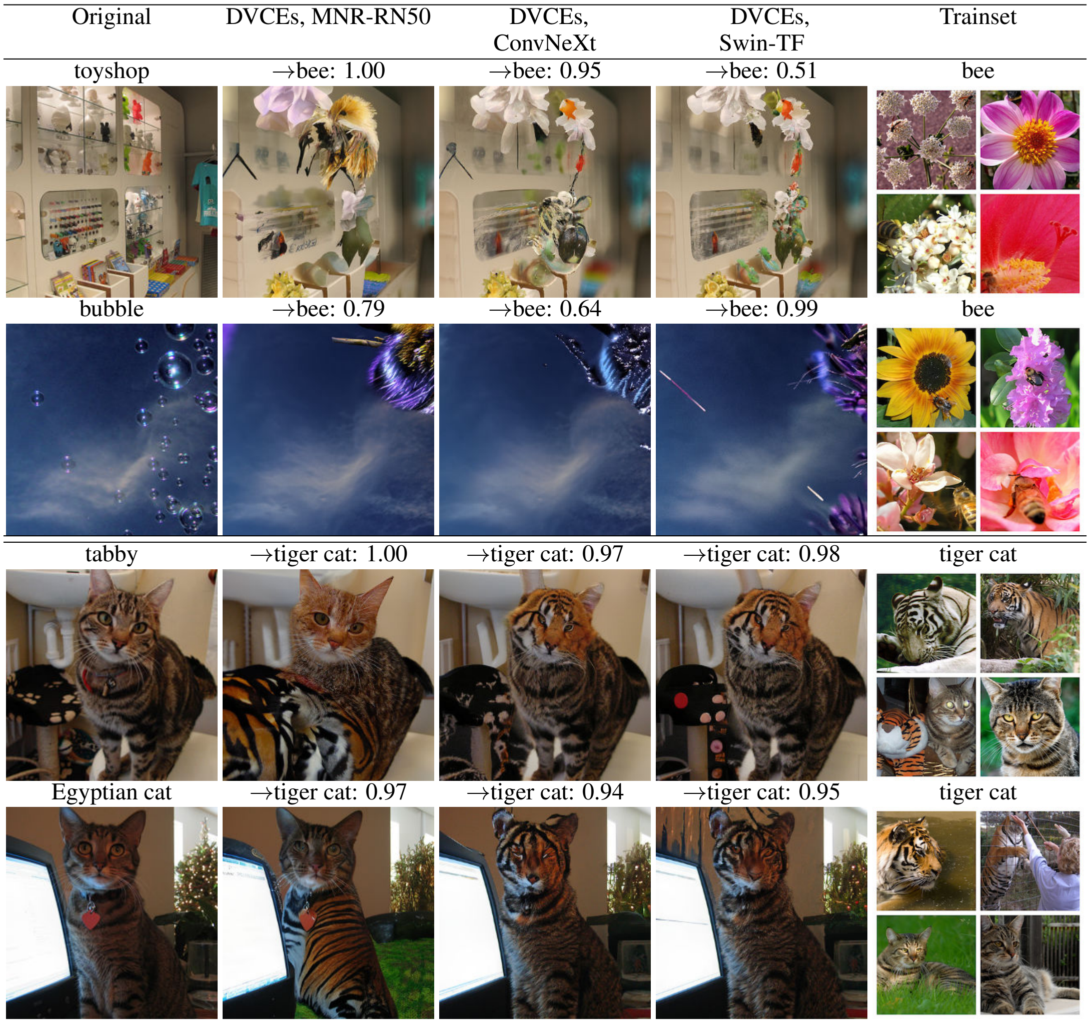


Figure 16: **New spurious features of [7] for the target classes “bee”, and “tiger cat”.** DVCEs for for ConvNeXt and Swin-TF have spurious features for the respective target classes: flowers (and not the bee itself) for “bee”, and tiger face for the “tiger cat”. Here, we show both the target class and the ground truth label for all the DVCEs.

New spurious features for non-robust models. We use our DVCEs here to find novel spurious features picked up by all classifiers (but to a mixed extent) MNR-RN50, ConvNeXt and Swin-TF: features of flowers (and not the bee itself) increase the confidence in the target class “bee” and features of the tiger face increase the confidence in the target class “tiger cat” as can be seen in Fig. 16. Here, we show both the target class and the ground truth label for all the VCEs. We also show in the rightmost column the samples from the trainset of ImageNet from the respective target classes. In both cases, they show why the classifier has picked up these spurious features. Images of bees often show flowers, most of the time much larger than the bee itself. Whereas the DVCE for “tiger cat” shows that the training set of this class is completely broken as it contains images of “tigers” (note that “tiger” is a separate class in ImageNet).

G Failure cases

From what we have seen, generally, there are two failure cases of our method: i) sometimes DVCEs can be blurry, which seems to be an artefact of the diffusion model as it happens for BDVCEs and is visible also in the original diffusion paper (see top left image in Fig. 15 of [14]), ii) DVCEs can fail when the change is difficult to realize (i.e. the original image is not similar to the target class).

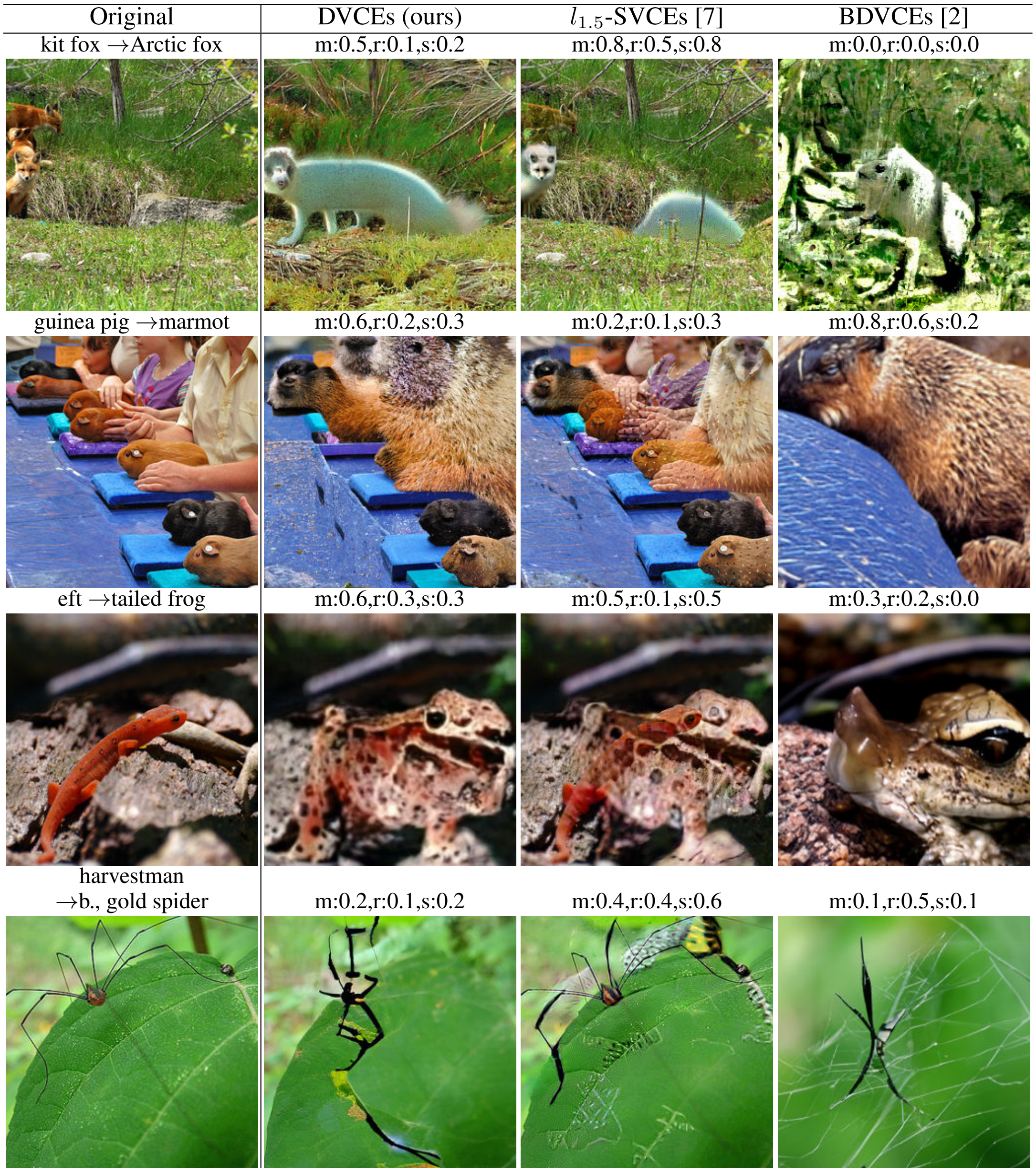


Figure 17: **Some failure cases of DVCEs for the robust MNR-RN50 from our user study in App. C**. The failure cases that we have seen are i) DVCEs (and BDVCEs) can be slightly blurry, or ii) change is difficult to realize. As in Fig. 14, we show our DVCEs (left column), the $l_{1.5}$-SVCEs of [7] (middle), and BDVCEs [2] (right) with the respective percentage of time the respective metric (introduced in App. C) - **meaningful** (**m**), **realism** (**r**), **subtle** (**s**) - is chosen by the users. Here, we show both the target class and the ground truth label for all the VCEs.