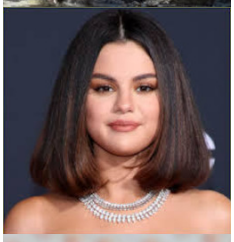
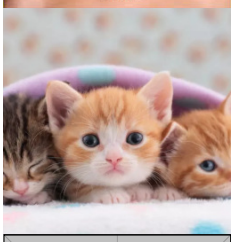
A PROMPTING THE IMAGE-TO-TEXT MODEL

To ensure that captions are descriptive and composed of short sentences, we prompt our image-to-text model with the following for all experiments:

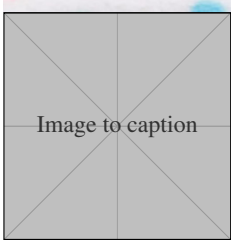
is a realistic picture of two penguins. They are holding hands. They are standing in front of the sea. The picture is mostly grey. The penguins are facing away from the camera. They take up most of the image.



is a portrait photograph of a famous person. She is wearing two necklaces. She has dark hair and is wearing makeup. She is facing the camera and the background is black.



is a cute photograph of three kittens. They are under a blanket. The background is blurred but it seems white and orange. The blanket is purple. The two cats on the right are orange and the one on the left is grey. The orange cats have open eyes and the grey cat has closed eyes. They are all super cute.



is a realistic photograph of a `[label name]`. `[...]`

These images are not from IMAGENET but downloaded from the web. Choosing these captions is only done once and then fixed for all experiments in the paper. We also set the decoding strategy to be *greedy* (as we did not observe significant improvements from using beam search). We highlight that any expert knowledge, if needed, is only required to annotate these three images. Compared to annotating a full test set, the cost is negligible.

B ADDITIONAL RESULTS

B.1 OPEN-ENDED FAILURE SEARCH

Large scale experiments. Similarly to Fig. 2 and Table 1, Fig. 6 and Table 3 show failure cases automatically found by our pipeline for a RESNET-50 available on TF-HUB in a fully open-ended manner (i.e., without leveraging an external dataset). The labels considered are a subset of the 200 labels present in IMAGENET-A. We let the reader interpret these failure cases themselves. The failures are diverse and are due to different factors, such as: *(i)* misleading color patterns (e.g., `sea amemone` → `daisy`), *(ii)* spurious context (e.g., `jeep` → `snowplow`), *(iii)* missing knowledge (e.g., `custard apple` → `mask`), or *(iv)* hallucinations (e.g., `feather boa` → `maltese dog`).

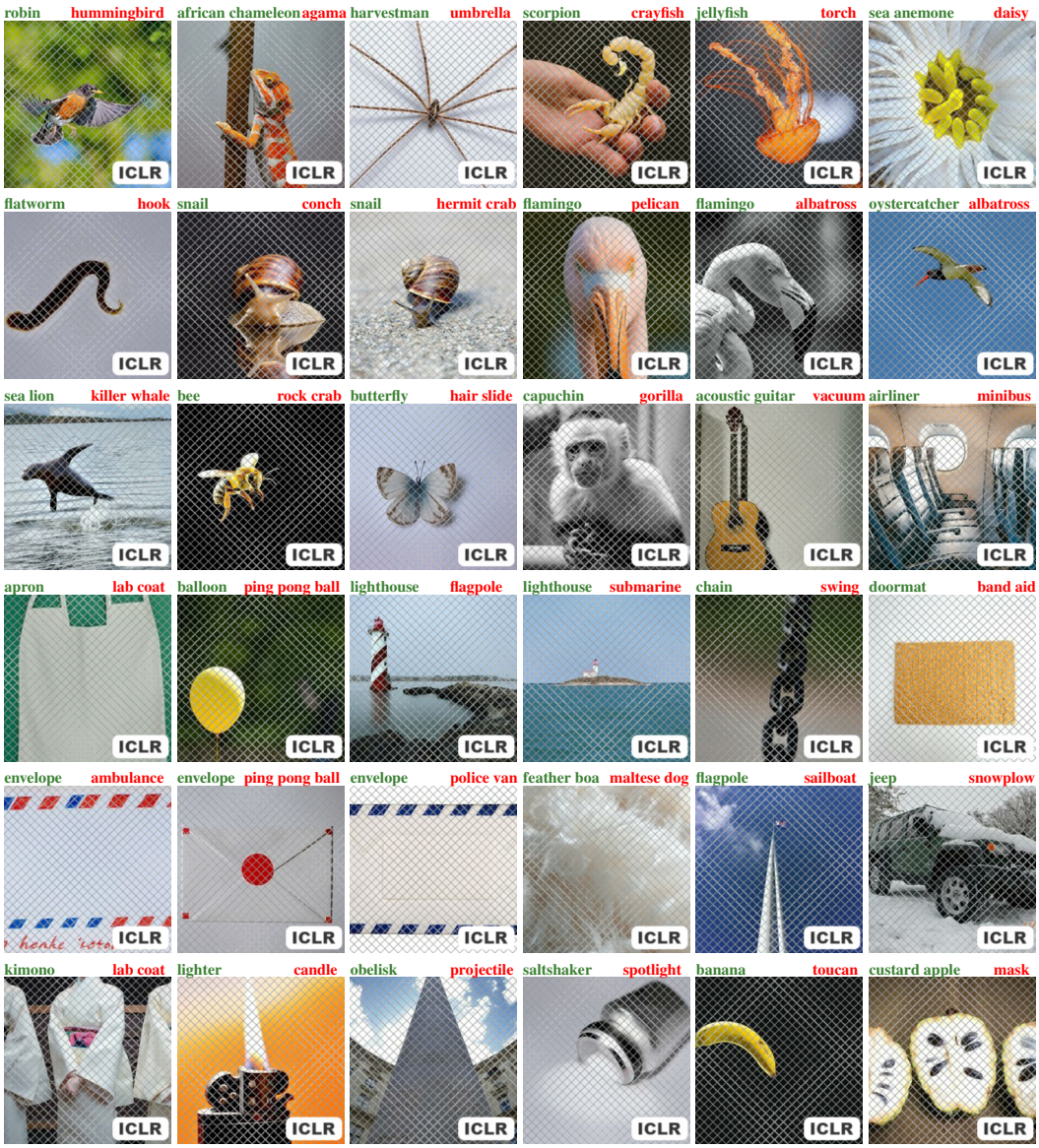


Figure 6: Illustration of failure cases listed in Table 3. The correct label is to the left in **green**. The incorrect prediction is to the right in **red**. The model used is a RESNET-50 found on TF-HUB.

True label	Target label	Caption	Failure rate (target)	
robin	hummingbird	a realistic photograph of a robin (oscine).	0.0032%	1×
		— " — It is flying.	7.35%	2264.3×
african chameleon	agama	a realistic photograph of an african chameleon (lizard).	0.15%	1×
		— " — He is holding a stick. The chameleon is orange and white.	1.01%	6.7×
harvestman	umbrella	a realistic photograph of a harvestman (arthropod).	0.45%	1×
		— " — It is shot from above. The harvestman is on a white background.	1.32%	2.9×
scorpion	crayfish	a realistic photograph of a scorpion (arthropod).	0.0042%	1×
		— " — It is on a person's hand.	0.13%	29.5×
jellyfish	torch	a realistic photograph of a jellyfish (invertebrate).	0.14%	1×
		— " — The background is black. The jellyfish is orange.	0.45%	3.2×
sea anemone	daisy	a realistic photograph of a sea anemone (coelenterate).	0.32%	1×
		— " — It is yellow and white. The background is blurred.	1.33%	4.2×
flatworm	hook	a realistic photograph of a flatworm (invertebrate).	0.61%	1×
		— " — It is on a white background.	1.58%	2.6×
snail	conch	a realistic photograph of a snail (mollusk).	0.039%	1×
		— " — It is on a black background. The snail is reflected on the floor.	0.88%	22.5×
snail	hermit crab	a realistic photograph of a snail (mollusk).	0.0082%	1×
		— " — It is on a grey road.	0.10%	12.2×
flamingo	pelican	a realistic photograph of a flamingo (aquatic bird).	0.023%	1×
		— " — It is a close up of the head. The flamingo is facing the camera.	0.87%	37.6×
flamingo	albatross	a realistic photograph of a flamingo (aquatic bird).	0.081%	1×
		— " — It is black and white. The flamingo is looking to the right.	1.30%	16.1×
oystercatcher	albatross	a realistic photograph of an oystercatcher (wading bird).	0.0025%	1×
		— " — It is flying.	0.52%	208.2×
sea lion	killer whale	a realistic photograph of a sea lion (seal).	0.14%	1×
		— " — It is jumping out of the water.	4.31%	31.3×
bee	rock crab	a realistic photograph of a bee (insect).	0.086%	1×
		— " — It is flying. The background is black.	0.18%	2.1×
cabbage butterfly	hair slide	a realistic photograph of a cabbage butterfly (butterfly).	0.099%	1×
		— " — It is on a white background. It is in the middle of the image.	1.98%	20.0×
capuchin	gorilla	a realistic photograph of a capuchin (monkey).	0.011%	1×
		— " — It is a black and white photograph.	0.84%	73.9×
acoustic guitar	vacuum	a realistic photograph of an acoustic guitar (stringed instrument).	0.20%	1×
		— " — It is leaning against a wall.	1.02%	5.0×
airliner	minibus	a realistic photograph of an airliner (heavier-than-air craft).	0.16%	1×
		— " — There are seats in the foreground.	3.05%	19.2×
apron	lab coat	a realistic photograph of an apron (clothing).	0.44%	1×
		— " — It is white.	3.57%	8.1×
balloon	ping-pong ball	a realistic photograph of a balloon (aircraft).	0.53%	1×
		— " — It is yellow. The background is blurred.	17.86%	33.8×
lighthouse	flagpole	a realistic photograph of a beacon (structure).	0.095%	1×
		— " — The lighthouse has red and white stripes.	2.12%	22.4×
lighthouse	submarine	a realistic photograph of a beacon (structure).	0.041%	1×
		— " — It is on a small island at the horizon.	12.5%	308.0×
chain	swing	a realistic photograph of a chain (attachment).	1.22%	1×
		— " — The chain is vertical. The chain is in focus.	12.5%	10.2×
doormat	band aid	a realistic photograph of a doormat (floor cover).	0.94%	1×
		— " — The doormat is rectangular and is on a white background.	5.68%	6.1×
envelope	ambulance	a realistic photograph of an envelope (instrumentality).	0.210%	1×
		— " — It has white and has red and blue stripes at the top and bottom.	17.86%	60.0×
envelope	ping-pong ball	a realistic photograph of an envelope (instrumentality).	1.04%	1×
		— " — It is white and has a red dot on it.	75.%	72.0×
envelope	police van	a realistic photograph of an envelope (instrumentality).	0.510%	1×
		— " — It has white and has white and blue diagonal stripes at the top and bottom.	6.94%	11.7×
feather boa	maltese dog	a realistic photograph of a feather boa (garment).	4.59%	1×
		— " — It is white and fluffy.	41.67%	9.1×
flagpole	sailboat	a realistic photograph of a flagpole (stick).	0.19%	1×
		— " — It is white and the sky is blue.	2.19%	11.5×
jeep	snowplow	a realistic photograph of a jeep (motor vehicle).	0.30%	1×
		— " — It is parked in the snow.	17.86%	59.3×
kimono	lab coat	a realistic photograph of a kimono (garment).	0.69%	1×
		— " — It is white.	2.84%	4.1×
lighter	candle	a realistic photograph of a lighter (instrumentality).	5.37%	1×
		— " — It has a flame coming out of it.	41.67%	7.8×
obelisk	projectile	a realistic photograph of an obelisk (structure).	0.14%	1×
		— " — It is pointing up. The sky is blue.	1.09%	7.7×
saltshaker	spotlight	a realistic photograph of a saltshaker (container).	1.02%	1×
		— " — It has a silver lid. The salt shaker is on a white background. The salt is spilling out of the jar.	13.89%	13.7×
banana	toucan	a realistic photograph of a banana (produce).	0.0058%	1×
		— " — It is yellow and is floating in the air. The background is black.	0.047%	8.2×
custard apple	mask	a realistic photograph of a custard apple (produce).	0.32%	1×
		— " — The fruit is cut in half.	4.46%	14.0×

Table 3: Absolute failure rates of a RESNET-50 for 36 additional true and target label pairs. We show the target failure rate (i.e., the model prediction is the target label). Captions are automatically discovered using the method detailed in Sec. 3. Note that to the contrary of Table 1, we consider an image to be misclassified when the top-1 prediction is wrong (and not from the same WORDNET parent) rather than when the true label is not part of the top-3 predictions.

Generalization to other vision architectures. We investigate whether our approach extends to other, more challenging backbones, such as a VIT-B/32 and VIT-B/8 which obtain much better performance than a RESNET50 on IMAGENET. We use the `fly` and `bee` failure case as our case study and run the same experiment as the one presented in Table 1. Results are reported in Table 4. First, we observe that both VIT models exhibit the same bias than the one found for the RESNET-50 model (that a fly on a flower is more often confused as a bee). Second, while the failure rates increase significantly compared to the baseline, the bias seems to be less pronounced (with only a 114× and 13.5× increase in failure rates compared to 497× for the RESNET) which highlights the qualities of both models.

Architecture	True label	Target label	Caption	Failure rate (target)	
VIT-B/32	`fly`	`bee`	a realistic photograph of a fly (insect). — " — it is on a flower.	0.0002% 0.02278%	1× 113.9×
VIT-B/8	`fly`	`bee`	a realistic photograph of a fly (insect). — " — it is on a flower.	0.0002% 0.0027%	1× 13.5×

Table 4: Absolute failure rates of a VIT-B/8, VIT-B/32 on the `fly`/`bee` failure case. We show the target failure rate (i.e., the model prediction is the target label). Captions are automatically discovered using the method detailed in Sec. 3.

Significance of results. We further investigate the significance of the results on two of the open-ended failure cases in the main paper (the ones exhibiting larger failure rates). Here, we evaluate our RESNET-50 and generate samples until we either find 10 images that cause the classifier to mispredict the class towards the target class (using top-1 accuracy instead of the typical top-3 to allow us to run many experiments efficiently) or find no misclassification towards the target class within 20K samples. We report the failure rate for the original and discovered captions and compute p-values using the Mann-Whitney U test (Mann & Whitney, 1947) at a significance level of 0.005 to determine if the differences in failure rates are statistically significant. Results are in Table 5. As the p-values (0.00015, $6.34 \cdot 10^{-5}$) are lower than 0.005, we find a *significant* result that images of the discovered caption (e.g., "... crayfish (crustacean). it is in a net.") are more often misclassified for the target class (e.g., `chainlink fence`) than the original caption (e.g., "... crayfish (crustacean).").

True label	Target label	Caption	Failure rate (target)	p-value
`fly`	`bee`	a realistic photograph of a fly (insect). — " — it is on a flower.	$0.00\% \pm 0.00$ $0.58\% \pm 0.15$	0.00015
`crayfish`	`chainlink fence`	a realistic photograph of a crayfish (crustacean). — " — it is in a net.	$0.00\% \pm 0.00$ $2.09\% \pm 0.56$	$6.34 \cdot 10^{-5}$

Table 5: Significance of failure rates. We report the mean and standard deviation of the target failure rate for the original and discovered captions over ten runs. We then compute the p-value to determine if the difference in failure rates between the original and discovered caption is statistically significant. We do this for two failure cases and find that our results are statistically significant.

B.2 ADDITIONAL COMPARISONS ON IN-G-RN, IN-G-VIT

Models considered. We collate a large set of models trained on IMAGENET with differing size, pretraining, augmentation, and architectures in addition to the two RESNETs and VITs we trained:

- VIT-B*, VIT-L*, VIT-S* (Dosovitskiy et al., 2020): VITs pretrained on IMAGENET21K.
- VIT-R* (Steiner et al., 2022): a hybrid VIT and RESNET model pretrained on IMAGENET21K.
- BIT-* (Kolesnikov et al., 2020): BIT models pretrained either on IMAGENET21K (BIT-M *) or not pretrained (BIT-S *).
- INCEPTION_RESNET V2 (Szegedy et al., 2017): a hybrid INCEPTION, RESNET model with no pretraining.
- INCEPTION* (Szegedy et al., 2015): INCEPTION models with no pretraining.
- RESNET* (He et al., 2016): RESNET models with no pretraining.

Transferability of errors on IN-G-RN and IN-G-VIT. We evaluate how often failures in IN-G-RN and IN-G-VIT transfer to these models in Fig. 7. We can see that failures from both IN-G-RN and IN-G-VIT transfer across model architectures. However, the VITs and BITs which are pretrained on IMAGENET21K and achieve lower error on IMAGENET are fooled the least often. Within a model class, larger versions of the model seem more robust. For example, VIT-B/16 is more robust than VIT-B/32 and similarly the larger BITs (those of size 101x1) are more robust than their smaller counterparts (those of size 50x1). Thus, stronger pretraining and larger models seem to lead to improved (but not complete) robustness against these generated datasets.

Error consistency on IN-G-RN and IN-G-VIT. Finally, we measure the error consistency of the models in Fig. 8. We combine IN-G-RN and IN-G-VIT into one dataset and evaluate how often models make similar errors while accounting for the accuracy of each model (see Eq. 3 in Geirhos et al., 2021). A value of 100% indicates that the errors two models make are perfectly correlated and -100% that they are perfectly anti-correlated. It is striking that errors are most consistent within a model class: RESNETS make similar errors to other RESNETS trained in a similar manner and similarly BITs make similar errors to other BITs, especially BITs trained in the same manner.

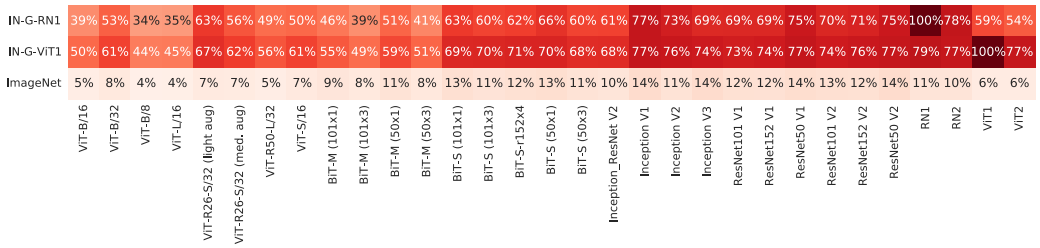


Figure 7: Failure rates (top-3) for different models on two generated datasets and IMAGENET. We report the failure rates of different models trained on IMAGENET on both IN-G-RN and IN-G-VIT as well as IMAGENET.

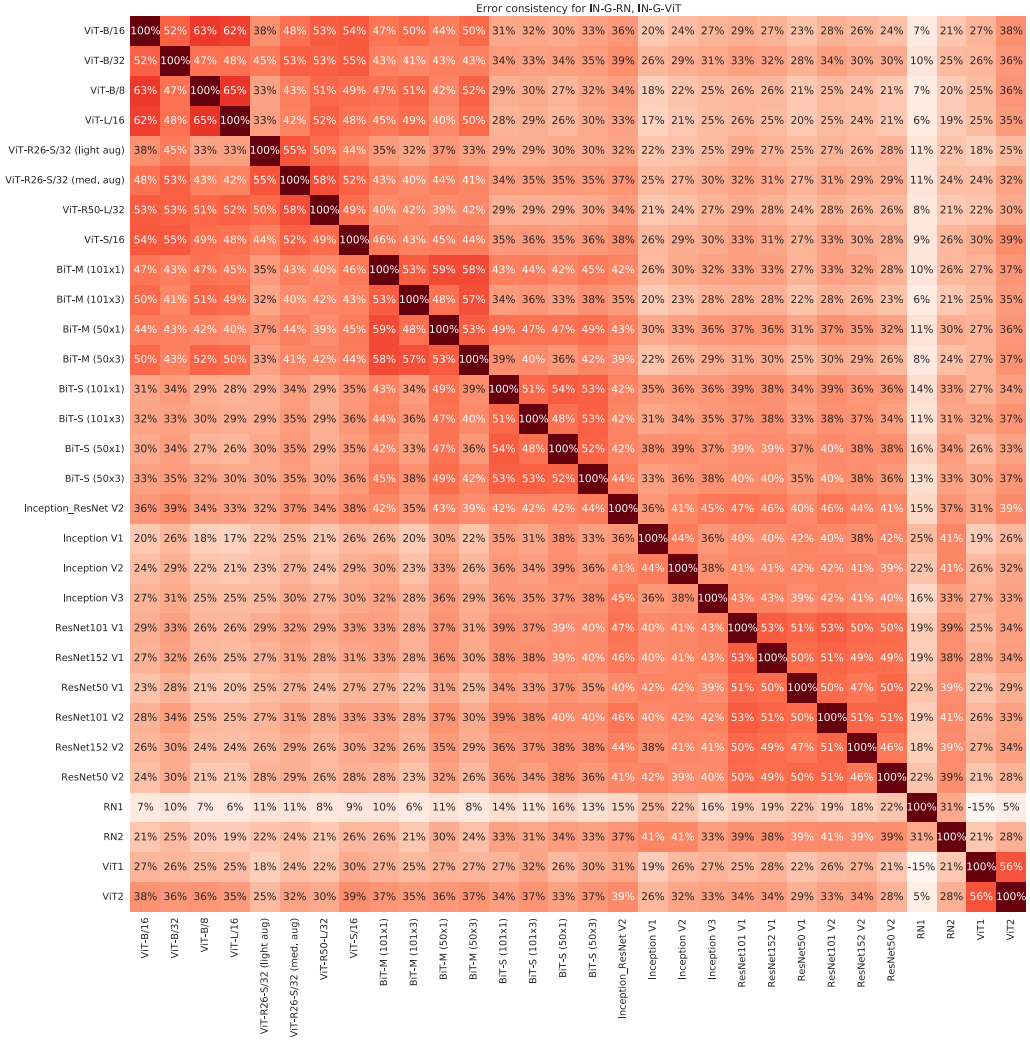


Figure 8: Error consistency for all models on the combined IN-G-RN and IN-G-VIT dataset.

C INTERPRETING FAILURE CASES

In this section, we aim at further characterizing failure cases by investigating why the models considered in our work yield wrong predictions on IN-G-RN and IN-G-VIT instances. For that, we compared a few failure cases with their respective nearest neighbors within the training set of IMAGENET in order to find patterns that shed light on the reasons behind wrong predictions.

We find the ten nearest neighbors of an IN-G-VIT instance in the embedding space induced by the second-to-last layer of a VIT trained on IMAGENET, using cosine similarity as the distance measure. This particular model achieves 82.7% top-1 accuracy on the IMAGENET validation set and has a failure rate of 100% on IN-G-VIT.

Fig. 9-11 show IN-G-VIT failure cases, along with their respective ten nearest neighbors within the *full* IMAGENET training set and the ten nearest neighbors with the same label. Results suggest that failure cases found by our approach induce errors by generating images that have elements in their background which are more often found in other classes within IMAGENET. We further observe that all failure cases are closer to examples containing objects semantically related with cues present in images that are not commonly found in the training set of IMAGENET for these classes. In Fig. 9(a), for example, we show a failure case labeled as `mushroom` for which the VIT predicts the label `snail`. All nearest neighbors shown in Fig. 9(b) are labeled as `snail` and contain elements such as human skin and grass in the background, which do not appear in the nearest neighbors from the label `mushroom`, as shown in Fig. 9(c). The VIT appears to be capturing spurious features in its representations (presence of human skin and grassy background) and relying on them to make predictions, which lead it to yield the wrong label for the IN-G-VIT instance presented in Fig. 9(a). In Fig. 10 and 11, we observe a similar pattern, where the VIT focuses on spurious cues such as the presence of a net in the background in Fig. 10(a) and snow in Fig. 11(a). Exploiting such correlations made the model mistake the particular instances of `cabbage butterfly` and `flagpole` as `barn spider` and `ski`, respectively.

(a) IN-G-VIT failure case.

(b) 10 nearest neighbors of (a) in the IMAGENET train set.

(c) 10 nearest neighbors of (a) in the IMAGENET train set and in its respective label (`mushroom`).

Figure 9: Interpreting failure cases by inspecting nearest neighbors in the train set of IMAGENET. We analyze the failure case in IN-G-VIT shown in panel (a). The example is labeled as `mushroom` and classified as `snail` by a VIT trained on IMAGENET. In panel (b), we show the 10 nearest neighbors of (a) in the train set of IMAGENET. All 10 neighbors are from the class `snail` and have similar features to the failure case, such as the background (e.g., the human hand), while the 10 nearest neighbors with the label `mushroom` showed in panel (c) do not have those features. This suggests that the VIT correlates such features with the label `snail`, and these spurious correlations likely induced it to misclassify the image in (a).

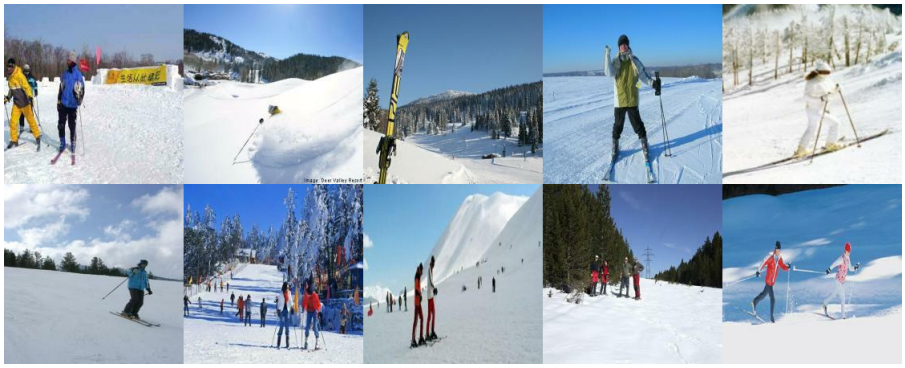
(a) IN-G-VIT failure case.

(b) 10 nearest neighbors of (a) in the IMAGENET train set.

(c) 10 nearest neighbors of (a) in the IMAGENET train set and in its respective class (`cabbage butterfly`).

Figure 10: Interpreting failure cases by inspecting their nearest neighbors in the train set of IMAGENET. We analyze the failure case in IN-G-VIT shown in panel (a). The example is labeled as `cabbage butterfly` and classified as `barn spider` by a VIT trained on IMAGENET. In panel (b) we show the 10 nearest neighbors of (a) in the train set of IMAGENET. All 10 neighbors are from the classes `barn spider` or `black and gold garden spider` and have similarities to the failure case such as the white net in the background, while the 10 nearest neighbors in the class `cabbage butterfly` shown in panel (c) do not present those common features. This suggests that the VIT correlates such features with instances from the labels `barn spider` and `black and gold garden spider`, and exploiting these spurious correlations likely induced the model to misclassify the image in (a).

(a) IN-G-VIT failure case.



(b) 10 nearest neighbors of (a) in the IMAGENET train set..

(c) 10 nearest neighbors of (a) in the IMAGENET train set and in its respective class (`flagpole`)

Figure 11: Interpreting failure cases by inspecting their nearest neighbors in the train set of IMAGENET. We analyze the failure case in IN-G-VIT shown in panel (a). The example is labeled as `flagpole` and classified as `ski` by a VIT trained on IMAGENET. In panel (b) we show the 10 nearest neighbors of (a) in the train set of IMAGENET. All 10 neighbors have the label `ski` and have similarities to the failure case, such as simultaneously having snow on the ground and blue sky in the background, while the 10 nearest neighbors with the label `flagpole` shown in (c) do not present those common features. This suggests that the VIT correlates such features with instances from the label `ski`, and exploiting such spurious correlations likely induced it to misclassify the image in (a).

D MALICIOUS USAGE AND MITIGATION STRATEGIES

This work demonstrates how to find failure cases in vision classifiers with the help of large-scale generative models. Much like adversarial examples (Biggio et al., 2013; Szegedy et al., 2013), malicious actors could leverage the proposed approach to build adversarial images that bypass automated online filtering mechanism. In this section, we discuss how to make classifiers robust to these failure cases.

First, classifiers can be trained with discovered failure cases to make them more robust to generated inputs. As a demonstration, we split the IN-G-VIT dataset into a train and test set (80% train, 20% test). We train the original VIT model in the exact same manner as before, except that batches are now made of 95% IMAGENET data and 5% IN-G-VIT data. We report results with and without additional synthetic data in Table 6. Training with additional generated data leads to a minimal loss of performance on IMAGENET while achieving nearly 90% top-1 accuracy on the IN-G-VIT test set. This demonstrates that adding the generated failure cases into the training set is an effective mitigation strategy.

Second, we note that our approach is computationally expensive. It requires hundreds to thousands of calls to the generative model and vision classifier to find a *single* failure case. Hiding the underlying classifier behind a rate-limited API can act as a first line of defense.

Training Set	top-1 on IMAGENET ↑	top-1 on IN-G-VIT ↑
IMAGENET (train)	**82.57 ± 0.09**	5.60 ± 2.80
IMAGENET (train) + IN-G-VIT (train)	82.11 ± 0.05	**88.11 ± 0.44**

Table 6: top-1 accuracy on IMAGENET and IN-G-VIT. We train a VIT model on either just IMAGENET or IMAGENET and IN-G-VIT. By training on IN-G-VIT, we achieve nearly 88% top-1 accuracy on IN-G-VIT (test) while minimally hurting performance on IMAGENET. To obtain standard deviations, we run the experiment with 5 random seeds.

E ADDITIONAL VISUALIZATIONS

Figure 12: Images from the text-to-image model used in this manuscript. Images are generated with captions identical to those used in Fig. 2(b) and Fig. 2(c). A comparison with DALL·E 2 is shown in Fig. 13.

Figure 13: DALL·E 2 images. Images are generated with captions identical to those used in Fig. 2(b) and Fig. 2(c). A comparison with the text-to-image model used in the paper is shown in Fig. 12.

Figure 14: STABLE-DIFFUSION images. Images are generated with captions identical to those used in Fig. 2(b) and Fig. 2(c). A comparison with the text-to-image model used in the paper is shown in Fig. 12.

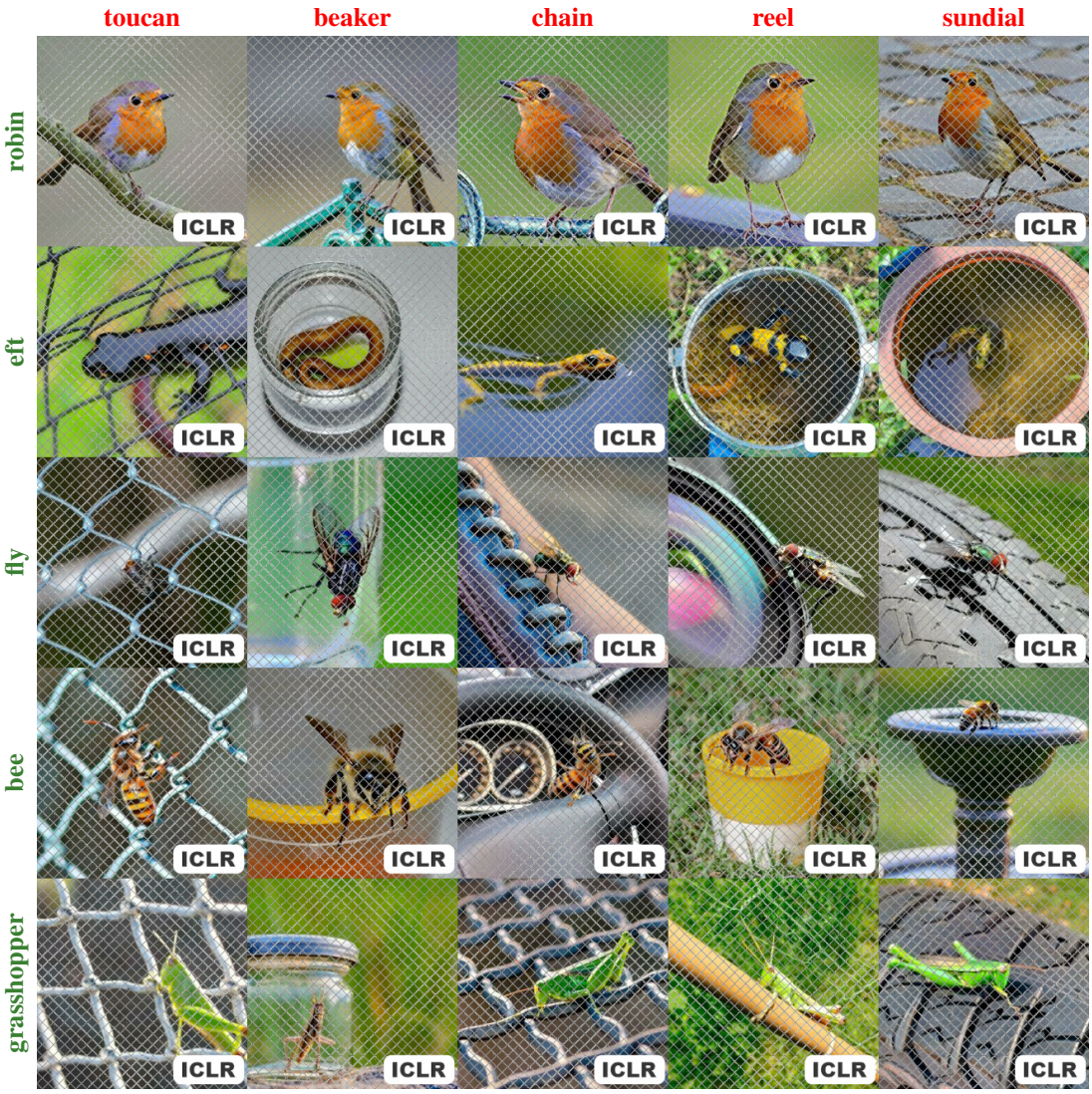


Figure 15: Further examples from IN-G-RN. The label at the top of the column is one of the incorrectly predicted top-3 labels and the label on the left is the true label.

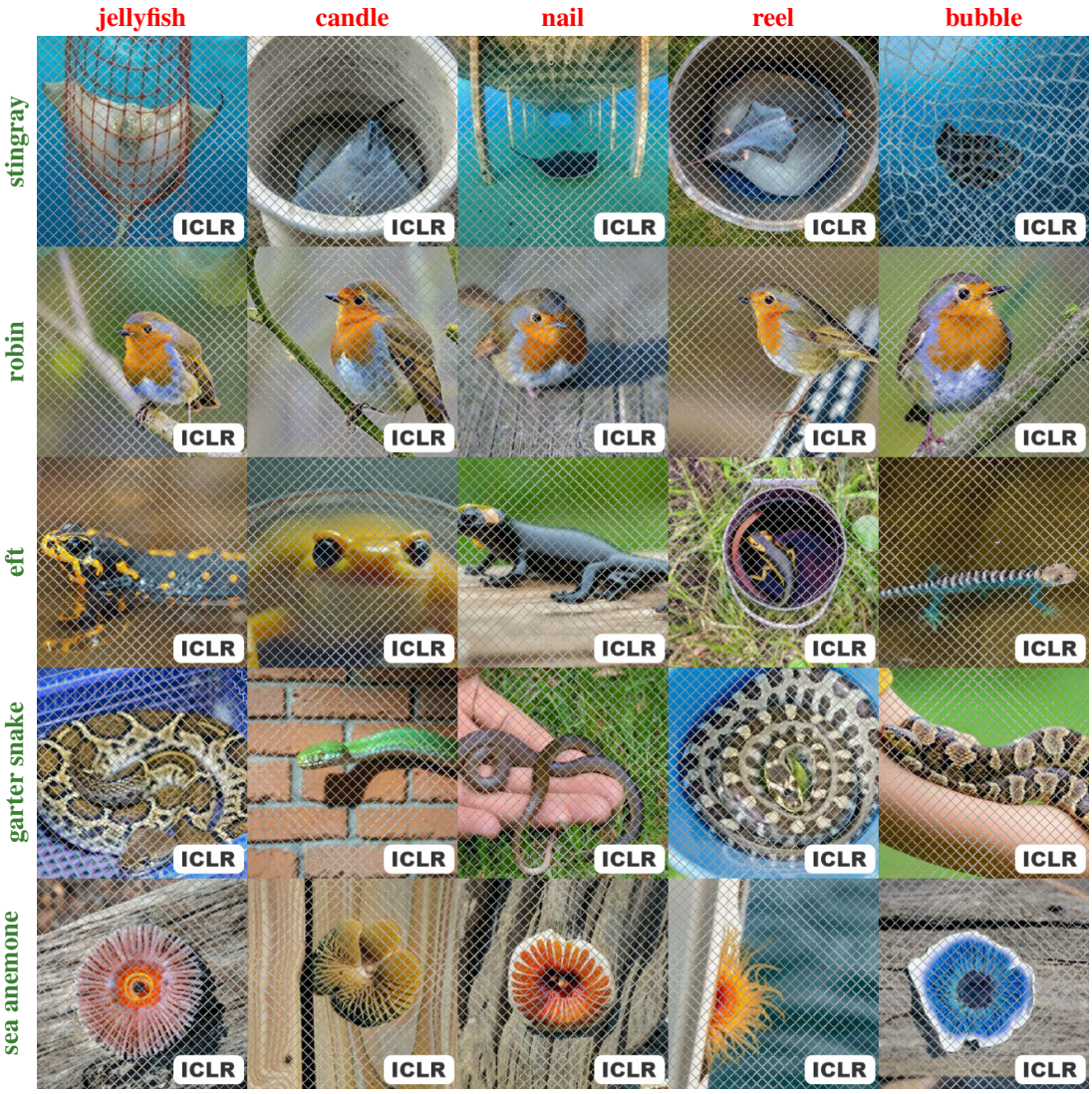


Figure 16: Examples from IN-G-VIT. The label at the top of the column is one of the incorrectly predicted top-3 labels and the label on the left is the true label.