Discovering Bugs in Vision Models using Off-the-shelf Image Generation and Captioning

Olivia Wiles*
DeepMind
oawiles@deepmind.com

Isabela Albuquerque
DeepMind
isabelaa@deepmind.com

Sven Gowal*
DeepMind
sgowal@deepmind.com

Abstract

Automatically discovering failures in vision models under real-world settings is an open challenge. This work describes how off-the-shelf, large-scale, image-to-text and text-to-image models, trained on vast amounts of data, can be leveraged to automatically find such failures. We detail a pipeline that demonstrates how we can interrogate classifiers trained on IMAGENET to find specific failure cases and discover spurious correlations. We also show that we can scale our approach to generate adversarial datasets targeting specific classifier architectures. This work serves as a proof-of-concept demonstrating the utility of large-scale generative models to automatically discover bugs in vision models in an open-ended manner.

1 Introduction

Machine learned models are known to exhibit numerous failures arising from using *shortcuts* and *spurious correlations* [20, 3, 60, 40]. As a result, they can fail catastrophically when training and deployment data differ [8]. Yet, only a few tools exist to automatically find failure cases of such models on unseen data (see expanded literature survey in App. A). Some methods analyze the performance of models by collecting new datasets (usually by scraping the web). These datasets must be large enough to obtain some indication of how models perform on a particular subset of inputs [29, 30, 50]. Other methods rely on expertly crafted, synthetic (and often unrealistic) datasets that highlight particular shortcomings [22, 63].

In this work, we present a methodology to automatically find failure cases of image classifiers in an open-ended manner, without prior assumptions on the types of failures and how they arise. We leverage off-the-shelf, large-scale, text-to-image, generative models, such as DALL·E 2 [49], IMAGEN [54] or STABLE-DIFFUSION [52], to obtain realistic images that can be reliably manipulated using a text prompt. We also leverage captioning models, such as FLAMINGO [2] or LEMON [33], to retrieve human-interpretable descriptions of each failure case. This provides a few advantages: *(i)* generative models trained on web-scale datasets can be re-used and have broad non-domain-specific coverage; *(ii)* they demonstrate basic compositionality, can generate novel data and can faithfully capture the essence of (most) prompts, thereby allowing images to be realistically manipulated; *(iii)* textual descriptions of failures can be easily interpreted (even by non-experts) and interrogated (e.g., by performing counterfactual analyses). Overall, our contributions are as follows:

- We describe a methodology to discover failures of image classifiers trained on IMAGENET [14]. To the contrary of prior work, we leverage off-the-shelf generative models, thereby avoiding the need to collect new datasets or to rely on expertly crafted heuristics to compose new images.
- Our approach surfaces human-interpretable failures by captioning inputs on which classifiers fail. These captions can be modified to produce alternative hypotheses of why failures occur.

*Equal contributions.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

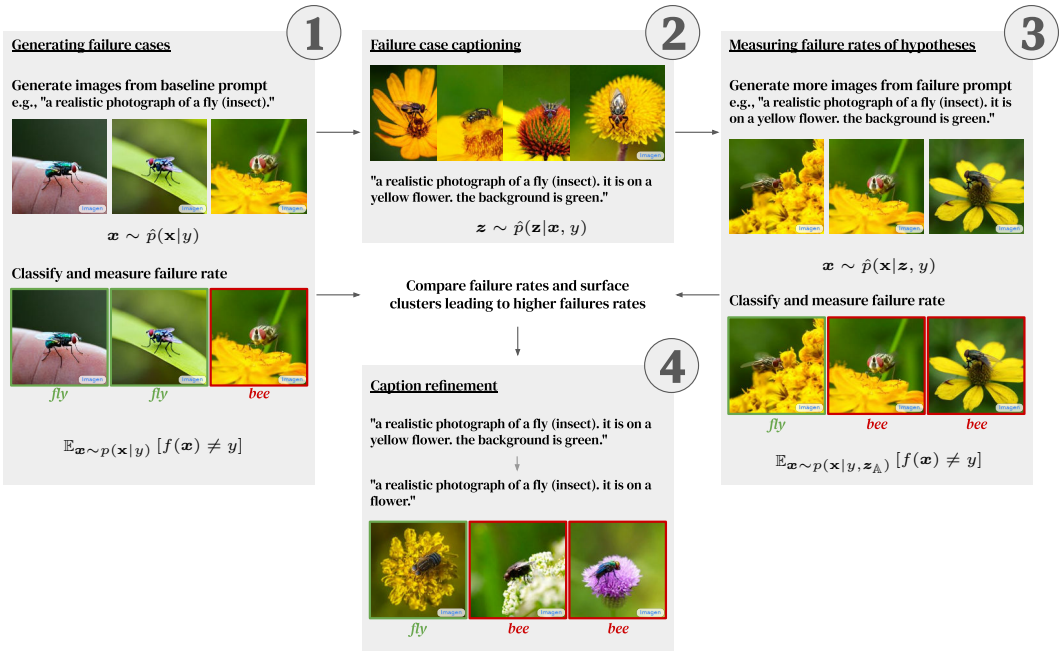


Figure 1: Diagram of our method. The method starts by generating images containing a given class y to measure the baseline failure rate of that class (right-hand side of Eq. 1). For misclassified image, we construct a detailed textual description. This description is used to produce new images and measure the failure rate on images corresponding to that description (left-hand side of Eq. 1). The final description can be edited (manually or automatically) to understand where the source of failures comes from.

- In Sec. B.2, we demonstrate the scalability of the approach by generating adversarial datasets (akin to IMAGENET-A; 29). In contrast to IMAGENET-A, our generated datasets align more closely with the original training distribution from IMAGENET and generalize to multiple classifier architectures.

Importantly, while this work focuses on vision models trained on IMAGENET, it is neither limited to IMAGENET nor the visual domain. It serves as a **proof-of-concept** that demonstrates how large-scale, off-the-shelf, generative models [6] can be combined to automate the discovery of *bugs* in machine learning models and produce compelling descriptions of model failures. The approach is agnostic to the model architecture, which can be treated as a black box.

2 Method

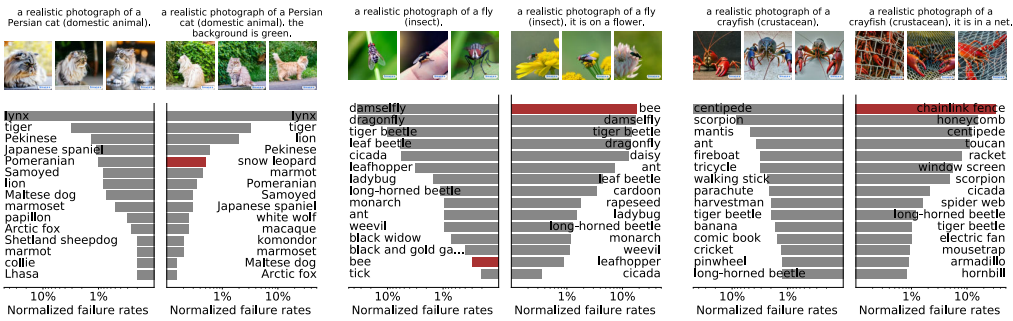
Notation. We consider a classifier $f : \mathbb{X} \to \mathbb{Y}$, where $\mathbb{X}$ is the set of inputs (i.e., images) and $\mathbb{Y}$ is the label set. We also assume that inputs $\boldsymbol{x} \in \mathbb{X}$ with label $y \in \mathbb{Y}$ are drawn from an underlying distribution $p(\mathbf{x}|\boldsymbol{z}, y)$ conditioned on latent representations $\boldsymbol{z} \in \mathbb{Z}$. In the context of this specific work, $\boldsymbol{z}$ is a textual description of the image $\boldsymbol{x}$. We are interested in discovering captions $\boldsymbol{z}$ corresponding to images $\boldsymbol{x} \sim p(\mathbf{x}|\boldsymbol{z}, y)$ with label y that lead to significantly higher misclassification rates than generic images drawn from the marginal distribution $p(\mathbf{x}|y)$ conditioned solely on the label. Formally, given a label y, we would like to find $\boldsymbol{z}$ with

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim p(\mathbf{x}|\boldsymbol{z},y)}\,[f(\boldsymbol{x}) \neq y] > \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p(\mathbf{x}|y)}\,[f(\boldsymbol{x}) \neq y] \tag{1}$$

where $[\cdot]$ represents the Iverson bracket.

As we do not have access to the true underlying distributions $p(\mathbf{x}|\boldsymbol{z}, y)$ and $p(\mathbf{x}|y)$, we leverage a large-scale text-to-image model (we use IMAGEN) to approximate them. Similarly, we approximate $p(\mathbf{z}|\boldsymbol{x}, y)$ with a captioning model (we use FLAMINGO). We denote approximations of these distributions with the symbol $\hat{p}$.

Generating failure cases. Our approach is described in Fig. 1. It consists of initially finding *baseline* failures for the underlying model f by sampling inputs $\boldsymbol{x}$ from $\hat{p}(\mathbf{x}|y)$. Given a label of



(a) `Persian cat` → `Snow leopard` (b) `Fly` → `Bee` (c) `Crayfish` → `Chainlink fence`

Figure 2: Distribution of failures of a RESNET-50 for the baseline and automatically discovered captions for three true and target label pairs. For each panel, failures resulting from the baseline caption are on the left and failures resulting from the discovered caption are on the right. We show the top-15 mistakes and three randomly sampled images for each caption. We highlight in red, the bar corresponding to the target label. Absolute failure rates, shown in Table 1, rise from 0.11% to 0.64%, 0.48% to 4.11% and 0.93% to 6.31%.

interest y, the output of this step is a set $\mathbb{D} = \{\boldsymbol{x}_i \sim \hat{p}(\mathbf{x}|y)\}_{i=1}^{N}$ (where N is the number of images we wish to generate), a set $\mathbb{D}_{\text{fail}} = \{\boldsymbol{x} \in \mathbb{D} | f(\boldsymbol{x}) \neq y\}$ and an estimate of the baseline failure rate $|\mathbb{D}_{\text{fail}}|/N$ (corresponding to the right-hand side of Eq. 1).[2] The conditioning on y is implemented using prompts such as *"a realistic photograph of a fly (insect)."*, which are automatically generated from the corresponding label name and WORDNET hierarchy.

Failure case captioning. For each input $\boldsymbol{x} \in \mathbb{D}_{\text{fail}}$, we would like to find a caption $\boldsymbol{z}$ that describes it. Ideally, we would like to find the caption $\boldsymbol{z}$ that maximizes the likelihood of sampling $\boldsymbol{x}$, i.e., $\boldsymbol{z} = \arg\max_{\boldsymbol{z}} \hat{P}(\boldsymbol{x}|\boldsymbol{z}, y)$. We may wish to impose constraints on $\boldsymbol{z}$, such as a maximum number of words or sentences.[3] Finding such a caption is computationally hard and measuring exact likelihoods $\hat{P}(\boldsymbol{x}|\boldsymbol{z}, y)$ can be challenging. Hence, we resort to sampling captions directly from a captioning model $\hat{p}(\mathbf{z}|\boldsymbol{x}, y)$ for each image.[4] To condition on y, we force the captioning model to only consider completions to the original baseline prompt. An example of resulting caption could be *"a realistic photograph of a fly (insect). the background is blurred. the fly is in focus. it is on a yellow flower. the background is green."* Each caption serves as a failure hypothesis.

Measuring failure rates of hypotheses. For each failure hypothesis or caption $\boldsymbol{z}$, we can measure its failure rate via sampling $\mathbb{E}_{\boldsymbol{x}\sim p(\mathbf{x}|\boldsymbol{z},y)}\left[f(\boldsymbol{x}) \neq y\right]$.[5] This step allows to surface captions $\boldsymbol{z}^\star$ that satisfy Eq. 1 by comparing the resulting failure rate with the baseline failure rate obtained initially.

Caption refinement and counterfactual analysis. Given a caption $\boldsymbol{z}^\star$, we would like to provide a shorter, self-contained caption that obtains a similar failure rate. For this step, we rely on handcrafted rules[6] and evaluate promising caption rewrites. Finally, as captions are human-readable, users can interact with the system and test alternative hypotheses.

3 Results on open-ended failure search

Setup. We evaluate a RESNET-50 [26] trained on IMAGENET and available online on TF-HUB.[7] We select three labels y at random, namely `Persian cat`, `fly` and `crayfish` and execute the

[2] In this work, we consider problematic misclassifications only and we restrict ourselves to failures where any of the top-3 predicted labels are not under the same parent as the true label y in the WORDNET hierarchy [45].

[3] These constraints guarantee that captions remain simple and not overly descriptive.

[4] This formulation implicitly assumes that any caption is as likely as another under a given label y, which in general does not hold true, but serves as a reasonable approximation.

[5] Grounding failures to simple textual descriptions allows to maintain the diversity of the generated images. It is important that generated images resemble images leading to the original failure without being exact copies.

[6] This may seem at odds with our claim on open-endedness. However, we note that this step is optional and its goal is simply to produce shorter failure descriptions. In this work, we evaluate all individual sentences and the most promising prompt is further refined by dropping adjectives.

[7] https://tfhub.dev/google/imagenet/resnet_v2_50/classification/5

protocol defined in Sec. 2. Fig. 2 (and Table 1 in Sec. B.1 of the appendix) shows the resulting automatically discovered failures. More failures for additional labels are also shown in Sec. B.1.

Discovered failures. Fig. 2(a) shows the distribution of failures for the baseline label `Persian cat`. We observe that the most frequent confusion, on images generated using the baseline caption *"a realistic photograph of a Persian cat (domestic animal)."* is with `lynx`. This mistake arises about 0.1% of the time and constitutes 87.3% of all failures. In comparison, the confusion with `snow leopard` is rather infrequent and arises only 0.00022% of the time. Our approach automatically discovers that adding *"the background is green."* to the caption results in a large increase in failure rates. Failures are 5.72× more likely and the model is 14.3× more likely to predict `snow leopard`. We observe that mistakes with wild animals are more prevalent when the cat is outdoors.

Similarly, Fig. 2(b) and Fig. 2(c) show failures on images of flies and crayfish, respectively. Flies on flowers are significantly more likely to be confused for bees when they are on flowers (497×), while crayfish in nets are more frequently confused as chainlink fences (3721×), honeycomb, window screens or spider webs. These highlight two shortcomings of the underlying classifier: *(i)* the over-reliance on spurious cues (such as the flower); *(ii)* the inability to determine which object is the main subject of a photograph (e.g., which of the net or crayfish is important).

Generalizability of failure descriptions. To verify that the discovered failures are not specific to the text-to-image model (IMAGEN) used in this manuscript and do not result from artifacts in the image generation process, we generate 30 images using the baseline and discovered captions with DALL·E 2 and STABLE-DIFFUSION (samples are shown in Fig. 7 and Fig. 8). We evaluate the failure rates for the `fly` and `crayfish` labels (which exhibited higher failure rates). With DALL·E 2, for the 30 images generated with the prompt *"a realistic photograph of a fly (insect)."*, 18 are correctly classified as flies and none are classified as bees. When adding *"it is on a flower."* to the prompt, the overall failure rate increases (only 14 images are correctly classified) and nine images are now classified as bees. Similarly, for *"a realistic photograph of a crayfish (crustacean)."*, 29 images are correctly classified as either `crayfish`, `spiny lobster`, `American lobster`, `Dungeness crab` or `king crab`, while none are classified as `chainlink fence`. When adding *"it is in a net."*, four are classified as `chainlink fence` (with `chainlink fence` appearing ten times in the top-3 predictions), while only 21 images are correctly classified.[8] Overall, we observe that discovered failures generalize across generative models. Finally, we also leverage *Google Image Search*[9] to find 30 images for each of the following queries: *"fly"*, *"fly on flower"*, *"crayfish"*, *"crayfish in net"* (images must have a resolution of at least 256×256 and should contain the true label). We classify all images and observe that the number of failures towards `bee` increases from zero to two and those towards `chainlink fence` increase from zero to four. This illustrates again that discovered failures are general and extend to real photographs.

4 Conclusion

The motivation behind our work is to develop a proof-of-concept demonstrating that today's large-scale text-to-image and image-to-text models can be leveraged to find human-interpretable failures in vision models. While we focus exclusively on IMAGENET, there are encouraging signs that these generative models could be used to probe models trained on specialized tasks such as medical imaging [37]. We note that there remain a number of keys challenge to address and detail some of them in more details in App. D.

Acknowledgments

We would like to thank Ali Eslami, Johannes Welbl for providing feedback, Marianne Monteiro for providing a simplified API to FLAMINGO and Taylan Cemgil, Sumanth Dathathri, Ira Ktena, Sylvestre-Alvise Rebuffi, Florian Stimberg for discussions throughout this work. Finally, we also thank the IMAGEN team – in particular, William Chan, Mohammad Norouzi and Chitwan Saharia.

[8] For STABLE-DIFFUSION, the number of failures increases from one to two and from four to 16, for the `fly` and `crayfish` labels, respectively.

[9] https://images.google.com

References

[1] A. Abid, M. Yuksekgonul, and J. Zou. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In *International Conference on Machine Learning*. PMLR, 2022. 9

[2] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. 1

[3] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 1

[4] S. Baluja and I. Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*, 2017. 9

[5] M. Binkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. 14

[6] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 2

[7] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, and S. Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018. 20

[8] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, 2018. 1

[9] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the ACM Workshop on Artificial Intelligence and Security*, 2017. 9

[10] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy*, 2017. 9

[11] S. Casper, M. Nadeau, D. Hadfield-Menell, and G. Kreiman. Robust feature-level adversaries are interpretability tools. *arXiv preprint arXiv:2110.03605*, 2022. 9

[12] M. J. Choi, A. Torralba, and A. S. Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 2012. 9

[13] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. *arXiv preprint arXiv:1311.3618*, 2013. 9

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009. 1

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*, 2020. 13

[16] S. Eyuboglu, M. Varma, K. Saab, J.-B. Delbrouck, C. Lee-Messer, J. Dunnmon, J. Zou, and C. Ré. Domino: Discovering systematic errors with cross-modal embeddings. *arXiv preprint arXiv:2203.14960*, 2022. 9

[17] S. Garg, V. Perot, N. Limtiaco, A. Taly, E. H. Chi, and A. Beutel. Counterfactual fairness in text classification through robustness. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2019. 9

[18] Y. Ge, J. Xu, B. N. Zhao, L. Itti, and V. Vineet. Dall-e for detection: Language-driven context image synthesis for object detection. *arXiv preprint arXiv:2206.09592*, 2022. 9

[19] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018. 9

[20] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. In *Nature Machine Intelligence*, 2020. 1, 9

[21] R. Geirhos, K. Meding, and F. A. Wichmann. Beyond accuracy: quantifying trial-by-trial behaviour of cnns and humans by measuring error consistency. In *Advances in Neural Information Processing Systems*. 2020. 9

[22] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2022. 1

[23] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 9

[24] S. Gowal, C. Qin, P.-S. Huang, T. Cemgil, K. Dvijotham, T. Mann, and P. Kohli. Achieving Robustness in the Wild via Adversarial Mixing with Disentangled Representations. *arXiv preprint arXiv:1912.03192*, 2019. 9

[25] F. Harder, M. J. Asadabadi, D. J. Sutherland, and M. Park. Differentially private data generation needs better features. *arXiv preprint arXiv:2205.12900*, 2022. 19

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3, 13

[27] J. Heek, A. Levskaya, A. Oliver, M. Ritter, B. Rondepierre, A. Steiner, and M. van Zee. Flax: A neural network library and ecosystem for JAX, 2020. URL `http://github.com/google/flax`. 20

[28] D. Hendrycks and T. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*, 2018. 9

[29] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019. 1, 2, 9

[30] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. *arXiv preprint arXiv:2006.16241*, 2020. 1, 9

[31] T. Hennigan, T. Cai, T. Norman, and I. Babuschkin. Haiku: Sonnet for JAX, 2020. 20

[32] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*. 2017. 14

[33] X. Hu, Z. Gan, J. Wang, Z. Yang, Z. Liu, Y. Lu, and L. Wang. Scaling up vision-language pre-training for image captioning. *arXiv preprint arXiv:2111.12233*, 2021. 1

[34] S. Jain, H. Lawrence, A. Moitra, and A. Madry. Distilling model failures as directions in latent space. *arXiv preprint arXiv:2206.14754*, 2022. 9

[35] S. Jain, H. Salman, A. Khaddaj, E. Wong, S. M. Park, and A. Madry. A data-based perspective on transfer learning. *arXiv preprint arXiv:2207.05739*, 2022. 18

[36] R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017. 9

[37] J. N. Kather, N. Ghaffari Laleh, S. Foersch, and D. Truhn. Medical domain knowledge in domain-agnostic generative ai. In *npj Digital Medicine*, 2022. 4, 18, 19

[38] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, 2012. 9

[39] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, 2020. 13

[40] A. Kuehlkamp, B. Becker, and K. Bowyer. Gender-from-iris or gender-from-mascara? In *IEEE Winter Conference on Applications of Computer Vision*, 2017. 1

[41] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 9

[42] C. Laidlaw, S. Singla, and S. Feizi. Perceptual Adversarial Robustness: Defense Against Unseen Threat Models. *arXiv preprint arXiv:2006.12655*, 2020. 9

[43] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking clever hans predictors and assessing what machines really learn. In *Nature Communications*, 2019. 9

[44] H. Mania, J. Miller, L. Schmidt, M. Hardt, and B. Recht. Model similarity mitigates test set overuse. In *Advances in Neural Information Processing Systems*. 2019. 9

[45] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995. 3

[46] E. Perez, S. Huang, F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, and G. Irving. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022. 9

[47] H. Qiu, C. Xiao, L. Yang, X. Yan, H. Lee, and B. Li. SemanticAdv: Generating Adversarial Examples via Attribute-conditional Image Editing. *arXiv preprint arXiv:1906.07927*, 2019. 9

[48] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *OpenAI Blog*, 2021. 9

[49] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[50] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do ImageNet Classifiers Generalize to ImageNet? *arXiv preprint arXiv:1902.10811*, 2019. 1, 9

[51] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. 9

[52] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1

[53] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2020. 9

[54] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1

[55] Y. Song, R. Shu, N. Kushman, and S. Ermon. Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems*, 2018. 9

[56] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022. 13

[57] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 9

[58] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. 13

[59] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 13

[60] A. Torralba, A. A. Efros, and others. Unbiased look at dataset bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2011. 1, 9

[61] E. Wong and J. Z. Kolter. Learning perturbation sets for robust machine learning. In *International Conference on Learning Representations*, 2021. 9

[62] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018. 9

[63] K. Xiao, L. Engstrom, A. Ilyas, and A. Madry. Noise or Signal: The Role of Image Backgrounds in Object Recognition. *arXiv preprint arXiv:2006.09994*, 2020. 1, 9

Appendix

A Related Work

Model failures. Spurious correlations can entice models to learn unintended shortcuts that obtain high accuracy on the training set but fail to generalize [43, 20]. Recht et al. [50] show that the accuracy of IMAGENET models is impacted by changes in the data collection process, while Torralba et al. [60], Khosla et al. [38], Choi et al. [12] explore how contextual bias affects generalization. Mania et al. [44] demonstrate that models trained on IMAGENET make consistent mistakes with one another and Geirhos et al. [21] show that these mistakes are not necessarily consistent with human judgment.

Evaluation datasets. Understanding how model failures arise and empirically analyzing their consequences often requires collecting and annotating new test datasets. Hendrycks et al. [29] collected datasets of natural adversarial examples (IMAGENET-A and IMAGENET-O) to evaluate how model performance degrades when inputs have limited spurious cues. Hendrycks et al. [30] collected four real-world datasets (including IMAGENET-R) to understand how models behave under distribution shifts. In many cases, particular shortcomings can only be explored using synthetic datasets [13]. Hendrycks and Dietterich [28] introduced IMAGENET-C, a synthetic set of common corruptions. Geirhos et al. [19] propose to use images with a texture-shape cue conflict to evaluate the propensity of models to over-emphasize texture cues. Xiao et al. [63], Sagawa et al. [53] investigate whether models are biased towards background cues by compositing foreground objects with various background images (IMAGENET-9, WATERBIRDS). In all cases, building such datasets is time-consuming and requires expert knowledge.

Automated failure discovery. In some instances, it is possible to distill rules or specifications that constrain the input space enough to enable the automated discovery of failures via optimization or brute-force search. In vision tasks, adversarial examples, which are constructed using ℓ_p-norm bounded perturbations of the input, can cause neural networks to make incorrect predictions with high confidence [9, 10, 23, 41, 57]. In language tasks, some efforts manually compose templates to generate test cases for specific failures [36, 17, 51]. Such approaches rely on human creativity and are intrinsically difficult to scale. Several works [4, 55, 62, 47, 61, 42, 24] go beyond hard-coded rules by leveraging generative and perceptual models. However, such approaches are difficult to automate as it is unclear how to relate specific latent variables to isolated structures of the original input. Finally, we highlight a concurrent work from Ge et al. [18], which leverages captioning and text-to-image models to construct background images to evaluate (and improve) an object detector. Their approach requires compositing the resulting images with foreground objects and is not open-ended, in the sense that it relies on the availability of a source dataset of background images. Perhaps, the work by Perez et al. [46] on *red-teaming* language models is the most similar to ours. Perez et al. demonstrate how to prompt a language model to automatically generate test cases to probe another language model for toxic and other unintended output.

Interpretability. A recent direction is to extract interpretable explanations of failures. Casper et al. [11] demonstrate how to use adversarial patches to add objects to images and fool a classifier; unlike our work, the resulting images are clearly not realistic to a human. Leveraging a dataset of images and auxiliary information in the form of attributes or image-to-text embeddings (e.g. from CLIP [48]), other works [1, 34, 16] aim to explain the spurious correlations or other factors that cause failures in the dataset. However, their analyses and conclusions are limited by the images present in the dataset.

B Additional results

B.1 Open-ended failure search

Table 1 shows the absolute failure rates of a RESNET-50 trained on IMAGENET for the three true and target label pairs highlighted in the main manuscript. For each row, we sample images from the text-to-image model until we obtain 20 images that are misclassified as the target label.[10] Overall, we observe that failure rates are between six and nine times more likely with our automatically discovered prompts. Failures towards specific labels become orders of magnitude more frequent.

True label	Target label	Caption	Failure rate (any)		Failure rate (target)	
`Persian cat`	`Snow leopard`	a realistic photograph of a Persian cat (domestic animal).	0.11%	1×	0.00022%	1×
		— " — the background is green.	0.64%	6×	0.0032%	14×
`Fly`	`Bee`	a realistic photograph of a fly (insect).	0.48%	1×	0.0014%	1×
		— " — it is on a flower.	4.11%	9×	0.72%	497×
`Crayfish`	`Chainlink fence`	a realistic photograph of a crayfish (crustacean).	0.93%	1×	0.00047%	1×
		— " — it is in a net.	6.31%	7×	1.73%	3721×

Table 1: Absolute failure rates of a RESNET-50 for three true and target label pairs. We show the total failure rate (i.e., the model prediction is different from the true label) as well as the target failure rate (i.e., the model prediction is the target label). Captions are automatically discovered using the method detailed in Sec. 2.[11]

Similarly to Fig. 2 and Table 1, Fig. 3 and Table 2 show failure cases automatically found by our pipeline for the same RESNET-50 on additional labels. 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`).

[10] This amounts to 9.1M samples for the first row of the table, 625K for the second and 1.4M, 2.8K, 4.3M, 1.2K for the subsequent rows.

[11] As a point of comparison, we can also evaluate the baseline failure rates on images from the IMAGENET test set. For `Persian cat`, `fly` and `crayfish` the baseline failure rates are 16%, 8% and 18%, respectively (the target failure rate is 0% for all labels). These failure rates are higher than on generated images. This is perhaps indicating that the generative model produces images that are more canonical and conservative.

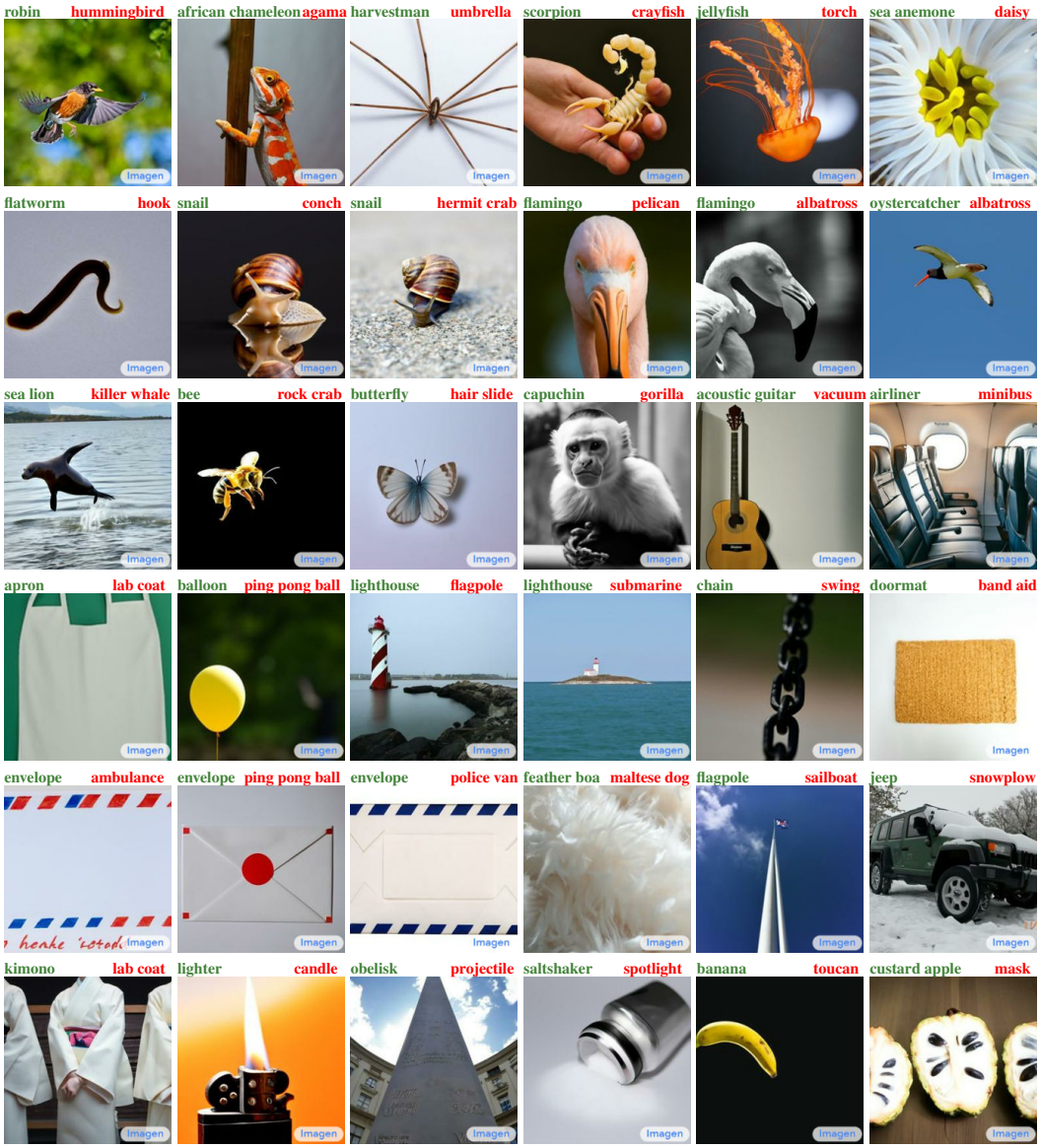


Figure 3: Illustration of failure cases listed in Table 2. The correct label is to the left in **green**. The incorrect prediction is to the right in **red**. The model used is a RESNET-50 found on TF-HUB.

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

Table 2: 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. 2. 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.

B.2 Adversarial dataset generation at scale

In this section, we demonstrate how to apply our automated pipeline to generate large datasets of failures. We seed our search by captioning a set of problematic images from IMAGENET-A. We show that the discovered failures generalize across initializations of a given model architecture and between models of different architectures.

Generating large-scale datasets. We assume that we have access to a set of problematic captions that describe potential failure cases.[12] These captions are automatically extracted from the 7,500 images of IMAGENET-A using FLAMINGO, limiting its output to two sentences maximum. For each caption, we sample up to a thousand images keeping those leading to misclassifications and limit the number of misclassified images kept per caption. We consider two models: a RESNET-50 (abbreviated hereafter by RN) and a Vision Transformer in its B/16 variant [15], abbreviated by VIT. Both models are trained solely on IMAGENET and achieve 76% and 80% top-1 accuracy, respectively. This yields two separate datasets of failures which we refer to as IN-G-RN and IN-G-VIT that are of size 12,332 and 9,536 respectively.

Visualizations of failure cases. Samples from IN-G-RN are shown in Fig. 4, along with true and predicted labels. We can see that while the images clearly show the correct class, the model erroneously predicts a different one. Additional samples for both IN-G-RN and IN-G-VIT are visible in Fig. 9 and Fig. 10 (at the end of this manuscript).

Figure 4: Examples of failures automatically found in IN-G-RN. The correct label is to the left in **green**. The incorrect prediction is to the right in **red**. More examples are given in Fig. 9 and 10.

Generalizability of failure cases. To investigate whether the discovered failures generalize across classification models, we train an additional Residual Network (RESNET) and VIT on IMAGENET with the exact same setup as our two original models but different random seeds. We also consider a large set of additional models trained on IMAGENET and optionally pre-trained on larger datasets obtained from TF-HUB:

- VIT-B*, VIT-L*, VIT-S* [15]: VITs pretrained on IMAGENET21K.
- VIT-R* [56]: a hybrid VIT and RESNET model pretrained on IMAGENET21K.
- BIT-* [39]: BIT models pretrained either on IMAGENET21K (BIT-M *) or not pretrained (BIT-S *).
- INCEPTION_RESNET V2 [59]: a hybrid INCEPTION, RESNET model with no pretraining.
- INCEPTION* [58]: INCEPTION models with no pretraining.
- RESNET* [26]: RESNET models with no pretraining.

Fig. 5 shows the failure rates induced by both datasets and IMAGENET on all models (failures are accounted when the top-3 predictions do not include the true label). First, we observe that failures transfer well between models of the same architecture. Indeed, about 80% of the failures in IN-G-RN transfer to the RESNET-50 we trained, while the ones in IN-G-VIT transfer with at least 75% chance to the other VIT. Second, we can observe that while there is a drop in performance, failures for a given model architecture often transfer across architectures. Even when large scale pretraining is used (e.g. the BIT-M * models and VIT models were pretrained on IMAGENET21K), failures transfer at a rate of about 35-55% for IN-G-RN and about 45-65% for IN-G-VIT. Within a model class,

[12] This assumption is not necessary. However, it accelerates our search by generating images that are more likely to induce failures.

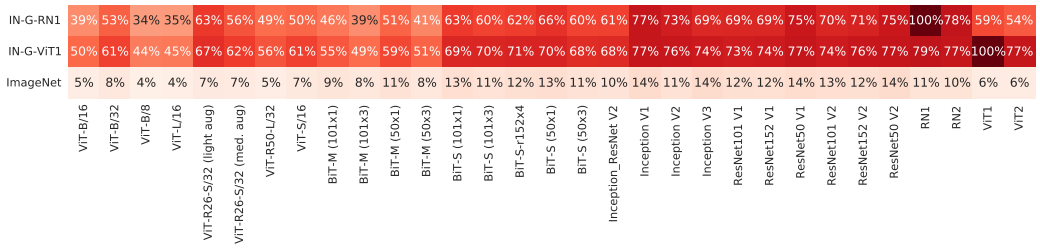


Figure 5: Failure rates (top-3) for different models on two generated datasets. We report the failure rates of different models trained on IMAGENET on both IN-G-RN and IN-G-VIT as well as on IMAGENET.

larger versions of the model seem more robust. For example, VIT-B/16 is more robust than VIT-B/32 (and correspondingly gets lower error on IMAGENET) 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.

Distribution shift. We compare our generated datasets to IMAGENET and IMAGENET-A. The aim is to validate that we generate images that are similar to those in IMAGENET. We compute the Fréchet Inception Distance (FID; 32) and Kernel Inception Distance (KID; 5) between the generated images and the IMAGENET test set. Table 3 also shows the FID and KID of a sample of size 50000 of the IMAGENET train set and IMAGENET-A. We find that our generated images are *more* similar to those from IMAGENET under both metrics than those from IMAGENET-A.

	FID ↓	KID ↓
IMAGENET-A	56.6	0.0460
IN-G-RN	48.3	0.0305
IN-G-VIT	53.9	0.0330
IMAGENET (train)	2.3	0.0003

Table 3: FID and KID scores. We report the FID and KID scores of IMAGENET (train), IMAGENET-A and our two generated datasets (IN-G-RN, IN-G-VIT) in relation to IMAGENET (test).

C Additional figures

Figure 6: Images from IMAGEN (which was 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. 7.

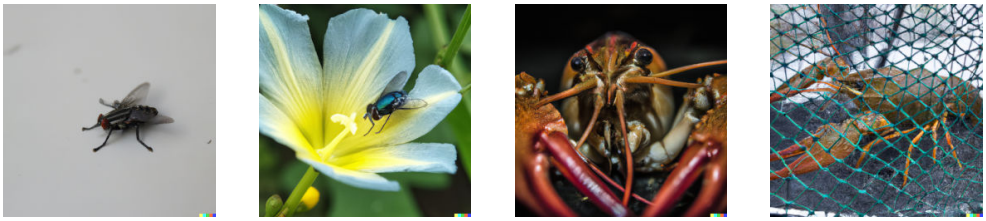


Figure 7: DALL·E 2 images. Images are generated with captions identical to those used in Fig. 2(b) and Fig. 2(c).

Figure 8: STABLE-DIFFUSION images. Images are generated with captions identical to those used in Fig. 2(b) and Fig. 2(c).

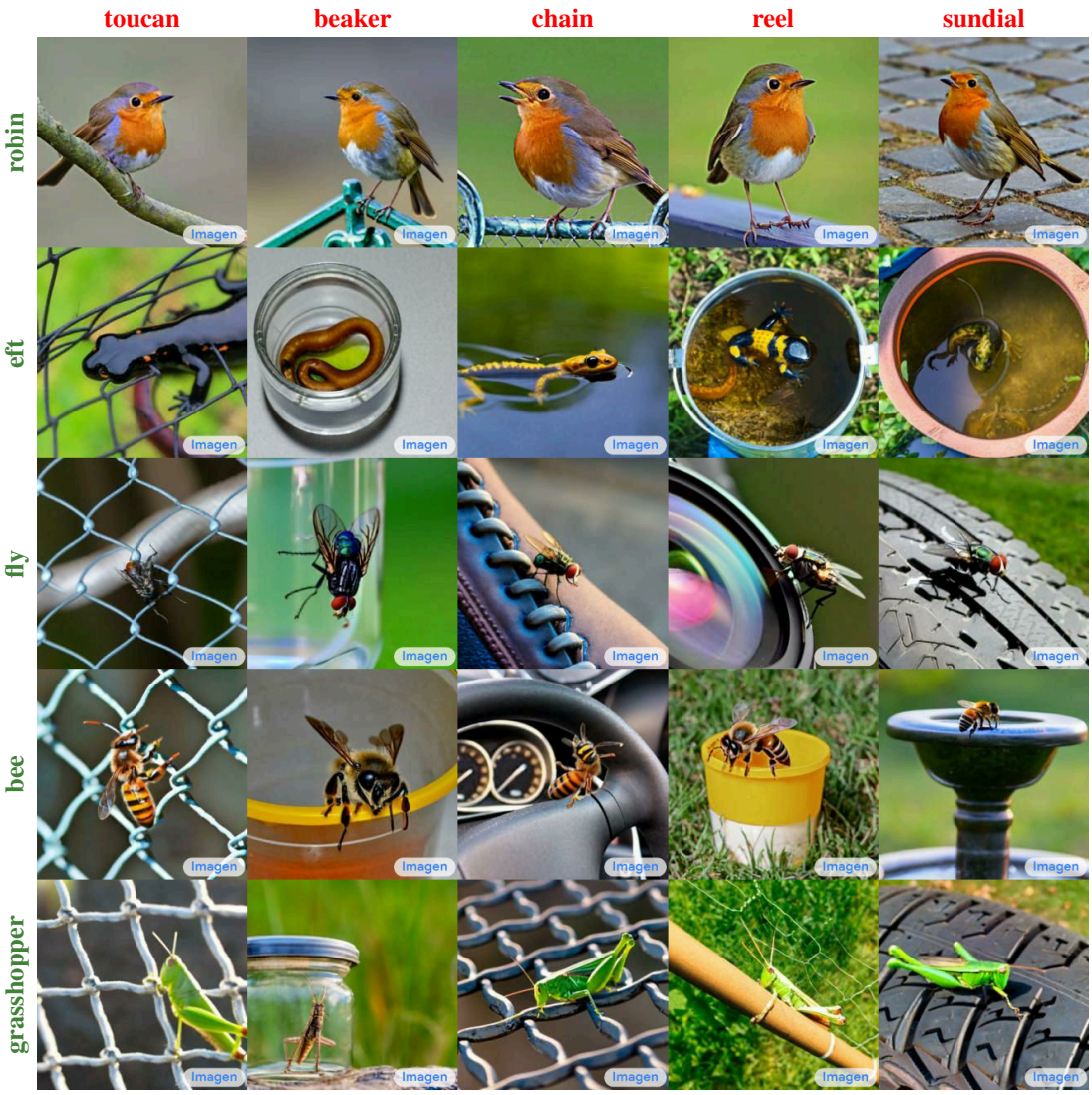


Figure 9: Further examples from IN-G-RN. The label at the top of the column is one of the incorrectly predicted top-3 labels and the label on the left is the true label.

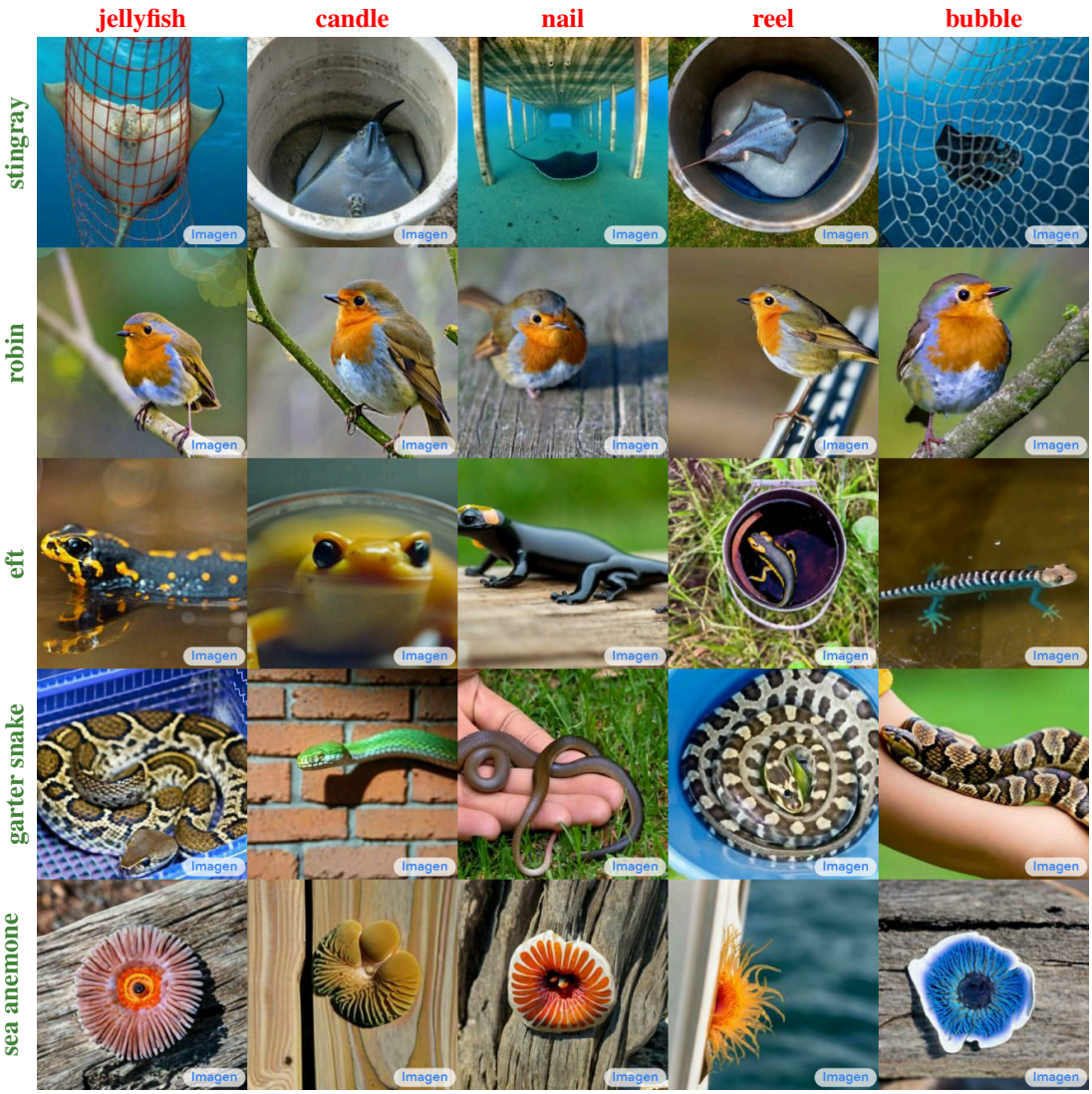


Figure 10: Examples from IN-G-VIT. The label at the top of the column is one of the incorrectly predicted top-3 labels and the label on the left is the true label.

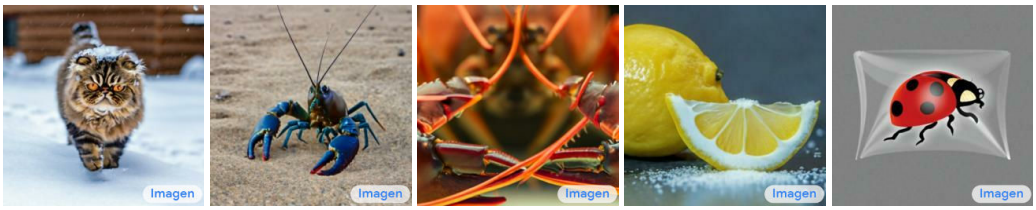
D Discussion and Limitations

The motivation behind our work is to develop a proof-of-concept demonstrating that today's large-scale text-to-image and image-to-text models can be leveraged to find human-interpretable failures in vision models. While we focus exclusively on IMAGENET, there are encouraging signs that these generative models could be used to probe models trained on specialized tasks such as medical imaging [37]. Overall, there remain a number of keys challenges to address.

Coverage. While our approach can successfully be used to demonstrate the presence of failures, it is important to understand that (just like scraping the web) it cannot prove their absence. In other words, there is no guarantee that it will discover all failures of a given model. Moreover, the generative model is only an approximation of the distribution of interest and may lack coverage. For example, it might almost never generate "a lawnmower falling down from the sky" (an actual image from the IMAGENET training set; 35) when prompted with *"a realistic photograph of a lawnmower"*. While this can help ground failures to scenes that are likely to occur in the real-world, it also means that rare failures are unlikely to be discovered (see Fig. 11(a)).

Bias. While we take the view here that off-the-shelf large-scale generative models are trained on diverse and unbiased data, the reality is far from it: these models mirror the distribution of images and captions seen on the web. The generative model may over-sample particular regions of the image manifold and, as a result, our approach is more likely to discover failures in these high-density regions and miss failures pertaining to other regions (see Fig. 11(b)). Possible solutions to reduce bias include clever prompting (which introduces expert knowledge) or discovering failure prompts more actively by avoiding random sampling (e.g., through adversarial techniques).

Captioning issues. Using captions as our latent representation allows our approach to produce human-interpretable explanations. This requirement constrains our search to failures that can be explained in words. Not only is it possible for the captioning model to miss important details or produce ungrounded captions, but some failures may simply be hard to describe (even by a human). As a result, newly generated images may look different from the set of images that induced the original failure. We note that efficiently enforcing consistency between the generated and original images (through a common caption) is an open problem since we would like to search over *reasonable*



(a) Coverage (b) Bias (c) Captioning issues (d) Generation issues (e) OOD sampling

Figure 11: Illustrative examples of various challenges. (a) Persian cats in snow (generated using *"a realistic photograph of a Persian cat (domestic animal). it is walking in the snow."*) are misclassified as snow leopards at a rate of 0.016%, which is significantly higher than the failure rate of 0.0032% induced by the automatically found caption (*"— " — the background is green."*); the total failure rate also increases twelve-fold to 8.15% (from 0.64%). (b) It is estimated that only 1 in 10,000 crayfish turn blue. However, 9% of the images generated using *"a realistic photograph of a crayfish (crustacean)."* contain a blue crayfish (estimated by manually looking at 100 samples). (c) This image of a crayfish is misclassified as a chainlink fence. The output of the captioning model for this particular image is *"a realistic photograph of a crayfish. the crayfish is very detailed. the crayfish is facing the camera. the crayfish is orange. it has two antennae."* While the caption describes the image, it does not provide enough details to reconstruct the image. (d) This image is generated from the caption *"a realistic photograph of a saltshaker (container). there is a lemon slice on the side of the salt shaker."* While the image contains a lemon, the main subject which corresponds to the true class y (`saltshaker`) is not visible. (e) Generated with the caption *"a realistic photograph of a ladybug (insect). it is in a plastic bag."*, this image illustrates that text-to-image models can create image that are not from the intended distribution (i.e., of realistic photographs).

captions that are likely to produce images corresponding to the original failure. Fig. 11(c) shows an example. The figure shows a `crayfish` misclassified as a `chainlink fence`. While the reason for that failure is immediately obvious to us, it remains difficult to describe with a succinct caption.

Image generation issues. While the text-to-image model may make occasional mistakes (such as generating the wrong object for unambiguous prompts), subtle errors can also arise from the interplay between the model and its prompt. The prompt may be ambiguous, such as using words that have multiple meanings (e.g., a *"walking stick"* can be both a cane or an insect), or may describe multiple objects with complex relationships that exacerbate failures (see Fig. 11(d)).

Out-of-distribution sampling. Ensuring that images sampled from an off-the-shelf generative model are part of the intended distribution (e.g., resembling IMAGENET) is difficult. We start our prompts with *"a realistic photograph"* in a bid to help steer the approximated distribution $\hat{p}(\mathbf{x}|y, \boldsymbol{z})$ away from artistic drawings and closer to the true distribution $p(\mathbf{x}|y, \boldsymbol{z})$. This approach is effective, but not always successful (see Fig. 11(e)). In some cases, finding a suitable prompt is not obvious (e.g., to output images from a particular medical domain; 37) and fine-tuning models on the dataset of interest may be necessary.

Privacy. As we are generating large amounts of images and captions, it is important to consider the privacy risks associated with our approach. While these risks can be mitigated by using generative models trained on public, non-sensitive data, we believe that more research on private generative modelling is necessary [25].

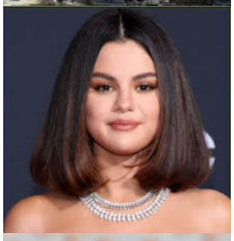
Despite these challenges, we foresee that large-scale generative models will increasingly be used as debugging tools. In this work, we introduced an automated pipeline that discovers failure cases in vision models. It constitutes a proof-of-concept that such a system allows for large-scale investigations of vision models in an open-ended manner.

E Experimental details

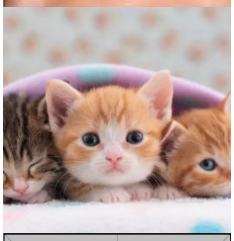
E.1 Prompting the image-to-text model (FLAMINGO)

To ensure that captions are descriptive and composed of short sentences. We prompt FLAMINGO with the following:

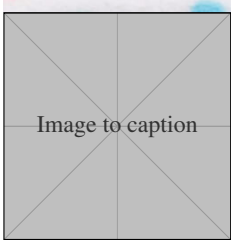
is a realistic picture of two penguins. They are holding hands. They are standing in front of the sea. The picture is mostly grey. The penguins are facing away from the camera. They take up most of the image.



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]. [...]`

We also set the decoding strategy to be *greedy* (as we did not observe significant improvements from using beam search).

E.2 Resource requirements

Each experiment in Sec. 3 and Sec. B.2 runs on twenty TPUv4s. We use JAX [7], Haiku [31] and Flax [27].