# A Experimental Details

All experiments were done on a single NVIDIA Titan Xp GPU. All of our code can be found attached in the supplementary material.

## A.1 Learning the Concept Bank

Before learning the concept bank, we ran an analysis to see how many samples from a concept we would need to get a reasonable performance. To test this, we follow [2, 1]. Concretely, for a given concept indexed by $i$, we collect two sets of embeddings $P_i = \{f(x_{p_1}), ..., f(x_{p_{N_p}})\}$, that contains the concept, and similarly negative examples $N_i = \{f(x_{n_1}), ..., f(x_{p_{N_n}})\}$ that do not contain the concept. We then train a linear SVM using $P_i$ and $N_i$ to learn the corresponding CAV. Since the sample sizes are limited for many concepts, we run leave-10-out cross-validation to test these linear SVMs, i.e. we have a held-out set of 10 images, and we run this test 25 times.

On Figure 1, on x-axis we give the minimum number of training samples used for a concept, i.e. for an x-axis value of 10 we used all concepts where there are $> 40 + 5$ images such that we can choose $40$ positive training and $5$ positive validation images, and similarly we sample $40 + 5$ negative images. On the y-axis, we report the accuracy on the held out $10$ images, averaged over different concepts and $25$ runs. We observe that performance saturates around $50$ samples, hence we use $50$ as a threshold to filter concepts in the later sections. We also report accuracy for different concept categories, grouped under Primary Descriptors, Secondary Descriptors, Shape, and Color features.

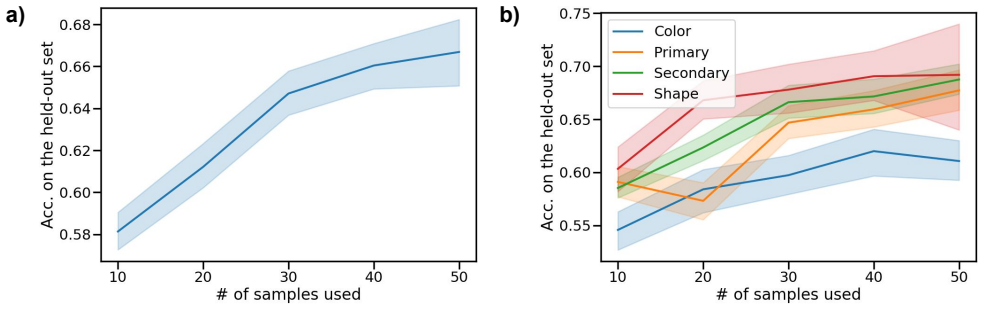


Figure 1: **Ablation on the number of images per a concept.**

## A.2 Generating Counterfactual Explanations

We use L2 regularization strength $0.0001$, L1 regularization strength $0.0001$, used mean margin statistics for validity constraints, and implemented the optimization procedure in PyTorch. We ran CCE individually over a batch of mistakes, and report examples in the main paper. For algorithmic details, we refer the reader to [1].

## A.3 PCBM

We use the Inception backbone from the PyTorch implementation. We then use scikit-learn library to fit the linear model (SGDClassifier). We use the hyperparameters reported in the PCBM paper without any adjustments. Particularly, we use the regularization strength $0.01$, sparsity ratio $0.9$, and for the learning rate we use scikit-learn's 'optimal' learning rate setting for the SGDClassifier. We trained the model on a subset of Fitzpatrick17k that was not used to label concepts. Then, we tested the model on the held-out DDI dataset. We ran our experiments with 5 seeds, and report the error bars in the main paper.

| Concept | DDI # Images | % of images | Fitz17k # Images | % of Images |
|---|---|---|---|---|
| Vesicle | 0 | 0.00 | 46 | 0.01 |
| Papule | 410 | 0.62 | 1.170 | 0.32 |
| Macule | 24 | 0.04 | 13 | 0.00 |
| Plaque | 168 | 0.26 | 1.967 | 0.53 |
| Abscess | 0 | 0.00 | 5 | 0.00 |
| Pustule | 0 | 0.00 | 103 | 0.03 |
| Bulla | 0 | 0.00 | 64 | 0.02 |
| Patch | 6 | 0.01 | 149 | 0.04 |
| Nodule | 46 | 0.07 | 189 | 0.05 |
| Ulcer | 13 | 0.02 | 154 | 0.04 |
| Crust | 53 | 0.08 | 497 | 0.13 |
| Erosion | 14 | 0.02 | 200 | 0.05 |
| Excoriation | 0 | 0.00 | 46 | 0.01 |
| Atrophy | 1 | 0.00 | 69 | 0.02 |
| Exudate | 13 | 0.02 | 144 | 0.04 |
| Purpura/Petechiae | 0 | 0.00 | 10 | 0.00 |
| Fissure | 0 | 0.00 | 32 | 0.01 |
| Induration | 0 | 0.00 | 33 | 0.01 |
| Xerosis | 0 | 0.00 | 35 | 0.01 |
| Telangiectasia | 5 | 0.01 | 100 | 0.03 |
| Scale | 103 | 0.16 | 686 | 0.19 |
| Scar | 4 | 0.01 | 123 | 0.03 |
| Friable | 14 | 0.02 | 153 | 0.04 |
| Sclerosis | 0 | 0.00 | 27 | 0.01 |
| Pedunculated | 8 | 0.01 | 26 | 0.01 |
| Exophytic/Fungating | 8 | 0.01 | 42 | 0.01 |
| Warty/Papillomatous | 18 | 0.03 | 46 | 0.01 |
| Dome-shaped | 67 | 0.10 | 146 | 0.04 |
| Flat topped | 0 | 0.00 | 18 | 0.00 |
| Brown(Hyperpigmentation) | 363 | 0.55 | 760 | 0.21 |
| Translucent | 7 | 0.01 | 16 | 0.00 |
| White(Hypopigmentation) | 22 | 0.03 | 257 | 0.07 |
| Purple | 2 | 0.00 | 85 | 0.02 |
| Yellow | 23 | 0.04 | 245 | 0.07 |
| Black | 29 | 0.04 | 90 | 0.02 |
| Erythema | 235 | 0.36 | 2.139 | 0.58 |
| Comedo | 3 | 0.00 | 24 | 0.01 |
| Lichenification | 1 | 0.00 | 25 | 0.01 |
| Blue | 0 | 0.00 | 5 | 0.00 |
| Umbilicated | 5 | 0.01 | 49 | 0.01 |
| Poikiloderma | 0 | 0.00 | 5 | 0.00 |
| Salmon | 3 | 0.00 | 10 | 0.00 |
| Wheal | 0 | 0.00 | 21 | 0.01 |
| Acuminate | 0 | 0.00 | 8 | 0.00 |
| Burrow | 0 | 0.00 | 5 | 0.00 |
| Gray | 0 | 0.00 | 5 | 0.00 |
| Pigmented | 1 | 0.00 | 5 | 0.00 |
| Cyst | 0 | 0.00 | 6 | 0.00 |
| Do not consider this image | 20 | 0.03 | 460 | 0.12 |

Table 1: Concept Statistics for the Dataset. In total, we have $3230 + 656 = 3886$ images.

## B Dataset Statistics

## References


[1] Abubakar Abid, Mert Yuksekgonul, and James Zou. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

[2] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.