

A DISTRIBUTION SHIFTS ON IMAGENET

ImageNet is a large-scale visual ontology of images built upon the backbone of the WordNet structure. ImageNet aims to populate the majority of the 80,000 synsets of WordNet with an average of 500–1000 clean and full resolution images, making it a roughly class-balanced, fully supervised dataset. [Deng et al. \(2009\)](#)

There now exist a wide range of distribution shifts on ImageNet. These are novel test datasets designed to overcome some of the limitations of the original benchmark. While they cannot remedy issues with the labeling scheme, these datasets do provide challenging new contexts in which to analyze classifier performance.

The four we focus on are Imagenet-Sketch, Imagenet-R, Imagenet-A, and Imagenet-V2.

ImageNet-V2 was designed to duplicate, as closely as possible, the original ImageNet test set. It was intended to answer the question of whether ImageNet-trained classifiers could successfully generalize even to the most mild of distribution shifts. [Recht et al. \(2019\)](#)

Imagenet-Sketch is a distribution shift covering sketches, paintings, drawings and illustrations of ImageNet classes. This test set is very large and comprehensive. [Wang et al. \(2019\)](#)

Imagenet-R is a 200-class subset of ImageNet-2012 focused on renditions of everyday objects, defined broadly as drawings, paintings, photographs of food art, etc. [Hendrycks et al. \(2021a\)](#)

Imagenet-A is a 200-class subset of ImageNet-2012 which was algorithmically selected – the natural distribution shift captured here is the set of ImageNet-class images which most often fool a RN50. This test is challenging, and tends to include a lot of images with challenges such as occlusion, changes in angle or position, and changes in brightness. [Hendrycks et al. \(2021b\)](#)

A.1 DIFFERENT SHIFTS RESPOND TO DIFFERENT INTERVENTIONS

While it is tempting to deal with distribution shifts as a kind of monolith, the truth is, perhaps unsurprisingly, more complex.

We found that ImageNet-V2 seemed to respond more to model architecture than other shifts, with the handful of non-ResNet models we evaluated outperforming nearly all other models, regardless of training objective.

ImageNet-R and ImageNet-Sketch both showed high sensitivity to the training data, with the CC12M and LAION-15m distributions considerably outperforming even the best YFCC-trained models. These types of shifts are particularly amenable to subset matching strategies. [8](#), [6](#)

On ImageNet-A, cross-entropy models significantly underperformed compared to VL models regardless of the data, and all models significantly underperformed compared to the ViT-L CLIP. [7](#)

We also note that there is no readily apparent logit-scaled linear trend in these distribution shifts when one considers models trained on a wide range of different datasets, underscoring the importance of a well-chosen baseline for comparison.

We find that different shifts tend to disadvantage different kinds of models, which makes improving on all of them simultaneously very challenging. The fact that ViT-L CLIP was able to do is both impressive and, given the vital importance of the underlying data distribution in such measures, a mystery which is unlikely to ever be solved. Even the massive public datasets such as LAION are unable to match the performance of the dataset CLIP was trained on, although other factors might possibly have played a role.

A standardized benchmark of distribution shifts on ImageNet would be a welcome contribution to this area of research.

B PRETRAINING DATASETS

Today, many SOTA models are pretrained on web-scale unsupervised data. We utilized three such datasets in our experiments. We observe that one major challenge of conducting research on unsupervised datasets is that the links provided as part of the dataset fail more and more over time, leading to each group getting a different version of the dataset. Therefore, to the extent possible, we report the details of each dataset in the appendix, and encourage other researchers working with these datasets to do the same.

CC-12M is a lightly supervised web-scale dataset created by Google. The image-caption pairs in CC-12M were filtered and selected for the purposes of training models to caption images. [Changpinyo et al. \(2021\)](#) Our version of CC12M contained 9703885 image-caption pairs.

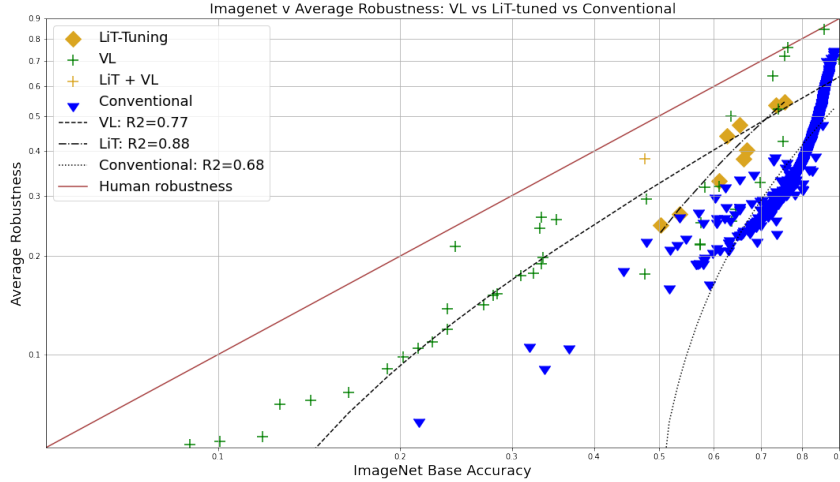


Figure 5: **The new picture of effective robustness.** This plot, which contains inference results from nearly 1000 models, shows a more complicated landscape of effective robustness than previous investigations. Ablations on the caption space show that even when the information in captions is aggressively transformed or reduced, robustness is preserved. In the low-accuracy regime, RN50s trained on integer-captioned yfcc and LAION data are able to match or exceed VL robustness. In the high accuracy regime, LiT-tuned VL models and wise-ft models approach, but do not reach, the VL-robust line. Cross-entropy loss models also approach the VL line at very high base accuracies.

YFCC-15M is a subset of YFCC-100M, which is 100M image-metadata pairs taken from Yahoo-Flickr in 2016. The subset was selected by OpenAI. This dataset contains images and metadata, which includes a "title" and a "description" field. These fields are combined and processed in various ways by researchers in order to generate captions for models to train on. Thomee et al. (2016) Our version of YFCC contained 14825134 image-caption pairs.

LAION is a 5B image-caption dataset recently created by LAION.ai. It is the first publicly available dataset which matches the scale of the datasets used by the large companies to train their best models. Schuhmann et al. (2021a) The subset of LAION we refer to as LAION-15m contained 13775512 image-caption pairs.

C LARGE-SCALE EVALUATION OF CROSS-ENTROPY VS VL ROBUSTNESS

In order to get a more complete picture of the current landscape of model robustness, we evaluated nearly 1000 models on our suite of distribution shifts, including all of the models with metrics reported by Taori et al. (2020); Wightman (2019); Feuer et al. (2022), models trained with LiT and Wise-FT objectives as described in Wortsman et al. (2021); Zhai et al. (2021) and all of the models trained for this paper. The results are shown in C. We find that in the very high accuracy regime, the logit-transformed linear fit of cross-entropy models fails to hold, and cross-entropy robustness increases faster than predicted, approaching VL robustness.

D COMPLETE TABLE OF MODEL RESULTS

For a complete table of results for all models trained and evaluated in the study, please see the supplemental materials.

Here, we include some abbreviated subsets of the results that are particularly useful.

D.1 SUBSET MATCHING AND VL COMPARISON, LOW AND MEDIUM ACCURACY REGIMES

D.2 CAPTIONNET COMPLETE RESULTS

E SUBSET MATCHING STRATEGIES

In this section, we define and describe the subset matching strategies we explored when preparing the experiments described in the main paper. All experiments fell into one of three main categories;

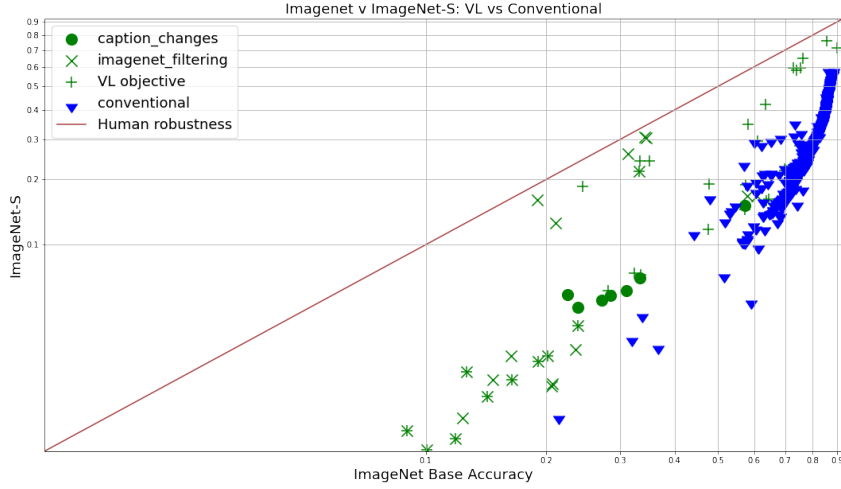


Figure 6: **Non-linearities in ImageNet-Sketch.** ImageNet-sketch performance is not linear, with only the very largest VL models showing a reliable improvement over conventionally trained models, when controlling for dataset size.

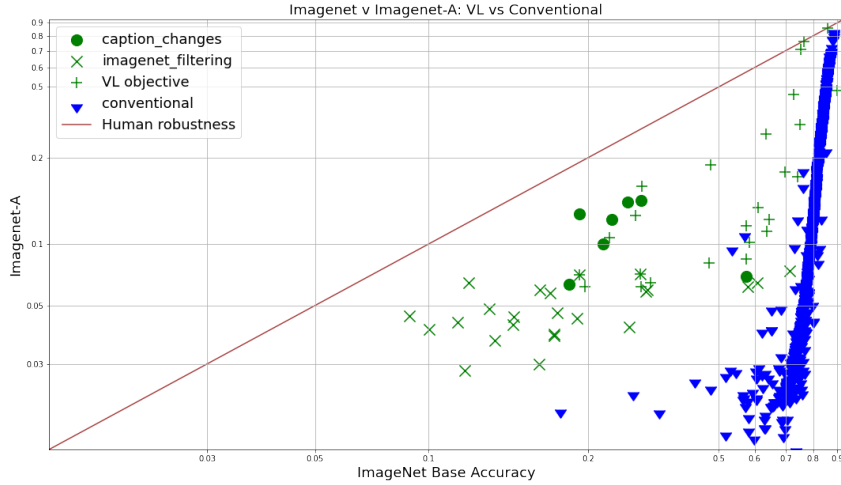


Figure 7: **ImageNet-A is learnable by all models at extremely high base accuracy.** Although VL models seem to learn ImageNet-A faster than conventional models, conventional models reach near-parity with VL models when base accuracy gets very high.

approx_samples	dataset	technique	matching terms	strategy	in-val	in-a	in-r	in-s	in-v2	avg-rob	eff-rob-rat
14.8m	yfcc	VL	None	Baseline	0.324	0.136	0.223	0.073	0.280	0.178	0.55
4m	yfcc	VL	None	Subsample	0.191	0.060	0.110	0.026	0.165	0.09	0.471
2m	yfcc	VL	None	Subsample	0.119	0.041	0.066	0.010	0.100	0.054	0.454
2.3m	yfcc	VL	default	Strict	0.143	0.048	0.097	0.017	0.123	0.071	0.497
2.3m	yfcc	Submat	default	Strict	0.124	0.023	0.059	0.013	0.107	0.051	0.411
3.9m	yfcc	VL	default	Multiclass	0.238	0.071	0.164	0.040	0.206	0.12	0.504
3.9m	yfcc	Submat	default	Multiclass	0.208	0.034	0.085	0.021	0.180	0.08	0.385
2.2m	yfcc	VL	ours	Strict	0.165	0.044	0.100	0.021	0.138	0.076	0.461
2.2m	yfcc	Submat	ours	Strict	0.148	0.033	0.074	0.021	0.130	0.065	0.439
3.1m	yfcc	VL	ours	Multiclass	0.202	0.058	0.127	0.028	0.180	0.098	0.485
3.1m	yfcc	Submat	ours	Multiclass	0.164	0.040	0.097	0.028	0.145	0.078	0.476
10m	yfcc	VL	NOT ours & default	Anticlass	0.127	0.065	0.071	0.023	0.118	0.069	0.543
13.7m	LAION	VL	None	Baseline	0.351	0.065	0.425	0.241	0.294	0.257	0.733
8m	LAION	Submat	openai	Strict	0.342	0.059	0.489	0.308	0.289	0.286	0.836
32m	LAION	Submat	openai	Strict	0.376	0.067	0.524	0.333	0.319	0.311	0.827
2.6m	LAION	VL	openai	Singleclass	0.332	0.072	0.398	0.217	0.281	0.242	0.73
2.6m	LAION	Submat	openai	Singleclass	0.313	0.039	0.299	0.259	0.268	0.216	0.69
9.7m	cc12m	VL	openai	Baseline	0.245	0.062	0.380	0.186	0.228	0.214	0.873
2.4m	cc12m	VL	openai	Singleclass	0.189	0.047	0.293	0.126	0.165	0.158	0.836
2.4m	cc12m	Submat	openai	Singleclass	0.211	0.046	0.324	0.127	0.182	0.17	0.806

Strict: Only match on samples which have exactly one class in them. Strict matching degrades as matching strategies grow more aggressive, and is also strongly affected by the number of classes evaluated; however, it tends to be the least noisy method, making it useful in some contexts.

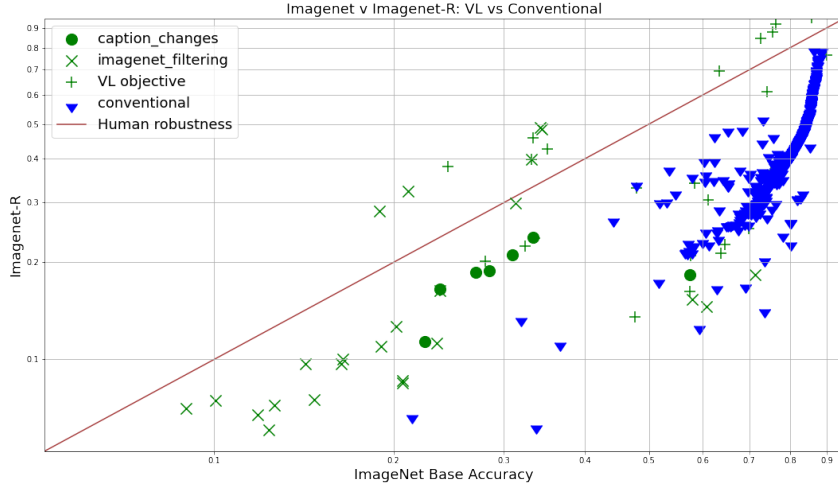


Figure 8: **VL performance on ImageNet-R outstrips base accuracy.** On ImageNet-R, which is a 200-class subset of ImageNet, VL models are able to achieve higher accuracy than on ImageNet itself. VL continues to outperform CE models on this dataset, even at very high accuracies.

name	dataset	technique	in-val	in-a	in-r	in-s	in-v2	avg-rob	eff-rob-rat
in100-vl	in100	VL	0.574	0.085	0.163	0.146	0.474	0.217	0.378
in100-sup	in100	Conventional	0.801	0.094	0.230	0.297	0.710	0.333	0.416
oi100-vl-bestcaps	oi100	VL	0.225	0.064	0.114	0.057	0.203	0.11	0.489
oi100-sup	oi100	Conventional	0.571	0.113	0.233	0.237	0.512	0.274	0.48
in100+oi100-vl	in100+oi100	VL	0.575	0.122	0.209	0.189	0.484	0.251	0.437
in100+oi100-sup	in100+oi100	Conventional	0.834	0.135	0.324	0.384	0.738	0.395	0.474
in100+laion100-vl	in100+laion100	VL	0.583	0.102	0.342	0.348	0.476	0.317	0.544
in100+laion100-submat	in100+laion100	Submat	0.811	0.129	0.418	0.478	0.725	0.438	0.54
in100+yfcc100+laion100-vl	in100+laion100+yfcc100	VL	0.620	0.152	0.301	0.291	0.544	0.322	0.519
in100+yfcc100+laion100-submat	in100+laion100+yfcc100	Submat	0.817	0.202	0.449	0.475	0.740	0.467	0.572
in100+yfcc22m-vl	in100+yfcc15m	VL	0.668	0.142	0.205	0.172	0.567	0.272	0.407
in100+yfcc22m-submat	in100+yfcc15m	Submat	0.553	0.047	0.114	0.119	0.443	0.181	0.327
in100+yfcc39m-vl	in100+yfcc15m	VL	0.698	0.216	0.252	0.220	0.620	0.327	0.468
in100+yfcc39m-submat	in100+yfcc15m	Submat	0.553	0.062	0.111	0.101	0.442	0.179	0.324
in1k-1.2m-in100	in1k	Conventional	0.736	0.017	0.144	0.187	0.605	0.238	0.323
in100+yfcc100-vl	in100+yfcc100	VL	0.637	0.115	0.213	0.160	0.526	0.254	0.399
in100+yfcc100-submat	in100+yfcc100	Submat	0.773	0.166	0.293	0.309	0.675	0.361	0.467
in100-submat-title	in100	Submat	0.579	0.062	0.154	0.168	0.470	0.214	0.37
in100-submat-ttd-size 256ep	in100	Submat	0.714	0.074	0.184	0.223	0.606	0.272	0.381
in100-submat-tags	in100	Submat	0.608	0.065	0.147	0.168	0.506	0.222	0.365
in100-submat-ttd	in100	Submat	0.651	0.065	0.179	0.194	0.537	0.244	0.375
in100-vl-blip	in100	VL	0.574	0.070	0.184	0.153	0.458	0.216	0.376
in100-tokenstrip	in100	VL	0.476	0.081	0.136	0.118	0.372	0.177	0.372
yfcc15m-in100	yfcc15m	VL	0.751	0.349	0.371	0.282	0.704	0.427	0.569
laion15m-in100	laion	VL	0.741	0.205	0.611	0.582	0.693	0.523	0.706
CLIP-WIT400m-in100	wit	VL	0.900	0.482	0.764	0.713	0.867	0.707	0.786
oi100-blipcaptop	oi100	VL	0.283	0.066	0.137	0.095	0.227	0.131	0.464
oi100-flickraption	oi100	VL	0.197	0.061	0.098	0.048	0.172	0.095	0.482
in100+oi100+laion100-vl	in100+laion100+oi100	VL	0.530	0.092	0.297	0.327	0.468	0.296	0.558
yfcc100-vl	yfcc100	VL	0.378	0.107	0.108	0.082	0.337	0.159	0.421
laion100-vl	laion100	VL	0.402	0.092	0.274	0.245	0.363	0.244	0.607
in100-cliplabels	in100	Conventional	0.778	0.099	0.225	0.286	0.660	0.318	0.409
in100+oi100+laion100-submat	in100+laion100+oi100	Submat	0.825	0.152	0.448	0.491	0.754	0.461	0.559

Single class: greedily take the first matching class as the true class and ignore all others. As a general matter, we found that single-class matching struck the best balance between dataset utilization and accuracy.

Multi class: Match on all matching classes, up to 25 classes per sample. We found that this approach tended to decrease accuracy, and that under most strategies, multiclass matches were uncommon anyway.

In addition to differences in matching strategy, we also experimented with different sets of terms for matching, which we refer to as in1k-openai (from [Radford et al. \(2021\)](#)), in1k-default (from Fox, E., and Guestrin, C. (n.d.). Coursera Machine Learning Specialization), and in1k-ours, created by us. We found that the heuristic changes to term matches often had substantial effect on accuracy; however, we leave the algorithmic discovery of optimal subset matching terms to future work.

E.1 DATASET UTILIZATION

We define dataset utilization as the number of correctly classified samples over all samples available to the model. We define accuracy as the number of correctly classified samples over correctly plus incorrectly classified samples.

Because ImageNet-100 and OpenImages-100 have ground truth labels, the optimal model would have a dataset utilization of 1.0 (100 percent).

E.2 PERFORMANCE OF SUBSET MATCHING STRATEGIES ON IMAGENET-100 AND OPENIMAGES-100

A detailed comparison of subset matching strategies indicates that there tends to be a trade-off between dataset utilization and accuracy. We describe our findings in detail below.

Name	TotalMatch	MC Acc	SC Acc	Strict Acc	MaxCorrS	MaxDU
in100-tags-ours	88597	0.76	0.86	0.92	76193	0.61
in100-tags-openai	83327	0.79	0.87	0.92	72494	0.58
in100-tags-openai-synset	99708	0.31	0.58	0.78	57831	0.47
in100-tags-default	93982	0.74	0.84	0.91	78945	0.63
in100-descr-ours	37685	0.7	0.8	0.83	30148	0.24
in100-descr-openai	35259	0.75	0.83	0.86	29265	0.24
in100-descr-openai-synset	51148	0.27	0.48	0.61	24551	0.2
in100-descr-default	40786	0.66	0.78	0.82	31813	0.26
in100-title-ours	69637	0.91	0.93	0.95	64762	0.52
in100-title-openai	64919	0.91	0.94	0.95	61024	0.49
in100-title-openai-synset	76299	0.39	0.76	0.87	57987	0.47
in100-title-default	74219	0.88	0.92	0.94	68281	0.55
in100-titletags-ours	103655	0.77	0.9	0.93	93290	0.75
in100-titletags-openai	97181	0.8	0.9	0.93	87463	0.7
in100-titletags-default	109192	0.75	0.88	0.92	96089	0.77
in100-ttd-ours	107559	0.73	0.89	0.92	95728	0.77
in100-ttd-openai	101210	0.77	0.89	0.92	90077	0.72
in100-ttd-default	113175	0.69	0.87	0.9	98462	0.79
oi100-tags-ours	19880	0.45	0.49	0.51	9741	0.07
oi100-tags-openai	17182	0.53	0.56	0.58	9622	0.07
oi100-tags-default	20371	0.51	0.56	0.58	11408	0.08
oi100-descr-ours	10286	0.32	0.37	0.35	3806	0.03
oi100-descr-openai	8600	0.4	0.44	0.42	3784	0.03
oi100-descr-default	12505	0.28	0.35	0.31	4377	0.03
oi100-title-ours	16437	0.54	0.56	0.51	9205	0.07
oi100-title-openai	14999	0.61	0.63	0.58	9449	0.07
oi100-title-default	18975	0.53	0.56	0.48	10626	0.08
oi100-title-openai-fuzzy	15492	0.56	0.61	0.59	9450	0.07
oi100-title-openai-synset	33685	0.24	0.29	0.3	9769	0.07
oi100-titletags-default	31440	0.47	0.53	0.52	10319	0.08
oi100-ttd-default	36968	0.4	0.48	0.46	11499	0.08

E.3 PERFORMANCE OF VARIATIONS ON SIMPLE SUBSET MATCHING

There are many ways one could conceivably improve on the simple strategies described in CITE ABOVE SECTION. We explore some of them and describe them here.

Fuzzy subset matching uses the Levenshtein distance between tokens to locate potential matches. CITATION. We used the standard Python library fuzzywuzzy with a setting of 55 (higher approaches found few matches). We found that this approach was considerably slower than simple subset matching, but offered only limited benefits.

We explore synset matching as well; we match on noun synsets only, with the NLTK toolkit. We include all synonym nouns, hyponyms, hypernyms, alsoeses and similartos, a broad matching strategy whose refinement we leave to future work. Synset matching also results in much slower subset matching, and also results in very small improvements; therefore, we use simple subset matching for the majority of our experiments.

E.4 PERFORMANCE COMPARISON OF SUBSET MATCHING STRATEGIES ON YFCC AND LAION

We train baseline 2M and 4M models on random subsamples of YFCC.

We then use subset matching strategies as described in CITE APPENDIX ABOVE.

We note that in1k-ours is more robust than in1k-default matching on YFCC, controlling for size.

We also try the opposite strategy – targeting the 10M samples in YFCC which were NOT matched. Even under this rather punishing transformation, we find that a VL model is able to learn a low-level representation of ImageNet (around 12 percent accuracy and an average of 8 percent under shift, around one third of a baseline YFCC model). The performance of a subset matched model under these circumstances would be zero, because it would not have any samples to train on.

Does subset matching favor certain datasets?

Subset matching strategies only reach parity with VL models when we approximately control for dataset utilization (the number of samples the model evaluates), indicating that this factor is important to consider when attempting to answer this question. Could it be that subset matching underperforms on non-robust datasets because it simply matches fewer samples?

This turns out not to be the case; we find that on YFCC and LAION, a subset match is found around 20 percent of the time, and on CC12M, around 25 percent of the time, resulting in roughly similar dataset utilization.

Single-class subset matching usually outperforms other filtering strategies

As shown earlier, subset matching strategy can impact both robustness and accuracy during training, and changes do not necessarily affect both measures in equal proportion.

We consider a range of variations on simple subset matching to see if they offer any improvement, such as fuzzy matching using Levenshtein distance and synset matching, but in our experiments, there were few if any advantages to these techniques.

We also experiment with changing the term-matching dictionary and the matching strategy; for more information on this, please refer to [E.2](#).

Overall, we find that single-class, non-strict matching on a relatively limited set of terms provides the best balance of accuracy, speed and dataset utilization, and that all subset matching strategies perform worse than annotated labels, even when controlling for dataset size.

Prefiltering samples improves caption quality

We find in [E.2](#) that the caption noise profiles of ImageNet-100 and OpenImages-100 differ dramatically.

- On ImageNet-100, simple subset matching on flickr-captions achieves high dataset utilization and accuracy
- On OpenImages-100, the same strategy shows lower accuracy and much lower dataset utilization

Since both datasets use the same caption source and the same image source, data alone is unlikely to explain this discrepancy.

Instead, we suspect the difference exists because of the strategies used to assemble these datasets.

OpenImages labelers applied labels to images which had not been selected with any particular objective in mind. [Kuznetsova et al. \(2020\)](#) ImageNet labelers applied labels to images which had been preselected with the intent of building a class-balanced dataset for 1000 classes chosen in advance. [Deng et al. \(2009\)](#)

We cannot verify the accuracy of subset-matching on YFCC-100 or LAION-100 because there are no human labels for these datasets. However, we know that LAION samples used a much stronger prefiltering strategy than YFCC. [Schuhmann et al. \(2021b\)](#); [Thomee et al. \(2016\)](#) Therefore, it is possible that subset matching’s strong performance on LAION can be attributed to this difference alone.

Subset matching hit rate is a good signal for subset matching accuracy

On OpenImages-100, we observed that as raw match count decrease dramatically, the accuracy of matched samples also decreased when matches took place. [E.2](#) This fact offers one possible guideline for when subset matching is better than VL; it performs better when there are a relatively high number of subset matches, possibly because hit rate correlates with accuracy.

The hit rate on YFCC-100 is around 2 percent, whereas the hit rate on LAION-100 is around 3.5 percent. This difference is certainly substantial enough that it could be a signal that subset matching will be a more successful approach.

Machine labels complement subset-matched labels

When classes are known in advance, machine labeling can be a good strategy for learning unsupervised image data. One advantage of machine labeling is that if labels are accurate, dataset utilization has the potential to be much higher.

In [4](#), we find that on ImageNet-100, 90 percent of the labels from a CLIP ViT-L model match the ground truth label. This is very comparable to that model’s zero-shot validation accuracy on ImageNet-val-subset. However, when we train a cross-entropy model on the CLIP labels, we find the machine labels outperform subset matching with size control, performing nearly as well as annotated labels.

The most likely explanation for this somewhat puzzling result is that the noise in subset-matched labels is more destructive to model learning than the noise in CLIP labels, perhaps because it is more diffuse and random. The question of what constitutes a ‘good’ error we leave to future work.

Machine labeling can also be useful for estimating the accuracy of subset matched labels. On LAION-100, we find that 60 percent of labels are in agreement between subset matching and labels from a CLIP ViT-L model.

We leave to future work the question of how models with a wider range of predefined classes would perform on such a task.

Overall, we find that machine labeling and caption supervision serve complementary roles; machine labeling does not directly rely on captions, but can still benefit from them. In particular, subset matching can prefilter incoming image data to target classes of interest.

The use cases for subset matching We find that simple subset matching is a powerful technique for learning on unsupervised image / caption data. We make the following recommendations for applying this technique;

1. Both flickr-style tags and descriptions and alt-text can be effective when used for subset matching
2. Subset matching techniques are much more effective when image-caption data has been roughly filtered or supervised, even if it has not actually been labeled; for instance, all images that match terms in a search engine, or all images that maximize the dot product of a clip model.
3. Subset matching relies on terms which are relatively common, and so works best with objects with unique names that happen to often be the subject of an image; English cocker spaniel, for instance
4. If the terms are uncommon or difficult to nail down, but the captions are still expected to contain them, then a VL model may perform better
5. When data is unfiltered, the hit rate of subset matching is a good barometer for how accurate the matches will be
6. If hit rate is low, so is accuracy, and VL will probably perform better than subset matching
7. Augmenting or ensembling with machine labels works well in conjunction with captioning; captions generated from machine labels can provide additional signal for VL and subset matching when captions are noisy

F DETAILS ON CAPTIONNET

CaptionNet Supervision Strategy In-Depth

In [F](#) we discuss in detail the supervision strategy used for CaptionNet, with a per-class breakdown of each class.

An overview of the supervision process follows;

- All samples were supervised by the authors of the paper
- Samples were sourced from flickr using the available API, sorted by ‘interesting’, with safesearch enabled, searching only samples with Creative Commons licenses
- Additional filtering terms were passed to the API in order to eliminate commonly encountered confounds in the search terms
- After the search term was selected, items were downloaded in bulk
- All downloaded samples were then individually tagged by the researchers as either “in-class” or “out-of-class”, using reference photographs from each class as a baseline comparison

We found that classes varied widely along several vectors;

- Some classes had far greater availability than others (ranging from 450,000 to 283 available samples)
- Some classes were much cleaner than others (ranging from 100 percent clean to around 25 percent)

- Some classes tended to be the ‘subject’ of photographs, such as dog breed, while others, such as mashed potato, tended to be featured as secondary items in the background of a photograph of something else

Dataset Construction The 100 classes in CaptionNet were selected randomly from a subset of all classes with more than 600 captions available in ImageNet-Captions (Fang et al., 2022). The list of classes selected is available in [\[1\]](#). We note that this approach introduces a potential bias in class selection, since it may be that captions were still available for those images ten years after ImageNet was originally constructed for some reason that correlates with properties we are interested in studying; however, we feel that the risk of this is outweighed by the many benefits of having such a dataset available for study.

in1k classname	search term	good samples	total samples	avail. samples	pct. good
lion	lion	962	1000	450000	0.96
wine bottle	wine bottle	925	1000	29500	0.93
book shop	bookstore	816	984	83000	0.83
parking meter	parking meter	377	1000	9500	0.38
african elephant	african elephant	885	1000	44000	0.89
bagel	bagel	699	988	20500	0.71
tarantula	tarantula	667	981	9000	0.68
ice cream	ice cream	741	984	154500	0.75
fig	fig	517	1000	46000	0.52
shoe shop	shopping shoes	425	1000	13000	0.43
french bulldog	french bulldog	887	996	7500	0.89
hen	hen	412	1000	73000	0.41
guacamole	guacamole	683	998	6500	0.68
broccoli	broccoli	679	997	19000	0.68
howler monkey	howler monkey	817	847	9000	0.96
scuba diver	scuba diver	827	1000	15000	0.83
spindle	”spindle wool, spindle -wool thread”	311	867	867	0.36
lhasa	lhasa dog	719	1000	2500	0.72
traffic light	stoplight	622	991	5500	0.63
lionfish	lionfish	552	897	6500	0.62
popsicle	popsicle -animal -sticks -animals -insect -insects -icicle -garden -sticks -icicles -gardens -toes -label -labels	638	943	7500	0.68
lampshade	lampshade	446	807	6500	0.55
spiderweb	spiderweb -spiderman -halloween -pumpkin -butterfly -pleiades -nebula -stars	832	996	17500	0.84
lifeboat	lifeboat	572	1000	13000	0.57
cucumber	cucumber -sea -spider -beetle -flower -spiral	730	999	26500	0.73
english springer	english springer spaniel	772	993	3500	0.78
macaw	macaw	972	1000	13500	0.97
mailbox	mailbox	900	1000	36500	0.9
peacock	peacock -butterfly	966	999	72000	0.97
bee	bumblebee OR wasp OR hornet -jet -airplane -helicopter -navy -aircraft -comic -RIAT -military -Helicopter -Helicopters -helicopters -aviation -Hudson -car -basketball -sports -Transformers -cosplay -disfrazado -costume -transformer AND flower	686	761	110000	0.9

dungeness crab	dungeness AND crab -restaurant -breakfast -lunch -dinner -shack -creels -traps -cannery	474	1000	1500	0.47
banana	banana -plant -blossom -flower -seed -seedlings -tree -spider -leaf -abstract -bay -band -festival -doll -sexy -sexiest -bread -soup -puree -smoothie -car -plantation -cake -cream -monkey -pudding -zoo -republic -boxes -buying -selling -vendor -bridge -scone -moon	793	995	65000	0.8
corn	corncob	354	1000	1000	0.35
lemon	lemon -plant -blossom -flower -seed -seedlings -tree -spider -leaf -abstract -bay -band -festival -doll -sexy -sexiest -bread -soup -puree -smoothie -car -plantation -scent -fresh -cleaner -butterfly -grove -shots -car -sunrise -paint -graffiti -origami -cake -cream -pudding -boxes -buying -selling -vendor -bridge -scone -don -lime	693	1000	65000	0.69
marimba	marimba instrument	127	283	283	0.45
orange	orange food fruit -plant -blossom -flower -seed -seedlings -tree -spider -leaf -abstract -bay -band -festival -doll -sexy -sexiest -bread -soup -puree -smoothie -car -plantation -cake -cream -monkey -pudding -zoo -republic -boxes -buying -selling -vendor -bridge -scone -moon -cupcake -cake -sales -seller -pancakes -crepes -crep -crepe -pancake -cookie -flavored -juice -soda -pop -beach -island -cove -grove -street -drive -tea -curd -marmalade -bars -cabs -chicken -cheesecake -pie -milk	744	1000	4000	0.74
bell pepper	bell pepper vegetable -plant -blossom -flower -seed -seedlings -tree -spider -leaf -abstract -bay -band -festival -doll -sexy -sexiest -bread -soup -puree -smoothie -car -plantation -cake -cream -monkey -pudding -zoo -republic -boxes -buying -selling -vendor -bridge -scone -moon -cupcake -cake -sales -seller -pancakes -crepes -crep -crepe -pancake -cookie -flavored -juice -soda -pop -beach -island -cove -grove -street -drive -tea -curd -marmalade -bars -cabs -chicken -cheesecake -pie -milk -market -spice	392	505	505	0.78
espresso	espresso coffee -maker -machine -beans -building -exterior -window	828	1000	22000	0.83
mashed potato	mashed potato	635	996	10000	0.64

stingray	stingray water -dolphin -shark -cruise -boat -scuba -fish	600	983	2000	0.61
flagpole	flagpole -lighthouse -church -bank -station	614	991	7000	0.62
teapot	teapot -tea -flower -tower -building -dome -art -fashion -vase -store -stores -shop -shops -Sagittarius -project365 -fountain -candle -mug -teacup -keg -vessel -amphora -urn -coffeepot	660	997	10500	0.66
umbrella	umbrella	911	1000	126000	0.91
beer bottle	beer bottle -house -door -brewery -glass -cap	909	1003	19000	0.91
barn	barn -swallow -owl -bird	980	1000	115000	0.98
christmas stocking	christmas stocking fireplace	317	779	779	0.41
magpie	magpie -screenshots -moth -butterfly -coprinopsis -thieving -mushroom	736	983	25500	0.75
mitten	mitten glove	800	995	1500	0.8
ram	ram sheep -Church -window -Window -church -school -dance -parade -festival -celebration -festivities -community -fair -ewe -fox -lamb -bird -cat -dog -Dodge	742	1000	3000	0.74
warthog	warthog animal -zebra -cheetah -leopard -giraffe -gazelle -hippo -rhino -donkey -armadillo -elephant -crocodile -lion -leopard -impala -cat -monkey -bird	946	997	2500	0.95
goose	geese	474	500	69000	0.95
bubble	soap bubble -dancer -dance -fairy -tree -leaf -leaves -flowers -water -toy -art -abstract -museum -dog -cat -butterfly -food -wine -beer -chocolate -Chocolate	414	500	5000	0.83
cougar	cougar animal -warthog -mascot -zebra -cheetah -leopard -giraffe -gazelle -hippo -rhino -donkey -armadillo -elephant -crocodile -lion -leopard -impala -cat -monkey -bird -lake -Lake -river -River -blonde -Blonde -woman -girl -milf -bear -cliff -Cliffs -cliffs -military -wallaby -horse -jet -print	297	500	1000	0.59
daisy	daisy flower	500	500	52000	1
menu	menu	431	500	92000	0.86
bald eagle	bald eagle	475	500	33500	0.95
necklace	necklace jewelry -brooch -pendant -creation -earring -earrings -bracelet -ring -Engraver -bauble -anklet	478	500	12500	0.96
chickadee	chickadee bird -Goldfinch -goldfinch -robin -thrush -jay -cardinal -woodpecker -wren -hawk -raven -titmouse -nuthatch	494	500	9000	0.99
stone wall	""""stone wall""""	424	500	32000	0.85
flamingo	flamingo bird	476	500	38500	0.95
gas pump	gas station	348	500	41000	0.7

vulture	vulture bird -hawk -crow -eagle	489	500	15500	0.98
pizza	""""pizza pie"""" -Fest -festival - summit -experience -party -band -moon -parade -Parade -harvard - mosaic -montage"	305	500	1000	0.61
wallaby	wallaby -warthog -mascot -zebra - cheetah -leopard -giraffe -gazelle -hippo -rhino -donkey -armadillo - elephant -crocodile -lion -leopard -impala -cat -monkey -bird -koala -sports -kangaroo -soccer -football - food -church -hills -stadium -tribute -grass -rugby -apartment -car	369	500	10000	0.74
hay	haystack field -hole -trail -poster - sign	360	500	1000	0.72
grand piano	"kawai grand piano, steinway grand piano"	312	455	455	0.69
laptop	laptop	443	500	98000	0.89
dishwasher	dishwasher appliance	191	268	268	0.71
cricket	cricket -batting -sports -team - match	337	500	44000	0.67
sea slug	nudibranch	468	500	12500	0.94
mongoose	mongoose -bike -bicycle -park -tree -joe -rocket -military -airplane -toy -car	379	500	5000	0.76
siamese cat	siamese cat -bangkok -flower - snake -campaign -wat -costume - cosplay -festival	416	500	13000	0.83
freight car	freight car	491	500	70500	0.98
vending machine	""""vending machine""""	411	500	13000	0.82
bottlecap	bottlecap -tab	448	500	3500	0.9
acorn	acorn -woodpecker -fairy -squirrel -weevil -travel -squash -street	352	500	25000	0.7
feather boa	feather boa	135	500	2000	0.27
macaque	macaque	485	500	14500	0.97
bolete	boletus	444	500	3500	0.89
border terrier	""""border terrier""""	422	500	1500	0.84
barbell	barbells	352	500	1000	0.7
fly	housefly	398	500	1500	0.8
suspension bridge	suspension bridge	432	500	33500	0.86
jellyfish	jellyfish	477	500	46500	0.95
barbershop	barbershop -quartet -singers	430	500	9000	0.86
koala	koala	458	500	32500	0.92
bannister	bannister staircase	174	183	183	0.95
pillow	pillow -talk -fight -cat -dog -moss -sky -cloud -sky	420	500	34500	0.84
bib	baby bib -shower -food	406	500	1500	0.81
juncto	juncto bird -finch -sparrow -thrush -cardinal -woodpecker -jay	475	500	7000	0.95
chainlink fence	chainlink fence	375	500	3500	0.75
soccer ball	""""soccer ball"""" -match -game - milky -beach -Lewes"	349	500	2500	0.7
stupa	stupa	418	500	23500	0.84

quail	quail bird -finch -sparrow -thrush -cardinal -woodpecker -jay - partridge -rabbit -hawk -avocet -deer -dog -wolf -coyote -gopher -eagle -vole -molerat -butterfly	396	500	11000	0.79
padlock	padlock	378	500	9500	0.76
great white shark	""""great white shark""""	309	500	2000	0.62
totem pole	""""totem pole"""" wood"	383	500	1000	0.77
ant	ant insect	447	500	18000	0.89
bison	bison	429	500	41500	0.86
greenhouse	greenhouse	407	500	82000	0.81

Adding BLIP Captions to CaptionNet

Since we could not find human-authored captions for ImageNet, we used BLIP [Li et al. \(2022\)](#) to generate descriptive captions on ImageNet-100. BLIP often uses word fragments to describe objects, so we used a spell checker as a simple intervention to improve the quality of BLIP captions. Finally, because BLIP’s vocabulary does not include many of the specialized classes available in ImageNet, we augmented the BLIP captions with Flickr image titles, the form of text which is most commonly available for an image. We used top p=0.9, max length=40, min length=5, repetition penalty=1.1.

We repeated the process for OpenImages-100. However, we used human-authored captions sourced from [Pont-Tuset et al. \(2020\)](#) instead of BLIP whenever available; around 16,000 out of the 135,000 OpenImages-100 samples had human-authored captions.

G CLIP DOES NOT LEARN A LANGUAGE MODEL

We show that, evaluated by the conventional definition of a language model in the literature, as well as by our expectations as human beings, CLIP’s text tower does not learn a language model.

[Fang et al. \(2022\)](#) noted that on most natural distribution shifts, models trained with language information from a captioned subset of ImageNet follow the same trend as models trained without it, with neither coming close to the robustness of VL models.

We conducted a range of experiments in [G](#) to verify this.

1. Zeroshot and fully trained scrambling: we randomly scrambled the word order of captions and trained a model for the full training duration. We saw only a 1% loss in zero-shot accuracy when the model was trained in this fashion, and no loss at all when we scrambled the word order of the prompts used to generate the text embeddings. This finding shows that despite the existence of a positional encoder, CLIP’s text tower is invariant w.r.t. word position on its input captions.
2. We train a "simple captions" model of yfcc, with "An image of a CLASSNAME", where the class name is all the wordnet-recognized nouns and adjectives in the captions, and see very little change in effective robustness. This indicates that the model pays very little attention to verbs, adverbs and non-wordnet tokens, at least for the purposes of ImageNet zero-shot accuracy.
3. We train a "simpler captions" model on YFCC, captioned "An image of a CLASSNAME", where the class name is a single version of an ImageNet-1k classname. We find that accuracy decreases, but even under this highly destructive transformation, robustness remains on the line.
4. We test prompt ensembling by eliminating all but one prompt during inference when computing CLIP’s predictions. We try scrambling this prompt and leaving it unscrambled. We find very little change in accuracy and none in robustness. This indicates that prompt selection and prompting methods may improve accuracy, but do not affect robustness.
5. We fully trained a language model with a shift cipher applied to all of the letters in the captions. While this has a significant effect on accuracy (a drop of 8%), effective robustness holds nearly constant. This shows that CLIP’s tokenizer aids accuracy but offers no robustness benefit (since a shift cipher destroys nearly all word-wise tokenization and forces the model to tokenize one letter at a time) and that the model is not more robust because of

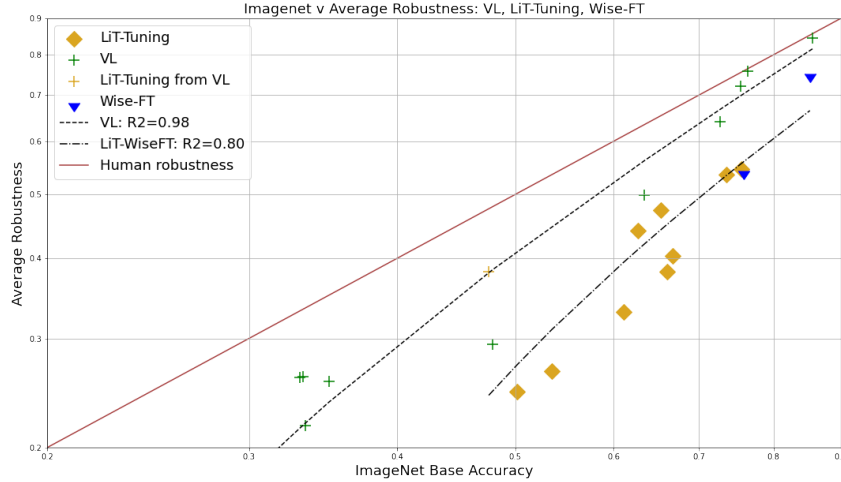


Figure 9: **Wise-FT, optimized to balance id/ood accuracy, fits the LiT-tuned robustness line.**

some special understanding of the text data itself, such as learning the frequency distribution of letters or tokens as a way of guessing captions.

Throughout our experiments, we find that interventions that affect the integrity of the caption space do have impacts on the overall accuracy of the model, they do not change the robustness trend line, indicating that the changes in robustness cannot be attributed to the properties of natural language which we ablate, such as the tokenizer, sentence-wise attention mechanisms, prompt ensembling or even class-independent caption content. [G](#)

name	dataset	in-val	in-a	in-r	in-s	in-v2	avg-rob	eff-rob-rat
Baseline	yfcc15m	0.324	0.136	0.2232	0.073	0.2804	0.178	0.55
Simple Captions	yfcc15m	0.31	0.1576	0.21	0.06	0.2737	0.175	0.566
Simple Captions & Filtering	yfcc15m	0.101	0.038	0.0736	0.008	0.0898	0.052	0.521
Simpler Captions & Filtering	yfcc15m	0.09	0.0441	0.0694	0.011	0.0804	0.051	0.573
Shift Cipher	yfcc15m	0.238	0.139	0.1657	0.049	0.2016	0.139	0.584
No Ensembling	yfcc15m	0.284	0.131	0.188	0.057	0.2394	0.154	0.541
Token Order Scrambled	yfcc15m	0.333	0.16	0.237	0.069	0.295	0.19	0.571

H ALTERNATE TRAINING SCHEMES

One fairly immediate explanation for the robustness of VL models would be that we are witnessing a kind of overfitting which happens whenever a model is fine-tuned on a training dataset. If this is the case, then if we had a method for reverting the overfitting, we would be able to shift models to a different line with respect to effective robustness.

In [H](#), we explore this possibility, focusing on two alternative training schemes in particular, Wise-FT and LiT-Tuning.

[Radford et al. \(2021\)](#) ran fine-tuning experiments on certain datasets and found that robustness declined as base accuracy increased, indicating that fine-tuning makes CLIP models (somewhat) less robust, fitting a middle line in-between VL and conventional models.

[Wortsman et al. \(2022\)](#) ran a series of experiments interpolating the weights of zero-shot CLIP with its fine-tuned counterparts and showed that for certain distribution shifts, it is possible to find a ‘sweet spot’ where both i.d. and o.o.d. accuracy increase. We test the wise-ft method on ViT-L and find that under this intervention, robustness does not scale proportionately with accuracy, instead holding essentially constant as base accuracy increases.

We also evaluated the pre-trained LiT-tuned models released by Zhai et al, and trained several LiT-tuned models ourselves in order to compare their performance to fully-trained VL models, extending the results in Beyer et al. Our findings are as follows –

1. Like Wise-FT, LiT-tuning produces models whose i.d. / o.o.d accuracy trade-off fits a line between that of traditional models and VL models – more robust than the former, less robust than the latter. The only exception we found was when we LiT-tuned the vision tower of a

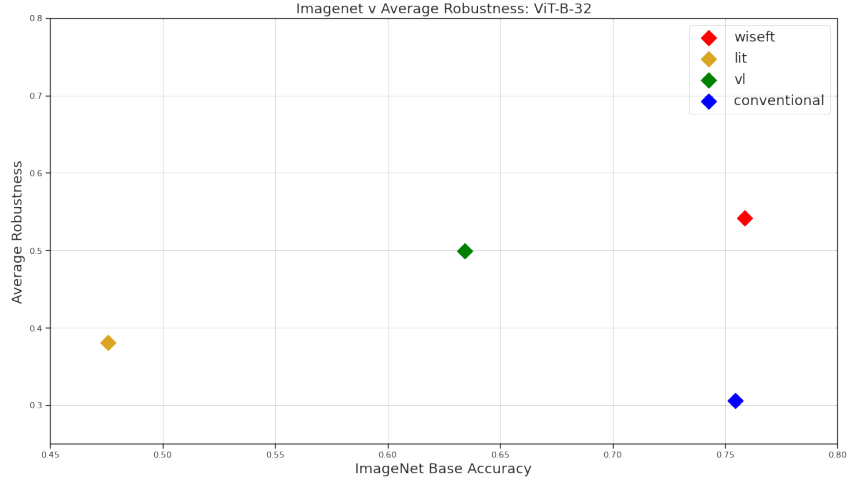


Figure 10: **LiT-tuning on a VL-trained image tower reduces accuracy without altering robustness, suggesting that VL pretraining is at least as robust as LiT-tuning.** *Wise-FT tuning greatly increases base accuracy and slightly improves robustness, at the cost of zero-shot capability. Conventional from-scratch training matches Wise-FT accuracy, but sacrifices robustness and zero-shot.*

ViT trained on the CLIP objective – in this case, LiT-tuning decreased base accuracy while holding effective robustness constant (the near-opposite effect of Wise-FT)

- LiT-tuning offers negative benefit for fully trained VL models, suggesting that it can only hope to approach, rather than exceed, the accuracy of its baselines
- LiT-tuning performance tends to closely correlate to the base accuracy of the underlying vision model
- Intriguingly, we find that this is true regardless of the specific dataset used for LiT-tuning – LiT-tuned models trained on small amounts of data are able to recover accuracy on out-of-distribution tasks even when very little data from that distribution shift appears in the pretraining data
- These experiments suggest that some degree of effective robustness is "locked away" in many vision models, but is lost during the training process, but that certain techniques are able to increase effective robustness disproportionate to the loss in base accuracy, pushing the model 'above the line' we would normally expect. Furthermore, if the distribution shift of interest is known and well-defined, it is possible to select a tuning to optimize for that shift

Taken together, we can conclude that effective robustness cannot be explained by overfitting alone – if it were, then we would expect interpolation-type interventions to be capable of lifting models to the effective robustness line, which they do not.

I CLASSNAMES USED FOR SUBSET MATCHING

We provide the classnames used for subset matching as a supplemental attachment.

J CLASS FREQUENCY COUNTS FOR IN100 SUPERVISED DISTRIBUTIONS

J.1 IMAGENET-100

'african bush elephant': 2082, 'mailbox': 1846, 'peafowl': 1844, 'macaw': 1689, 'barn': 1689, 'wine bottle': 1648, 'lion': 1643, 'umbrella': 1608, 'mitten': 1595, 'warthog': 1565, 'french bulldog': 1550, 'spider web': 1544, 'beer bottle': 1531, 'scuba diver': 1514, 'bookstore': 1493, 'ice cream': 1475, 'traffic light': 1468, 'koala': 1456, 'espresso': 1450, 'banana': 1444, 'magpie': 1439, 'bottle cap': 1437, 'howler monkey': 1428, 'orange': 1425, 'teapot': 1414, 'english springer spaniel': 1407, 'lhasa apso': 1404, 'bagel': 1401, 'tarantula': 1378, 'ram adult male sheep': 1377, 'lemon': 1370, 'cucumber': 1360, 'bee': 1355, 'popsicle': 1351, 'broccoli': 1339, 'guacamole': 1335, 'fig': 1323, 'menu': 1291, 'mashed potatoes': 1283, 'stingray': 1270, 'flagpole': 1268, 'chickadee': 1253,

'lionfish': 1221, 'goose': 1220, 'junco': 1216, 'daisy': 1198, 'freight car': 1196, 'lifeboat': 1194, 'hay': 1193, 'bubble': 1173, 'dungeness crab': 1168, 'sea slug': 1165, 'vulture': 1158, 'suspension bridge': 1152, 'laptop computer': 1146, 'chain link fence': 1142, 'bald eagle': 1142, 'flamingo': 1137, 'padlock': 1133, 'jellyfish': 1128, 'stupa': 1125, 'ant': 1119, 'parking meter': 1115, 'wallaby': 1114, 'fly': 1108, 'border terrier': 1107, 'greenhouse': 1104, 'necklace': 1100, 'macaque': 1095, 'stone wall': 1093, 'barbell': 1093, 'lampshade': 1087, 'siamese cat': 1085, 'barbershop': 1072, 'pillow': 1071, 'spindle': 1069, 'bolete': 1067, 'vending machine': 1066, 'totem pole': 1054, 'bison': 1049, 'quail': 1049, 'hen': 1045, 'shoe store': 1045, 'christmas stocking': 1043, 'baby bib': 1034, 'mongoose': 1026, 'bell pepper': 1022, 'cricket insect': 1018, 'acorn': 1015, 'gas pump': 1009, 'soccer ball': 1005, 'great white shark': 1002, 'pizza': 983, 'corn': 969, 'grand piano': 962, 'cougar': 920, 'marimba': 914, 'dishwasher': 810, 'baluster handrail': 804, 'feather boa': 759

J.2 OPENIMAGES-100

'laptop computer': 55813, 'menu': 13159, 'grand piano': 10836, 'bee': 3688, 'umbrella': 3575, 'goose': 3477, 'wine bottle': 2318, 'necklace': 2224, 'traffic light': 1915, 'pizza': 1699, 'ice cream': 1618, 'hen': 1499, 'lion': 1300, 'pillow': 1205, 'banana': 1119, 'siamese cat': 1058, 'orange': 1035, 'lemon': 961, 'jellyfish': 797, 'stone wall': 718, 'baluster handrail': 697, 'broccoli': 614, 'cucumber': 609, 'greenhouse': 597, 'chain link fence': 594, 'cougar': 593, 'macaque': 588, 'vulture': 560, 'scuba diver': 559, 'beer bottle': 550, 'teapot': 549, 'french bulldog': 546, 'espresso': 528, 'soccer ball': 527, 'lampshade': 491, 'fly': 491, 'macaw': 488, 'peafowl': 487, 'stupa': 484, 'suspension bridge': 476, 'ant': 475, 'flamingo': 462, 'barn': 458, 'bell pepper': 458, 'padlock': 434, 'african bush elephant': 432, 'spider web': 421, 'feather boa': 416, 'koala': 415, 'mongoose': 409, 'bottle cap': 388, 'freight car': 387, 'vending machine': 370, 'bison': 363, 'bald eagle': 351, 'hay': 340, 'bubble': 339, 'daisy': 339, 'totem pole': 335, 'stingray': 333, 'chickadee': 322, 'mailbox': 315, 'barbell': 313, 'bagel': 308, 'bookstore': 307, 'gas pump': 303, 'wallaby': 300, 'marimba': 294, 'spindle': 290, 'bolete': 262, 'popsicle': 258, 'shoe store': 255, 'guacamole': 250, 'barbershop': 248, 'lionfish': 234, 'parking meter': 234, 'lhasa apso': 218, 'christmas stocking': 211, 'mashed potatoes': 193, 'great white shark': 186, 'magpie': 186, 'warthog': 170, 'sea slug': 166, 'cricket insect': 156, 'lifeboat': 141, 'baby bib': 138, 'junco': 138, 'dishwasher': 113, 'mitten': 101, 'acorn': 98, 'quail': 96, 'tarantula': 89, 'howler monkey': 66, 'english springer spaniel': 61, 'fig': 54, 'ram adult male sheep': 54, 'dungeness crab': 51, 'border terrier': 49

K CLASS FREQUENCY COUNTS FOR IN100 SUBSET MATCHING DISTRIBUTIONS, OPENAI LABELS, MC MATCHING

K.1 IMAGENET-100

'orange': 1820, 'lion': 1788, 'barn': 1695, 'macaw': 1684, 'umbrella': 1583, 'banana': 1500, 'mitten': 1500, 'warthog': 1488, 'magpie': 1438, 'lemon': 1437, 'koala': 1435, 'espresso': 1400, 'bagel': 1376, 'howler monkey': 1337, 'tarantula': 1331, 'broccoli': 1299, 'fig': 1295, 'ice cream': 1285, 'cucumber': 1272, 'goose': 1231, 'daisy': 1224, 'junco': 1207, 'chickadee': 1193, 'teapot': 1175, 'french bulldog': 1166, 'vulture': 1150, 'stingray': 1142, 'guacamole': 1134, 'flamingo': 1126, 'lifeboat': 1120, 'ant': 1114, 'suspension bridge': 1109, 'greenhouse': 1100, 'lhasa apso': 1093, 'wallaby': 1073, 'stupa': 1073, 'bald eagle': 1063, 'lionfish': 1057, 'fly': 1055, 'english springer spaniel': 1051, 'necklace': 1048, 'bison': 1047, 'barbell': 1042, 'mailbox': 1041, 'quail': 1037, 'macaque': 1032, 'padlock': 1026, 'hen': 1024, 'pizza': 995, 'pillow': 995, 'acorn': 993, 'vending machine': 976, 'bottle cap': 969, 'stone wall': 968, 'popsicle': 955, 'spider web': 949, 'totem pole': 934, 'spindle': 920, 'bookstore': 903, 'bubble': 893, 'border terrier': 889, 'mongoose': 888, 'corn': 874, 'parking meter': 866, 'flagpole': 864, 'dungeness crab': 862, 'marimba': 862, 'peafowl': 848, 'bee': 840, 'bell pepper': 821, 'menu': 758, 'wine bottle': 734, 'great white shark': 733, 'jellyfish': 703, 'dishwasher': 701, 'soccer ball': 700, 'beer bottle': 663, 'grand piano': 600, 'bolete': 576, 'hay': 547, 'gas pump': 541, 'christmas stocking': 534, 'traffic light': 479, 'cougar': 471, 'scuba diver': 470, 'feather boa': 435, 'african bush elephant': 408, 'siamese cat': 358, 'lampshade': 352, 'barbershop': 349, 'baby bib': 258, 'freight car': 119, 'laptop computer': 46, 'sea slug': 37, 'shoe store': 32, 'cricket insect': 19, 'baluster handrail': 2

K.2 YFCC-100

'grand piano': 62610, 'orange': 37182, 'fly': 30889, 'lion': 16043, 'bee': 14164, 'pizza': 12084, 'barn': 11854, 'goose': 11200, 'ice cream': 10556, 'greenhouse': 9479, 'menu': 7463, 'umbrella': 7164, 'banana': 6933, 'bubble': 6838, 'corn': 6835, 'cougar': 6619, 'lemon': 6439, 'daisy': 5386, 'scuba

diver': 5044, 'cricket insect': 4702, 'laptop computer': 4601, 'ant': 4465, 'hay': 4427, 'peafowl': 4140, 'pillow': 4137, 'flamingo': 3735, 'bookstore': 3668, 'necklace': 3201, 'bald eagle': 2912, 'ram adult male sheep': 2617, 'jellyfish': 2482, 'vulture': 2462, 'suspension bridge': 2439, 'espresso': 2189, 'mailbox': 2125, 'bison': 2073, 'flagpole': 2012, 'fig': 1973, 'hen': 1896, 'cucumber': 1815, 'bagel': 1746, 'koala': 1592, 'magpie': 1366, 'stone wall': 1337, 'spider web': 1296, 'acorn': 1277, 'popsicle': 1226, 'baluster handrail': 1182, 'vending machine': 1118, 'broccoli': 1114, 'junco': 1113, 'quail': 1108, 'stupa': 1043, 'feather boa': 1018, 'stingray': 971, 'macaw': 961, 'wallaby': 942, 'sea slug': 832, 'chickadee': 783, 'lifeboat': 781, 'baby bib': 774, 'mitten': 748, 'teapot': 728, 'macaque': 661, 'traffic light': 638, 'mashed potatoes': 625, 'african bush elephant': 600, 'tarantula': 593, 'barbershop': 537, 'gas pump': 520, 'padlock': 517, 'beer bottle': 433, 'warthog': 430, 'mongoose': 407, 'siamese cat': 395, 'guacamole': 393, 'parking meter': 381, 'spindle': 379, 'wine bottle': 370, 'dishwasher': 361, 'lampshade': 358, 'lhasa apso': 356, 'howler monkey': 314, 'lionfish': 296, 'shoe store': 285, 'soccer ball': 260, 'marimba': 168, 'freight car': 114, 'great white shark': 104, 'christmas stocking': 100, 'dungeness crab': 97, 'french bulldog': 94, 'bottle cap': 85, 'bolete': 80, 'chain link fence': 51, 'barbell': 23, 'english springer spaniel': 22, 'border terrier': 19

K.3 LAION-100

'spindle': 68186, 'necklace': 59079, 'orange': 52221, 'pillow': 41020, 'laptop computer': 23319, 'lion': 14246, 'lemon': 12769, 'ram adult male sheep': 11550, 'bubble': 11079, 'barn': 9909, 'pizza': 9597, 'daisy': 8331, 'umbrella': 8323, 'banana': 7690, 'corn': 7140, 'menu': 6788, 'cougar': 6714, 'ice cream': 6539, 'cricket insect': 5696, 'peafowl': 4662, 'espresso': 4227, 'flamingo': 4029, 'goose': 3532, 'soccer ball': 3532, 'barbershop': 2963, 'dishwasher': 2853, 'bald eagle': 2678, 'fig': 2635, 'greenhouse': 2460, 'broccoli': 2348, 'teapot': 2298, 'acorn': 2164, 'cucumber': 2053, 'hay': 2023, 'wine bottle': 1824, 'scuba diver': 1818, 'bison': 1736, 'lampshade': 1497, 'mitten': 1457, 'french bulldog': 1435, 'stone wall': 1402, 'koala': 1394, 'bee': 1296, 'mailbox': 1199, 'padlock': 1126, 'stingray': 1115, 'bookstore': 1069, 'spider web': 976, 'macaw': 964, 'barbell': 913, 'christmas stocking': 887, 'traffic light': 825, 'vending machine': 808, 'popsicle': 780, 'quail': 768, 'chickadee': 744, 'bagel': 714, 'baluster handrail': 713, 'jellyfish': 706, 'bottle cap': 648, 'beer bottle': 603, 'flagpole': 589, 'bell pepper': 553, 'grand piano': 544, 'guacamole': 520, 'magpie': 481, 'suspension bridge': 477, 'african bush elephant': 459, 'baby bib': 451, 'wallaby': 423, 'stupa': 399, 'macaque': 350, 'gas pump': 335, 'great white shark': 333, 'mongoose': 308, 'junco': 302, 'siamese cat': 291, 'marimba': 289, 'hen': 272, 'tarantula': 257, 'lifeboat': 236, 'lionfish': 205, 'totem pole': 199, 'english springer spaniel': 192, 'warthog': 186, 'shoe store': 166, 'border terrier': 145, 'vulture': 118, 'feather boa': 116, 'lhasa apso': 105, 'sea slug': 90, 'howler monkey': 85, 'fly': 83, 'parking meter': 54, 'freight car': 50, 'ant': 44, 'dungeness crab': 36, 'chain link fence': 33, 'bolete': 14

L PER CLASS ACCURACY FOR SUBSET MATCHING, OPENAI CLASSNAMES, SC

L.1 IMAGENET-100

'macaw': 0.81, 'barn': 0.9, 'umbrella': 0.85, 'lion': 0.92, 'mitten': 0.89, 'warthog': 0.9, 'magpie': 0.87, 'koala': 0.88, 'banana': 0.88, 'espresso': 0.89, 'bagel': 0.88, 'howler monkey': 0.87, 'tarantula': 0.87, 'orange': 0.86, 'lemon': 0.87, 'fig': 0.87, 'broccoli': 0.85, 'cucumber': 0.84, 'ice cream': 0.84, 'junco': 0.83, 'goose': 0.83, 'chickadee': 0.83, 'teapot': 0.82, 'daisy': 0.82, 'french bulldog': 0.82, 'vulture': 0.82, 'stingray': 0.81, 'guacamole': 0.81, 'flamingo': 0.81, 'lifeboat': 0.81, 'suspension bridge': 0.81, 'greenhouse': 0.8, 'lhasa apso': 0.81, 'ant': 0.8, 'stupa': 0.8, 'wallaby': 0.8, 'bald eagle': 0.8, 'lionfish': 0.82, 'english springer spaniel': 0.82, 'bison': 0.82, 'barbell': 0.82, 'macaque': 0.82, 'mailbox': 0.85, 'necklace': 0.84, 'quail': 0.84, 'padlock': 0.85, 'hen': 0.84, 'acorn': 0.83, 'pillow': 0.82, 'fly': 0.83, 'vending machine': 0.83, 'stone wall': 0.83, 'bottle cap': 0.83, 'popsicle': 0.83, 'spider web': 0.82, 'totem pole': 0.82, 'pizza': 0.82, 'spindle': 0.81, 'bookstore': 0.79, 'mongoose': 0.79, 'border terrier': 0.78, 'parking meter': 0.77, 'marimba': 0.77, 'flagpole': 0.77, 'dungeness crab': 0.78, 'peafowl': 0.77, 'bubble': 0.74, 'bell pepper': 0.74, 'bee': 0.7, 'corn': 0.68, 'menu': 0.68, 'great white shark': 0.67, 'wine bottle': 0.67, 'dishwasher': 0.65, 'soccer ball': 0.65, 'jellyfish': 0.65, 'beer bottle': 0.59, 'grand piano': 0.56, 'bolete': 0.55, 'gas pump': 0.52, 'christmas stocking': 0.51, 'hay': 0.48, 'traffic light': 0.45, 'scuba diver': 0.45, 'cougar': 0.45, 'feather boa': 0.42, 'african bush elephant': 0.4, 'siamese cat': 0.35, 'lampshade': 0.35, 'barbershop': 0.35, 'baby bib': 0.26, 'freight car': 0.11, 'laptop computer': 0.05, 'sea slug': 0.04, 'shoe store': 0.03, 'cricket insect': 0.02, 'baluster handrail': 0.0

L.2 OPENIMAGES-100

'bee': 0.03, 'pizza': 0.08, 'goose': 0.09, 'menu': 0.23, 'lion': 0.21, 'banana': 0.14, 'umbrella': 0.21, 'jellyfish': 0.21, 'ice cream': 0.24, 'orange': 0.21, 'ant': 0.2, 'koala': 0.2, 'necklace': 0.22, 'flamingo': 0.24, 'vulture': 0.25, 'fly': 0.24, 'lemon': 0.21, 'wine bottle': 0.22, 'broccoli': 0.26, 'bison': 0.29, 'barn': 0.27, 'bald eagle': 0.3, 'hen': 0.27, 'stupa': 0.27, 'spider web': 0.24, 'pillow': 0.24, 'padlock': 0.23, 'macaw': 0.24, 'totem pole': 0.23, 'traffic light': 0.23, 'laptop computer': 0.23, 'bubble': 0.23, 'chickadee': 0.23, 'cucumber': 0.24, 'daisy': 0.24, 'warthog': 0.24, 'parking meter': 0.24, 'teapot': 0.24, 'junco': 0.22, 'spindle': 0.22, 'lionfish': 0.22, 'bagel': 0.22, 'cougar': 0.22, 'french bulldog': 0.21, 'mailbox': 0.19, 'hay': 0.19, 'stingray': 0.19, 'magpie': 0.19, 'wallaby': 0.19, 'vending machine': 0.19, 'macaque': 0.19, 'greenhouse': 0.19, 'espresso': 0.18, 'quail': 0.17, 'bottle cap': 0.17, 'grand piano': 0.16, 'acorn': 0.15, 'siamese cat': 0.14, 'guacamole': 0.13, 'gas pump': 0.13, 'mitten': 0.12, 'bell pepper': 0.12, 'fig': 0.12, 'bookstore': 0.11, 'barbershop': 0.11, 'lifeboat': 0.11, 'peafowl': 0.11, 'great white shark': 0.11, 'mongoose': 0.11, 'suspension bridge': 0.11, 'tarantula': 0.11, 'marimba': 0.11, 'dishwasher': 0.11, 'stone wall': 0.1, 'christmas stocking': 0.09, 'bolete': 0.09, 'lhasa apso': 0.09, 'soccer ball': 0.09, 'beer bottle': 0.1, 'border terrier': 0.09, 'howler monkey': 0.09, 'lampshade': 0.09, 'african bush elephant': 0.05, 'scuba diver': 0.06, 'mashed potatoes': 0.05, 'english springer spaniel': 0.04, 'cricket insect': 0.04, 'feather boa': 0.04, 'dungeness crab': 0.05, 'shoe store': 0.04, 'freight car': 0.04, 'barbell': 0.04, 'baby bib': 0.03, 'sea slug': 0.03