CHiLS: ZERO-SHOT IMAGE CLASSIFICATION WITH HIERARCHICAL LABEL SETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Open vocabulary models (e.g. CLIP) have shown strong performance on zero-shot classification through their ability generate embeddings for each class based on their (natural language) names. Prior work has focused on improving the accuracy of these models through prompt engineering or by incorporating a small amount of labeled downstream data (via finetuning). In this paper, we propose **C**lassification with **Hi**erarchical **L**abel **S**ets (or CHiLS), an alternative strategy that proceeds in three steps: (i) for each class, produce a set of subclasses, using either existing label hierarchies or by querying GPT-3; (ii) perform the standard zero-shot CLIP procedure as though these subclasses were the labels of interest; (iii) map the predicted subclass back to its parent to produce the final prediction. Across numerous datasets, CHiLS leads to improved accuracy yielding gains of over 30% in situations where known hierarchies are available and more modest gains when they are not. CHiLS is simple to implement within existing CLIP pipelines and requires no additional training cost.

1 INTRODUCTION

Recently, machine learning researchers have become captivated by the remarkable capabilities of pretrained *open vocabulary models* (Radford et al., 2021; Wortsman et al., 2021; Jia et al., 2021; Gao et al., 2021; Pham et al., 2021; Cho et al., 2022; Pratt et al., 2022). These models, like CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021), learn to map images and captions into shared embedding spaces such that images are close in embedding space to their corresponding captions but far from randomly sampled captions. The resulting models can then used to assess the relative compatibility of a given image with an arbitrary set of textual "prompts". Notably, Radford et al. (2021) observed that by inserting each class name directly within a natural language prompt, one can then use CLIP embeddings to assess the compatibility of an images with each among the possible classes. Thus, open vocabulary models are able to perform zero-shot image classification, and do so with high rates of success (Radford et al., 2021; Zhang et al., 2021b).

Despite the documented successes, the current interest in open vocabulary models poses a new question: **How should we represent our classes for a given problem in natural language?** As class names are now part of the inferential pipeline (as opposed to mostly an afterthought in traditional scenarios) for models like CLIP in the zero-shot setting, CLIP's performance is now directly tied to the descriptiveness of the class "prompts" (Santurkar et al., 2022). While many researchers have focused on improving the quality of the prompts into which class names are embedded (Radford et al., 2021; Pratt et al., 2022; Zhou et al., 2022b;a; Huang et al., 2022), surprisingly little attention has been paid to improving the *richness of the class names themselves*. This can be particularly crucial in cases where class names are not very informative or are too broad to sound match the sort of descriptions that might arise in natural captions. Consider, for an example, the class "large man-made outdoor things" in the CIFAR20 dataset (Krizhevsky, 2009)).

In this paper, we introduce a new method for Zero-Shot prediction with CLIP models, which we refer to as **C**lassification with **Hi**erarchical **L**abel **S**ets (CHiLS for short). Our method utilizes a hierarchical map to convert each class into a list of subclasses, performs normal CLIP zero-shot prediction across the union set of all *subclasses*, and finally uses the inverse mapping to convert the subclass prediction to the requisite superclass. We additionally include a reweighting step wherein

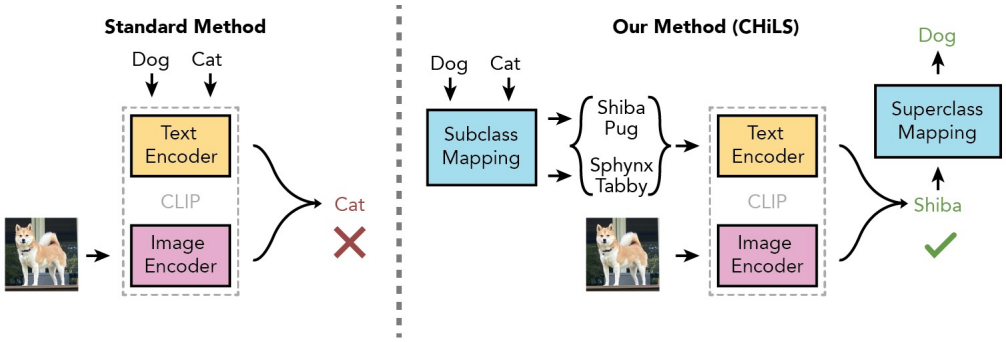


Figure 1: **(Left)** *Standard CLIP Pipeline for Zero-Shot Classification*. For inference, a standard CLIP takes in input a set of classes and an image where we want to make a prediction and makes a prediction from that set of classes. **(Right)** *Our proposed method CHiLS for leveraging hierarchical class information into the zero-shot pipeline*. We map each individual class to a set of subclasses, perform inferences in the subclass space (i.e., union set of all subclasses), and map the predicted subclass back to its original superclass.

we leverage the raw superclass probabilities in order to make our method robust to less-confident predictions at the superclass and subclass level.

We evaluate CHiLS on a wide array of image classification benchmarks with *and* without available hierarchical information. In the former case, leveraging preexisting hierarchies leads to strong accuracy gains across all datasets. In the latter, we show that rather than enumerating the hierarchy by hand, using GPT-3 to query a list of *possible* subclasses for each class (whether or not they are actualy present in the dataset) still leads to consistent improved accuracy over raw superclass prediction. We summarize our main contributions below:

- We propose CHiLS, a new method for improving zero-shot CLIP performance, which requires no labeled data or training time and is flexible to both existing and synthetically generated hierarchies.
- We show that CHiLS consistently performs as well or better than standard practices in situations with only synthetic hierarchies, and that CHiLS can achieve up to 30% accuracy gains when ground truth hierarchies are available.

2 Related Work

2.1 Transfer Learning

While the focus of this paper is to improve CLIP models in the zero-shot regime, there is a large body of work exploring improvements to CLIP's few-shot capabilities. In the standard fine-tuning paradigm for CLIP models, practitioners discard the text encoder and only use the image embeddings as inputs for some additional training layers. This however, leads to certain problems.

Wortsman et al. (2021) and Ding et al. (2022) focus on robustness issues that arise when fine-tuning CLIP models, specifically through the lenses of distribution shift and catastrophic forgetting respectively. Wortsman et al. (2021) proposes to linear interpolate the weights of a fine-tuned and a zero-shot CLIP model, which succeeds in making fine-tuned CLIP models robust under standard distribution shifts. In Ding et al. (2022), they propose to fine-tune both the image encoder and the text encoder, where the latter draws from a replay vocabulary of text concepts from the original CLIP database.

There is another line of work leveraging the adapter framework (Houlsby et al., 2019) from parameter-efficient learning; specifically, in Ding et al. (2022) they fine-tune a small number of additional weights on top of the encoder blocks, which is then connected with the original embeddings through residual connections. Zhang et al. (2021a) builds on this method by removing the need for additional training and simply uses a cached model.

Additionally, some have looked at circumventing the entire process of prompt engineering. Zhou et al. (2022a) and Zhou et al. (2022b) tackle this by treating the tokens within each prompt as learnable vectors, which are then optimized within only a few images per class. Huang et al. (2022) echoes these works, but instead does not utilize any labeled data and learns the prompt representations in an unsupervised manner. In all the above situations, *some* amount of data, whether labeled or not, is used in order to improve the predictive accuracy of the CLIP model.

2.2 Zero-Shot Prediction

The field of Zero-Shot Learning (ZSL) has existed well before the emergence of open vocabularly models, with its inception traced to Larochelle et al. (2008). With regards to non-CLIP related methods, the ZSL paradigm has shown success in improving multilingual question answering (Kuo & Chen, 2022) with large language models (LLMs), and also in image classification tasks where wikipedia-like context is used in order to perform the classification without access to the training labels (Bujwid & Sullivan, 2021; Shen et al., 2022).

With CLIP models, ZSL success has been found in a variety of tasks. Namely, Zhang et al. (2021b) expands the CLIP 2D paradigm for 3D point clouds. Tewel et al. (2021) shows that CLIP models can be retrofitted to perform the reverse task of image-to-text generation, with a strong ability for semantic arithmetic in the image domain. Both Yu et al. (2022) and Cho et al. (2022) expand CLIP's zero-shot abilities through techniques drawn from reinforcement learning, with the former using CLIP for the task of audio captioning. Zeng et al. (2022) in particular show the capabilities of composing CLIP-like models and LLMs together to extend the zero-shot capabilities to novel tasks like assitive dialogue and open-ended reasoning. Unlike our work here, these prior directions mostly focus on generative problems or, in the case of Bujwid & Sullivan (2021) and Shen et al. (2022), require rich external knowledge databases to employ their methods.

In the realm of improving CLIP's zero-shot capabilities for image classification, we particularly note the contemporary work of Pratt et al. (2022). Here, authors explore using GPT-3 to generate rich textual prompts for each class rather than using preexisting prompt templates, and show improvements in zero-shot accuracy across a variety of image classification baselines. In another work, Ren et al. (2022) proposes leveraging preexisting captions in order to improve performance, though this is restricted to querying the pre-training set of captions. In contrast, our work explores a complementary direction of leveraging hierarchy in class names to improve zero-shot performance of CLIP. with a fixed set of preexisting prompt templates.

3 Proposed Methods

In this paper, we are primarily concerned with the problem of zero-shot image classification in CLIP models. For CLIP models, zero-shot classification involves using both a pretrained image encoder and a pretrained text encoder (see the left part of Figure 1). To perform a zero-shot classification, we need a predefined set of classes written in natural language. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$ be such a set. Given an image and set of classes, each class is embedded within a natural language prompt (through some function $\mathrm{T}(\cdot)$) to produce a "caption" for each class (e.g. one standard prompt mentioned in Radford et al. (2021) is *"A photo of a* $\{\}$*"*). These prompts are then fed into the text encoder and after passing the image through the image encoder, we calculate the cosine similarity between the image embedding and each class-prompt embedding. These similarity scores form the output "logits" of the CLIP model, which can be passed through a softmax to generate the class probabilities.

As noted in Section 2, previous work has focused on improving the $\mathrm{T}(\cdot)$ for each class label c_i. With CHiLS, we instead focus on the complementary task of directly modifying the set of classes $\mathcal{C}$, while keeping $\mathrm{T}(\cdot)$ fixed. At a high level, our method involves into two main steps: (1) using hierarchical information to perform inference across *subclasses*, and (2) leveraging raw superclass probabilities to combine the best of subclass and superclass prediction probabilities.

3.1 Zero-Shot Prediction with Hierarchical Label Sets

Our method CHiLS slightly modifies the standard approach for zero-shot CLIP prediction. As each class label c_i represents some concept in natural language (e.g. the label "dog"), we acquire a

Algorithm 1 **C**lassification with **Hi**erarchical **L**abel **S**ets (CHiLS)

input : data point $\boldsymbol{x}$, class labels $\mathcal{C}$, prompt function T, label set mapping G, CLIP model f

1: Set $\mathcal{C}_{\text{sub}} \leftarrow \cup_{c_i \in \mathcal{C}} G(c_i)$ ▷ `Union of subclasses for subclass prediction`
2: $\widehat{\boldsymbol{y}}_{\text{sub}} = \sigma(f(\boldsymbol{x}, \text{T}(\mathcal{C}_{\text{sub}})))$ ▷ `Subclass probabilities`
3: $\widehat{\boldsymbol{y}}_{\text{sup}} = \sigma(f(\boldsymbol{x}, \text{T}(\mathcal{C})))$ ▷ `Superclass probabilities`
4: **for** $i = 1$ to $|\mathcal{C}|$ **do**
5: $\quad S_{c_i} = G(c_i)$
6: $\quad$ **for** $s_{c_i,j} \in S_{c_i}$ **do**
7: $\quad\quad \widehat{\boldsymbol{y}}_{\text{sub}}[s_{c_i,j}] = \widehat{\boldsymbol{y}}_{\text{sub}}[s_{c_i,j}] * \widehat{\boldsymbol{y}}_{\text{sup}}[c_i]$
▷ `Combining subclass and superclass prediction probability`
8: $\quad$ **end for**
9: **end for**

output : $G^{-1}(\arg\max \widehat{\boldsymbol{y}}_{\text{sub}})$

subclass set $\mathcal{S}_{c_i} = \{s_{c_i,1}, s_{c_i,2}, \ldots, s_{c_i,m_i}\}$ through some mapping function G, where each $s_{c_i,j}$ is a linguistic *hyponym*, or subclass, of c_i (e.g. corgi for dogs) and m_i is the size of the set S_{c_i}.

Given a label set S_{c_i} for each class, we proceed with the standard process for zero-shot prediction, but now using the *union* of all label sets as the set of classes. Through this, CHiLS will now produce its guess for the most likely *subclass*. We then leverage the inverse mapping function G^{-1}[1] to coarse-grain our prediction back into the corresponding superclass. Our method is detailed more formally in Algorithm 1.

In our work, we experiment with two scenarios: (i) when hierarchy information is available and can be readily queried; and (ii) when hierarchy information is *not* available and the label set for each class must be generated, which we do so by prompting GPT-3.

3.2 Reweighting probabilities with Superclass Confidence

While the above method is able to effectively utilize CLIP's ability to identify relatively fine-grained concepts, by predicting on only subclass labels we lose any positive benefits of the superclass label, and performance may vary widely based on the quality of the subclass labels. Given recent evidence (Kadavath et al., 2022) that large language models (like the text encoder in CLIP) generally predict correct labels with *high* probability, we modify our initial algorithm to leverage this behavior and utilize *both* superclass and subclass information. We provide empirical evidence of this property in Appendix A.

Specifically, we include an additional reweighting step within our main algorithm (see lines 4-9 in Algorithm 1). Here, we reweight each set of subclass probabilities by its superclass probability. In this way, we can attempt to avoid the behavior in which CLIP makes an incorrect subclass prediction despite a confident and correct superclass prediction, and thus bias our model to never do *worse* than the raw superclass predictions.

4 Experiments

In this section, we first lay out the experimental set-up. We then discuss, in order, the efficacy of our proposed method in situations with available class hierarchy information and in situations *without* any preexisting hierarchy. After these main results, we present a series of ablations over various design choices showing where our method is robust and what might be crucial for its performance.

4.1 Setup

Datasets. We test our method on the following image benchmarks: the four BREEDS imagenet subsets (living17, nonliving26, entity13, and entity30) (Santurkar et al., 2021), CIFAR20 (the

[1]We use G^{-1} to refer to an inverse mapping from a subclass to superclass, i.e., G^{-1} takes a subclass and maps it back to the superclass.

Dataset	Superclass Accuracy	CHiLS Accuracy (Existing Map)	CHiLS Accuracy (GPT-3 Map)
Nonliving26	79.82	90.67 (+10.85)	81.51 (+1.69)
Living17	91.08	93.80 (+2.72)	91.43 (+0.35)
Entity13	77.46	92.59 (+15.13)	78.11 (+0.65)
Entity30	70.32	88.87 (+18.55)	71.75 (+1.43)
CIFAR20	59.54	85.30 (+25.76)	65.90 (+6.36)
Food-101	91.73	N/A	91.56 (−0.17)
Fruits-360	58.71	61.41 (+2.70)	61.38 (+2.67)
Fashion1M	42.24	N/A	46.01 (+3.77)
Fashion-MNIST	68.49	N/A	70.84 (+2.35)
LSUN-Scene	88.20	N/A	88.97 (+0.77)
Office31	87.27	N/A	88.37 (+1.10)
OfficeHome	88.85	N/A	88.76 (−0.09)
ObjectNet	53.10	85.34 (+32.24)	53.52 (+0.42)

Table 1: Zero-shot accuracy performance across image benchmarks with superclass labels (baseline), CHiLS with existing hierarchy (whenever available), and CHiLS with GPT-3 generated hierarchy. CHiLS improves classification accuracy in all situations with given label sets and all but 2 datasets with GPT-3 generated label sets.

coarse-label version of CIFAR100) (Krizhevsky, 2009), Food-101 (Bossard et al., 2014), Fruits-360 (Mureşan & Oltean, 2018), Fashion1M (Xiao et al., 2015), Fashion-MNIST (Xiao et al., 2017), LSUN-Scene (Yu et al., 2015), Office31 (Saenko et al., 2010), OfficeHome (Venkateswara et al., 2017), and ObjectNet (Barbu et al., 2019). These datasets constitute a wide range of different image domains and include datasets with and without available hierarchy information. We also examine CHiLS's robustness to distribution shift within a dataset by averaging all results for the BREEDS datasets, Office31, and OfficeHome across different shifts (see Appendix C for more information). We additionally modify the Fruits-360 and ObjectNet datasets to create existing taxonomies. In the former we group classes together based on major fruit species (e.g. apples) and fix some ill-formed subclass labels. In the latter we use only the subset of ObjectNet that overlaps with ImageNet and utilize the resulting hierarchy to construct a coarser version of this subset. More details for dataset preparation are detailed in Appendix C.

Model Architecture. Unless otherwise specified, we use the ViTL/14@336px backbone (Radford et al., 2021) for our CLIP model, and used DaVinci-002 (with temperature fixed at 0.7) for all ablations involving GPT-3. For the choice of the prompt embedding function $\mathrm{T}(\cdot)$, we follow the procedure laid out for ImageNet in Radford et al. (2021) by averaging the text embeddings of 75 different prompts for each class label.

Choice of Mapping Function G. In our experiments, we primarily look at how the choice of the mapping function G influences the performance of CHiLS. In Section 4.2, we focus on the datasets with available hierarchy information. Here, G and G^{-1} are simply table lookups to find the list of subclasses and corresponding superclass respectively. In Section 4.3, we explore situations in which the true set of subclasses in each superclass is unknown. In these scenarios, we use GPT-3 to generate our mapping function G. Specifically, given some label set size m and superclass name `class-name`, we query GPT-3 with the prompt:

`Generate a list of` **`m`** `types of the following:` **`class-name`**

The resulting output list from GPT-3 thus defines our mapping G from superclass to subclass. Unless otherwise specified, we fix $m = 10$ for all datasets. Additionally, in 4.4 we explore situations in which hierarchical information is present but noisy, i.e. the label set for each superclass contains the subclasses present *and* subclasses not present in the data.

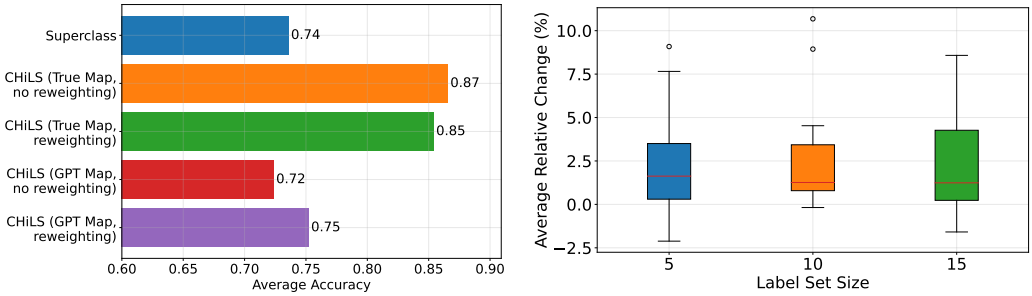


Figure 2: (Left) Average accuracy across datasets for the baseline superclass predictions, our CHiLS method, and CHiLS *without* the reweighting step. While when given the true hierarchy omitting the reweighting step can slightly boost performance beyond CHiLS, in situations without the true hierarchy the reweighting step is crucial to improving on the baseline accuracy. (Right) Average relative accuracy change from the baseline to CHiLS (across all datasets), for varying label set sizes. In all, there is not much difference in performance across label set sizes.

4.2 Leveraging Available Hierarchy Information.

We first concern ourselves with the scenario in which there is hierarchy information already available (or readily accessible) for a given dataset. In this situation, the set of subclasses for each superclass is exactly specified and correct (i.e. every image within each superclass falls into one of the subclasses). We emphasize that here we do *not* need information about which example belongs to which subclass, we just need a mapping of superclass to subclass. For example, each class in the BREEDS dataset living17 is made up of 4–8 ImageNet subclasses at finer granularity (e.g. "parrot" includes "african grey" and "macaw").

Results In Table 1, we can see that our method performs better than using the baseline superclass labels alone across all 7 of the datasets with available hierarchy information, in some cases leading to +15% improvement in predictive accuracy.

4.3 CHiLS in Unknown Hierarchy Settings

Though we have seen considerable success in situations with access to the true hierarchical structure, in some real-world settings our dataset may not include any available information about the subclasses within each class. In this scenario, we turn to using GPT-3 to approximate the hierarchical map G (as specified in Section 4.1). It is important to note that GPT-3 may sometimes output suboptimal label sets, most notably in situations where GPT-3 chooses the wrong wordsense or when GPT-3 only lists modifiers on the original superclass (e.g. producing the list `[red, yellow, green]` for types of apples). In order to account for these issues in an out-of-the-box fashion, we automatically append the superclass name (if not already present) to each generated subclass label, and also include the superclass itself within the label set.

Results In this setting, our method is still able to beat the baseline performance in 11 out of 13 datasets, albeit with lower accuracy gains generally (see Table 1). Thus, while knowing the true subclass hierarchy can lead to large accuracy gains, it is enough to simply enumerate a list of possible subclasses for each class with no prior information about the dataset in order to improve the predictive accuracy.

4.4 Ablations

Is Reweighting Necessary? Though the reweighting step in CHiLS is motivated by the evidence that CLIP generally assigns higher probability to *correct* predictions rather than incorrect ones (see Appendix A for empirical verification), it is not immediately clear whether it is truely necessary. Averaged across all documented datasets, in Figure 2 (left) we show that in the true hierarchy setting, not reweighting the subclass probabilities can actually slightly *boost* performance (as the label sets are adequately tuned to the distribution of images). However, in situations where the true hierarchy

Dataset	Superclass Accuracy	CHiLS Accuracy (Existing Map)	CHiLS Accuracy (Existing Map + Noise)
nonliving26	79.82	90.67 (+10.85)	89.48 (+9.66)
living17	91.08	93.80 (+2.72)	92.47 (+1.39)
entity13	77.46	92.59 (+15.13)	90.34 (+12.88)
entity30	70.32	88.87 (+18.55)	87.56 (+17.24)

Table 2: CHiLS zero-shot accuracy when G includes *all* subclasses in the ImageNet hierarchy descended from the respective root node. Even in the presence of noise added to the true label sets, CHiLS is able to make large accuracy gains.

is not present, omitting the reweighting step puts accuracy below the baseline performance. Thus, as the latter situation is considerably more likely in the wild, the reweighting step is imperative to utilizing CHiLS to its fullest.

Noisy Available Hierarchies While the situation described in Section 4.3 is the most probable in practice, we additionally investigate the situation in which the hierarchical information is present but *overestimates* the set of subclasses, where each label set for a class includes the in-distribution subclasses and subclasses not present in the dataset. To do this, we return to the BREEDS datasets presented in Santurkar et al. (2021). As the BREEDS datasets were created so that each class contains the same number of subclasses (which are ImageNet classes), we modify G such that the label set for each superclass corresponds to *all* the ImageNet classes descended from that node in the hierarchy. As we can see in Table 2, CHiLS is able to improve upon the baseline performance even in the presence of added noise in each label set.

Label Set Size In previous works investigating importance of prompts in CLIP's performance, it has been documented that the number of prompts used can have a decent effect on the overall performance (Pratt et al., 2022; Santurkar et al., 2022). Along this line, we investigate how the size of the *subclass set* generated for each class effects the overall accuracy by re-running our main experiments with varying values of m (namely, 5, 10, and 15). In Figure 2 (right), there is overall not much variation across label set sizes that is consistent over all datasets. In all, we observe that the optimal label set size is context-specific, and depends upon the total number of classes present and the semantic granularity of the classes themselves. Individual dataset results are available in Appendix B.

Model Size In order to examine whether the performance of CHiLS only exists within the best performing CLIP backbone (e.g. ViT-L/14@336), we measure the average relative change in accuracy performance between CHiLS and the baseline superclass predictions across all datasets for an array of different CLIP models. Namely, we investigate the RN50, RN101, RN50x4, ViT-B/16, ViT-B/32, and ViT-L/14@336 CLIP backbones (see Radford et al. (2021) for more information on the model specifications). In Figure 3, we show that across the 6 specified CLIP backbones, CHiLS performance leads to relatively consistent relative accuracy gains, which shows that CHiLS's benefits are not an artifact of large model size.

Alternative Ensembling Methods While the core of CHiLS is based on a *set-based* ensembling approach for subclasses and a linear averaging approach for prompt templates (based on Radford et al. (2021)'s procedure for ImageNet), we experimented with alternative ensembling methods for different parts of the CHiLS pipeline. Namely, we replaced the set-based mapping for subclasses with a simple linear average (i.e. every class embedding is the linear average of subclass embeddings), and changed the ensembling method for prompts into a set-based method. Note in the latter case we only experiment with how this effects raw *superclass* prediction (where each class maps to a set of 75 different prompt embeddings), as using set-based ensembling for *both* prompts and subclasses within CHiLS quickly becomes computationally expensive. In 4, we see that using our initial ensembling methods (i.e. linear averaging for prompts and set mappings for subclasses) achieves greater accuracy on average.

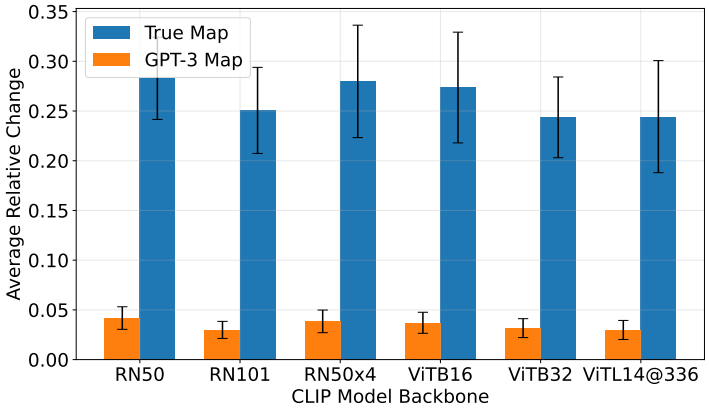


Figure 3: Average relative change between CHiLS and baseline for true mapping and GPT-3 generated mapping. Across changes in CLIP backbone size and structure, the effectiveness of CHiLS at improving performance only varies slightly.

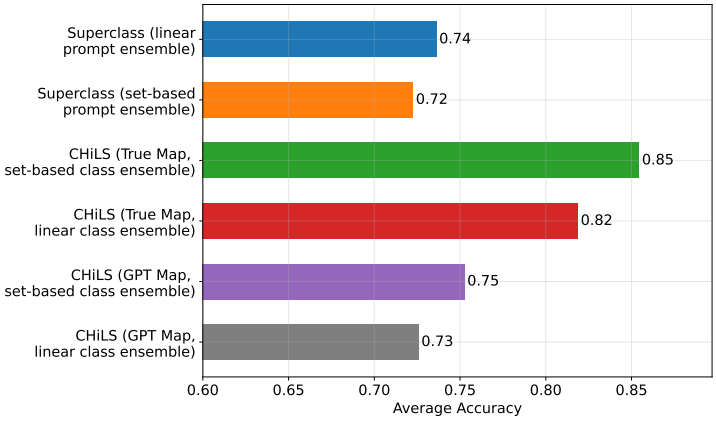


Figure 4: Average accuracy across datasets for varying ensembling methods on both the prompt and subclass steps of the zero-shot pipeline. In general, linear averaging for subclasses performs worse than our proposed set-based method, while linear averaging for prompts (for raw superclass prediction) performs better thant using a set-based mapping.

5 Conclusion

In this work, we demonstrated that the zero-shot image classification capabilities of CLIP models can be improved by leveraging hierarchical information for a given set of classes. When hierarchical structure is available in a given dataset, our method shows large improvements in zero-shot accuracy, and even when subclass information *isn't* explicitly present, we showed that we can leverage a LLM (e.g., GPT-3) to generate subclasses for each class and still improve upon the baseline (superclass) accuracy.

We remark that CHiLS may be quite beneficial to practitioners using CLIP as an out-of-the-box image classifier. Namely, we show that even without existing hierarchical data accuracy can be improved with a *fully automated* pipeline (via querying GPT-3), yet CHiLS is flexible enough that any degree of hand-crafting label sets can be worked into the zero-shot pipeline. Our method has the added benefit of being both *completely zero-shot* (i.e. no training or fine-tuning necessary) and is resource efficient.

Limitations and Future Work As with usual zero-shot learning, we don't have a way to validate the performance of our method. Given CHiLS's empirical successes, we hope to perform more

investigation to develop an understanding of *why* CHiLS is able to improve zero-shot accuracy and whether there is a more principled way of reconciling superclass and subclass predictions.

REPRODUCIBLITY STATEMENT

The source code for reproducing the work presented here is available at `https://github.com/anonOpenReview1/clip-hierarchy`. We implement our method in PyTorch (Paszke et al., 2017) and provide an infrastructure to run all the experiments to generate corresponding results. We have stored all models and logged all hyperparameters and seeds to facilitate reproducibility. Additionally, all necessary data preprocessing details are present in Appendix C.

REFERENCES

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, 2014.

Sebastian Bujwid and Josephine Sullivan. Large-scale zero-shot image classification from rich and diverse textual descriptions. In *Proceedings of the Third Workshop on Beyond Vision and LANguage: inTEgrating Real-world kNowledge (LANTERN)*, 2021.

Jaemin Cho, Seunghyun Yoon, Ajinkya Kale, Franck Dernoncourt, Trung Bui, and Mohit Bansal. Fine-grained image captioning with clip reward. In *Findings of NAACL*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

Yuxuan Ding, Lingqiao Liu, Chunna Tian, Jingyuan Yang, and Haoxuan Ding. Don't stop learning: Towards continual learning for the clip model, 2022.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning (ICML)*, 2019.

Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models, 2022.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Chia-Chih Kuo and Kuan-Yu Chen. Toward zero-shot and zero-resource multilingual question answering. *IEEE Access*, 2022.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2008.

Horea Mureşan and Mihai Oltean. Fruit recognition from images using deep learning. *Acta Universitatis Sapientiae, Informatica*, 2018.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, Mingxing Tan, and Quoc V. Le. Combined scaling for open-vocabulary image classification, 2021.

Sarah Pratt, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Shuhuai Ren, Lei Li, Xuancheng Ren, Guangxiang Zhao, and Xu Sun. Rethinking the openness of clip, 2022.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, 2010.

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. In *International Conference on Learning Representations (ICLR)*, 2021.

Shibani Santurkar, Yann Dubois, Rohan Taori, Percy Liang, and Tatsunori Hashimoto. Is a caption worth a thousand images? a controlled study for representation learning, 2022.

Sheng Shen, Chunyuan Li, Xiaowei Hu, Yujia Xie, Jianwei Yang, Pengchuan Zhang, Anna Rohrbach, Zhe Gan, Lijuan Wang, Lu Yuan, Ce Liu, Kurt Keutzer, Trevor Darrell, and Jianfeng Gao. K-lite: Learning transferable visual models with external knowledge, 2022.

Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic, 2021.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*, 2021.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, 2015.

Youngjae Yu, Jiwan Chung, Heeseung Yun, Jack Hessel, JaeSung Park, Ximing Lu, Prithviraj Ammanabrolu, Rowan Zellers, Ronan Le Bras, Gunhee Kim, and Yejin Choi. Multimodal knowledge alignment with reinforcement learning, 2022.

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv*, 2022.

Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021a.

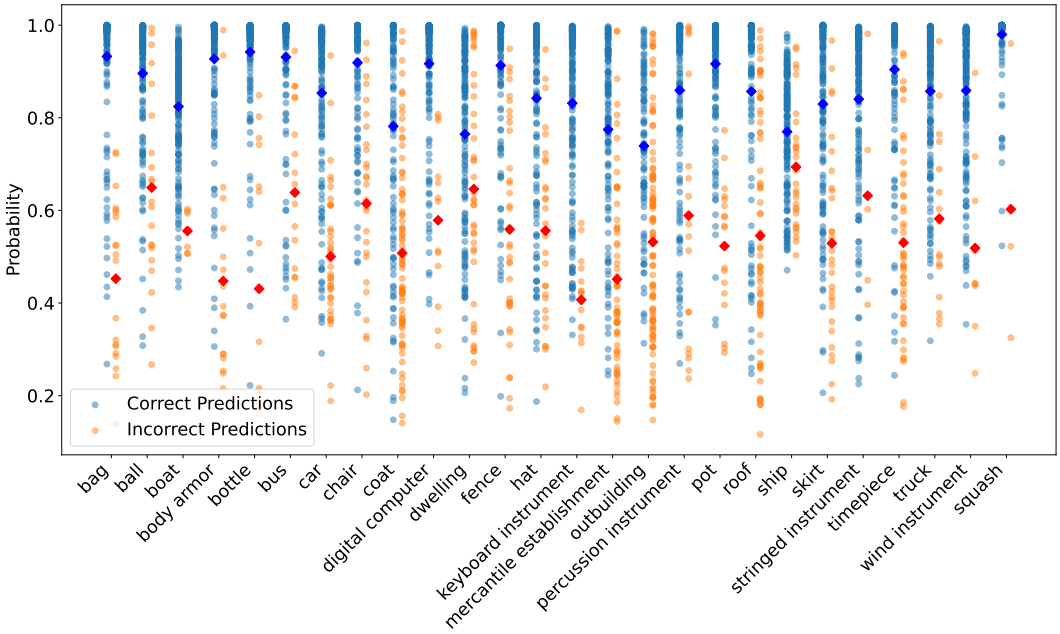
Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. *arXiv preprint arXiv:2112.02413*, 2021b.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

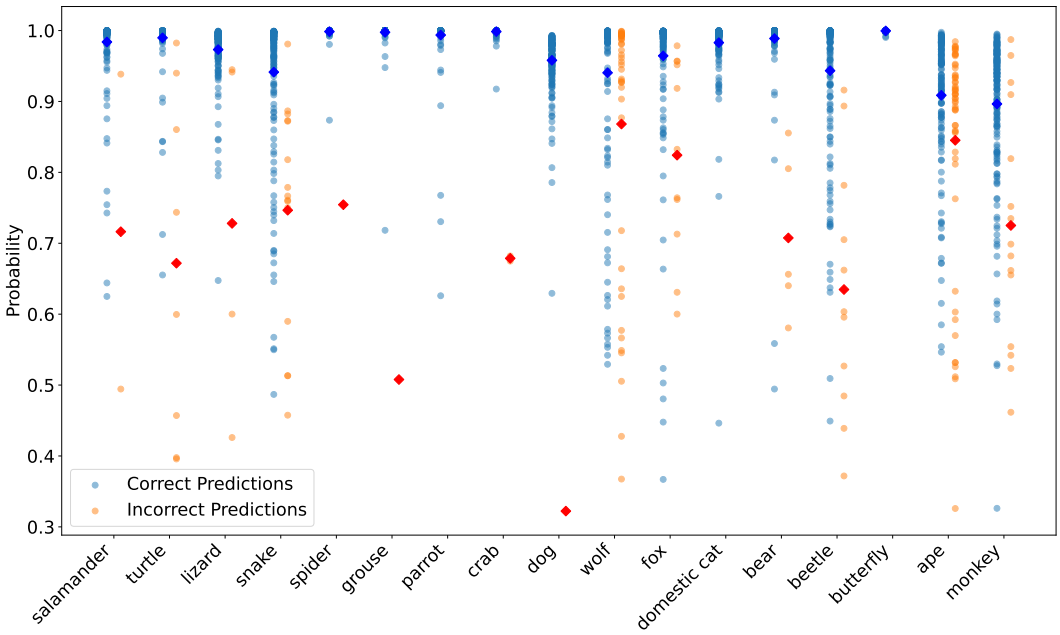
Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022b.

APPENDIX

A EMPIRICAL EVIDENCE OF CLIP CONFIDENCE



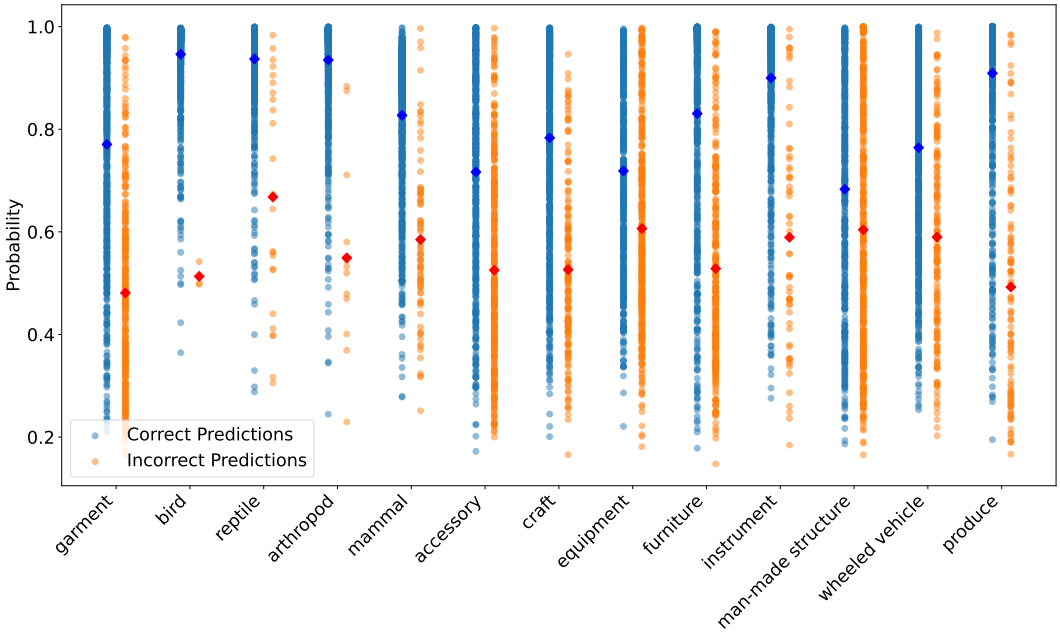
(a) nonliving26



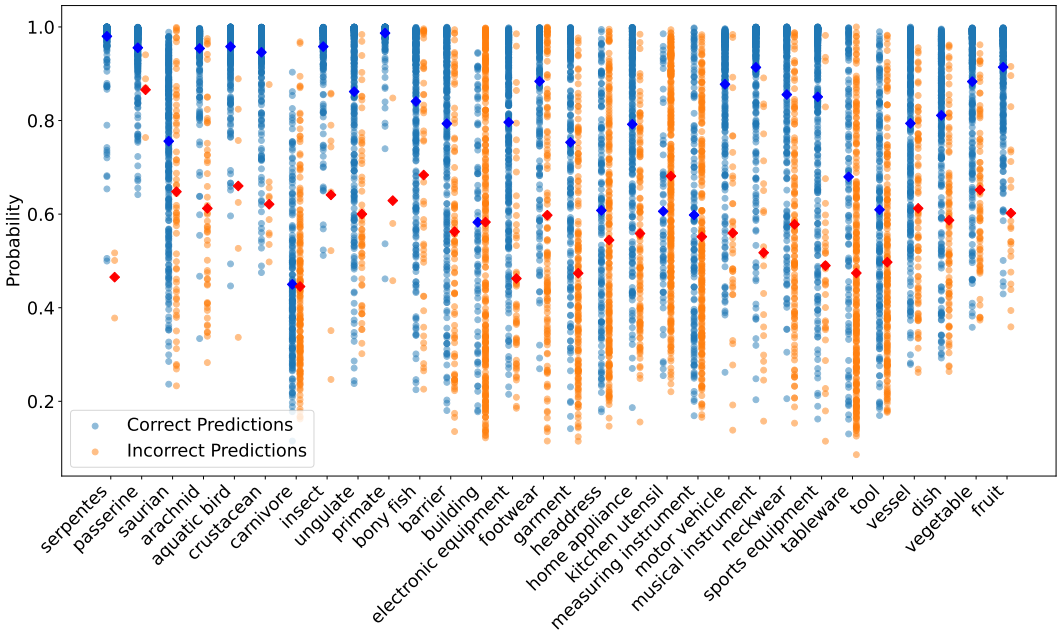
(b) living17

Figure 5: Distribution of argmax probabilities across ImageNet BREEDS datasets for correctly and incorrectly classified data points, with the diamonds representing average probability for each class. Correctly classified probabilities are on average higher than the misclassified probabilities.

The motivation behind the reweighting step of CHiLS primarily comes from the heuristic that LLMs make correct predictions with high estimated probabilities assigned to them (Kadavath et al., 2022). However, we also verify whether there is some evidence of this behavior in CLIP models. Given



(c) entity13



(d) entity30

that the output of a CLIP model is a probability distribution over the provided classes, we care specifically about the probability of the *argmax* class (i.e. the predicted class) when the model is correct and when it is incorrect. Across the BREEDS datasets for the standard ImageNet domain, in Figure 5 we show the distribution of the correct and incorrect argmax probabilities for each class (i.e. for each class c_i, we show the output probabilties for c_i when it was correctly classified and the output probabilities of the predicted classes when the true class is c_i). Whenever CLIP is correct, the associated probability is on average much higher than the probabilities associated with misclassification.

B Label Set Ablation Accuracy

Dataset	Superclass	CHiLS ($m = 5$)	CHiLS ($m = 10$)	CHiLS ($m = 15$)
Nonliving26	79.82	81.12(+1.30)	81.51 (+1.69)	81.79 (+1.97)
Living17	91.08	92.69 (+1.61)	91.43 (+0.35)	91.55 (+0.48)
Entity13	77.46	78.15 (+0.70)	78.11 (+0.65)	78.42 (+0.96)
Entity30	70.32	71.47 (+1.15)	71.75 (+1.43)	73.38 (+3.06)
CIFAR20	59.54	64.95 (+5.41)	65.90 (+6.36)	62.80 (+3.26)
Food-101	91.73	91.62 (−0.11)	91.56 (−0.17)	91.51 (−0.22)
Fruits-360	58.72	60.77 (+2.06)	61.38 (+2.66)	61.22 (+2.51)
Fashion1M	42.24	45.47 (+3.23)	46.01 (+3.78)	43.29 (+1.06)
Fashion-MNIST	68.49	70.93 (+2.44)	70.84 (+2.35)	69.09 (+0.60)
LSUN-Scene	88.20	86.33 (−1.87)	88.97 (+0.77)	86.80 (−1.40)
Office31	87.27	86.27 (−1.01)	88.37 (+1.10)	86.77 (−0.51)
OfficeHome	88.85	89.12 (+0.27)	88.76 (−0.09)	89.06 (+0.21)
ObjectNet	53.10	53.29 (+0.18)	53.52 (+0.42)	57.66 (+4.56)

Table 3: Accuracy across different label set sizes generated by GPT-3. In general, there is no consistent trend related to label set size and zero-shot performance across datasets.

Table 3 displays the raw accuracy scores for CHiLS across different label set sizes.

C Dataset Details

Dataset	Domains
BREEDS	ImageNet, ImageNet-Sketch, ImageNetv2, ImageNet-c {Fog-1, Contrast-2, Snow-3, Gaussian Blur-4, Saturate-5}
Office31	Amazon, DSLR, webcam
OfficeHome	Clipart, Art, Real World, Product

Table 4: Domains used for BREEDS, Office31, and OfficeHome.

CHiLS Across Domain Shifts For each of the BREEDS datasets (Santurkar et al., 2021), Office31 (Saenko et al., 2010), and OfficeHome (Venkateswara et al., 2017), all results presented are the average over different domains. The specific domains used are show in Table 4.

Fruits-360 For zero-shot classification with CLIP models, Fruits-360 (Mureşan & Oltean, 2018) in its raw form is somewhat ill-formed from a class name perspective, as there are classes only differentiated by a numeric index (e.g. "Apple Golden 1" and "Apple Golden 2") and classes at mixed granularity (e.g. "forest nut" and "hazelnut" are separate classes even though hazelnuts are a type of forest nut). We thus manually rename classes using the structure laid out in Table 6, which results in a 59-way superclass classification problem, with 102 ground-truth subclasses.

ObjectNet The ObjectNet dataset (Barbu et al., 2019) has partial overlap (113 classes) with the ImageNet (Deng et al., 2009) hierarchical class structure. From this subset of ObjectNet, we use the

BREEDS hierarchy (Santurkar et al., 2021) to generate a coarse-grained version of ObjectNet that is shown in Table 5. In this 11-way classification task, the true subclasses are the original ObjectNet classes.

Table 5: Class Structure for ObjectNet experiments.

Superclass	Subclasses (Original ObjectNet)
garment	{Dress, Jeans, Skirt, Suit jacket, Sweater, Swimming trunks, T-shirt}
soft furnishings	{Bath towel, Desk lamp, Dishrag or hand towel, Doormat, Lampshade, Paper towel, Pillow}
accessory	{Backpack, Dress shoe (men), Helmet, Necklace, Plastic bag, Running shoe, Sandal, Sock, Sunglasses, Tie, Umbrella, Winter glove}
appliance	{Coffee/French press, Fan, Hair dryer, Iron (for clothes), Microwave, Portable heater, Toaster, Vacuum cleaner}
equipment	{Cellphone, Computer mouse, Keyboard, Laptop (open), Monitor, Printer, Remote control, Speaker, Still Camera, TV, Tennis racket, Weight (exercise)}
furniture	{Bench, Chair}
toiletry	{Band Aid, Lipstick}
wheeled vehicle	{Basket, Bicycle}
cooked food	{Bread loaf}
produce	{Banana, Lemon, Orange}
beverage	{Drinking Cup}

Table 6: Mapping from original class names to new subclass and superclasses for Fruits-360.

Original Class	Cleaned Subclass	Cleaned Superclass
Apple Braeburn	braeburn apple	apple
Apple Crimson Snow	crimson snow apple	apple
Apple Golden 1	golden apple	apple
Apple Golden 2	golden apple	apple
Apple Golden 3	golden apple	apple
Apple Granny Smith	granny smith apple	apple
Apple Pink Lady	pink lady apple	apple
Apple Red 1	red apple	apple
Apple Red 2	red apple	apple
Apple Red 3	red apple	apple
Apple Red Delicious	red delicious apple	apple
Apple Red Yellow 1	red yellow apple	apple
Apple Red Yellow 2	red yellow apple	apple
Apricot	apricot	apricot
Avocado	avocado	avocado
Avocado ripe	avocado	avocado
Banana	banana	banana
Banana Lady Finger	lady finger banana	banana
Banana Red	red banana	banana
Beetroot	beetroot	beetroot
Blueberry	blueberry	blueberry
Cactus fruit	cactus fruit	cactus fruit
Cantaloupe 1	melon	melon
Cantaloupe 2	melon	melon
Carambula	star fruit	star fruit
Cauliflower	cauliflower	cauliflower
Cherry 1	cherry	cherry
Cherry 2	cherry	cherry
Cherry Rainier	rainier cherry	cherry
Cherry Wax Black	black cherry	cherry
Cherry Wax Red	red cherry	cherry
Cherry Wax Yellow	yellow cherry	cherry
Chestnut	nut	nut
Clementine	orange	orange
Cocos	cocos	cocos
Corn	corn	corn
Corn Husk	corn husk	corn husk
Cucumber Ripe	cucumber	cucumber
Cucumber Ripe 2	cucumber	cucumber
Dates	date	date
Eggplant	eggplant	eggplant
Fig	fig	fig
Ginger Root	ginger root	ginger root
Granadilla	granadilla	passion fruit
Grape Blue	blue grape	grape
Grape Pink	pink grape	grape
Grape White	white grape	grape
Grape White 2	white grape	grape
Grape White 3	white grape	grape
Grape White 4	white grape	grape
Grapefruit Pink	pink grapefruit	grapefruit
Grapefruit White	white grapefruit	grapefruit
Guava	gauva	gauva
Hazelnut	nut	nut
Huckleberry	huckleberry	huckleberry

Kaki	kaki	persimmon
Kiwi	kiwi	kiwi
Kohlrabi	kohlrabi	kohlrabi
Kumquats	kumquat	kumquat
Lemon	lemon	lemon
Lemon Meyer	meyer lemon	lemon
Limes	lime	lime
Lychee	lychee	lychee
Mandarine	orange	orange
Mango	mango	mango
Mango Red	red mango	mango
Mangostan	mangostan	mangostan
Maracuja	maracuja	passion fruit
Melon Piel de Sapo	melon	melon
Mulberry	mulberry	mulberry
Nectarine	nectarine	nectarine
Nectarine Flat	flat nectarine	nectarine
Nut Forest	forest nut	nut
Nut Pecan	pecan nut	nut
Onion Red	red onion	onion
Onion Red Peeled	red onion	onion
Onion White	white onion	onion
Orange	orange	orange
Papaya	papaya	papaya
Passion Fruit	passion fruit	passion fruit
Peach	peach	peach
Peach 2	peach	peach
Peach Flat	flat peach	peach
Pear	pear	pear
Pear 2	pear	pear
Pear Abate	abate pear	pear
Pear Forelle	forelle pear	pear
Pear Kaiser	kaiser pear	pear
Pear Monster	monster pear	pear
Pear Red	red pear	pear
Pear Stone	stone pear	pear
Pear Williams	williams pear	pear
Pepino	pepino	pepino
Pepper Green	green pepper	pepper
Pepper Orange	orange pepper	pepper
Pepper Red	red pepper	pepper
Pepper Yellow	yellow pepper	pepper
Physalis	groundcherry	groundcherry
Physalis with Husk	groundcherry	groundcherry
Pineapple	pineapple	pineapple
Pineapple Mini	mini pineapple	pineapple
Pitahaya Red	dragon fruit	dragon fruit
Plum	plum	plum
Plum 2	plum	plum
Plum 3	plum	plum
Pomegranate	pomegranate	pomegranate
Pomelo Sweetie	pomelo	pomelo
Potato Red	red potato	potato
Potato Red Washed	red potato	potato
Potato Sweet	sweet potato	potato
Potato White	white potato	potato
Quince	quince	quince
Rambutan	rambutan	rambutan
Raspberry	raspberry	raspberry

Redcurrant	redcurrant	redcurrant
Salak	salak	snake fruit
Strawberry	strawberry	strawberry
Strawberry Wedge	strawberry	strawberry
Tamarillo	tamarillo	tamarillo
Tangelo	tangelo	tangelo
Tomato 1	tomato	tomato
Tomato 2	tomato	tomato
Tomato 3	tomato	tomato
Tomato 4	tomato	tomato
Tomato Cherry Red	cherry tomato	tomato
Tomato Heart	heart tomato	tomato
Tomato Maroon	maroon tomato	tomato
Tomato Yellow	yellow tomato	tomato
Tomato not Ripened	unripe tomato	tomato
Walnut	nut	nut
Watermelon	melon	melon